Representation Learning in Irregular Domains

Michael Edwards

Submitted to Swansea University in fulfilment
of the requirements for the Degree of Doctor of Philosophy

Department of Computer Science
Swansea University

October 13, 2018

Declaration

This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed .... (candidate)

Date ....

Statement 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed .... (candidate)

Date ....

Statement 2

I hereby give my consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed .... (candidate)

Date ....

I would like to dedicate this work to Molly and Sam.

Abstract

The use of representation learning has grown rapidly over the last decade, replacing previous methods of producing hand-crafted feature descriptors in favor of machine learning algorithms which are able to form their own representation of the problem domain. Deep learning approaches have seen performance increases in numerous applications, utilizing advances in computational power and data collection to produce feature descriptor generating architectures. The introduction of Convolutional Neural Networks (CNNs), and their ability to learn spatially localized features, has seen representation learning in image analysis grow unlike any other. Although the convolution operator in CNN architecture is well defined for the regular spatial domain of the Cartesian grid, there are many application domains that do not exhibit such regular spatial topology. Such domains may still contain features that exhibit some spatial relationship, upon which a localized filtering operation like the one presented in CNNs could provide further benefit. This work explores the use of representation learning in such domains, presenting methods for learning features in irregular domains with applications in human action recognition and medical segmentation.

Firstly, we present the use of unsupervised clustering as a method for learning primitive behaviors for supervised action and interaction classification. The representation of primitive gestures that compound to form higher-level behaviors is learned by an evolutionary algorithm approach, identifying informative joints and samples that facilitate overall classification.

Following this, we explore the use of deep learning for producing learned feature descriptors, focusing on their use in irregular spatial domains. We present a generalized formulation of convolution and pooling operators that utilize a graph structure to represent underlying spatial relationships between input features. We evaluate the proposed method in learning localized features in domains where conventional CNNs are unable.

The presented Graph-CNN operators are then used for learning multi-scale features across multiple resolutions, relating information from global context to detailed local structures. By creating a hierarchical sampling approach we are able to densely sample raw information in the central region of focus, while sampling coarser information from a wider contextual area. A multi-resolution Graph-CNN architecture is then able to learn descriptive features from across multiple scales. An evaluative case study is provided in the context of medical segmentation, identifying pose of small-scale anatomical structures.

To conclude, we use a Graph-CNN architecture to learn features on temporal information, providing evaluation on human skeletal motion and evaluating its suitability as a representation learning method for human action recognition on the irregular spatial topology of skeleton models.

Acknowledgements

I would like to extend my sincerest gratitude to Dr. Xianghua Xie for his supervision during the course of my studies. Over the years he has been a constant source of advice and guidance, for which I am always grateful. I want to thank him for the discussions and support that has lead to my professional and personal development. Thank you for pushing me to do my best.

During my studies, I have been fortunate to work with many individuals in the Computer Vision group at Swansea University who have helped me bounce ideas and frustrations, and have helped with numerous tasks and endeavors. My gratitude is extended to Dr. Jingjing Deng, Dr. Gary Tam, Dr. Robert Palmer, Dr. Joss Whittle, and David George for their support throughout the years.

I would like to thank Dima Damen and Reyer Zwiggelaar for their valuable feedback.

Finally, I want to thank Alyssia Edwards for her patience. She has always supported me in pursuing something that I love, even when things have been difficult.

Contents

List of Publications

The following is a list of published papers as a result of the work in this thesis. An outline of the contributions from each can be found in Section 1.3.

1. M. Edwards, X. Xie, R. Palmer, G. K. L. Tam, R. Alcock, and C. Roobottom. Graph Convolutional Neural Network for Multi-scale Feature Learning. Under Review at Computer Vision and Image Understanding, 2018.
2. M. Edwards, X. Xie. Graph-Based CNN for Human Action Recognition from 3D Pose. Deep Learning on Irregular Domains Workshop, 2017. [1]
3. M. Edwards and X. Xie. Graph Based Convolutional Neural Networks. British Machine Vision Conference, 2016. [2]
4. M. Edwards, J. Deng, and X. Xie. From Pose to Activity: Surveying Datasets and Introducing CONVERSE. Computer Vision and Image Understanding, 2016. [3]
5. M. Edwards and X. Xie. Generating Local Temporal Poses from Gestures with Aligned Cluster Analysis for Human Action Recognition, British Machine Vision Workshop, 2015. [4]

In addition, there are a number of publications related to applying machine learning techniques to natural problems in face detection and biometric identification.

1. M. Edwards, J. Deng, and X. Xie. Labelling Subtle Conversational Interactions. Annotation of User Data for Ubiquitous Systems Workshop, 2017. [5]
2. J. Deng, X. Xie, and M. Edwards. Combining Stacked Denoising AutoEncoders and Random Forests for Face Detection. Advanced Concepts for Intelligent Vision Systems, 2016. [6]
3. M. Edwards and X. Xie. Footstep Pressure Signal Analysis for Human Identification. Biomedical Engineering and Informatics, 2014. [7]

List of Acronyms

ACA Aligned Cluster Analysis

AE AutoEncoder

AMG Algebraic Multigrid

ASM Active-Shape-Modelling

BoKP Bag of Key Poses

CMU Carnegie Mellon University

CAE Convolutional AutoEncoder

CNN Convolutional Neural Network

CT Computerised Tomography

CTW Canonical Time Warping

DMW Dynamic Manifold Warping

DTAK Dynamic Time Alignment Kernel

DTW Dynamic Time Warping

EA Evolutionary Algorithm

FFD free-form-deformation

GAN Generative Adversarial Network

GFT Graph Fourier Transform

Graph-CNN Graph-based Convolutional Neural Network

Graph-CNN-MSL Graph-CNN-Based Marginal Space Learning

HACA Hierarchical Aligned Cluster Analysis

HAR Human Action Recognition

HOF Histograms of Optical Flow

HoG Histogram of Oriented Gradients

IMU Inertial Measurement Unit

LOSO Leave-One-Subject-Out

LSTM Long Short-Term Memory

MHAD Multimodal Human Action Database

MLP Multilayer Perceptron

MoCap Motion Capture

MSL Marginal Space Learning

MSR Microsoft Research

NN Neural Network

PReLU Parametric ReLU

RF Random Forest

ReLU Rectified Linear Unit

RNN Recurrent Neural Network

SBU Stony Brook University

SIFT Scale Invariant Feature Transform

SSM Statistical-Shape-Modelling

STIP Space-Time Interest Point

SURF Speeded-Up robust Features

SVM Support Vector Machine

Tanh Hyperbolic Tangent

List of Tables

List of Figures

Chapter 1

Introduction

Contents

1.1 Motivations

In recent decades the use of machine learning methods has been widespread, utilizing a plethora of approaches in order to learn underlying information within an application domain. Numerous commercial and academic advances have made use of statistical pattern recognition methods, such as Random Forests (RFs) [8], Neural Networks (NNs) and Support Vector Machines (SVMs) [9], for a range of applications; including object recognition, medical diagnosis, content recommendation, and language translation. An increase in computational power and a growing pace of data acquisition has lead to great advances in machine learning approaches [10]. Previous state-of-the-art methods made use of hand-crafted features to represent the underlying structure of the data, however the advent of representation learning methodologies has lead to the development of machine learning techniques which are able to self-learn and optimize their feature descriptors based on the observed data. Such methods saw a second resurgence in popularity with the introduction of spatially localized feature representations provided by Convolutional Neural Networks (CNNs) [11], especially in image domain applications. In the years since, so called ‘deep learning’ architectures have been shown to continually achieve strong results in numerous topics of research, such as the impressive object recognition abilities of models such as ResNet [12] and GoogLeNet [13]. The development of the deep learning strategy for representation learning stems from not developing hand-crafted feature descriptors, instead relying on observed data to describe the function space of the problem to be solved. Many deep learning architectures developed for image domain problems make use of the regularly spaced array structuring of the input domain, such as the 2D Cartesian image grid, in order to define a notion of spatial locality. Localized features are learned which are generalizable with regards to their translation across the input domain, providing a reduced set of parameters to optimize. This formulation of localization has benefited problems such as image and volume recognition, however the benefits are less prominent for input data which does not reside on a grid domain, limited in its ability to learn spatially localized information due to the array input assumption.

The overarching motivation for this thesis aims to continue the current trend in delegating production of feature descriptors to the machine learning pipeline, avoiding the necessity to produce hand-crafted descriptors. We describe several domain applications, namely non-uniform sampling and human action recognition, upon which there are non-Euclidean spatial relationships inherent to the problem. Previously such methods would require either that the explicit spatial information be disregarded with Fully Connected Networks, or that the original space be embedded into a regular Euclidean domain for use with Convolutional Neural Networks. We explore the use of representation learning on domains which do not exhibit the regular spatial topology assumed in conventional CNN implementations, presenting methods for learning spatio-temporal features on the human skeleton model and a novel multi-resolution sampling method for multi-scale features.

1.1.1 Human Action Recognition

The human action recognition field aims to develop understanding from the observation of human behaviors for a variety of purposes, encompassing surveillance [14, 15] to human-computer interactions [16, 17]. Human Action Recognition (HAR) is the process in which a system attempts to label an observed sequence with a given action class label, and is closely related to the wider field of understanding human behaviors or actions; which can include the detection, segmentation and classification of behaviors within an observation [18–23]. The use of image based data collection is common in human action recognition applications, however the use of skeletal models to represent the structure of the human skeleton also helps to reduce the data into a space parameterized by the spatial location of key points on the human body. Numerous hand-crafted feature descriptors have been developed for human action recognition, in both appearance and pose modalities, with promising results on numerous datasets within the community [24,25]. Gains in the image processing field via deep learning approaches have seen large improvements in accuracy for the image modality formulations of HAR [26], however the same methods are less prominent with pose based approaches. Such representation learning methods on skeletal models have either ignored localized information in the input feature space [27, 28], or have made the assumption that such spatial information resides on the 2D Cartesian grid, in an attempt to use image processing operators [29,30]. This has motivated the following work to explore the use of representation learning with skeletal information, learning relationships between the joints on the human body and how such features relate to the observed behaviors. Chapter 2 gives an overview of the field and various methods used to develop models of human behavior for such applications, Chapters 4 and 7 aim to identify methods for the data driven construction of reliable feature descriptors in human action recognition; first via an evolutionary approach to feature selection in generating a bag of low level gestures, and then by applying deep learning methods generalized to the irregular topology of the human skeleton.

1.1.2 Deep Learning on Irregular Domains

The growth of deep learning approaches in image processing has produced numerous feature mining techniques to learn relationships between observed input data for tasks such as face detection and object recognition. The development of convolutional neural networks has seen the use of representation learning in image based problems become a prominent approach in both research and industry. The main operators of the architectures, convolution and pooling operations, exploit a well-defined and optimized kernel based scheme for learning localized relationships of features on the Cartesian grid. Such formulation allows for translation of a receptive field across an array domain, with efficient processing of convolutional neural networks on dedicated graphics processor units increasing the productivity and use of CNN architectures [10]. Convolutional neural networks have been used for understanding problems in images and video, learning spatio-temporal features from the grid, [26, 31, 32], however the application of deep learning on non-uniformly structured inputs has either ignored spatial information by using standard neural networks, or used a spatial embedding on the image grid which may make assumptions regarding the localized relationship between features which may be unsuitable [28–30]. Chapter 3 introduces the current state of deep learning approaches, highlighting the need for developing methods able to learn spatially localized features in applications that do not fit the assumption imposed by convolutional neural networks. Chapters 5, 6, and 7 are motivated by this aim to explore the utilization of deep learning methods for identifying a feature representation which is able to compute spatially localized, informative features on input domains which do not exhibit regular Cartesian grid topologies. Evaluation is provided in numerous example applications and case studies; applying the proposed Graph-CNN to signal classification tasks, 3D anatomical detection and segmentation, and human action recognition from skeletal pose.

1.1.3 Multi-scale Representation Learning

Much like conventional feature extractors, the use of deep learning for spatial feature mining has made attempts to incorporate methods for learning features from multiple scales into the descriptors they learn [33, 34]. Due to the regular Cartesian grid required by the convolutional neural network operators, the use of multi-scale features has been limited to using either multiple branches, each dedicated to a given scale [35, 36], or the use of shared or expanding filters, in which a learned kernel is interpolated to cover a wider area of interest [37]. Recent methods have used cascading approaches to gradually refine the resolution of prediction in detection problems [38], whilst others use ‘Multi-Scale Blob’ modules containing multiple filters of different receptive sizes [39]. Applications of such multi-scale approaches are vast, however notable examples include face detection in crowded scenes [39], medical segmentation [36] and time series understanding [40]. Methods with branching networks and modules introduce an increased parameter set to optimize, and neglect explicit spatial relationships between the data on separate branches in the network. Shared kernels work well to represent the same feature at different scales, which handle variation in feature scale well, but do not learn singular filters which describe information across differing scales. Chapter 6 was motivated by the aim to incorporate the spatial relationship between multiple scales into a deep learning framework able to learn spatial features across numerous scales. The study utilizes the Graph-based Convolutional Neural Network (Graph-CNN) method introduced in Chapter 5 to allow spatial information across different resolutions to be analyzed within a single irregularly sampled domain, avoiding the need for optimization of numerous kernels at varying scales whilst retaining information about how the original resolution input has been sampled.

1.1.4 Objective

The current state of deep learning has led to numerous advances in a variety of domains, and the use of Convolutional Neural Networks has seen improvements within the image processing and pattern recognition community. The assumption of a grid-based input domain for Convolutional Neural Networks provides a constraint on the representations that can be learned, producing kernels representing a local sampling and filtering operator. The issue with such an assumption is that it places a prior requirement on the input data, with the array structure defining the spatial relationships between elements in the input space. This assumption holds well for images and other grid-based domains, where we consider a pixel to be closely correlated with its neighbors, however there are domains in which there exists spatial relationships between input elements but this array-based assumption does not hold. By dropping the assumption of a regular domain topology we intend to open the learning of localized features to such domains, instead providing a method of learning localized feature descriptors on the graph representation of the domain. By relaxing this assumption we are able to maintain the spatial information within the input domain without requiring such a domain to be arbitrarily projected onto the Cartesian grid.

To this end, the overall objective of this work is to develop a method for learning feature descriptors on domains that exhibit an irregular spatial topology. In evaluation of this approach we specifically apply this irregular domain representation learning in the tasks of learning features across multiple scales and from temporal information. The proposed methods are generalizable to a wide range of applications, given their use of the graph representation for spatial localization, however several case study examples are used for evaluation. First, the use of a generalized approach to localized filter learning is given in an example domain of the irregularly sampled 2D grid, highlighting the shortcomings in convolutional neural network assumptions in relation to domains with an irregular spatial topology. Multi-scale feature learning of the relationships between highly detailed localized information and wider contextual structure is then explored in the context of medical segmentation, detecting the aortic valve of the human heart. Learning features from temporal motion on an irregular domain is then evaluated in the context of human action recognition on the joints of the skeletal model.

Although we evaluate the use of the proposed Graph Convolutional Neural Network on three specific problem applications, the method is applicable to numerous other domains. The use of a graph representation of the input domain allows a given application to formulate the neighborhood relationships on their input space. Using generalized graph construction methodologies or node connectivity defined by the specific domain it is possible to construct a graph representation of the problem domain which can be used within the Graph Convolutional Neural Network architecture for the purpose of learning localized feature descriptors and graph-based pooling operations. This objective follows our motivation in providing a localized feature representation learning scheme for domains which do not hold to the array-based input space assumption of CNNs.

1.2 Overview

With the motivations introduced in section 1.1, the aim of this study is to explore the use of representation learning in domains with an irregular spatial structure, with applications in HAR and medical segmentation problems. Common approaches in previous years have focused on the production of specifically hand-tuned feature descriptors that are based on extensive expert knowledge of the problem domain. Recent developments in representation learning have utilized deep learning architectures to learn such features from a large collection of observations. Despite their prominent usage in image domain problems, including recognition from images and videos, the use of deep learning in determining informative spatial features from topologies outside the Cartesian grids has been limited. In Chapter 4, we will present the use of unsupervised clustering to learn localized temporal features for the purpose of supervised action and interaction recognition, with the gesture primitives used in a bag-of-words approach being generated via a learned unsupervised clustering. In Chapter 5, we introduce the use of Graph-CNNs for the task of learning localized features in irregular spatial topologies, exploiting the deep learning approach by generalizing it to domains that do not satisfy the assumptions required by convolutional neural networks. Chapter 6 utilizes such a Graph-CNN for efficient simultaneous, end-to-end localized feature learning on local and global scales. In Chapter 7, we then utilize the presented Graph-CNN operators to learn features from temporal information, returning to the human action recognition problem by using the human skeleton as an input domain for the purpose of human action recognition.

1.3 Contributions

The main contributions of this study can be seen as follows:

- **A gesture learning scheme for a bag-of-words approach to action recognition.**
  We present a method for producing gestures for use in a bag-of-words approach to human action recognition. We dub the method 'bag-of-gestures', due to the use of spatio-temporal action primitives in the classification of higher-level actions and interactions from sequences of skeletal pose. The generation of action primitives is driven by a hierarchical unsupervised clustering mechanism with an evolutionary hyper-parameter optimization scheme. Observed sequences are selected and clustered based on the population's identification of informative body locations, learning a generalized representation for the target action classes which is based on the observed data. Evaluation is provided on the tasks of human action recognition and interaction recognition between two individuals via the modality of human skeleton pose. The utilization of hierarchical clustering as a mechanism for gesture generation provides improved stability in regards to variation in observation timings when compared to previous methods in [41].

- **A deep learning approach to learning localized features on irregular spatial domains.**
  An approach to deep learning of localized feature representations is presented which utilizes a graph representation of the underlying spatial relationships between input features in the domain of application. Graph-based Convolutional Neural Network operators are introduced with a stable weight update scheme. The presented contributions provide smoother weight optimization and an increased stability in regards to the derivatives obtained when creating smooth spectral multipliers. An evaluation on a proof-of-concept

domain is given, utilizing an irregularly sampled 2D grid upon which regular convolutional operators cannot function.

- **Local and global feature learning via a multi-resolution Graph-CNN.**
  We present a method for representation learning which incorporates local and global scale features into a single spatial domain, producing a multi-resolution sampling scheme of the input resolution. Previous methods of learning spatially related local and global features have relied on either expanding kernels to learn the same feature at different scales, or the use of branches for each scale. Both methods come with their own shortcomings, including increased network parameters or dissociated features. The introduced method combines local and global information into a single filtering operation, maintaining spatial relationships between features whilst avoiding the need to create branching networks or filters for separate scales, such as those seen in [35–39]. A case study is provided in the context of medical segmentation, utilizing multi-resolution Graph-CNNs for segmentation of the aortic root in human body scans.

- **Learning temporal features on domains with an irregular spatial topology.**
  We utilize the Graph-CNN operators to develop an architecture for learning features from temporal information. An evaluation is provided on a graph of the human skeleton model for the purpose of action recognition, learning features based on observed localized motion of bodily joints. A feature representation is learned by the Graph-CNN approach, which relates a given frame to a class based on multi-scale temporal information from preceding frames. Utilizing the proposed Graph-CNN operators enables localized feature descriptors to be constructed on the irregular spatial topology of the human skeleton without resorting to an assumption of a regular Cartesian embedding as seen in [29, 30].

Outcomes from this thesis have also contributed to several publications as outlined in the List of Publications. The key contributions of each paper related to the main body of work can be summarized as follows:

M. Edwards, J. Deng, and X. Xie. From Pose to Activity: Surveying Datasets and Introducing CONVERSE.

A detailed survey and discussion regarding the current state of the art in Human Action Recognition and the datasets available to the community. Contributes to Chapter 2.

M. Edwards and X. Xie. Generating Local Temporal Poses from Gestures with Aligned Cluster Analysis for Human Action Recognition.

By utilizing hierarchical aligned cluster analysis and evolutionary selection of input features we propose a method for learning a bag of low level gestures for the task of human action and interaction recognition. Contributes to Chapter 4.

M. Edwards and X. Xie. Graph Based Convolutional Neural Networks.

We introduce a method that utilizes a graph representation of the spatial relationships with a given input domain to provide a representation learning approach that learns localized features without requiring the array-based constraint of classical Convolutional Neural Networks. Contributes to Chapter 5.

M. Edwards, X. Xie, R. Palmer, G. K. L. Tam, R. Alcock, and C. Roobottom. Graph Convolutional Neural Network for Multi-scale Feature Learning.

We propose a method for learning features across multiple scales, producing a multi-resolution patch which contains irregularly sampled spatial features. By representing the multiple scales as a singular graph we are able to utilize the Graph Convolutional Neural Network operators presented in Chapter 5 to learn feature descriptors without defining multi-branched neural network architectures. Contributes to Chapter 6.

M. Edwards, X. Xie. Graph-Based CNN for Human Action Recognition from 3D Pose.

We propose a method that utilizes the Graph Convolutional Neural Network operators presented in Chapter 5 to learn feature representations on a graph of the human skeleton as identified by motion capture markers. Contributes to Chapter 7.

1.4 Outline

The rest of this study is outlined as follows:

Chapter 2 Human Action Recognition:

We introduce the background to current works in human action and interaction recognition, looking at the development of feature representations for actions.

Chapter 3 Deep Learning:

This chapter introduces deep learning as a method of representation learning, highlighting considerations required in domains which show irregular spatial topology.

Chapter 4 Unsupervised Learning of Gestures for Human Action Recognition:

This chapter explores the use of an evolutionary unsupervised segmentation method for the extraction of primitive gestures for use in supervised classification of human actions. Experimental evaluation of the method is given and discussed.

Chapter 5 Deep Learning in Irregular Domains:

This chapter introduces a method for end-to-end learning of feature representations on problems exhibiting an irregular spatial domain. Graph convolutional and pooling operators are introduced as an analogue to convolutional neural networks. Experimental evaluation of the method is given in comparison to previous state-of-the-art methods.

Chapter 6 Graph Convolutional Neural Networks for Multi-Scale Feature Learning:

In this chapter we utilize the presented Graph-CNN operators to learn spatially related local and global features in an end-to-end deep learning approach. We define a multi-resolution graph to associate information at different scales for efficient feature learning. A focal case study is given in the domain of medical segmentation. Evaluation and results are given in comparison to state-of-the-art methods

Chapter 7 Graph Convolutional Neural Network for Temporal Feature Learning:

In this chapter we utilize Graph-CNN methodology to learn temporal features from motion capture sequences for action classification. An evaluation and discussion is given on a public dataset in comparison to numerous state-of-the-art methods.

Chapter 8 Conclusions and Future Work:

We draw concluding remarks on the presented studies, looking to future use cases and the potential for development.

Chapter 2

Human Action Recognition

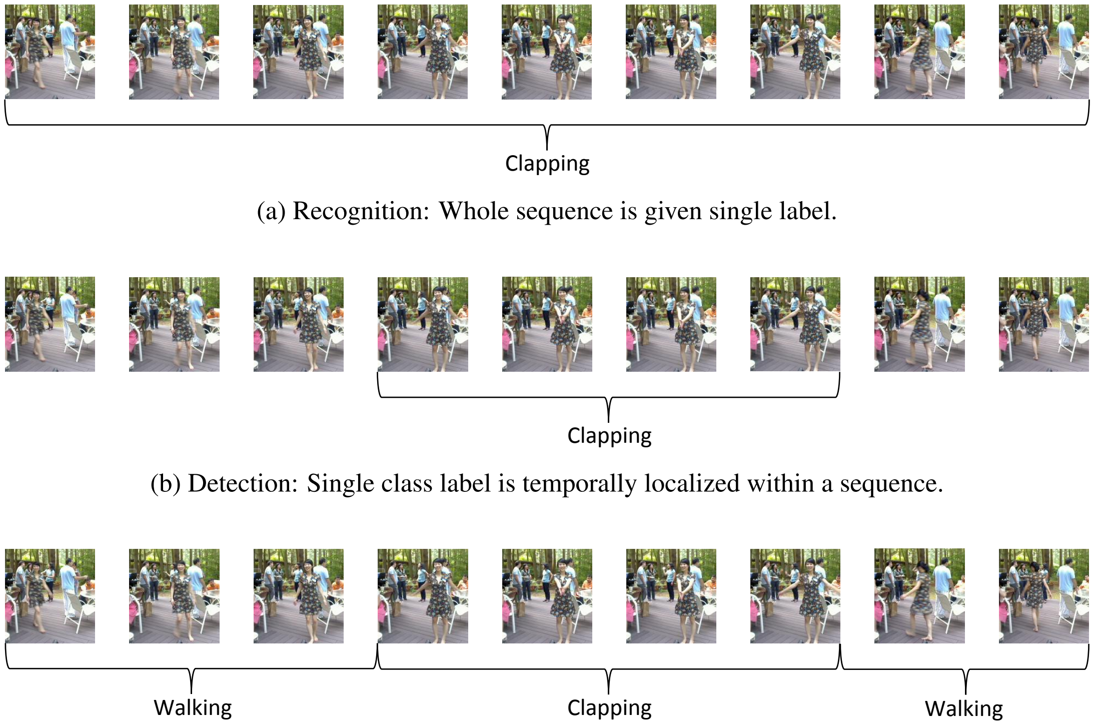
Contents

2.1 Introduction

Recent advances in human motion, action and interaction recognition have shown a range of practical real-world applications; including surveillance, synthesis of computer generated imagery, and human-computer interfaces [42]. The focus of the field has shifted several times through the years, however there are still problems towards understanding complex classes and maintaining high accuracy rates on significantly large datasets. Data modalities within the field have been strongly aimed at understanding behavior from appearance based information (RGB and grayscale images or videos), with the advent of the commercial depth sensor providing resurgence in the study of pose-based understanding. The field has recently moved from the previous methods of utilizing hand-crafted features for classifier training, towards more deep learning driven approaches, learning feature representations from a set of observed data. The use of both appearance- and pose- based information has been present since the early years of the community, with numerous datasets produced for both modalities. Such datasets cover a wide range of human actions, from low-level gestures up to complex activities or interactions and have been implemented into numerous real-world applications. Given the popularity and strength of deep learning methods such as Convolutional Neural Networks (CNNs), which utilize learned localized features optimized for image and video domain problems, it is understandable that Human Action Recognition (HAR) from appearance based data remains prominent in the field.

The problem of understanding human behavior has lead to several approaches be adopted in regards to a specific desired tasks; including recognition, segmentation, or detection systems. Recognition, which we mostly focus on in this piece of work, aims to classify an entire observation, whether a sequence or a static image, with a single label [20, 43]. Detection methods aim to localize a single action class within an observation; such systems may provide spatial annotations, temporal annotations, or both [20, 44, 45]. Similar to this, segmentation methods look to localize and label potentially numerous action classes within an observation, providing a label for each time frame [46–48]. Different application domains make use of the different approaches in various ways, for example surveillance applications will often make use of video segmentation, whilst video retrieval tasks may query a database using keyword matching via detection. Figure 2.1 gives an overview of the three different approaches.

The following chapter introduces the background to Human Action and Interaction Recognition; discussing the use of differing modalities, the use of feature descriptors and the current state of the art in the field. With this understanding of the field, we will look at the use of hand-



(a) Recognition: Whole sequence is given single label.

(b) Detection: Single class label is temporally localized within a sequence.

(c) Temporal Segmentation: Sequence is labeled on a frame-wise basis.
Note that spatial segmentation can also occur to localize within frames.

Figure 2.1: Comparison of human behavior tasks. Note that these methods can be expanded to handle multiple subjects within view, interactions with objects, and multi-class labels for individual frames.

crafted features and how the use of deep learning has been exploited to learn features from both appearance- and pose- based information. We discuss the growth of the field from consideration of generic emphasized actions towards the understanding of interactions between numerous individuals. Datasets are analyzed based on a variety of key properties that influences their use for various HAR techniques, including number of action classes, complexity of events and their application domain. Differing levels of abstraction within the understanding of human behavior are described, detailing the nature between pose, gesture, action, interaction, and activity. This will lead us into Chapter 4 and learning primitive gestures as descriptive features, and Chapters 5 and 7 in which we explore the use of deep learning on the irregular domains as a method for learning feature descriptors on the irregularly spaced human skeleton graph.

2.2 The Development of Human Action Recognition

In the 1970s, Johansson presented a model of the human form that closely followed the biological interpretation of human movement. The human skeleton representational model, based in Gestalt principles that provide key interest points in the movement, allowed for human actions to be accurately and consistently classified by observers in a reduced representation space [49, 50]. Clearly information from this reduced space was still highly beneficial for classification tasks. Such human representation schemes were then expanded by [51–56] to develop computational models that are able to identify human walking behaviors. A review of the field by [57] focused on the recognition of articulated movement and acknowledged the benefit *a priori* shape models provided in solutions to HAR applications. In a further reduction of the feature space, [58] used 3D coordinate locations of 14 joints on the human body to perform event recognition from a continuous sequence of ballet moves.

In following years, the use of pose-based information waned in favor of video and image analysis, due in part to their ease of acquisition and relatively lower cost compared to the use of motion capture systems at the time. Half a decade later, Aggarwal produced another review of the field [59], discussing the fusion of both body part representation and motion of the whole body. The review recognized the need for accurate tracking of body parts in human action recognition tasks. Utilizing 3D estimation from 2D projections revealed the difficulty in estimating the position of joints in the scene when using appearance based pose extraction methods, which in turn was deemed to lower the generalization and predictive abilities of the models developed. The review then draws light on the use of tracking motion without the need to directly identify body parts; making use of image processing methods for appearance based tracking such as bounding box locality [60] and mesh features [61–65]. By removing the need to identify specific structures of the human model it is possible to learn information from the generalized appearance in a localized scene about the body, representing actions in terms of localized motion. Motion features from appearance based modalities became widespread in human action recognition, developing descriptors of action classes which included motion fields [66, 67], motion histories [67], Histogram of Orientation Gradients [68, 69], and space-time interest points [70–72]. The subsequent introduction of the KTH and Weizmann human action recognition datasets, providing a collection of sequences with which to evaluate developed methodologies, resulted in a growth of appearance based approaches [71, 73]. The two datasets provided numerous action classes for recognition and have seen vast use as a comparison dataset in the community. Despite this, both sets were representative of the time in the

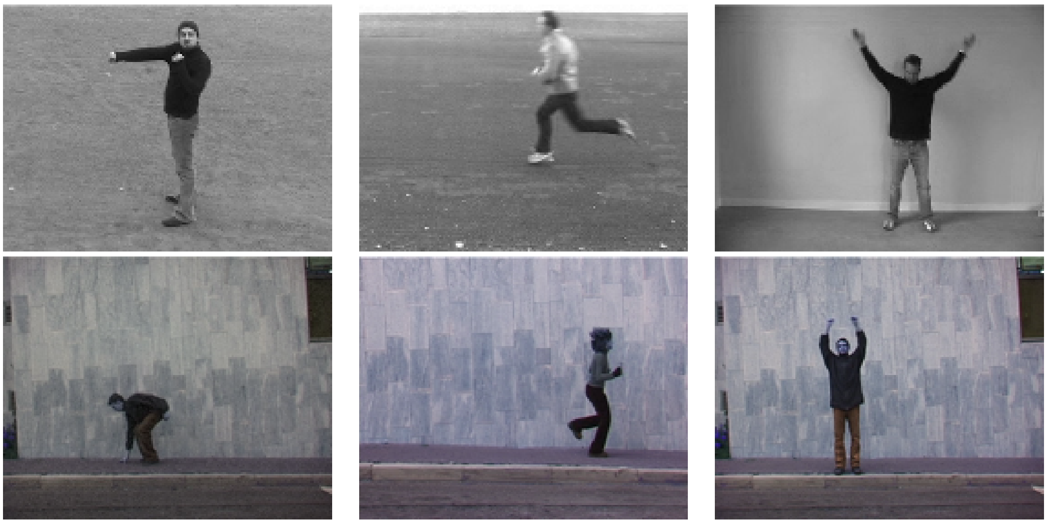


Figure 2.2: Example images from top) KTH dataset - punch, run, hand waving. bottom) Weizmann dataset - bend, jump forwards, two-handed wave.

field's development, being a composition of single camera recordings of individual subjects performing discrete actions which are readily defined by a sequence of poses they contain. As such, the feature descriptors developed during this phase of human action recognition worked well in generalizing information on such observations, yet in more natural settings the descriptors were less successful [74].

Following the release of the KTH and Weizmann action sets, Figure 2.2, recent appearance based HAR approaches have moved towards attempting to understand complex interactions, multiple subjects, and natural environments. Contextual understanding of the scene as a whole has been explored in recent years, with [75] utilizing the behaviors of multiple subjects in the scene to help obtain accurate classification of a given individual's action. Further appearance datasets are reviewed in [76] with identification of sets that provide classes for specific domains and describing complex scenarios; including meta-source sets, multi-view recordings, and repositories of long observations.

2.3 Human Action Recognition from Pose

In general machine learning applications there is the intent to model an underlying function space via some feature representation embedding. Both high and low level features have been shown to provide their own benefits to given applications of machine learning, especially with the development of the deep learning approaches identified in Chapter 3. Lower level features in image domain problems tend to look towards relating pixel intensities with the underlying problem, whilst higher level features attempt to learn relationships between lower level features in order to develop some semantic understanding in how such features relate to one another. Low and high level feature representations in human action recognition methodology has been explored for decades, with low level features typically limiting the recognition of classes to those with distinct spatio-temporal poses, such as the jumping jack, handshake and high-five. Higher-level features further generalize on the lower level features to a point, providing a deeper understanding of the observed data on a function to be modeled.

For the advancement of human action recognition problems, [25] suggested the consideration of higher level 3D pose features as a benefit over lower level appearance features. Whilst Yao acknowledged previous difficulties in obtaining accurate 3D pose features, it has since become much easier to obtain relatively accurate pose tracking via the commercialization of depth sensors. Such advances in pose capture and estimation techniques has enabled the collection and release of numerous 3D human pose datasets for HAR. Much as the release of the KTH and Weizmann datasets lead to an increase in appearance based feature descriptors, the release of several depth based pose datasets has resulted in various features being developed for the body pose domain; including joint-joint/joint-plane distances, motion velocities, and histograms of joint orientations [25, 77–79]. Current human action recognition methods often make use of numerous modalities for recognition [80–83], with particular highlight on the benefit of audio-visual fusion [84–86]. The development of depth-based datasets has grown in recent years, however the use of appearance based methods is still prominent. Datasets in both modalities have moved towards representing more complex human behavior and more realistic environments and observations. It is worth noting that appearance based datasets are still advanced in comparison to the depth based sets, mostly due to the ease of capturing and the modality's prominent use in several application domains.

2.4 Human Action Recognition Methods

Human action recognition methods and feature descriptors have been well studied in both image and depth focused methodologies. Pose rich actions, those with a readily identifiable sequence of key poses such as waving, clapping and walking, have shown strong performance gains through the development of feature descriptors which utilize localized motion information and template matching [87–89]. The number of public datasets on which to evaluate developed methodology on such primitive classes is vast, with sets beginning to trend towards more realistic natural observations in recent years. Current feature descriptors produced for the HAR domain have focused on the use of spatio-temporal features, such as Space-Time Interest Points (STIPs) [24]. Schuldt *et al.* [71] utilizes space-time interest points for action classification from video, developing a vocabulary of action primitives that train a Support Vector Machine (SVM) classifier. Such STIP features identify spatio-temporal corners within an XYT volume, located in regions of high image intensity variation along all three axial directions. In a video this identifies spatial corners that exhibit high motion between consecutive observed frames. The STIP extractor presented by Laptev [24] detects such spatio-temporal corners across several spatial and temporal scales for a given in an observed sequence and has been utilized extensively in action recognition [90, 91]. Blank and Gorelick *et al.* [43, 73] presented the action event as an XYT volume utilizing silhouette masking of the observed frame, extracting local saliency and orientation combined with global space-time features to perform spectral clustering based classification. The authors note that the method works well for scenarios where a known background is present, given the requirement to silhouette mask the subject in the scene. Methods designed for action representation, segmentation and recognition via appearance information has been reported in [92]; identifying the spatial features, temporal model, temporal segmentation, and view invariance provided by each method for appearance based recognition. Appearance based information representation is once again evaluated in [93], concluding that low level appearance features are able to achieve high accuracies on benchmark datasets, with mid to high level temporal features providing further gains in scenarios featuring strong temporal structure. The introduction of learned feature representations provided by deep learning methodologies has since seen favorable usage within human action recognition from appearance. Indeed several architectures incorporating Convolutional Neural Network operations to learn localized features on the image and video domains has shown strong performance gains in numerous benchmark datasets [31, 94–96]. A study of recent deep learning approaches in HAR for appearance modalities [26] shows that deep learning based approaches are able to learn spatio-temporal features from a given dataset with good generalization capabilities, however they note that a limiting factor is the reliance on large datasets in order to train models which avoid the overfitting problem. Further insight into the state of deep learning for the purpose of human action recognition is discussed in Section 2.7.

For pose based recognition [97], the commercial depth sensor has become a well utilized method of identifying human skeletal structures within a scene, providing representations of subjects with a standard underlying structure [98–101]. Often such pose estimation techniques can make use of appearance and depth based information to predict the human pose in an observation using regressors such as Random Forests [102, 103], others make use of traditional motion capture rigs to provide accurate 3D tracking [84]. The exploration into pose based HAR lead to a resurgence in pose estimation techniques [25, 104–110], with numerous public datasets for pose based method evaluation being released [84, 102, 108, 111, 112]. In addition, many recognition methods have been developed which are more generic in their ability to use both appearance and skeletal model derived features; focusing on the learning of similar representative sub-action primitives, which are then verified using both appearance and skeletal features [113–116]. Recent skeleton features have included relative position and motion information, such as joint-to-joint and joint-to-plane descriptors [25, 117–121], histograms of 3D joints [77], and manifold projections [122, 123]. During the rapid growth of representation learning methods in recent years, various methods for feature descriptors on skeleton data have been proposed, with varying success. [124] proposes a method for learning feature combinations on low level motion and positional features, with aggregations of features being fed to an optimization strategy which looks to minimize classification error returned by a non-parametric k-NN model. Learning temporal features from skeletal information via recurrent neural network architectures are presented in [27] and [28], however neither make use spatially informative feature learning such as those provided in convolutional networks, instead relying on Long Short-Term Memory and recurrent neural network approaches to learn motion information between features.

2.5 Difficulty in Human Action Recognition

During the field's development, some core problems have revealed themselves; namely, variation in execution style and appearance. Appearance artifacts are reduced in methods that only consider the human skeleton model and the articulation of such a model, removing all external appearance based stimuli. Despite the benefit of stripping away anecdotal image domain information, it is argued that this lack of appearance features may remove higher level contextual information within a scene [117]. The use of a skeletal model of bodily motion is also dependent on the reliable capture of such information, and that modality's robustness to noise. Motion Capture (MoCap) tracking systems often provide a high degree of fidelity under well constrained conditions, with modern systems providing accurate spatial localization of points on the surface of the body with a high temporal sampling frequency [125, 126]. Other methods make use of pose estimation techniques, attempting to extract accurate pose of the human skeletal structure from image or video [21, 127, 128]. Such pose estimation methods are themselves subject to noise within the scene, including occlusions and variance, and their subsequent use in action recognition can be dependent on the viability of the pose estimation portion of the pipeline [129, 130].

In recent years the field has returned to appearance based information as the main domain for action recognition, utilizing deep learning operators optimized towards image domain problems to learn generalized features which aim to be more robust to small variations in appearance. Such representation learning schemes have shown promising results, just as they have in other applications within the field of image processing [26]. In addition to spatial appearance differences, temporal execution variation has a strong impact on the ability to recognize events. Variations in execution speed of sequences and the ordering of primitive gestures in higher-level activity classes can have damaging effects on the performance of numerous algorithms. Several methods attempt to utilize some scale invariant features for detection tasks, producing features which are more robust to the effects of zooming or temporal stretching [43, 115, 131], however such methods can present computationally expensive multi-scale extractors.

In activity recognition problems where a sequence of primitive actions are compounded into some larger semantically meaningful task, such as ‘setting a table’ or ‘cooking a meal’, there can be significant variation in the execution styles. Certain activities or sub-activities may exhibit a definitive order in which action primitives and gestures must be executed, often described as a sequence of key poses [101, 132–134]. Such behaviors can often be classified well by methods that consider sequence matching in some form, and indeed pattern recognition approaches work well in these classes. The more variation execution style of higher level activities is more challenging, with bag of words style approaches attempting to handle the numerous possible paths an observed event can take in order to complete its goal. Current datasets are moving towards representing more complex interactions and activities, however a plethora of datasets are available for the action recognition problem in which basic primitive classes are observed, such as ‘punching’ and ‘kicking’. The problem of execution length variation soon became a key focus of HAR methods, and methods such as Dynamic Time Warping (DTW) have been used to align two sequences of actions whilst taking temporal variance into account. Utilization of DTW for action recognition, adapted from [135] by [136, 137] and expanded on by [115, 138–140] has been used to compare sequences of differing execution length. Despite DTW’s success in time series analysis problems, the method has also come under criticism, especially when representing highly periodic actions [111] or actions where the temporal execution rate is considered a defining feature, such as walking versus running [134]. As a response, exemplar based methods make use of key poses analogous as a series of checkpoints frames which must be observed to compose a given action, and therefore are believed to not require a time warping alignment phase [141, 142]. These developed methodologies seem to provide reliable accuracy for the publicly available datasets on which they are often validated, despite their variance in execution rates and styles, by removing superfluous frames from the representation space.

Another key issue in the community is the lack of methods able to extend beyond the recognition of simplistic behavioral classes. [92] reports upon the predictive accuracy of methods that are evaluated on the KTH, Weizmann and IXMAS datasets; showing that in recent years the level of accuracy can often reach over 90%. State of the art performance accuracy is also reported within [143], with older datasets often reporting the highest number of correct classifications. Indeed it can be observed that current deep learning approaches applied to appearance information often perform well on current datasets in comparison to previous methods which utilize hand-crafted feature descriptors [26]. In [143] and [26] it is shown that those datasets more representative of real world observations tend to challenge current methods within the community; such as Hollywood1/2, HMDB51 and Olympic Sports. This suggests that current HAR methods are now able to easily classify the relatively simplistic classes presented in established datasets, but that the community requires challenging with complex scenarios. Previously the generation of datasets has been the issue, however recent advances in generating large datasets which are representative of real-world observations has been shown to improve the robust training of models within applications such as deep learning [144, 145]. Such datasets have at first been challenging for their respective fields, however such advances are required to further understand the topics at hand and the release of larger human action recognition datasets is becoming more prominent [112].

2.6 Evaluating the Current State of Human Action Recognition Datasets

Numerous HAR datasets have been produced and publicly released in the last two decades for the purpose of detecting and identifying action events in an observed scene. Many of these sets allow cross-comparison of proposed methodologies, becoming key benchmark sets for the field. Due to the vast number of available datasets and their intended usage, it is important to carefully consider which sets allow an appropriate validation of the proposed method. Variations between sets can be seen in many attributes of the sets, from number of class to the modality of capture. Numerous modalities for observing human behaviors are now possible; including RGB videos, depth maps, accelerometers and marker based motion capture. The field has developed over time to represent higher-level behaviors, from simple gestures to discrete actions, activities to complex interactions. Some datasets make use of original data collection, allowing a degree of control over certain parameters within the data collection methodologies, others use meta-data collected from video clips from publicly available sources. There tends to be a correlation between sets that collect their own original data, an expensive task, versus those able to collect a vast quantity of samples from films and online videos. Meta-sets tend to have large amounts of variation between individual sequences, however they are also among the largest of the datasets, with some meta-sets containing thousands of sequences [146, 147]. Labeling of observations also impacts on the intended usage of the set. Numerous sets have ground truth labels for an entire sequence; however many are either manually segmented out of a continuous sequence of multiple actions, or are left for users to perform labeling before their use. Ground truth labeling on a frame-by-frame basis is rare, due to the complexity in determining the exact frame at which an action begins, how methods for automating such processes are increasing in the field, further expanding available data [148].

In the following sections we discuss key datasets shown in Table 2.1 and their composition, exploring the coverage of the field and highlighting potential areas for new datasets. By looking at the data collection modality, the size of sets and the types of behaviors observed we present a baseline for why the field should move towards larger datasets of more complex behaviors in order to fully push the abilities of the deep learning methodologies currently being used.

2.6.1 Modalities

In Table 2.2 we cluster the datasets based on how they represent the observed behavior events; video, depth maps, skeletal tracking, MoCap marker tracking, IMU, and audio. The vast majority of sets in HAR make use of vision, especially since the resurgence of appearance based information resulting from the growth of deep learning and CNNs. Feature descriptors investigated are often domain specific, therefore understanding the modality presented by a dataset

Table 2.1: Comparisons of key action recognition datasets, detailing the presented modalities, download location, associated descriptive publications, and number of simultaneous viewpoints.

Name	Modality	URL	Description	Views
50 Salads	RGB-D, IMU	[149]	[150]	1
BEHAVE	RGB	[151]	[152]	2
Berkeley MHAD	RGB-D, IMU, Audio, MoCap	[153]	[84]	14
BIT Interaction	RGB	[154]	[155]	1
CAD120	RGB-D	[156]	[45]	2
CAD60	RGB-D	[156]	[157]	2
CASIA	RGB	[158]	[159]	3
CAVIAR	RGB	[160]	[161]	1, 2
CMU MMAC	RGB, MoCap, IMU	[162]	[113]	6
CMU MoCap	MoCap	[163]	-	1
CONVERSE	RGB-D	[164]	[118–120]	1
Drinking/Smoking	RGB	[165]	[166]	1
ETISEO	RGB	[167]	[15]	1, 3, 4
G3D	RGB-D	[168]	[169]	1
G3Di	RGB-D	[170]	[171]	1
HMDB51	RGB	[172]	[146]	1
Hollywood	RGB	[173]	[174]	1
Hollywood-2	RGB	[175]	[176]	1
Hollywood3D	RGB-D	[177]	[178]	1
HumanEVA-I	RGB, MoCap	[179]	[180]	7
HumanEVA-II	RGB, MoCap	[179]	[180]	4
IXMAS	RGB, Silhouette	[181]	[182]	5
JPL	RGB	[183]	[184]	1
K3HI	RGB-D	[185]	[186]	1
KTH	RGB	[187]	[71]	1
LIRIS	RGB-D	[188]	[189]	1
MPI08	RGB, IMU, Laser Scan	[190]	[191, 192]	8
MPII Cooking	RGB	[193]	[194]	1
MPII Composite	RGB	[195]	[196]	1
MSR Action-I	RGB	[197]	[198]	1
MSR Action-II	RGB	[197]	[199]	1
MSR Action3D	RGB-D	[197]	[102]	1
MSR DA3D	RGB-D	[197]	[111]	1
MSR Gesture3D	RGB-D	[197]	[200]	1
MuHAVi	RGB, Silhouette	[201]	[202]	8
Olympic Sports	RGB	[203]	[204]	1
POETICON	RGB, MoCap	[205]	[206]	7
Rochester AoDL	RGB	[207]	[208]	1
SBU Kinect Interaction	RGB-D	[209]	[108]	1
Stanford 40 Actions	Image	[210]	[25]	1
TUM Kitchen	RGB, Markerless MoCap, RFID	[211]	[212]	4
UCF101	RGB	[213]	[147]	1
UCF11	RGB	[214]	[215]	1
UCF50	RGB	[216]	[217]	1
UCF Sport	RGB	[218]	[219]	1
UMPM	RGB, MoCap	[220]	[221]	1
UT Interaction	RGB	[222]	[223]	1
ViHASi	RGB, Silhouette	[224]	[225]	40
VIRAT	RGB	[226]	[227]	-
Weizmann	RGB, Silhouette	[228]	[43, 73]	1
WVU MultiView	RGB	[229]	[230, 231]	8

Table 2.2: Comparison of provided data and presence of dedicated validation sets.

	Datasets
Data	
RGB/Greyscale	All sets except CMU MoCap
MoCap	Berkeley MHAD, CMU MMAC, CMU MoCap, HumanEVA-I, HumanEVA-II, POETICON, TUM Kitchen, UMPM
Depth	50 Salads, Berkeley MHAD, CAD120, CAD60, G3D, G3Di, Hollywood3D, K3HI, LIRIS, MSR Action3D, MSR DA3D, MSR Gesture3D, SBU Kinect Interaction, CONVERSE
Skeleton	Berkeley MHAD, CAD120, CAD60, G3D, G3Di, K3HI, MSR Action3D, MSR DA3D, SBU Kinect Interaction, CONVERSE
IMU	50 Salads, Berkeley MHAD, CMU MMAC, MPI08, TUM Kitchen
Audio	Berkeley MHAD, POETICON
Laser Scan	MP108

	Appearance sets	Pose sets
Train/Test split		
Yes	Drinking/Smoking, ETISEO, Hollywood, Hollywood 2, IXMAS*, KTH, Olympic Sports, Rochester AoDL*, Stanford 40 Actions, UCF101, UCF11*, UCF50*, UCF Sport*, UT Interaction, ViHASi*, VIRAT*, Weizmann*, WVU MultiView-I, WVU MultiView-II	Hollywood3D, HumanEVA-I, HumanEVA-II, LIRIS, MSR Action3D, SBU Kinect Interaction, TUM Kitchen*, CONVERSE*
No	BEHAVE, BIT-Interaction, CASIA, CAVIAR, HMDB51, JPL, MPII Cooking, MPII Composite, MSR Action-I, MSR Action-II, MuHAVi	50 Salads, Berkeley MHAD, CAD120, CAD60, CMU MMAC, CMU MoCap, G3D, G3Di, K3HI, MPI08, MSR DA3D, MSR Gesture3D, POETICON, UMPM

[1] provided in description paper via Leave Out cross validation methodology

will impact on the choice of features used to describe each sequence.

Video

Appearance based HAR makes use of datasets that are often collected via still images or video, as cameras can provide a relatively cost effective method of obtaining both real-world and staged execution samples from both a laboratory or real-world environment. In Table 2.1 it can be seen that all of the datasets presented contain some form of video or appearance based data (except CMU MoCap, K3HI and UCF iPhone), therefore in Table 2.2 we omit the video data. The quality of the recordings varies greatly between sets, with some specializing in evaluating action detection and recognition in low quality or small-scale recordings. High intra-set and inter-sequence variation in image quality, camera motion, scale and viewpoint are common in meta-data sets that collect observations from multiple sources, such as UCF101, UCF50, UCF11, Hollywood, Hollywood-2 and HMDB51, and these pose a more realistic problem to the community. Visual based HAR can provide an intuitive representation of the scene, however there can often be superfluous information contained within an observation that negatively impacts on the reliable global recognition of a given action; therefore, appearance based modalities can often make use of subject localization and background removal, coupled with the extraction of descriptors such as STIPs, Histogram of Oriented Gradients (HoG), Histograms of Optical Flow (HOF) or local regions of motion features to enable the global recognition of actions regardless of background information or subject-specific appearance. Many depth-based datasets also provide simultaneously captured video representations of their data; this appearance data can either be omitted from the learning, or combined to form a multi-modal system. Of the appearance-based datasets, the KTH and Weizmann datasets have been cited the most for single action recognition method evaluation. For appearance based interaction recognition the CAVIAR, Hollywood and UT Interaction datasets have been used frequently by the community.

MoCap

Motion capture concerns the recording of numerous markers placed upon the body by multi-camera systems, providing accurate tracking of the markers within a volume over time. MoCap often provides a method of capturing a spatial ground truth for the marker locations within the scene, being used as a stand-alone modality or augmenting datasets captured through other methods. MoCap systems are often calibrated using built in software and a calibration tool, allowing all cameras to be spatially and temporally synchronized, increasing confidence in the marker tracking. Placement of the markers varies between datasets and as such datasets which make use of MoCap provide details of the marker placement on the body, allowing semantic affordance to be applied to each marker. MoCap can be seen as a cost-expensive method of data collection, often requiring dedicated systems, however the generation of a spatial ground truth and reliable pose tracking method is of great benefit when developing pose from appearance or pose based action recognition methodologies. Despite this, an implementation of marker based MoCap systems in a real world environment is impractical, requiring individuals to wear a motion capture suit to be detected by the system would provide little benefit to the user; as such there has been some effort has also been made to produce human skeletal tracking without the use of markers from simple RGB image recording [212] and from depth maps [102].

Of the HAR datasets that utilize MoCap, the HumanEVA, Berkeley Multimodal Human

Action Database (MHAD) and Carnegie Mellon University (CMU) MoCap datasets are most commonly used. The HumanEVA dataset provides a set of evaluation metrics for the purpose of action recognition, Berkeley MHAD provides a detailed dataset containing multiple modalities for fusion based action recognition, and the CMU MoCap dataset contains a vast number of continuous sequences which can be used for action detection and sequence segmentation.

Depth

The production of a consumer level depth sensor, most notably the Microsoft Kinect, coupled with efficient and accurate joint tracking software has provided the HAR community with an inexpensive method of collecting 3D poses of a subject performing actions within a scene [99–101]. This has allowed for the development of methods that represent the action as a series of key poses or bag of words model [118, 132, 133], extracting the key frames that describe the overall action event. Datasets such as 50 Salads, Berkeley MHAD, CAD120, CAD60, G3D, G3Di, K3HI, LIRIS, MSR Action3D, MSR DA3D, MSR Gesture3D, and SBU Kinect Interaction all make use of the Kinect depth sensor to collect data providing the depth map of the scene. The Hollywood3D set utilizes commercial films that have been recorded using a 3D stereo camera system to provide depth maps. By obtaining a 3D pose estimation of the subjects within the scene users are able to, given accurate tracking, generate pose, scale, and appearance invariant features for the purpose of HAR that include joint trajectories, joint-joint distances, joint-plane distances, and joint motion histories. Many of the depth datasets captured using the Kinect provide the associated estimated skeleton representation of the individual, tracking a number of joints across the scene. The number of joints tracked and the position of the provided markers often depends on the method used to extract the skeleton; those using the Microsoft Kinect SDK often provide 20 points, whilst those using the OpenNI standard track 15 joints on the body. The selection of joints often aligns with the major joints of the human body, and so provides an estimation of limb motion. Currently the use of depth sensors are limited to a viewpoint that is in a roughly front-on position due to the method of estimating depth, using distortions of infra-red projections into the scene which is then captured by a receiving sensor. This method has little ability to handle scene occlusions that can cause shadowed regions in the depth map, resulting in lost or noisy tracking in the extracted skeletons.

The most prominent depth datasets for single person actions include those presented by the Microsoft Research group, namely the Action3D and DA3D datasets. Despite the small number of samples and action classes provided by the MSR Action3D dataset there has been

Table 2.3: Comparison of dataset interaction types. Note that datasets can contain instances of several types of behaviors based on the labeling it provides.

	Appearance sets	Pose sets
Event type		
Action	CASIA, CAVIAR, Drinking/Smoking, ETISEO, HMDB51, Hollywood, Hollywood-2, IXMAS, KTH, MSR Action-I, MSR Action-II, MuHAVi, UCF11, UCF Sports, ViHASi, VIRAT, Weizmann, WVU MultiView-I, WVU MultiView-II	50 Salads, Berkeley MHAD, CAD120, CAD60, CMU MoCap, G3D, Hollywood3D, HumanEVA-I, HumanEVA-II, LIRIS, MPI08, MSR Action3D, MSR Gesture3D, POETICON, TUM Kitchen, UMPM
Interaction: Person - Person	BEHAVE, BIT Interaction, CASIA, CAVIAR, ETISEO, Hollywood, Hollywood-2, JPL, UT Interaction	CMU MoCap, G3Di, Hollywood3D, K3HI, LIRIS, POETICON, SBU Kinect Interaction, UMPM, CONVERSE
Interaction: Person - Object	ETISEO, MPII Cooking, MPII Composite, VIRAT	50 Salads, CAD120, CMU MMAC, LIRIS, POETICON, TUM Kitchen, UMPM
Activity	CASIA, MPII Composite, MuHAVi, Olympic Sports, Rochester AoDL, Stanford 40 Actions, UCF101, UCF11, UCF50, UCF Sports, ViHASi	50 Salads, MSR DA3D, CAD60, LIRIS, TUM Kitchen, CONVERSE

a vast number of citations for its use as an evaluation dataset. For person-person interactions there are few datasets available which make use of depth based data; the K3HI and SBU Kinect Interaction datasets provide sequences of single executions of a given interaction, analogous to those provided by the BIT Interaction and UT Interaction appearance datasets, however their recent release may reflect their low citation and usage for evaluation of pose based methods.

Other

Various other methods of data capture have been used for HAR purposes, including the use of audio recordings [84, 232] and IMUs [84, 233, 234]. These methods can provide reasonable classification results on their own, however they are often used in a multi-modality system to improve the accuracy rates of single modality methods. These datasets are beyond the scope of this survey and omitted for brevity.

2.6.2 Behavior Types

Human behaviors are often a set of events with differing levels of abstraction and complexity, producing a problem for human action recognition methods to learn. Identifying primitive gestures with definitive poses can be straightforward in comparison to learning semantically broad compositions of unordered interactions. Therefore to aid comparison between HAR class types we shall first define some assumptions made about terminology we wish to use. Many class labels provided within HAR datasets can often be relabeled to fit within a different level of abstraction, however we attempt to use common terminology found across the community, with an overview provided in Figure 2.3 and a summary of the datasets in Table 2.3. Example images from datasets that describe differing levels of abstraction are given in Table 2.4.

Pose An atomic observation of the spatial arrangement of a human body at a single temporal instance, e.g. 'Arm above head'.

Gesture A temporal series of poses on a sub-action scale, sometimes described as action primitives e.g. 'Arm moves left'.

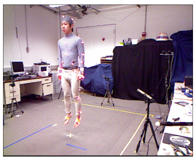
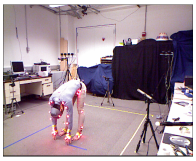
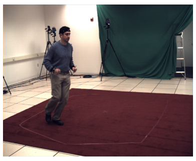
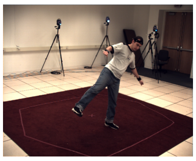
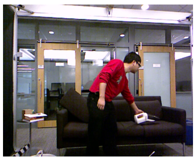
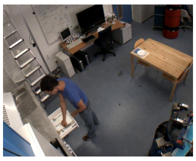
Action A series of gestures which form a contextual event, e.g. Repeated gestures of arm moving left and then right can be contextual described as an 'overhead wave action'. These are the most commonly used class labels found within current datasets, describing single actions executed by a subject including 'run', 'jump', and 'wave'.

Interaction A pairwise or reciprocal action is committed by two entities on each other. Each entity therefore has a single action that reflects its state compared to the other entity, i.e. consider the action of person A shaking the hand of person B; A executes the action of shaking the hand of B, B executes the action of having their hand shaken by A, together this pairwise action execution can be described as that of a 'handshake' interaction. For the purpose of action recognition interactions are often further divided into differing interaction types based on if the entities include people, objects or groups. For this study we have omitted group interaction datasets due to space limitations.

Person-Person An action is committed directly by one individual upon another. This definition does not include crowded scenes in which an individual performs a single person action with other subjects in the environment. The class labels in a P-P interaction treats the interaction as a single entity, rather than two separate single person actions, e.g. we consider the class 'punching' as an interaction between person A, the puncher, and person B, the individual being punched.

Person-Object An action is committed directly by one individual upon an object. This includes the manipulation of objects. We consider class labels such as 'lift chair' and 'open box' as person-object interactions as the actions 'lift' and 'open' are performed on the objects 'chair' and 'box' respectively.

Table 2.4: Example frames of currently available depth based human action recognition datasets. Images are provided here to give insight into the types of classes provided by pose based data.

Action Type	Dataset	Example frames		
Action	Berkeley MHAD			
Action	HumanEva			
P-P Interaction	SBU Kinect Interaction			
P-O Interaction	50 Salads			
Activity	MSR DA3D			
Activity	TUM Kitchen			

Groups Characterized as interactions carried out between a collected entity of more than two individuals. Group interactions can include inter- and intra-group behaviors and the interaction of the group on other objects, individuals, or even other groups. These often form their own subsets of group behaviors.

Activity A collection of actions and/or interactions that compound to describe a high level event. These are common within the sets that describe daily behaviors, e.g. 'cook a meal' and 'tidy room' can often include numerous actions and interactions that are executed. Each action and interaction can therefore be thought of a sub-activity event in such scenarios. Activity is also used to describe the daily activities, a more realistic observation execution than the exaggerated instances such as 'punch' and 'kick'.

A common scenario presented within HAR instances is that of a single person executing a singular action, in which an individual actor performs an action with no interaction to other individuals or objects, such as within KTH, Weizmann, MSR Action, and MSR Action3D. In recent years, interaction datasets have become more prominent, often displaying actions where one actor performs an action upon which another actor is the recipient. These interaction sets can still exhibit behaviors that are quite well defined, with a single instigator and a single recipient, such as punching, pushing and move towards. The most notable interaction sets include BIT Interaction, UT Interaction, K3HI, and SBU Kinect Interact datasets. There also exists interaction classes that are more complex in their composition, involving multiple entities, object manipulation or requiring higher level semantics; these are prominent in the TUM, BEHAVE, VIRAT, ETISEO, and POETICON datasets. The higher-level activity datasets often provide observations of an entire task being carried out and require the understanding of the sub-activity actions and interactions being carried out over the course of the recording. In the current sets there are often annotations of low-level actions that are encompassed within a higher-level activity context, with sets such as MPII Composite, 50 Salads and TUM Kitchen providing annotations of both levels of abstraction and the objects that are subject to interactions during the course of the activity.

The choice of classes that are performed by the actors is a key motivation in the generation and usage of the proposed dataset. Often the actions executed are those of a visually definable nature, comprising single executions of a discrete action that contains key poses and gestures. The complexity of the problem can then be increased by observing multiple executions of actions in a sequence, either with distinct boundaries between the classes or with a natural flow

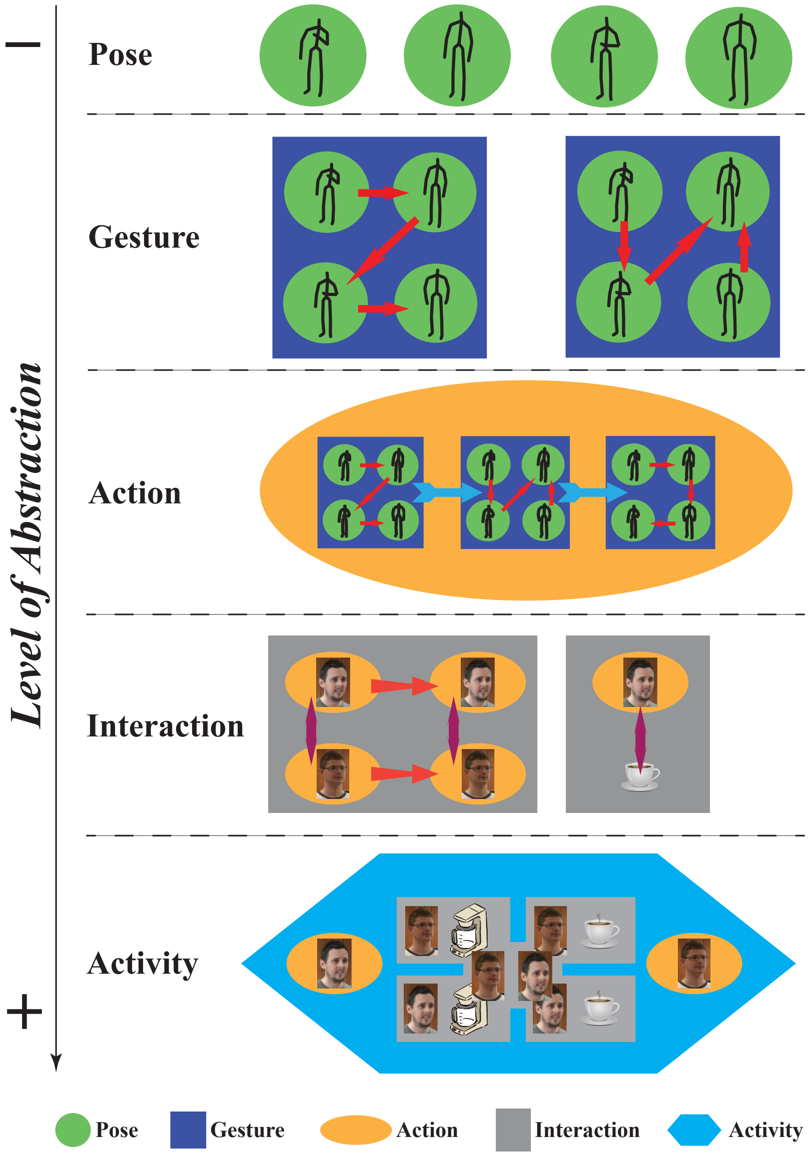


Figure 2.3: Levels of abstraction within human action recognition.

between different classes. These are all complex issues that are the focus of the community, with segmentation methods often utilized to separate out actions from a continuous sequence. Judging the difference in complexity between two classes can be subjective, depending upon the subtlety of gestures, the context of any interactions, and the spatio-temporal rigidity of the executions; subtle gestures, for example, may well present a more complex recognition problem than the simplest of activity classes. We can however make some generalized assumptions about the complexity within the different abstraction levels. Lower levels of abstraction such as pose and gesture should provide fewer challenges to the field in its current state, while higher levels of abstraction, especially those involving interactions between two or more entities, still remain a challenging issue.

Obviously with the definitions of the action types presented there can be some overlap in how to handle events in which an entity is not only interacted with, but also pivotal to the context of the label. Consider the class label ‘smoking’, this event can fit both into the definition of a singular action in which the object is explicit to the action, a person-object interaction between the person and cigarette, and also into its own activity class in which smoking is the task executed. Consider also the class label of ‘pushing’; this may be a class label that can be readily classified as a single action, person-person interaction, or person-object interaction depending upon the entities present, and also as an activity if there is a contextual background to the event. This highlights the complexity in describing class labels and requires the careful consideration of overlaps that appear to be presented between datasets with similar action classes. To further this point, we ask should the community consider an interaction as its own complete class, or should the system understand the states occupied by all entities within the interaction, i.e. the class label of ‘pushing’ may be deconstructed into sub-classes that describe the action of the instigator and the reaction of the recipient. Many interaction datasets handle the class labeling as a single complete unit of interaction, often reliant on the action committed by the instigator, e.g. K3HI, SBU Kinect Interaction, and UT Interaction. However the TUM Kitchen, 50 Salads and MPII Composite sets explicitly annotate the states of both entities to define the person-object interactions for the purpose of activity recognition. The use of a single interaction class that encompasses all sub-divisions of that interaction may provide learning that is broad and resistant to variation of intra-class behaviors; however by learning the sub-divisions of an interaction class, considering the different actions and reactions as their own states, there may be an ability to learn more effective boundaries for execution variations. For this study we have considered and evaluated upon the class labels provided by the original datasets; however we invite the community towards potentially defining multi-scale class labeling for the purpose of action and activity recognition.

2.6.3 Sizes

The size of a dataset, not just in the number of sequences but also in the range of different action classes and participants, can impact on its suitability for method evaluation. Training of deep learning classification networks, especially in situations with complex function spaces, often requires large datasets to provide generalized features which are representative across observation variation. Testing on a small-scale dataset can provide misleading results during analysis that may not be replicated when introducing more class labels or observations, due in part to the highly variable nature of inter- and intra-instance executions. Contrarily there are implications in the usage of large datasets; not only the collection and storage of data, but also in the processing of features, class learning and validation. Due to the inherent issues in obtaining a large number of participants, action classes, and sequences, the largest sets tend to be meta-sets, which collect action sequences from various sources, such as YouTube and films, containing large variation between sequences; this often makes meta-sets highly variable and challenging problems to be solved. A summary of dataset sizes is given in Table 2.5.

Number of Classes

Datasets with a small number of action classes, such as MSR Action-I, MSR Action-II, and Drinking/Smoking, can often provide strong recognition results in part due to the low number of partitions needed to divide the actions provided within the set. Those sets that contain a large number of action classes, namely HMDB51, UCF101, and UCF50, provide a difficult challenge to HAR methods due to the need to find partitioning information within each class that allows for inter-class partitioning, whilst preserving intra-class similarity. Due to the inconceivable number of possible actions and interactions that can exist in the real world it can be beneficial to evaluate methodologies on datasets with a large number of distinct action classes. Much as image classification datasets may contain the labels $y \in \{$dog, cat, ...$\}$, $y \in \{$Alaskan Malamute, Beagle, German Shepherd, ...$\}$, or even $y \in \{$dog left leg, dog right leg, cat left leg, cat right leg, ...$\}$, it is possible for such coarse- or fine-scaled granularity to be applied to labeling of observations of human action. Dependent on the task being undertaken it may be suitable to provide a coarse labeling of behavior, while others may require more in-depth annotation and understanding. Datasets provided in Table 2.1

contain a coarse labeling, with little fine-scale annotation. Finer scale class labeling can often come a requirement for more discriminative ability to produce accurate and stable predictions, and requires an extensive and accurate ground truth labeling approach in order to produce a meaningful set of observations.

Number of Subjects

Datasets that are able to provide more individual subjects performing an action are able to portray the variability in inter- and intra-subject execution of a given class. Observations of the same action class can often differ greatly in both their temporal rate and spatial occupancy, leading to complexity in learning the action for recognition purposes. Methods that are able to provide subject invariant action recognition should provide consistent results on a dataset that contains a large number of subjects. Again, the meta-sets tend to provide the highest number of subjects, almost capturing a new subject per sequence, representing a large range of inter-subject variation.

Number of Samples per Class

The number of observations per class, and the balancing of samples available for each class, can impact on the ability of a system to suitable learn a given class. A low number of observed instances of a class can result in weak recognition of unobserved instances of the same class. Unbalanced datasets can often result in models trending toward a class of which it observes more samples; this is often the case with supervised methods which optimize their parameters based on a loss function which compares their predictive ability against the true labels of a set of observed sequences they are provided with. Balancing, sampling, and stratification of the observed dataset can often have a large effect on a model's training [235, 236]. Some datasets provide observations with a large class imbalance, such as the Smoking/Drinking dataset, which provides an action detection problem with a large number of negative samples where neither class occurs. Other sets aim to balance the number of observations within each class; HMDB51 provides over 100 instances of each action class it contains, providing a range of observations across differing viewpoints, quality and executions, as such it can provide a useful benchmark for the recognition of actions from a subject and observation invariant methodology. Current pose based datasets contain few repeated instances of an action class, often with 3-5 repetitions per subject per class. To increase the number of instances per class it is possible to segment those datasets containing continuous recordings of multiple executions into discrete single execution clips, this includes the KTH dataset. It is also possible to utilize data augmentation methods such as spatial dimension flips, shifting and scaling, and appearance channel permutation, in an attempt to increase the variance of observations [237, 238].

Number of Sequences

The total number of sequences within a dataset should be a factor of the number of subjects, classes, and number of class executions, and as such can impact on the reliability of the results produced. Larger datasets can provide larger testing sets for which to evaluate a system, allowing for more confidence in the results of the validation. Size alone however is only one parameter in the selection of evaluation benchmark, with domain, class complexity and modality impacting on the application of methodologies to real world implementations.

2.6.4 Application Domains

The intended application domain of a dataset can provide certain intrinsic features in the data collection methodology and action classes captured, from low resolution images of CCTV surveillance footage to more complex action sequences of daily living. Some actions are representative of the domain from which they are intended; for example the UCF-Sports dataset, [219], makes use of numerous actions from various sports, such as javelin throws and long jumps. We classify the datasets into 4 action class domains; generic actions, daily living, surveillance, and sport. Generic action datasets have no overall theme, instead providing classes that are pan-domain; these include the classes ‘running’, ‘jumping’, ‘punching’, and also more complex interactions such as ‘handshake’ or ‘play guitar’. Daily living datasets often include actions and activities that are more natural in their execution and environment, this includes classes based on assisted living and household tasks. Surveillance datasets often make use of elevated view points and lower resolution images, mirroring the common camera setups in the security industry [227, 239]. Sports based action recognition often makes use of previously captured data from multiple sources, often containing varying image quality and varying levels of camera motion. A summary of the domains for each of the datasets is provided in Table 2.6.

Generic

Many action recognition datasets often contain generic action classes that are observable in numerous domains. The intention is to cover a wide variety of actions to allow domain in-

Table 2.5: Comparison of dataset sizes.

	Appearance sets	Pose sets
# Actions		
≤ 5	Drinking/Smoking, MSR Action-I, MSR Action-II	
6 - 10	BEHAVE, BIT Interaction, CAVIAR, Hollywood, Hollywood-2, JPL, KTH, Rochester AoDL, UCF Sport, UT Interaction, Weizmann, WVU MultiView-II	CMU MMAC, HumanEva-I, HumanEva-II, K3HI, LIRIS, MPI08, POETICON, SBU Kinect Interaction, UMPM, CONVERSE
11 - 15	CASIA, ETISEO, IXMAS, UCF11, VIRAT, WVU MultiView-I	Berkeley MHAD, CAD60, G3Di, Hollywood3D, MSR Gesture3D, TUM Kitchen
16 - 20	MuHAVi, Olympic Sports, ViHASi	50 Salads, CAD120, G3D, MSR Action3D, MSR DA3D
≥ 21	HMDB51, MPII Cooking, MPII Composite, Stanford 40 Actions, UCF101, UCF50	CMU MoCap
# Subjects		
≤ 5	Rochester AoDL	CAD120, CAD60, HumanEVA-I, HumanEVA-II, MPI08, POETICON, TUM Kitchen
6 - 10	MSR Action-I, MSR Action-II, UT Interaction, ViHASi, Weizmann	G3D, MSR Action3D, MSR DA3D, MSR Gesture3D, SBU Kinect Interaction
11 - 20	IXMAS, MPII Cooking, MuHAVi	Berkeley MHAD, G3Di, K3HI, CONVERSE
≥ 21	CASIA, KTH, MPII Composite	50 Salads, CMU MMAC, CMU MoCap, UMPM
Undefined	BEHAVE, BIT Interaction, CAVIAR, Drinking/Smoking, ETISEO, HMDB51, Hollywood, Hollywood-2, JPL, Olympic Sports, Stanford 40 Actions, UCF101, UCF11, UCF50, UCF Sport, VIRAT, WVU MultiView-I, WVU MultiView-II	Hollywood3D, LIRIS
# Sequences		
≤ 20	BEHAVE, CAVIAR, MSR Action-I,UT Interaction, WVU MultiView-II	HumanEVA-II, TUM Kitchen, CONVERSE
21 - 100	ETISEO, JPL, MPII Cooking, MSR Action-II, Weizmann	50 Salads, CAD60, CMU MMAC, G3Di, HumanEVA-I, MPI08, POETICON, UMPM
101 - 500	BIT Interaction, Drinking/Smoking, Hollywood, MPII Composite, Rochester AoDL, UCF Sport, ViHASi	CAD120, G3D, K3HI, MSR DA3D, MSR Gesture3D, SBU Kinect Interaction
501 - 1000	KTH, Olympic Sports, WVU MultiView-I	Berkeley MHAD, Hollywood3D, LIRIS, MSR Action3D
≥ 1001	CASIA, Hollywood2, HMDB51, IXMAS, MuHAVi, Stanford 40 Actions, UCF101, UCF11, UCF50, VIRAT	CMU MoCap

variant action recognition, with generic datasets being the most widely used for validation purposes, including the KTH [71], Weizmann [73] and MSR Action3D [111] sets. Many generic datasets are collected in a laboratory environment; with static cameras, static backgrounds and calibrated data-capture setups, including Berkeley MHAD and CMU MoCap. Others may be collected outdoors with a controlled clutter free setting, such as Weizmann and KTH. Others are collected within cluttered environments, featuring non-participatory subjects that complicate the scene, such as MSR Action-I and Action-II. Pose based datasets which make use of a depth sensor and the pose estimation technique of extracting the 3D skeleton are often captured in a relatively clutter free scene due to the limitations of the skeletal tracking methodology used.

Daily living

Daily living sets are designed to closely represent the natural world in both the environmental surroundings and the natural style of action classes executed. The Tum Kitchen [212], MSR DA3D [102, 111], MPII Cooking [194], and Rochester AoDL [208] sets are commonly used for the analysis of methodology in the recognition of day-to-day activities. Activities include ‘having a conversation’, ‘phone calls’, ‘laying down’, ‘drinking’ and ‘eating’, but may also include sub-actions within a higher level task, such as ‘setting a table’ or ‘cooking a meal’. The executions may be allowed to occur naturally as in the 50 Salads, MPII Cooking, and MPII Composite datasets; or the observations may be more scripted, such as in the POETICON and the robotic class of the TUM Kitchen set [150, 206, 212]. By understanding the actions and interactions within a daily activity dataset the field is moving towards learning higher-level semantics of human behavior via natural representations.

Surveillance

Surveillance is a domain concerned with detecting and identifying activity within a continuous observation of a scene, often making use of video-based action recognition samples that are taken from a distance, prone to crowding, and contain poor resolution recordings [240–243]. A surveillance domain sequence may contain more frames of empty or redundant information, sporadically interspersed with temporally short regions of interest. Datasets such as UT-Interaction, CASIA, and BEHAVE make use of surveillance style setups to capture emphasized person-person interaction classes such as ‘come together’ and ‘fight’. The CAVIAR, ETISEO, and VIRAT datasets all make use of detailed ground truth annotations to provide information regarding persons and objects within the scene, enabling the evaluation of methods in detecting varies entities and their interactions within a scene for higher semantic understanding of the events. Surveillance problems can often contain an element of individual tracking as well as action and interaction classification [244–246]

Table 2.6: Comparison of dataset domain applications.

	Appearance sets	Pose sets
Domain		
Generic	BIT Interaction, HMDB51, Hollywood, Hollywood-2, IXMAS, JPL, KTH, MSR Action-I, MSR Action-II, MuHAVi, Stanford 40 Actions, UCF101, UCF50, UCF11, ViHASi, Weizmann, WVU MultiView	Berkeley MHAD, CMU MoCap, G3D, G3Di, Hollywood3D, HumanEVA, K3HI, MPI08, MSR Action3D, MSR Gesture3D, SBU Kinect Interaction, UMPM
Daily Living	Drinking/Smoking, MPII Cooking, MPII Composite, Rochester AoDL	50 Salads, CAD120, CAD60, CMU MMAC, LIRIS, MSR DA3D, POETICON, TUM Kitchen, CONVERSE
Surveillance	BEHAVE, CASIA, CAVIAR, ETISEO, UT-Interaction, VIRAT	
Sport	Olympic Sports, UCF Sports	

Sport

The UCF-Sports, [219], and Olympic Sports, [204], datasets are focused explicitly on sports related action examples. These sets contain samples that are collected from various sources of TV and online recordings, providing samples that vary in their recording quality and containing both static and dynamic camera movements. As such these can often be challenging datasets. In both cases the intent of the dataset is to be able to recognize the sport being performed, this can be more challenging than in the case of learning sports related actions, such as in the case of ‘tennis serve’ and ‘boxing’ from some of the generic action datasets. A sport as a high level class can contain numerous action and interaction actions that make up the overall activity and learning a sporting class may require learning vastly different observations that belong to the same class. 3D pose based HAR in the sports domain has few datasets due to the complexity in capturing a large volume in which the activity can be played. The G3Di dataset provides interactions between two people in the context of a sporting game played through a console, however we treat the provided classes as being generic actions rather than true sporting based actions.

2.6.5 Ground Truth Labeling Approaches

Providing consistent and accurate ground truths is a key stage of supervised learning and general evaluation of model validation within data analysis and machine learning. The evaluation of performance is important for developing benchmarks against which to test developed methodologies, aiding in the generation of a metric score that can be used to compare implementations [247–249]. The production of accurate labeling, and what is considered accurate labeling, is an on-going area of research [250–253]. The impact of bias, inaccuracy, and granularity of label are just several lines of study within the dataset analysis community [254–256], with data quality often being a concern in the development of datasets for supervised training. Table 2.7 outlines various ground truths provided with each dataset previously identified, both for spatial ground truths and labeling of action classes.

Class label ground truths and scene annotations of a dataset can provide a clear benchmark for quantifying the performance of a developed methodology. Some datasets provide frame-by-frame labeling of the scene, whilst others label an entire sequence as containing a given class label, see Table 2.7. These annotations allow quantification of results obtained from various methodologies, with predicted class labels and detections being compared against the ground truth. There are numerous method for collecting the ground truth annotations for a given dataset, including manual and machine learning based approaches [257–260]. In some datasets, manual annotation can provide detailed descriptions of the entire scene, with locations and affordances being given to persons and objects within the scene, as can be seen with the ETISEO and HMDB51 datasets. These can be extremely useful when tracking the states of multiple entities within the scene, or for the understanding of a high level abstracted class. An issue with manual labeling is that annotation can be highly subjective and may introduce observer bias in relation to numerous key components of the data [261, 262]. It is common for the identification and labeling of start and end frames for a given class to vary between observers [247], and for more subtle class labels to introduce variance in annotation quality, with the crossover period between two action classes being an area for high variance in observer labeling. Such differences in observer behavior requires stricter objective criterion to gain consistent ground truths across different observers. Meta-context based annotations can be combined into the data labeling approach to rapidly provide ground truths to large datasets, e.g. the Hollywood and Hollywood-2 datasets are partially annotated by learning textual descriptions within the film's scripts. The use of meta-sources gives an assumption that such external data is accurate and useful for the required task, and is by no means free of bias; movie scripts are often produced ahead of an action being performed and may be subjectively interpreted by the individual carrying out the task. The use of automated ground truth annotation methods may require subsequent manual verification to minimize the incorrect annotations or refine boundary edges; but again this may be a subjective task, depending on the problem. The simplest form of ground truth labeling provided by HAR datasets is by attributing the entire sequence to a single specific label for recognition tasks, acknowledging that a given action occurs at some point within the observation, as is the case with CASIA, CMU MMAC, MSR Action3D, and many more. Having simplistic whole sequence labeling can make it hard to use such datasets for detection and segmentation purposes, as evaluating the beginning and end frames of an action can be problematic to determine manually. For action recognition purposes, the learning of background or non class-specific frames from a sequence may also provide some level of noise to the generalization of features for that class.

Spatial truth can be provided by explicitly locating the subjects and objects within the environment or by highlighting regions of interest in which the subject, object or event resides by using bounding boxes or silhouette masks. Calibrated ground truth methods can be used to determine the spatial locations of the subjects within a scene, often using motion capture suits and markers to explicitly track the body through a capture volume, providing either a raw point cloud or the predicted skeletal frame of the body. The accuracy of motion capture systems can vary from method to method, however the resolution accuracy is often within a range of a few millimeters, providing superior body tracking than using machine learning based pose extraction. Marker based motion capture systems, such as those used in CMU MoCap and Berkeley MHAD, require the application of each marker to the individual at certain predetermined locations, and variation in placement of the markers on the body from sequence to sequence can introduce small errors in obtaining truly explicit spatial truths. The use of depth maps to extract an estimated 3D pose of the subject in the scene has become a prominent inclusion in depth based HAR datasets such as MSR Action3D, K3HI, SBU Kinect Interaction, CAD120, and CAD60. The observation is fed into a skeleton extractor, such as the OpenNI, Microsoft Kinect SDK softwares, or custom methods [263–265], in which a subject is located and a human skeleton model is fitted, predicting the 3D coordinates for a number of joints. Although an approximation of true 3D spatial orientation of the joints, depth sensors and joint tracking has been shown to be relatively accurate in the tracking of humans [99, 101]. The use of bounding boxes to describe regions of interest in a scene are common within appearance based datasets, such as BEHAVE, CAVIAR, ETISEO and MSR Action, especially those that consider person-object interactions or belong to the surveillance domain. They simply provide an area of focus that contains relevant annotated information, such as object and subject location. The use of silhouette masks also provides a region of interest, whilst simultaneously removing external and internal appearance information, representing the subject as a binary classification as either belonging to the background or foreground. These regions of interest can also be utilized to validate action detection and localization methodologies, removing the unwanted information from the overall observation.

As has been described above, the ability to reliably and accurately annotate a dataset is a complex process, where methods have been developed to assist in streamlining the task or attempt in utilizing automated labeling systems. The concept of identifying boundaries of an action class can be complex and highly subjective, whilst classification of more subtle behaviors can also be problematic. Annotation of the “ground truth” upon which supervised techniques are trained against, and to which model performances are evaluated, requires careful consideration and design. The goal of an underlying task will drive how a collected dataset should be annotated and used, and this is reflected in the spread of approaches identified in Table 2.7.

2.6.6 Viewpoints

Camera based methods can also make use of various viewpoints, from single camera to multi-camera simultaneous viewpoint capture [266–268]. Viewpoints can also differ greatly, capturing events from roughly a parallel plane with the ground, elevated above head height, or from an almost top-down viewpoint. Often events are captured from a viewpoint that is roughly parallel to the ground, producing observations that are almost representative of a human-eye view of the event, examples can be found in MSR Action3D, K3HI, and CMU MoCap. A summary of dataset viewpoint representation is given in Table 2.8. Sets such as BEHAVE, UT Interaction and CASIA contain events recorded from an elevated angle; these viewpoints are common within the surveillance domain due to the positioning of surveillance cameras for capturing a large scene at once. Recently there has been work towards the recognition of actions from a first person perspective [269–272], with data captured from the viewpoint of the observer [184, 273, 274]. This field is often working towards the understanding of interactions by robots for the purpose of human-robot interaction. Such a viewpoint is believed to provide more meaningful information when the observer has an active role in the interaction rather than simply observing a scene, as is the case in human-robotics interactions. There are also datasets that attempt to capture simultaneous multi-camera views of an event for the purpose of evaluating supposedly pose-invariant methodologies. Sets such as WVU MultiView, Berkeley MHAD and TUM Kitchen all contain numerous cameras located in differing positions capturing the same scene. Depth based data, such as tracked skeletons and motion capture marker coordinates, can be orientated arbitrarily about its three axes to develop multi-view methodology, with some pose alignment used to reduce the effect of orientation discrepancies, [41]. However this is dependent upon accurate pose estimation in order to provide data with confident tracking. Due to the nature of extracting pose estimation from depth based methods there are limited numbers of datasets that utilize multiple depth sensors; however Berkeley MHAD provides multiple Kinect recordings alongside its vast number of appearance views, with the sensors located in positions from which the infrared sensors are not causing occlusions. Advances have also been made in providing view-invariant approaches to pose estimation and action understanding [275–279].

2.6.7 Use in Community

Popularity of a dataset within the community can be difficult to evaluate, however here we attempt to identify the number of citations that are made to the dataset's description publication via Google Scholar. Using this count as a measure of how well adopted a given dataset has become, we rank each set in Table 2.9. Note that older sets can often show higher citation due in part to their steady accumulation of references over time. Similarly, the number of citations made may not explicitly reflect the use of dataset as a benchmark, as often the datasets are published in parallel with a novel methodology that may accrue its own citations. It can be seen from Table 2.9 that the pose based datasets show considerably fewer citations, most likely due to the relative age of the rapidly growing field and the developments in appearance based deep learning methods and their use in deep learning based approaches which utilize image based inputs, such as the CNNs outlined in Section 2.7 and Chapter 3.

Table 2.7: Description of ground truth labeling provided by human action recognition datasets.

Name	Spatial ground truth labels	Class ground truth labels
50 Salads	-	Frame labeling
BEHAVE	Bounding boxes	Frame annotation
Berkley MHAD	MoCap tracking	File labeling
BIT Interaction	-	File labeling
CAD120	Extracted skeleton, bounding boxes	Frame labeling
CAD60	Extracted skeleton	File labeling
CASIA	-	File labeling
CAVIAR	Bounding box	Frame labeling
CMU MMAC	MoCap tracking	File labeling
CMU MoCap	MoCap tracking	File labeling
CONVERSE	Extracted skeleton	Frame labeling
Drinking/Smoking	Bounding box	Frame labeling
ETISEO	Bounding box	Frame labeling including calibration parameters, scene descriptions, object affordance
G3D	Extracted skeleton	File labeling
G3Di	Extracted skeleton	File labeling
HMDB51	Bounding boxes	File labeling including view, camera motion, visible body parts, quality, and number of subjects
Hollywood	-	Frame labeling
Hollywood-2	-	Frame labeling
Hollywood 3D	-	File labeling
HumanEVA-I	MoCap tracking	File labeling
HumanEVA-II	MoCap tracking	File labeling
IXMAS	Silhouette masks	Frame labeling
JPL	-	Frame labeling
K3HI	Extracted skeleton	File labeling
KTH	-	Frame labeling including scenario labeling
LIRIS	Bounding boxes	Frame labeling
MPI08	MoCap tracking and 3D scan	File labeling
MPII Cooking	-	Frame labeling
MPII Composite	-	Frame labeling
MSR Action-I	Bounding box	Frame labeling
MSR Action-II	Bounding box	Frame labeling
MSR Action3D	Extracted skeleton	File labeling
MSR DA3D	Extracted skeleton	File labeling
MSR Gesture3D	Extracted skeleton	File labeling
MuHAVi	Silhouette masks	Frame labeling
Olympic Sports	-	File labeling
POETICON	MoCap tracking	File labeling
Rochester AoDL	-	File labeling
SBU Kinect Interaction	Extracted skeleton	File labeling
Stanford 40 Actions	Bounding box	File labeling
TUM Kitchen	Markerless MoCap tracking	Frame labeling including body trunk, left arm, right arm, and object affordance
UCF101	-	Frame labeling
UCF11	-	Frame labeling
UCF50	-	Frame labeling
UCF Sport	-	File labeling
UMPM	MoCap tracking	File labeling
UT Interaction	Bounding box	Frame labeling
ViHASi	Silhouette masks	File labeling
VIRAT	Bounding box	Frame labeling including object affordance
Weizmann	Silhouette masks	File labeling
WVU MultiView-I	-	File labeling
WVU MultiView-II	-	File labeling

Table 2.8: Comparison of viewpoints and scenario constraint in human action recognition datasets.

	Appearance sets	Pose sets
Simultaneous Views		
Monocular	BIT Interaction, Drinking/Smoking, HMDB51, Hollywood, Hollywood-2, JPL, KTH, MPII Cooking, MPII Composite, MSR Action-I, MSR Action-II, Olympic Sports, Rochester AoDL, Stanford 40 Actions, UCF101, UCF11, UCF50, UCF Sport, UT Interaction, Weizmann	50 Salads, CMU MoCap, G3D, G3Di, Hollywood3D, K3HI, LIRIS, MSR Action3D, MSR DA3D, MSR Gesture3D, SBU Kinect, UMPM
Multi-view	BEHAVE, CASIA, CAVIAR, ETISEO, IXMAS, MuHAVi, TUM Kitchen, ViHASi, WVU MultiView-I, WVU MultiView-II	Berkeley MHAD, CAD120, CAD60, CMU MMAC, HumanEVA-I, HumanEVA-II, MPI08, POETICON, CONVERSE
Environment		
Interior Natural	CAVIAR, Drinking/Smoking, HMDB51, Hollywood, Hollywood-2, JPL, MuHAVi, Olympic Sports, Stanford 40 Actions, UCF101, UCF11, UCF50	Hollywood3D
Interior Controlled	IXMAS,MPII Cooking, MPII Composite, Rochester AoDL, ViHASi, WVU MultiView-I, WVU MultiView-II	50 Salads, Berkeley MHAD, CAD120, CAD60, CMU MMAC, CMU MoCap, G3D, G3Di, HumanEva-I, HumanEva-II, K3HI, LIRIS, MPI08, MSR DA3D, MSR Gesture3D, POETICON, SBU Kinect Interaction, TUM Kitchen, UMPM, CONVERSE
Exterior Natural	BEHAVE, BIT Interaction, Drinking/Smoking, ETISEO, HMDB51, Hollywood, Hollywood-2, MSR Action-I, MSR Action-II, Olympic Sports, Stanford 40 Actions, UCF101, UCF11, UCF50, UT Interaction, VIRAT	Hollywood3D
Exterior Controlled	BIT Interaction, KTH, Weizmann	

Table 2.9: Citation count for dataset description paper. Correct at time of submission. Note: CMU MoCap has no attributed publication

Name	Year of Publication	Total Citations
Appearance		
KTH	2004	2013
Hollywood	2008	1772
Weizmann	2005	1182
UCF11	2009	602
IXMAS	2006	590
UCF Sport	2008	584
Hollywood-2	2009	580
Drinking/Smoking	2007	327
UT Interaction	2009	303
Olympic Sports	2010	283
Rochester AoDL	2009	266
HMDB51	2011	265
MSR Action-I	2009	189
UCF101	2012	155
VIRAT	2011	144
Stanford 40 Actions	2011	137
UCF50	2013	131
ETISEO	2007	103
CAVIAR	2004	90
MSR Action-II	2011	82
MPII Cooking	2012	67
MuHAVi	2010	60
MPI08	2010	48
JPL	2013	38
ViHASi	2008	33
BEHAVE	2010	33
MPII Composite	2012	32
BIT Interaction	2012	19
CASIA	2009	12
WVU MultiView	2011	0
Pose		
HumanEVA	2010	373
MSR Action3D	2010	333
MSR DA3D	2012	311
CAD120	2012	159
TUM Kitchen	2009	117
CAD60	2013	81
MSR Gesture3D	2012	75
Berkeley MHAD	2013	50
CMU MMAC	2008	48
SBU Kinect Interaction	2012	33
Hollywood3D	2013	32
G3D	2012	28
POETICON	2011	8
UMPM	2011	7
50 Salads	2013	6
LIRIS	2014	5
CONVERSE	2015	4
K3HI	2013	2
G3Di	2014	0
CMU MoCap	-	-

2.7 The Use of Deep Learning in Human Action Recognition

The use of deep learning within human action recognition has grown in recent years [280–282], with the development of CNN architectures utilizing representation learning to identify spatial, temporal, and spatio-temporal features from the large number of appearance based datasets available to the community [26, 95]. In [31], Baccouche *et al.* present a two stage model for representation learning in human action recognition, in which a standard CNN learns spatio-temporal information from video, with a following Recurrent Neural Network used to classify the observed sequences based on the learned appearance features. Ji *et al.* [32] and Tran *et al.* [283] introduce 3D CNNs to capture motion information from video sequences, providing a convolution across time for extracting spatio-temporal features. A factorized approach of applying 2D spatial kernels, with subsequent 1D temporal kernels was presented in [284], providing a network which focuses on spatial information in lower layers of the network and temporal features in the deeper layers. This approach reduces the amount of parameters within the network significantly, and reduces the amount of data needed to separate the conflation of space and time. The separation of space and time has been utilized in several methods, such as producing a multi-modality CNN which has a branch dedicated to the appearance information of singular frames, and the temporal information across multiple frames [96,285,286], Figure 2.5. Understanding of temporal information via recurrent nets has been popular, with numerous papers exploring the use of Long Short-Term Memory (LSTM) modules to learn over appearance based motion [94, 287–289].

In addition to the development of using representation learning in appearance based action recognition, the use and extraction of pose-based features has also been explored via deep learning approaches [290–293]. Pose estimation is used in [294] to identify key regions of focal interest on the human body, extracting localized patches around semantically meaningful body parts. Appearance and flow images are then used to train a multi-branch CNN for action recognition. The method does not consider pose as an input feature to the network, however it exploits pose as a pre-processing sampling step and utilizes appearance based information as the feature modality. Rahmani *et al.* apply a 3D human model to motion capture data in order to create standardized appearance information to develop feature descriptors [295]. This approach expands the motion capture stick figure representation to a fuller representation of the human body, in essence embedding the motion capture information in an appearance domain. The use of purely pose-based information for deep learning in action recognition is more lim-

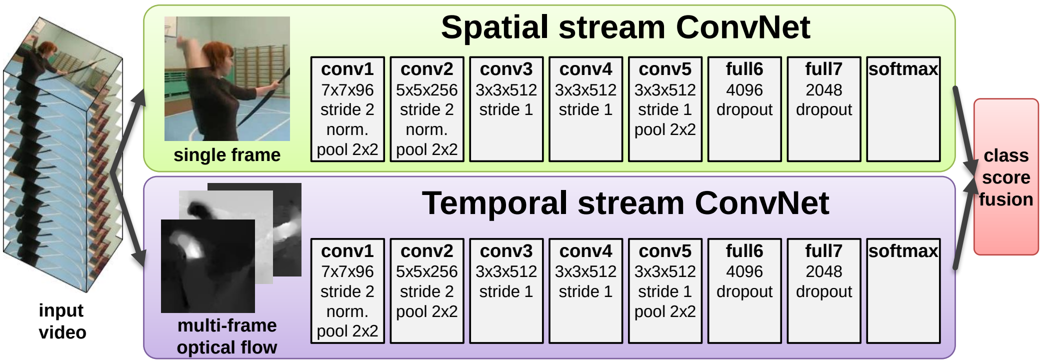
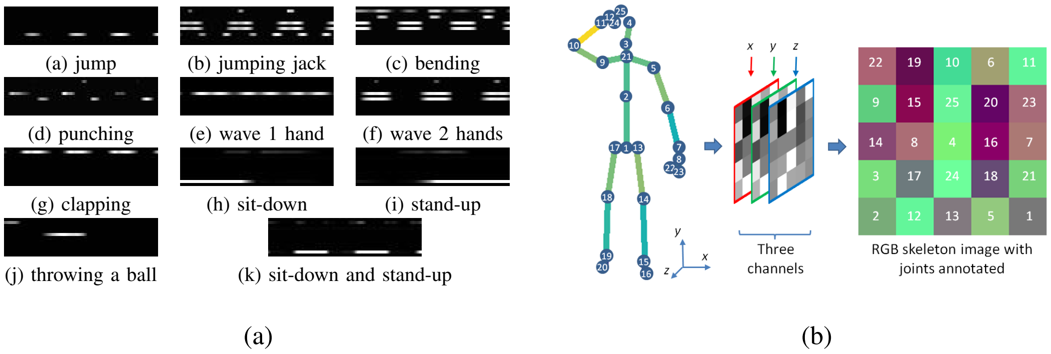


Figure 2.4: Pipeline of the Two-Stream Convolutional Neural Network as presented by [96]. Two distinct branches of the network learn convolutional filters from input appearance and temporal information before prediction fusion provides a final output class label.

ited, with Ijjina and Mohan suggesting the learning of motion capture information by CNNs on a spatial embedding of joint features [29, 30]. The presented approaches use hand-crafted feature descriptors on a hand-tuned set of tracked joints, forcing the feature vectors into a 2D image using arbitrary vector ordering. The methods present strong results on the evaluation dataset, however the use of hand-crafted features and heavy tuning of joints of interest may not generalize well to further problems, classes or observations. A similar embedding of the human skeleton model is presented by “Skepxels” in [296], in which joints are allocated to pixels within a 2D image in order to utilize CNN operators. In order to handle the arbitrary nature in which joints can be mapped to pixels within an image ($\sim 1.55 \times 10^{25}$ permutations for a skeleton with 25 joints, or $J!$ where J is the number of joints in the skeleton), a number of different permutations of the joint ordering are selected for passing to the CNN. The use of 1D convolutional kernels in these approaches learn temporal information in the horizontal direction of the image, and cross-feature information from the image columns, where the ordering of rows within the image enforces an assumption of localization between features. Such methods show the embedding of irregular domains into a regular space in order to utilize the CNN approaches found in image recognition literature, Figure 2.5. Using deep learning on the skeletal model representation of pose is less common but growing. Zhu *et al.* , [28], present a method for learning temporal features from motion capture in which LSTM networks take joint information as input. Huang *et al.* , [297], also utilize skeletal information, generating a ‘LieNet’ architecture which learn Lie representations of the action. The use of hierarchical structuring



22	19	10	6	11
9	15	25	20	23
14	8	4	16	7
3	17	24	18	21
2	12	13	5	1

Figure 2.5: Embedding motion capture information into an image to utilize Convolutional Neural Network operators defined on the grid domain. Using: a) vertical concatenation of feature vectors [29] b) Skepxels [296]. Images are then fed into a CNN architecture to learn features. Images used from original papers.

of the human skeleton is presented in [298], in which the skeletal model is decomposed into related anatomical components and an LSTM approach is taken to mine features from each. The utilization of recurrent nets to model human skeletal motion is extensive, however the recurrent models often focus on joints individually, omitting the explicit spatial relationships between the joints which are linked by bones [299], or require more multi-dimensional LSTM network structuring to learn both the spatial and temporal information [300].

As can be seen from the current state of the literature, the use of deep learning in human action recognition is well explored for appearance based information. Using deep learning approaches optimized from image and video processing is well suited to such a task, learning spatio-temporal features from observations in order to identify discriminative features for the classification of behaviors. Deep learning approaches applied to pose information are more diverse. Often pose is used as a method for sampling within an appearance space, utilizing standard CNN operations optimized to tasks in the image domain. Some studies have taken skeletal information and forced an embedding in the image domain, whilst others have ignored all inherent spatial information between the joints; instead relying on manifold embedding operations, or optimization of fully connected weightings between features. The generation of spatially localized descriptors on the irregular topology of the human skeleton precludes its use as raw input into standard convolution neural network architectures, however such spatial relationships between joints on the human body can be informative for the classification of behaviors. Although these methods do provide promising results, the forced embedding of spatial domains into a image based representation purely for the ability to utilize standard convolutional operations is an approach which raises several questions. What impact does the order of embedded pixels have on performance, and are methods robust to the choice of embedding? Can feature maps produced by these methods be reliably analyzed in their original spatial topology to find features that may be beneficial to domain experts? Is the choice of embedding method suitable for all applications, or are specific embedding approaches required? Further work is required to utilize the localized features on the skeletal model, without the need for spatial embeddings, and in Chapter 5 we present a method for learning features on an irregular topology, with Chapter 7 applying the method to the problem of Human Action Recognition.

2.8 Summary

As has been shown, the field of human action recognition contains a number of problems under constant study. As with other fields, the use of certain methods fall in and out of favor and the available data follows such trends. The community have a wealth of datasets to draw upon in order to evaluate the benefits and drawbacks of a given method; however further advances are required to explore our understanding of more complex and subtle interactions. Appearance based information, or more specifically observations from domains that reside on regular Cartesian grids such as images and videos, are once again being heavily explored due to the gains witnessed in the deep learning community. As a result, image and video datasets are showing growth in all areas of our evaluation, including their representation of complex scenarios. Pose based datasets are less common, and their coverage of more complex events requires development. Dataset sizes continue to grow in an attempt to reliably train and evaluate deep learning architectures, with large image and video sets becoming popular in the community. The pose-based sets still require expansions on similar scales. The introduction of the NTU RGB-D dataset may signal the begging of the development of large-scale sets that capture skeletal pose of observed actions.

As can be seen from the previous sections, datasets that are able to capture human action using appearance based modalities, such as RGB videos, have developed from representing non-realistic emphasized actions to considering more complex interactions between individuals and their surrounding environment. The field has moved from actions which are easily distinguishable in the visual domain, e.g. ‘waving’ and ‘jumping’, to those of interactions, although still recognizable, e.g. ‘hug’ and ‘kiss’ [72, 301]. Due to the availability of these datasets many methods have been produced and evaluated for the purpose of action recognition and detection, including the use of Scale Invariant Feature Transform (SIFT) [302], temporal Harris corner features [24] or STIPs [174]. Meanwhile, the pose-based methodologies which have grown rapidly over the past decade show far fewer publicly available datasets which consider the problem of person-person interactions, with most considering either emphasized actions or interactions. Representation learning on appearance based information is a hot topic within the current HAR community, however we propose a feature representation learning method on the irregular spatial topology of the human skeleton.

In the following chapters we will introduce the deep learning methodologies that have become so prominent in the machine learning field, explore the use of representation learning on features in irregular domains such as the human skeleton model, and explore the benefits of generalizing current spatial feature mining algorithms to domains beyond Cartesian grid systems.

Chapter 3

Deep Learning

Contents

3.1 Introduction

The growth of machine learning as a tool for analyzing and utilizing data has exploded over recent decades, introducing the use of a wide range of methodologies to exploit information from observed data [9, 303, 304]. Applications vary from recommendations provided on retail websites, to object detection and semantic scene segmentation. There are now machine learning techniques to translate text to natural speech [305], translate between languages [306], and even translate pictures into descriptive sentences [307]. Common machine learning techniques have often been reliant on the production of hand-crafted features over the use of raw data, with informative features carefully developed through domain-specific knowledge to generalize information across the distribution of observations. A tuned feature extractor would allow an observation to be represented as a feature vector that is utilized to train some model for the purpose at hand, often some classification or regression problem. The classical machine learning method would then be trained to recognize patterns within this feature space embedding. Such methodologies are sensitive to the feature descriptors used, with focus needed to generalize feature extractors to accommodate intra-class variance and still provide suitable discriminative power between classes.

Feature descriptors can often be domain specific, and their selection can have a drastic impact on the performance of machine learning implementations in a given application. Deciding upon a feature embedding is often not a trivial task, and can require significant user input in order to select suitable descriptors. Some features generalize well to image domain problems, developed from an understanding of image processing principles and their generalization to suitable a variety of tasks; such as Histogram of Oriented Gradients (HoG), Space-Time Interest Point (STIP)s, and Scale Invariant Feature Transform (SIFT). Others have been developed to make use of relationships between semantic relationships between 3D points, such as the joint-based features for Human Action Recognition (HAR) described in Chapter 2. Such feature descriptors can often have hyper-parameters that require optimization for a given task, which may again impact on the observed performance in real-world applications.

One alternative to developing such hand-crafted feature descriptors for machine learning applications is to enable a model to learn their own descriptor set for the application domain from the raw input data [308]. This technique, dubbed ‘representation learning’, has exploded in the last decade with the growth of neural network and deep learning methodologies [309].

Representation learning algorithms bypass the requirement to hand-craft a feature embedding for the target application, instead providing the model with the tools to construct its own feature extractor and a large enough training corpora to develop generalized features for the observed distribution. One such family of representation learning approaches is that of deep learning. Deep learning is a hierarchical representation learning methodology, and involves the non-linear mixing of raw input features into successively higher level and more abstracted representations. Subsequent layers of non-linear mixing will allow more complex functions to be learned and modeled, with each layer learning a function of the underlying input feature distribution based on the observed data. These representations are then learned via a parameter optimization scheme, rather than being hand-crafted features which can require considerable human influence.

This chapter presents an insight into the development of neural network and deep learning algorithms as methods of representation learning. An introduction to neural networks and the use of feature mixing with no spatial information seen in fully connected networks is discussed in Section 3.2. The localization of features on regular Cartesian grids exhibited by convolutional neural networks is introduced in Section 3.3, discussing motivations, benefits and shortcomings of existing methods. Key developments and network architecture construction methods will be discussed in Section 3.4, with seminal applications providing insight into the use of such representation learning schemes in real world applications. Highlighting assumptions made by methods focused on the regular Cartesian domain, we will explore the use of deep learning techniques in domains that exhibit irregular spatial topologies in Section 3.5, identifying the accompanying problems and solutions for utilizing deep learning on such application domains.

3.2 Neural Networks

The principle concept behind the development of deep learning methods is the delegation of learning a representation that approximates some function to the optimization of parameters denoting the weighting of various input features in relation to one another. Taking inspiration from neurobiology, the summation of input stimuli and the resulting activation response of the human synaptic neurons in the brain was the influence for the precursor of the Neural Network (NN), the artificial neuron or ‘perceptron’. In the following section we look to discuss the development of the field of deep learning and the task of representation learning. We will identify the field's development from its origins as a non-linear feature weighting approach, to the utilization of constraints on learning schemes for more specific tasks. We consider the use of fully connected networks with no explicit spatial relationships between features, to the addition of a localized receptive filtering constraint within convolutional networks. We will discuss the suitability of a regularly spaced kernel based convolutional neural network to the mining of features in domains that do not exhibit such a topology naturally. We explore advances in recent literature and the direction taken in relaxing the spatial topology constraint in order to apply such localized filtering on irregular spatial domains.

3.2.1 The Perceptron and Feedforward Networks

In order to learn a representation between an input and a target output [310, 311] introduce a perceptron unit, a function unit which computes a weighted summation of the inputs to a cell, adds a bias value and passes it to a non-linear activation function, see Figure 3.1. The output of the activation function is the overall output from the perceptron after activation, describing a response to a given stimuli set based on the current receptive state of the perceptron, mapping the input feature space to a new feature representation. An individual neuron can then be incorporated into a multi-layer network, an Multilayer Perceptron (MLP), in which the output from one neuron becomes the input to a subsequent neuron in the following layer of the network. This notion produces a sequential network of multiple layers, each learning weightings on the outputs of previous layers, resulting in ability to model increasingly complex function spaces. A network's depth is represented by the number of layers it is composed of, and a layer's width is defined as the number of neurons it holds.

A given neuron, i, in layer l of a MLP is connected to neuron j in layer $l+1$. Neuron j produces an activation output

$$a_j^{l+1} = f(b_j^{l+1} + \sum_{i=1}^{I} x_i^l w_{ij}^l) \tag{3.1}$$

where the activation response a_j^{l+1} is a non-linear function of the weighted summation of inputs. I inputs to the current neuron are weighted with their corresponding weight parameter w_{ij}^l. These weighted inputs are summed and a bias value for that neuron b_j^{l+1} is then added to shift the activation function space, increasing non-linearity between neurons in a layer. Subsequent layers utilize the output of a neuron as their inputs, learning high-level representations by having a_j^{l+1} as one of the I inputs to the next layer. Increasing the neuron count in a given layer

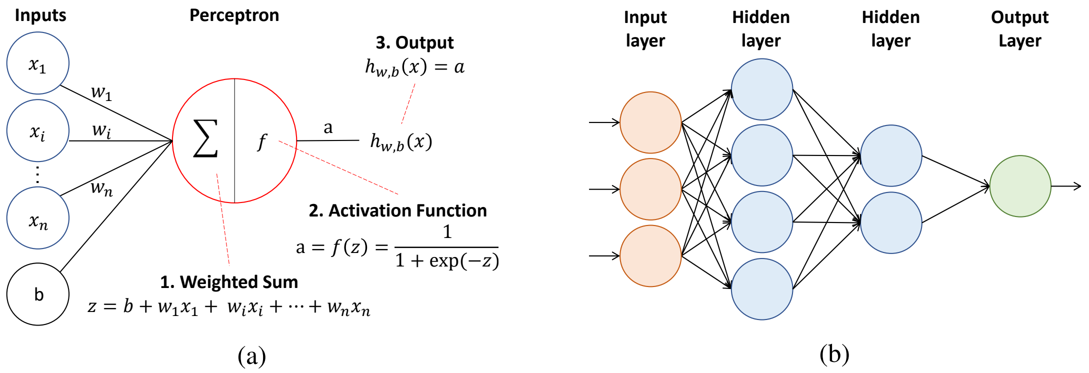


Figure 3.1: (a) Perceptron function behavior. The output of the neuron, a is the activation function response to the weighted sum of the inputs x plus a neuron bias b. By learning the weights w for each input, we can learn a new representation embedding. (b) Multiple layer network.

increases the possible representation capacity for that current set of features, but struggles to enable higher level generalization. Increasing the depth of a network develops higher levels of representation; however networks that are too shallow risk underfitting the data, whilst naively going too deep can lead to overfitting [12,13]. The overall architecture optimization of a neural network model is an open research topic, but some steps have been taken towards architecture design, hyper-parameter search and post-training [312–315].

3.2.2 Backpropagation and Weight Optimization

In order to learn the function space from the observed samples, MLP or feedforward network methods optimize their weights via the process of backpropagation. In supervised learning settings, a set of training instances are fed forward through the network and the predicted output is compared to the expected output [316,317]. The loss between the output and ground truth is passed backwards through the network to calculate the derivative in regards to the current set of network weights. Weights are then tuned to minimize this error by an optimization scheme, such as stochastic gradient descent. Given a set of t training observations, with each observation comprised of a $\mathbb{R}^n$ input feature vector and a target y_j, where j corresponds to a given class, we can perform a forward pass of the sample through the network, yielding a predicted output vector a^L, with a_j^L being the output activation for each neuron in the output layer. An application specific loss function is then able to calculate the difference in the expected target value and the predicted output vector. Such a loss is the cross-entropy for classification tasks, given by

$$E = -\frac{1}{t}\sum_x \sum_j y_j \ln a_j^L + (1 - y_j)\ln(1 - a_j^L) \tag{3.2}$$

where t is the total number of training samples, and x is a given training input. The loss, E, between the input sample and the output can then be fed backwards through the network. For each neuron we can obtain the partial derivative in respect to a given neuron's inputs, and weightings associated with those inputs, w;

$$\Delta w_{ij} = -\alpha \frac{\partial E}{\partial w_{ij}} \tag{3.3}$$

This allows the network to utilize a weight update scheme such as gradient descent, with a given learning rate α, to adjust the feature weighting with $w_{ij} = w_{ij} + \Delta w_{ij}$. Computing this for all neurons and weightings in the network allows weights to be optimized for the task at hand, for all layers and interactions between features in the network. Similarly the partial derivative in relation to the neuron bias, $\partial E/\partial b$, can be computed to affect the impact a given neuron has within the network. Various objective loss functions can be used to optimize the network training scheme, dependent on the task at hand [318]. The incorporation of the auto-differentiation approach to derivative calculation in regards to applying the chain rule to functions is now common within deep learning toolboxes, allowing layers to be implemented as a feed-forward operation and the derivative in respect to the weights and biases are obtained [319,320]

As networks became deeper, attempting to learn high level non-linear representations of the input function space, the backpropagation algorithm for calculating the derivative of the network parameters becomes less effective. As errors are passed back through the network and derivatives are obtained for the neurons in each layer of the network it was observed that gradients identified for use in stochastic gradient descent updates were rapidly diminishing [321]. The Hyperbolic Tangent (Tanh) and sigmoid activation functions required for creating the non-linear mappings would present a derivative trending quickly towards 0 at their extremities. Neurons in such a layer would quickly saturate, with ever decreasing gradients and ever-marginal updates. This would feed backwards through the network layers with backpropagation and lead to shallow layers in a network making little optimization gain during training, impacting on final model performance [322]. To counter the effect a number of strategies were adopted. The usage of Rectified Linear Unit (ReLU) style activations over sigmoid and Tanh functions were preferred, providing a stable derivative for all values over 0. Regularization, such as batch normalization [323], dropout layers [324], and weight initialization schemes were developed to aid the generalization of features learned during training [325,326]. Parameter optimization schemes were also developed to extend beyond that of stochastic gradient descent, [327]. Mini-batch gradient descent, [328], allows stable convergence by performing a gradient step over a small batch of input observations, while momentum allows faster convergence and a reduced oscillation around ravines of local minima. Recently more complex gradient-based optimizers have been proposed, including Adagrad [329], Adadelta [330], and Adam [331], which incorporate parameters for learning rate adaptation and thus impact on the stability of the gradient steps that are made during weight updates.

3.2.3 Activation Functions

The application of an activation function allows for networks to represent more complex non-linear function spaces, overcoming the limitations of linear regression models. The sigmoid function was commonly used for NNs, mapping the input x between 0 and 1 by

$$f(z) = \frac{1}{1+exp(-z)} \tag{3.4}$$

This simple activation function allowed more complex problems to be modeled by a network, however the saturation of the sigmoid function and vanishing gradient problem lead to the rise of the Tanh and ReLU functions. Tanh provides a zero-centered activation, resolving the saturation around 0 by

$$f(z) = \frac{1-exp(-2z)}{1+exp(-2z)} \tag{3.5}$$

Although it was now preferred over the sigmoid activation, the Tanh function still suffered from the vanishing gradient problem. In recent years the ReLU activation mapping, has provided improved convergence of model training over previous activations [145,332]. ReLU combines a linear activation with a thresholding at 0, removing information that is weighted down by the weight optimization scheme.

$$f(z) = max(0,z) \tag{3.6}$$

Such an activation brings us back to the similarities with neuroscience; in which the biological neuron will only fire in response to certain strength of input stimuli, and upon firing will give a proportional output to connected neurons. One downside to the standard ReLU activation is that a 0 threshold of signals can push weight optimization to result in neurons which never fire, termed ‘dead neurons’, rendering such neurons useless to the overall network. To avoid this case, various flavors of ReLU have been introduced, including Leaky ReLU, Parametric

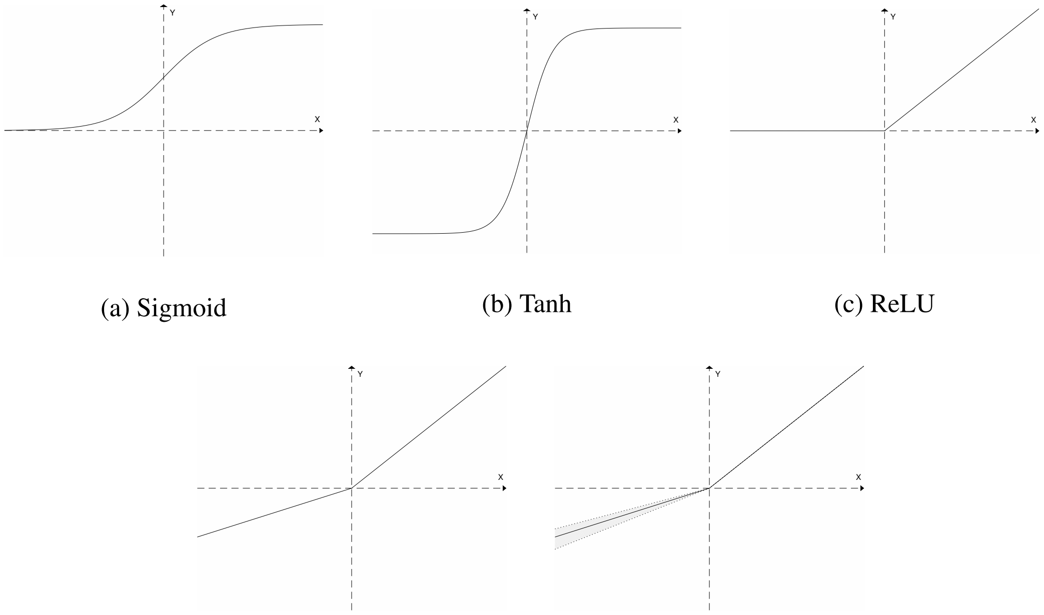
ReLU (PReLU), and Randomized Leaky ReLU. Leaky ReLU incorporates some leaking of signals below 0 by multiplying these signals with some leak factor, allowing some information from down-weighted signals to pass to the next layer

$$f(z) = \begin{cases} z, & \text{if } z \geq 1 \\ \frac{z}{\alpha}, & \text{otherwise} \end{cases} \tag{3.7}$$

In Leaky ReLU α is a constant, whereas PReLU takes α from hyper-parameter to trainable parameter, the same α leak factor is applied to all channels for an input observation. Randomized Leaky ReLU, [333], introduces random leak factors for each input channel, sampled from a uniform distribution, these random leak factors are fixed at testing time, averaging the leak alphas to obtain a deterministic leak factor. ReLU has been shown be favorable in providing a non-linearity in the mapping of the input space to a representation space for classification. The effect of differing ReLU variants on output activation can be seen in Figure 3.2. [334] demonstrated that untrained, randomly initialized weights with a ReLU activation can provide sufficient information propagation in a feedforward network, with [325] showing that given further training data it is beneficial to refine these randomly initialized weightings for the task at hand. These findings further added to the utilization of rectified linear activations over those of sigmoid and Tanh, with ReLU activation being commonly found in modern deep learning implementations.

3.2.4 Regularization

The development of deeper and more complex network architectures, combined with non-linear activation functions such as those presented in 3.2.3, have resulted in complex changes in the data distributions within a network as information is passed forward through the layers. Such internal covariate shift within a network can result in drastic performance degradation, including vanishing gradients, dead neurons and poor weight optimization [323]. Over recent years numerous methods have been presented to reduce the internal covariate shift within a network to allow stable training of increasingly larger networks; including various activation functions [145, 332, 333, 335], weight initialization schemes [325, 326], batch normalization [323], and dropout layers [324]. Batch normalization, [323], presents a method of realigning the covariance within a layer by providing a normalization scheme based on the observed batches, updating shifting and scale parameters. During training the means and variances of an input to a layer are fixed relative to the current batch and normalization parameters, with the pa-



(a) Sigmoid (b) Tanh (c) ReLU

(d) PReLU (e) RReLU

Figure 3.2: Examples of common activation functions in deep learning.

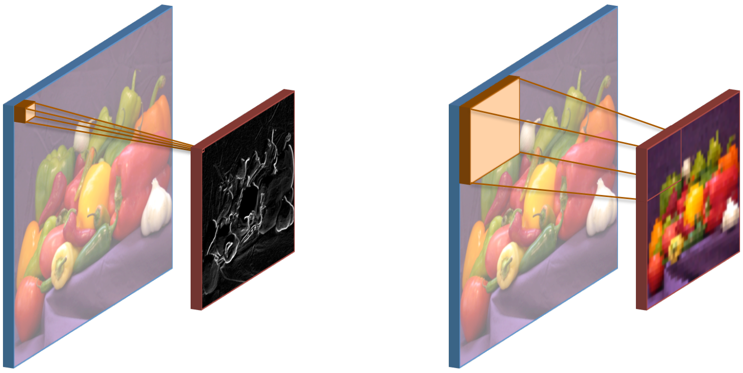
rameters being updated via backpropagation over numerous mini-batches. At test time these parameters are fixed, with new observations being normalized against the learned parameters. Such normalization of layer inputs aids stability during training, allowing increased network learning rates and avoiding gradient explosion, and also enables more generalized models to be produced. In order to reduce the chance of overfitting in large networks, Srivastava [324] suggested to randomly drop neurons from within a layer during training, pushing the network to further generalize learned representations in an attempt to stifle co-adaptation of neurons. A dropout probability hyper-parameter denotes the chance with which any given neuron within a layer will be evaluated during training time, but at testing time the probability is set to 1, evaluating all neurons in the network. By introducing noise into the training process the technique effectively strengthens neurons in their response to outputs from previous layers, improving the generalization of learned features. Such regularization techniques have been used to help stabilize learning within neural network approaches and increase performances on numerous application domains. This use of model and layer regularization has seen continued usage in deep learning approaches, especially in networks developed to learn generalized locally informative features, such as the convolutional neural networks.

3.3 Convolutional Neural Networks

The growth of deep learning algorithms for representation learning saw a large resurgence with the advent of the Convolutional Neural Network (CNN) [11] and the development of efficient training schemes. So called ConvNets were designed to make use of the regular spatial domain of multidimensional arrays, such as those seen in the 2D and 3D Cartesian grids, in order to learn localized information with some translation invariance. This approach is in contrast to standard NN architectures, which do not enforce a spatial relationship between the input feature, instead relying on a learned weighting between dimensions within the input feature space. CNNs are able to utilize locally connected kernel based convolutions to extract localized features from regions on the grid. Such a formulation allows weight sharing across the grid space, providing a smaller parameter set to optimize via backpropagation CNNs developed two key layer types for incorporation into the deep learning zoo; the convolutional layer and the pooling layer, which can be seen in Figure 3.3.

3.3.1 Convolutional layers

Convolutional layers provide the main driving force of the representation learning scheme for ConvNets, defining a bank of filters which are convolved across the input feature maps in order to produce an output feature map describing the response to the learned filter. Such filters are formulated as a kernel of a fixed size, which strides across the input spatial domain, returning a singular response to the filter for the output map. These kernels represent a localized weighting of features within the fixed neighborhood of the filter's region of interest. Via the optimization of these kernel weights, we are able to learn kernels which represent various localized features rather than hand-craft descriptors such as Haar ad HoG representations in conventional image processing applications. Multi-layer ConvNets are able to learn weightings of these localized features, incorporating more informative spatial features together into a higher semantic representation of the problem domain. We can often observe the lower layers of CNNs describing low level spatial features, such as edges in an image recognition setting. As we go deeper these low level spatial features are incorporated to represent curves and corners. Further layers may represent textures, then part structures of a class, and then more composite structures. This representation learning is entirely machine-driven, removing the need for a user to define a set of high-level semantic features. The main assumption a regular array-like spatial domain allows the definition of a localized filter and its translation across the spatial domain to be given.



(a)

(b)

Figure 3.3: Key operators of the CNN architecture: (a) Convolutional layer extracting features from an input array. (b) A pooling layer downsampling an input feature map.

We define a filter kernel with the same spatial regularity, with an array structure containing the weights to be optimized, and the convolutional feature map is given as

$$a^{l+1}_{i'j'k'} = \sum_{ijk} w_{ijkk'} a^{l}_{i+j',j+j',k} \tag{3.8}$$

where i, j, and k are indexes into the array height, width and channel space respectively. In 3.8, kernel w is convolved with the input feature map a^l to return the filtered feature map a^{l+1}.

3.3.2 Pooling layers

The introduced pooling layers provide two main benefits to the CNN feature learning approach. Firstly, they provide a method for reducing spatial complexity of a model with an increasing number of feature maps generated by successive convolutional layers [11]. The pooling operation strides a receptive field across a given input map, reducing the underlying signal in the receptive field into a single pooled representation on the output feature map. Numerous pooling schemes have been proposed over the years, with average and max pooling operations showing notable use [336]. The second impact of the pooling operation is the generalization of the learned feature set. The pooling operation takes a local neighborhood and reduces the representation down into a singular value, driving the discriminator to learn a more generalized representation of the underlying function. This operation allows the model to react to localized trends, as opposed to the explicit element-wise feature weightings in fully connected NNs. As with the convolutional operator, the pooling operator is defined as a spatially regular region of interest, or ‘cell’, upon which the pooling operation is performed.

3.4 Advances in Deep Learning

Key developments in the field of CNNs have led to the development of several seminal approaches and their utilization in major problems within machine learning applications. Basic ConvNets have been shown to give strong performances on image domain problems, such as object recognition [11, 95, 145, 337–339]. These architectures were expanded to incorporate ‘inception’ modules, a parallel collection of convolution and pooling layers which are able to incorporate narrow and wide receptive fields for learning multi-scale features, [13], where the architecture was able to achieve promising results whilst increasing the computational efficiency. With the ResNet architecture, [12] introduces the residual block, a method of computing an additive delta to the input signal rather than an unreferenced mapping on the input space. This residual mapping architecture has shown major performance gains on several benchmark tests, highlighting benefits over the over-use of network depth. The study showed that the incorporation of residual information improved both training and testing performance, and that deeper networks can often suffer from overfitting.

The incorporation of temporal information for deep learning architectures resulted in the development of the Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) module [322, 340]. These approaches utilize previously obtained outputs from a layer as partial inputs to the same layer, producing a feedback loop within the architecture. This feedback allows the network to have some notion of memory, with each step containing some information about previous steps before it. In their early development, the recurrent nets fell foul of the vanishing and exploding gradient problem, due to their dependency on the ‘back propagation through time’ method of gradient calculation. In this scenario the vanishing and exploding gradient problems are magnified, further degrading through multiple feedback loops. LSTMs alleviate this issue by providing gated memory units which act as a form of storage within a network over time, allowing the error being backpropagated through time to be more robust to degradation. Such networks have shown strong performance in time-series problems such as speech recognition, synthesis and tracking problems [305, 340, 341].

Unsupervised representation learning has also been exploited, learning feature embeddings from an unlabeled set of observations [342]. Such AutoEncoder (AE) techniques have been shown to produce informative high-level class descriptors for a variety of applications, and has been expanded to utilize the localized feature learning of CNN operators with the Convolutional AutoEncoder (CAE) [343]. The introduction of the Generative Adversarial Network (GAN) framework presents a method for training networks by utilizing adversarial interactions between two agents; generative and discriminative models [344–346]. A generative network produces novel samples from across the distribution of the data, whilst the discriminative network attempts to maximize its ability to discern between synthetic samples produced by the generative network and those from the true data distribution.

Some studies have been made into understanding the learned feature descriptors obtained via convolutional neural networks, [347, 348]. Such methods attempt to extract the learned feature descriptors from within a network to identify the activation responses within the architecture. Zeiler and Fergus [337] utilize a decovolutional and unpooling approach to map the activations of intermediate layers back to a pixel space, producing maps which describe how a given filter responds to a particular input image. They show that the lower layers of the CNN respond to low level gradients and edges, while deeper layers provide exaggeration of the discriminative features for a given class, identifying structures such as eyes, noses and wheels. In a similar approach, Mahendran and Vedaldi [349] invert feature embeddings to reconstruct input images, exploring the learned representation space. Many deep learning approaches utilize raw input features to produce a learned representation, however some applications have been made in using hand-crafted features as input to a CNN approach [350]. The extraction of features which have been defined by domain experts allows machine learning techniques to utilize previous years of research and understanding in a given domain, and can often produce favorable results by either concatenating hand-crafted features along the channel axis [351], or by training separate networks on raw and designed input features [352–354]. Such methods have made of image domain features, using the localized filtering of the CNN as with any raw input which resides on the grid. Other methods, such as [355] Figure 3.4, extract feature vectors from an input space and reshape these vectors into 2D arrays for input into the CNN. Such an approach makes the assumption that there is a 2D relationship with the vector. With the concatenation of numerous feature vectors such an assumption may not be appropriate. The approach extracts features such as shape curvature and diameter about a face, and geodesic and medial distances between faces at numerous scales. These vectors are then concatenated and reshaped into a 2D grid domain before being used to train a CNN for the task of labeling faces in a mesh. The ordering of the feature vectors, and the dimensions of the 2D grid embedding are explored in [355] and found to have no impact on the network performance, which indicates that the localized filters are able to learn some mapping between the input “image” and the target output, but that the actual ordering of the elements within the grid are not important, a finding which seems at odds with the original concept of convolutional neural networks learning localized features. A similar approach is seen within [29, 30], where six very specific membership functions are defined which describe the behavior of only 3 joints of the possible 20 joints captured in their chosen skeletal model; the two hands and the pelvis. The hand-crafted descriptors produce 6 feature vectors of an equal temporal length, $v_i = \mathbb{R}^{1\times t}$, where a feature vector v_i is a vector of length t. The six vectors are then stacked into a $6 \times t$ 2D array and fed to a CNN with 1D filters, one configuration containing filters which learn across features (filters of size 3×1), and another which learns across time (filters of size 1×3). The results from this study show that learning filters that convolve across feature vectors that may not hold a spatial relationship can reduce model accuracy. Learning temporal filters that only consider the one feature vector in each row of the image provides a lower error rate in comparison. This study provides an arbitrary ordering of the feature vectors as rows of the image embedding, but does not evaluate the impact of permuting the order of the features. These studies show that embedding of features into the 2D grid domain is often possible, given some vector concatenations and reshapes, however the ordering of disjoint feature vectors and how such an assumption of 2D relationships between them is still an area which requires careful analysis and understanding.

The utilization of domain expert knowledge can be invaluable, and the use of hand-crafted features can provide a level of insight into data which may not result from the optimization of weights within a deep learning architecture. The question then arises as to whether the embedding of such features into a grid structured domain is a suitable assumption to make in order to make use of CNN operators. If there is no spatial relationship within the feature vectors, is it appropriate to enforce a spatial relationship through the use of a kernel’s receptive field. If there is a natural spatial relationship between elements that doesn’t reside on a regular grid, is it appropriate to enforce a constraint which considers a regular domain topology. By relaxing the array-based spatial relationship assumption there is the ability to consider how elements within a vector are spatially related in their natural domain.

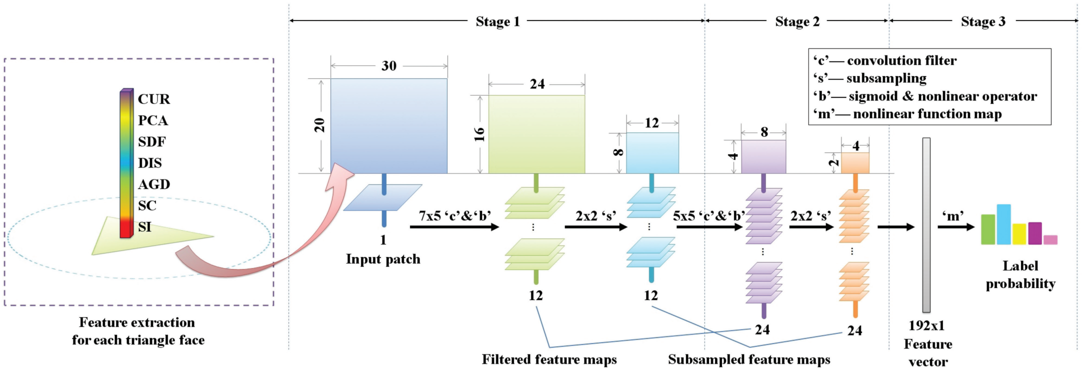


Figure 3.4: Pipeline of [355], concatenating and reshaping numerous feature vectors into the 2D grid for feeding as input to a Convolutional Neural Network.

3.5 Deep Learning in Irregular Domains

3.5.1 Regular Domain Assumptions

One common theme in the development of CNNs is the requirement for a regular Cartesian array as an input domain, upon which the convolution and pooling operations are executed. The regular grid allows for kernel-based filters to be defined with a regular local receptive field, and for these filters to be translated across the grid in a regular form. This works well for the problem domains which exhibit an array based spatial domain; 1D signals and sequences, 2D images, and 3D volumes and videos. This assumption does not hold however for a vast number of other domains, where spatial relationships between points in an input feature map may not be regularly spaced on a Cartesian grid; including sensor networks, social networks, text corpora, meshes, and the human skeleton, Figure 3.5. Such domains may benefit from a learned representation space that incorporates some localized information from the spatial domain, yet current CNN deep learning methods are unable to make appropriate use of such detail. Definition of the filter and a convolution operation is non-trivial when considering a domain in which the use of a spatially localized kernel and its translation across a grid is not regular. The notion of a localized neighborhood in the spatial domain is intuitive, however producing a localized kernel-based filter for a undefined neighborhood topology is not. We would still like to have analogous operators to those introduced by CNNs; the locally receptive filtering of the convolutional layer, and the feature generalization of pooling. However, defining such operations is a challenge. As discussed briefly in Section 2.7 it is possible to attempt some form of embedding of the irregular domain into a regular spatial grid. Such approaches look to utilize conventional CNN operators and architectures for representation learning on some reconfiguration of the spatial domain, usually through unrolling, projection or resampling [29, 30, 356–358]. These methods rely on the transform operation between the original domain and the new Cartesian embedding providing a suitably faithful reconstruction of the relationships between elements of the input space. Methods which introduce padding to support the embedding will introduce spatial locations in which no signal is present, and this will subsequently impact on the learned CNN filter kernels. The development of CNN layers came about as a method of adding a localized filtering constraint to fully connected neural networks, designed to optimize the architectures to the problem of image and volume recognition by exploiting the regular nature of the grid. The field of deep learning on the irregular domain aims to provide the same localized filtering constraints, but without the assumption of using the grid. These approaches instead intend to utilize the intrinsic spatial structure of the domain itself and redefine the filtering operation.

In Equation 3.8 we can see that indices i and j provide a relationship between pixels situated on the 2D grid and the weighted kernel convolved across them. For each pixel in a 2D feature map, a^l_{ij}, we are able to index into the appropriate neighboring pixels in order to perform the filtering operation and return the corresponding pixel in the output feature map, $a^l + 1_{ij}$. In domains that do not exhibit such a regularly formed Cartesian space it is difficult to produce such a regular kernel with which to sample the input space. Take for example the human skeleton in Figure 3.5, we could define a kernel which looks at the two adjacent nodes to a joint. This would work well for the majority of the joints on the skeleton; however the extremities of the skeleton (fingers, toes, head) have only one adjacent node, and some joints have more than two adjacent nodes (torso, hips). For these nodes, we would have to describe how to sample the input space, which would not work under standard CNN operators. Clearly some generalization of the convolution and pooling operation to irregular spatial domains is required to allow such problem domains to make use of the strong performance growth seen in representation learning. In the following sections we will discuss current approaches leading towards the use of deep learning in such irregular domains, identifying routes to generalizing learned localized filtering operations for non-Cartesian topologies which can be applied to different domains by defining spatial relationships between elements in the domain.

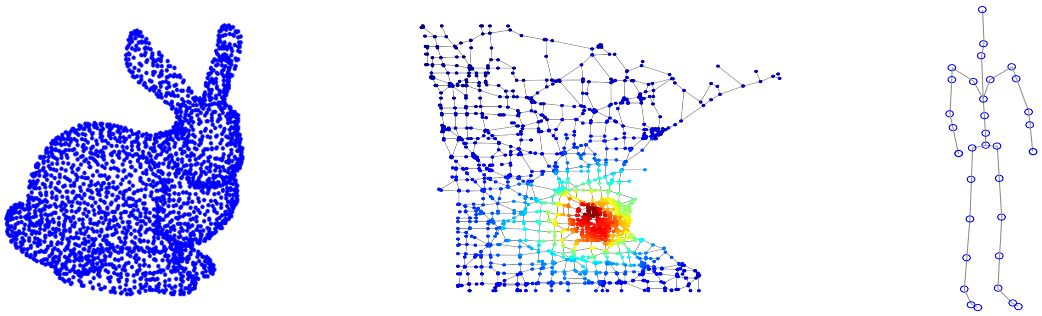


Figure 3.5: Examples of irregular spatial domains. Left to right: Meshes, sensor networks, human skeletons. Designing and convolving a localized filter for such domains is non-trivial.

3.5.2 Previous Approaches

Common previous methods in applying deep learning to such situations fall into two categories. First, the spatial information can be ignored entirely, utilizing standard neural networks to learn non-linear mappings on input features with no intrinsic spatial relationship between the inputs [28]. An alternative is to embed the input space into a regular Cartesian grid, such as an image, and make use of the usual CNN operators which are optimized for such a regular domain. The second approach has seen growing popularity in light of performance gains seen within the image processing community [29, 30, 359]. Both approaches have their own issues that require acknowledgment and discussion. In ignoring spatial information present in the input domain, we could be missing some underlying relation between inputs, limiting the performance of feature representation in such domains. Conversely, by enforcing a regular spatial topology on which we project an input domain, we may be making inappropriate assumptions about how certain input features are related. The power of deep learning has shown that such methods are often still able to display promising performance in such domains, however a more sensible approach to domains which exhibit these irregular topologies would remove unwanted assumptions and retain the information encoded in the spatial relationships between features.

3.5.3 Graph Signal Processing

In order to generalize current convolutional style approaches to graph domains, it is necessary to identify a representation upon which we can determine localized neighborhoods and compute filtering operations without the assumption of a regular spatial kernel. Graph-based signal processing techniques, [360], allow the application of common signal processing techniques to graph representations of an input domain with predefined graph filters. By formulating such a domain as a graph representation, we are able to define a structure upon which we observe graph signals and perform filtering operations. A graph G is composed of vertices and edges, $\{V,E\}$, where vertices of the graph represent a given input feature at a specific location in the input space, and the edges detail inherent spatial relationships between vertices. In an edge weighted graph of N vertices, the adjacency matrix $A \in 0,1^{N\times N}$ is a binary matrix representation of the edge list where $a_{i,j} = 1$ indicates an edge between vertices v_i and v_j, given $i \neq j$. The weight matrix $W \in \mathbb{R}^{N\times N}$ denotes the edge weights of the connected, undirected, non self-looping edges between vertices v_i and v_j. Such edge weighting metrics require defining for a graph construction method on a given domain; with certain domains providing a natural definition for the relationship between two nodes, such as connectivity between joints on the skeleton or distance between cities on a map. In applications where such a definition is not so readily available, one common approach is to threshold a Gaussian weighting function which utilizes the Euclidean distance between nodes within a given locality as a weighting measure

$$W_{i,j} = \begin{cases} \exp(-\frac{[\mathrm{dist}(i,j)]^2}{2\theta^2}), & \text{if } \mathrm{dist}(i,j) \leq k \\ 0, & \text{otherwise} \end{cases} \tag{3.9}$$

where parameters θ and k define parameters regarding the falloff provided by the Gaussian weighting scheme, providing a localized neighborhood of connectivity. It is possible to think of the Cartesian grid domain, for array inputs such as images or videos, as a graph which describes vertex connectivity via the Von Neumann (4-way) or Moore (8-way) neighborhoods, Figure 3.6. In this domain our pixels are formulated as nodes on the graph, with the adjacencies describing the connectivity to neighboring vertices. It is possible to use this generalized representation in numerous other less regular domains, such as using K-Nearest Neighbor connectivity in point clouds, or the connectivity of a human skeleton based on the adjacency of joints and bones. This graph construction step is critical, as it provides a underpinning description of the spatial relationships between the elements of the irregular domain problem, however approaches in generating such a representation for a given domain are still a matter of ongoing research [361, 362].

Once a graph representation of the spatial domain is defined, $G = \{V,E,W\}$, we are able to represent observations in the problem domain as graph signals $x \in \mathbb{R}^N$ residing on the graph structure. This can be further expanded to included multi-channel signals, where each node on the graph has C observed dimensions, $x \in \mathbb{R}^{N\times C}$. We could therefore represent a 28×28 RGB image as a graph of 784 nodes, where each node has a 3 dimensional vector detailing its red,

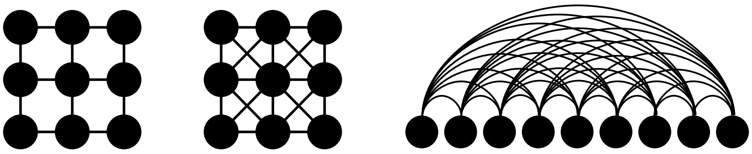


Figure 3.6: A few examples of graph constructions for a 3×3 regularly spaced grid structure. Left to right: Von Neumann Neighborhoods (4 way), Moore Neighborhoods (8-way), Fully Connected (non-spatial). Choice of a suitable graph construction approach is required as the graph represents the underlying spatial relationship between the spatial locations within the domain.

blue and green color intensities. Each given image on such a graph can then be represented by a 784×3 graph signal. We can further represent a graph structure as its non-normalized combinatorial graph Laplacian matrix

$$Ł = D - W \tag{3.10}$$

where D is a diagonalized degree matrix summing the adjacencies a_i for a given vertex i. The structure of Ł is a real symmetric matrix, exhibiting a complete set of orthonormal eigenvectors, examples of which are given in Figure 3.7, representing the spectral structure of an observed graph [363]. Providing a full eigen decomposition of the graph Laplacian returns a set of eigenvectors and their corresponding eigenvalues $\lambda_{l=0...N-1}$, which provide a method for projecting an observed graph signal x into the frequency space of the graph via the order eigenvector set u. The forward Graph Fourier Transform (GFT) expansion

$$\hat{x}(\lambda_l) = \sum_{i=1}^{N} x(i) u_l^T(i) \tag{3.11}$$

and the corresponding inverse

$$x(i) = \sum_{l=0}^{N-1} \hat{x}(\lambda_l) u_l(i) \tag{3.12}$$

provide spectral representation of the observed signal x. We can present the forward and inverse GFT functions in matrix form, by using the matrix of Laplacian eigenvectors, $U \in \mathbb{R}^{N \times N}$ where each column of U corresponds to a single eigenvector. Both transforms can be defined as matrix multiplication operations,

$$\hat{x} = U^T x \tag{3.13}$$

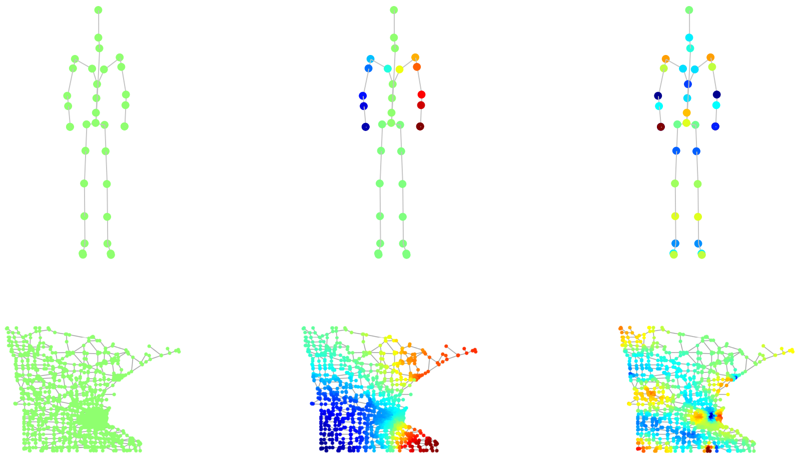


Figure 3.7: Example Laplacian eigenvectors taken from the human skeletal and Minnesota road network graph. The notion of frequency on the graph is shown by the intensity of each node. Left to right: 1st, 4th, and 20th eigenvectors. Note that lower frequencies are encoded by eigenvectors associated with lower eigenvalues.

and

$$x = U\hat{x} \tag{3.14}$$

where U^T denotes the transpose form of U.

By utilizing the graph Fourier transform, we are able to represent the same observed signal in both the spatial and frequency space of the graph, making use of spectral signal processing techniques to provide operators which may not be easily defined in the spatial domain, such as filtering, convolution and translation [360, 364]. Using the convolution theorem outlined by [365], it is possible to formulate the convolution of a filter g with a signal x on the graph in the spectral domain as an element-wise multiplication with a spectral filter. A fixed graph representation of the domain, and its Fourier basis, allows all observed signals to be projected into a common frequency space, and frequency filtering can be applied. Shuman identifies numerous continuous spatial filtering operations developed for spatial domains which can be generalized to graphs via the spectral filtering formulation [360, 366]; including Gaussian smoothing and bilateral filtering. Such spectral filtering operations have been shown to work well in generic graph domains [367–369]. It is also possible to describe an analogue to the convolutional operation of filter h on signal x in the graph domain, describing a localized filtering as a spectral operation

$$(x * h)(i) := \sum_{l=0}^{N-1} \hat{x}(\lambda_l)\hat{h}(\lambda_l)u_l(i) \tag{3.15}$$

where $\hat{x}$ is the spectral form of our input graph signal, and $\hat{h}$ denotes a spectral form of the transfer function. Thus we can generalize our convolutional filtering operation as

$$y = U(U^T x \odot h) \tag{3.16}$$

utilizing the graph Fourier forward and inverse transforms identified in equations 3.13 and 3.14. Figure 3.8 shows the projection of a graph signal, in this case an example image from the MNIST dataset, into the frequency domain, followed by a filtering of high frequency signals and an inverse graph Fourier transform back into the spatial domain. The removal of high frequency information provides a smoothed signal in the spatial domain, as is expected. The ringing artifacts present are produced by the interactions between the eigenvectors representing the frequencies across the graph topology, and as such the application of filtering on the spectral information equates to the scaling of frequency bands. Similar ringing effects are seen during the spectral filtering of images via the Fourier space due to the relationships and representation of harmonics in the Fourier transform [370]. In Chapter 5 we look at an approach to constrain spectral filtering to occur with spatial domain localization.

The graph signal processing techniques detailed in [360] detail ways to apply a given defined filtering operator to the graph but do not discuss how such filters can be learned or optimized, such as with representation learning approaches such as deep learning architectures. With the advent of self-learned filtering operations and the motivation to develop such filters on irregular domain applications, the logical progression is towards the development of graph filters which are able to optimize given a observed dataset as with CNNs. Such a formulation of graph based signal processing can be used to define graph convolutional layers, which will be explored in the coming chapters.

3.5.4 Graph Coarsening

In addition to filtering operations based on the graph domain, there are numerous methods of down-sampling graph structure in order to represent a given graph topology at a coarser scale [371–375]. Such methods aim to reduce the number of vertices within a graph, whilst retaining the general spatial topology of the finer resolution graph. Coarsening can assist in signal generalization and increase the efficiency of computing filtering operations, providing

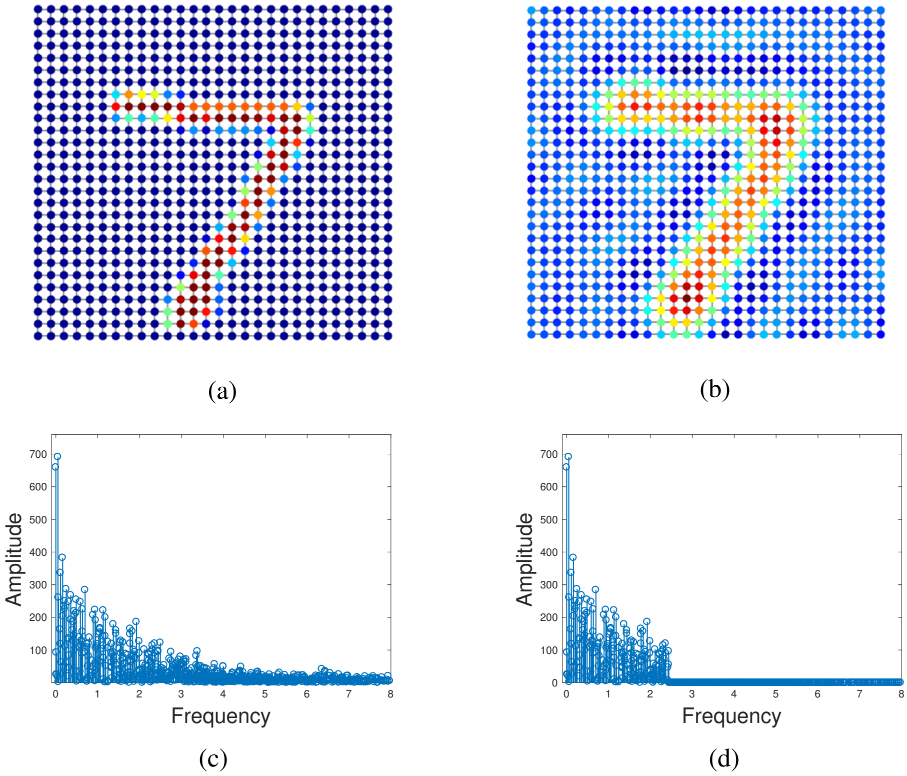


Figure 3.8: Spectral filtering of the graph signal residing on the graph representation of the 2D grid. a) Graph signal residing on the graph spatial domain. b) Resulting filtered signal in the spatial domain, notice the smoothing effect resulting from the removal of high frequency information. c) Spectral signal representation of the input signal from (a). d) Spectral signal after thresholding to remove high frequency signals.

analogous benefits to the pooling operations within the image processing applications. Just as the methods for signal filtering in array domains has been given a generalized form with graph signal processing techniques, pooling has an analogous operation in graph coarsening. Methods of reducing the resolution of a graph have been presented as an on-going area of research, focusing on several major classifications of coarsening schemes including cuts, contraction algorithms and agglomeration schemes [374]. In essence, graph coarsening looks to aggregate nodes and their corresponding graph signals. Contraction methods create a new node x by 'contracting' an edge connecting two nodes, u and v, subsequently connecting all neighbors of u and v to the new node, [376, 377]. Multi-resolution methods, [372, 378–380], construct coarser graphs via a iterative linear solver approach, aggregating nodes and signals together based on

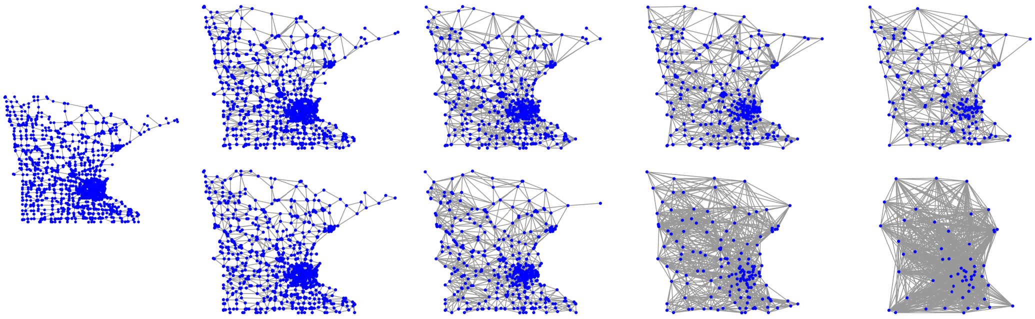


Figure 3.9: Graph coarsening of the Minnesota road network. Left - Original graph structure. Top - Kron's pyramid based pooling. Bottom - Algebraic Multi-grid based pooling.

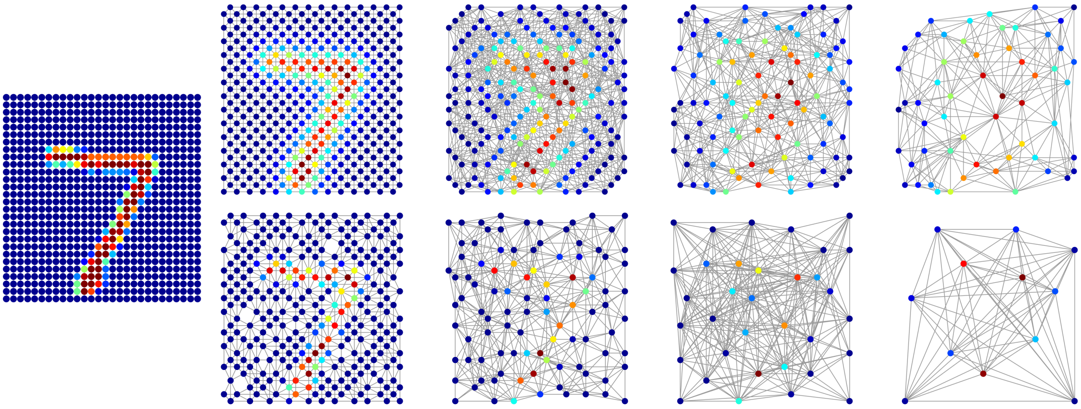


Figure 3.10: Graph based coarsening of an example MNIST signal residing on the 2D grid graph. The graph signal must also be pooled alongside the graph representation. From left - Original graph and signal followed by increasing levels of coarsening. Top - Kron's reduction. Bottom - Algebraic MultiGrid reduction.

the weighting between vertices as presented in the edge weight matrix W. Such methods often return matrices for the restriction and projection of a signal from a finer graph onto a coarser representation and vice-versa. Figure 3.9 highlights two graph coarsening methods, Kron's reduction [381] and Algebraic Multigrid (AMG), and their impact on a graph representation of the Minnesota road network. Figure 3.10 shows the impact of such graph pooling operations on an observed graph signal alongside the graph structure.

The selection of pooling methodology for a given domain application is one focus of study within the graph signal processing community, with [371, 374, 382–385] providing insights into different approaches and their uses for different classes of graphs. Conventional pooling within the regular domain of CNNs consists of translating a locally receptive field across the input and taking an average or a maximum of the underlying input map, as described in Section 3.3. As discussed, such translation and sampling via a regular kernel-based field is ill-defined for domains with an irregular spatial topology. The graph coarsening schemes approximate this behavior, but such methods are varied and behave with certain properties which may or may not be preferential for a given application domain. The selection of a suitable pooling method for a given domain is still an open problem, and the selection of pooling approach can vary for a given application domain. Figure 3.9 shows that the AMG pooling approach has greatly reduced the number of vertices within the graph, however the edge count has increased dramatically, leading to spatial relationships between vertices to span a much wider region of the input spatial domain. Compare this with the result of Kron's pooling and we can see that the reduction in vertex count is much less, but the overall edge connectivity is a more faithful representation of the original domain. It may appear that the choice of using a multi-resolution agglomerative method for Graph-CNN pooling layers appears to be a straightforward choice when compared to Kron's reduction, however in Chapter 7 we will explore the impact of applying both methods to the human skeleton for the purpose of human action recognition. In this application domain the use of AMG provides variable pooling performance, often condensing the representation too far and with low robustness to starting initialization of the greedy agglomeration.

The use of a graph representation provides a domain for applying common signal processing operations, defined as graph operators, producing analogous behaviors to those observed in CNN architectures. In essence the CNN convolution and pooling operators are a subset of such operations, where an observed image represents the array-like input domain of the regular spatial Cartesian space. The CNN operators however are optimized purely for the regular assumption, and as such the approximations provided by signal processing techniques may not achieve ideal performance on the image domain applications. Despite this drawback in image domain problems, by representing the spatial domain as a graph it is possible to generalize the learning of localized features to a wide range of domains which previously would have either ignored spatial information or enforced an assumed spatial embedding in an attempt to achieve the performance gains seen in image applications.

3.6 Summary

This chapter has outlined the overall state of deep learning methodology, from the beginnings in feedforward networks to the utilization of regular spatial information within CNNs. We have highlighted some shortcomings when approaching domains with irregular spatial topologies, identifying a direction in which the community can generalize current layers in order to accommodate a wider range of application domains. A graph based approach to signal processing methods is introduced and in the coming chapters we will develop a method to bring the benefit of representation learning of spatially localized information to the irregular domain. The following chapter presents a method of learning low level descriptors of motion information by an unsupervised clustering mechanism.

Following chapters will introduce the development of feature learning processes in irregular topologies; developing Graph-based Convolutional Neural Network architectures and applying them in the domains of signal classification, and multi-scale and temporal feature learning. Chapter 4 first looks at clustering observed skeletal sequences into primitive gestures, providing a bag-of-words approach to human action recognition in which the gestures are optimized over time via an evolutionary algorithm approach. Chapter 5 then introduces a deep learning approach on irregular domain problems, which is then used in Chapters 6 and 7 for feature learning across multiple scales and spatio-temporal motion.

Chapter 4

Unsupervised Learning of Gestures for Human Action Recognition

Contents

4.1 Introduction

As discussed in Chapter 2, Human Action Recognition (HAR) is a field concerned with the detection and identification of different human behaviors observed within a scene. As such, it is a topic applied to numerous problem domains; including surveillance, human-computer interaction and medical diagnostics [386]. Events are often categorized based on complexity in gesture, action, interaction, or group activity, yet an activity can potentially be a mixture of lower-level gesture types, e.g., ‘walk’ contains gestures including ‘lift leg’, ‘swing leg forward’, and ‘lower leg’ [387]. Current appearance and pose-based methods have lead to the accurate recognition of simplistic actions and gestures, such as ‘waving’, ‘running’, and ‘jumping’ [25,43,388]. Until very recently, methods have focused on developing hand-crafted feature extractors, utilizing descriptors of spatio-temporal information such as Space-Time Interest Points (STIPs) in space-time volumes or joint motion from skeletal models. Recent advances have been made in using representation learning within images and videos for human action recognition. There has been recent renewed interest in the use of pose-based HAR, partly due to the availability of commercial depth sensor systems which are able to track body joint locations with reasonable accuracy [25, 101], with some work towards using representation learning for skeletal information descriptors [29,30,94,295].

There are several key issues to consider in HAR problems, and often individuals may perform the same actions with both intra- and inter-subject level variation in their spatial or temporal execution [389]. Therefore, it is necessary to develop methodologies that are able to deal with the impact of spatio-temporal variation on an intra- and inter-subject level, whilst maintaining partitioning information at the inter-class level. Recent approaches have made use of sequence alignment to allow temporal comparison between actions, key pose representation to study the underlying gesture composition of an action, and segmentation to identify gestures within a sequence [120, 131].

Previous study on the use of pose estimation has promoted the use of a Bag of Key Poses (BoKP) model, in which representative key spatial poses form a bag of words which can be compounded to describe higher-level actions [120, 388, 390–392]. To achieve this, k key poses are generated by clustering similar frames from a whole sequence. Transforming a sequence into a key pose representation reduces the impact of minor frame-to-frame spatial variations, provided that sensible key poses are generated [131], as representative poses are produced for each class and stored within one bag [388, 390, 392].

To align actions composed of linear pose sequences which may vary in temporal execution, sequence alignment techniques such as Dynamic Time Warping (DTW) [135, 136], Dynamic Manifold Warping (DMW) [393, 394], and Canonical Time Warping (CTW) [395], have been employed to reduce the impact of temporal variations, [116,137,396,397]. These methods have however been criticized in situations where the temporal execution rate may provide some key information between two classes, e.g. ‘run’ and ‘walk’ [134], or there are repeated cyclic gestures within the action [111]. In some cases, an action can be defined by its accumulated composition of primitive poses, forming a bag of words representation [120]. In both of these situations we believe it is beneficial to first segment the observation to identify any repeated primitive gestures. This will assist in identifying cyclical or compound gestures that form a higher-level action without over-simplifying the key gestures an individual makes. To segment an observation, [115, 131] utilize a DTW approach to group varying length segments into k clusters by dynamic programming via Aligned Cluster Analysis (ACA) or Hierarchical Aligned Cluster Analysis (HACA). ACA methods provide a temporal clustering, utilizing a Dynamic Time Alignment Kernel (DTAK) proposed by [398] to construct a distance metric between two sequences for clustering.

The use of Dynamic Time Warping allows similarity between two sequences of varying length to be compared by creating a mapping between them which provides alignment and a non-linear warping in the temporal dimension, [399, 400]. Given two sequences, DTW uses a dynamic programming approach to produce a warping path which minimizes the distance between elements on the two sequences. Given the vector sequences $S \in \mathbb{R}^{1..i..n}$ and $\hat{S} \in \mathbb{R}^{1..j..m}$, we compute the warp path W through an $n \times m$ grid formed from the arrangement of S and $\hat{S}$. Each point (i, j), where $1 \leq i \leq n$,$1 \leq j \leq m$, on the grid describes a possible alignment between the element s_i and $\hat{s}_j$. An element w_i on the path W indicates the minimal path by identifying which pairing (i, j) provides the minimal distance. W is populated by searching through possible warping paths, with constraints to avoid an exhaustive search of a combinatorial problem. The warping path is monotonically ordered temporally, with $i_k \geq i_{k-1}$ and $j_k \geq j_{k-1}$, this enforces that the alignment of S and $\hat{S}$ does not become entangled and ensures that comparison of time flows forward through the two sequences. The warping path must be continuous, such that the distance between i_k and i_{k-1} is less than or equal to 1, with the same holding for elements in $\hat{S}$. The path is also bound by the two ends of each sequence, such that $w_1 = (1, 1)$ and $w_K = (n,m)$, thus aligning the starting and finishing elements. By selecting a suitable distance metric between two elements on the time series we can define the total DTW distance as

$$DTW(S,T) = min\sum_{k=1}^{K} \delta(w_k) \tag{4.1}$$

where $\delta(w_k) = \delta(i, j)$ denotes the chosen distance metric between elements from the two sequences, such as the squared difference $\delta(i, j) = (s_i - \hat{s}_j)^2$. The computation of the DTW distance has since been optimized and formulated as a recursive matrix populating problem, in which the matrix $D \in \mathbb{R}^{n,m}$ denotes an accumulated cost matrix such that $D(n,m) = DTW(S,\hat{S})$. D, as represented in [400], is populated via dynamic programming as follows

$$D(n,1) = \sum_{k=1}^{n} \delta(s_k, \hat{s}_1) \tag{4.2}$$

$$D(1,m) = \sum_{k=1}^{m} \delta(s_1, \hat{s}_k) \tag{4.3}$$

$$D(n,m) = \min(D(n-1,m-1), D(n-1,m), D(n,m-1)) + \delta(s_n, \hat{s}_m) \tag{4.4}$$

Further approximations of the DTW similarity measure have been introduced to provide a more efficient metric, including the use of the DTAK discussed in Section 4.2.

Finding an optimal set of classification parameters is non-trivial, and optimization requires the selection of informative training samples and features to reduce the impact of outliers in the action space. Evolutionary programming methods have been used to provide an optimum selection of training instances [116], and informative features for a given observed action class [41]. Such methods aim to remove sequence observations which hinder the recognition of a given action class, ignoring those sequences which do no provide benefit in partitioning the function space. As new classes and samples are introduced to the model the population genetics are updated to adapt towards the new conditions. This online learning has an attractive application for HAR, learning new action classes without having to retrain a classifier from scratch as in an offline fashion.

To efficiently recognize interactions between two people, whilst providing a method of key pose generation that reflects the composition of higher level actions in terms of their shared gesture dictionary, we present a means of using sequence alignment to obtain sub-action gesture segmentation across all training observations. The ability of ACA to cluster similar segments of frames from a sequence, combined with the benefit of recognizing repeated sub-action segments via temporally flexible DTW, presents a method of segmenting similar sub-actions between multiple observations of an action class. These segmented gestures are then represented as key poses in an evolving bag representation, thus identifying key poses of a local temporal region that is repeated across training instances. By moving towards the recognition of more complex scenarios we hope to eventually lead towards recognition of higher-level, complex interactions between individuals; such as the context specific interactions discussed by [120, 121].

The rest of the chapter is organized as follows: In Section 4.2 we describe our method of using ACA to identify cross-subject gestures before extracting key poses for each gesture cluster. Section 4.3 details the evaluation undertaken in an application on the problem of human action and interaction recognition from skeletal pose. In Sections 4.4 and 4.5 we draw conclusions on the predictive abilities of the proposed method, evaluating performance with two publicly available datasets.

4.2 Proposed Approach

We propose the use of sequence alignment and segmentation methodology to identify cross-subject gestures, generating a key pose representation for action and interaction recognition. By identifying descriptive poses within each gesture we are able to more accurately represent fine scale sub-action primitives which compound to form a given gesture, reducing the information loss within conventional whole-sequence compression. We propose that understanding these gestures may in turn benefit the learning of higher-level actions. We utilize ACA to generate segments for a given action, using these gesture clusters to identify key poses. The key pose space allows reduction of spatial variation within the observations, providing more accurate sequence alignments. The sequences of key poses are then used to generate a nearest neighbor classifier for predicting labels of newly observed sequences. In order to identify suitable pose generation parameters we utilize evolutionary programming to select informative training observations and features from the input data.

4.2.1 Segmentation of Gestures

The observation of an action is often the compounding of numerous poses into a sub-action gesture, with multiple gestures then forming the given class. Therefore a set of observed sequences for an action, X_a, where A is all possible actions, often contains a set of sub-action gestures which best describe X_a. These gestures are a temporally ordered sequence of key poses, the frame-by-frame pose of the human body. With the intention to recognize similar primitive gestures across all observations of X_a, we represent all training instances of the given action a as a single continuous sequence. Unlike previous methods, which use k-means as a method of determining the k key poses, we make use of the ACA methodology presented by [115, 131] to first cluster similar action primitives into k' gesture clusters common across all training instances of a given action. A minimum and maximum segment length is selected and all possible segment sizes within that range are iteratively clustered using DTAK. Using DTW, possible segments are aligned to current members of each cluster and allocated to the most similar cluster, with each iteration minimizing within-cluster variance, segmenting out similar repeated gestures across subjects. Once we obtain the ACA segmentation, we find the k poses which describe a gesture by k-medoids clustering over the gestures.

Usual key pose generation draws representative poses from all frames of observed sequences, which may cause motion within a gesture to be lost in key pose representations taken from the overall action observation. In comparison, by identifying key poses within sub-action gestures we are formulating a key pose representation that reflects the gestures that in turn compound to form an action. This produces key poses from gestures that are observable across numerous subjects, providing informative poses that compose each gesture.

Given an observed sequence, $x \in \mathbb{R}^{d \times n}$, the ACA algorithm produces a segmentation dividing x into m disjoint segments, each clustered into one of k clusters. The iterative clustering mechanism minimizes the energy function

$$\operatorname{aca}(G,s) = \sum_{c=1}^{k} \sum_{i=1}^{m} g_{ci} ||\psi(x_{[s_i,s_{i+1})}) - z_c||^2 \tag{4.5}$$

where $G \in \{0,1\}^{k \times m}$ is a one-hot encoding denoting which of the k clusters a given segment, $x_{[s_i,s_{i+1})}$, is allocated to. A vector S denotes the start position of each segment, where a given segment contains the time frames from s_i up to, but not including, s_{i+1}, limited by the hyper-parameter $n_{\max}$. The vector S is then iteratively updated via the aligned cluster analysis approach, updating the start and end positions of a segment via 4.5.

The distance metric from (4.5) defines the squared distance between the i^{th} segment and a given cluster centroid z_c, utilizing the DTAK metric

$$\begin{aligned} dist_{\psi}^2(Y_i, z_c) &= ||\psi(x_{[s_i,s_{i+1})}) - z_c||^2 \\ &= \tau_{ii} - \frac{2}{m_c} \sum_{j-1}^{m} g_{cj} \tau_{ij} + \frac{1}{m_c^2} \sum_{j_1 j_2=1}^{m} g_{cj_1} g_{cj_2} \tau_{j_1 j_2} \end{aligned} \tag{4.6}$$

where τ defines the DTAK function mapping value between the current segment and the current cluster centroid.

In order to cluster varying length segments, the DTAK function τ aligns two sequences $x \in \mathbb{R}^{d \times n_x}$ and $\hat{x} \in \mathbb{R}^{d \times n_{\hat{x}}}$ by populating the cumulative kernel matrix $U \in \mathbb{R}^{n_x \times n_{\hat{x}}}$ via

$$u_{1,1} = 2k_{1,1},\; u_{i,j} = max \begin{cases} u_{i-1,j} + k_{i,j} \\ u_{i-1,j-1} + 2k_{i,j} \\ u_{i,j-1} + k_{i,j} \end{cases} \tag{4.7}$$

where K is the frame kernel, a matrix denoting the similarity between frames in the two sequences. In this case we assume a Gaussian kernel based similarity metric (4.8).

$$k_{ij} = exp\left(- \frac{||x_i - \hat{x}_j||^2}{2\sigma^2} \right) \tag{4.8}$$

The output value given by the aligning kernel is the final cumulative sequence similarity value from U, normalized by the sum of the two sequence lengths,

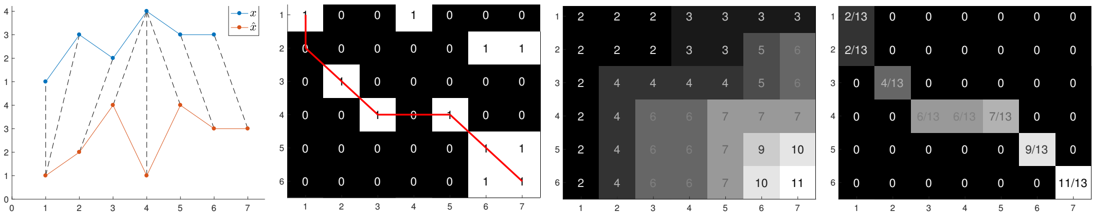
$$\tau(x, \hat{x}) = \frac{u_{n_x, n_{\hat{x}}}}{n_x + n_{\hat{x}}} \tag{4.9}$$

providing a distance metric which accommodates varying length sequences.

The overall segmentation pipeline is described as follows. Given sequence with some initial segmentation a forward pass of the sequence is made which computes the DTAK value between the current segment $X_{[i..v]}$ and each of the segments in the currently identified clusters, Fig. 4.1. The head position of each segment, the label predicted for that time-frame, and the minimal energy for the segment. A backward pass then traverses the sequence in reverse and segmentation cuts are made for each of the stored head positions. Iterating between forward and backward passes continues until convergence of the ACA energy. Such an approach to sequence clustering allows improvements over the use of other unsupervised methods, such as k-means and kernalized k-means, by providing an ability to cluster varying length sequences and a similarity based on the mapping between two observations [131].

4.2.2 Sequence Alignment and Prediction

The gestures identified by the aligned clustering are used to generate a DTW nearest neighbor classifier to provide label predictions for new observations of a given action class. For each of the training sample classes we produce a segmented sequence, returning clustered gesture primitives from across numerous subjects and observations of a given class, Figure 4.2. k key



(a) (b) (c) (d)

Figure 4.1: Computation of the Dynamic Time Alignment Kernel between two sequences. a) Two signals, x and $\hat{x}$, b) Frame kernel matrix K where $\sigma \to 0$, c) Cumulative kernel matrix (U), d) Normalized signal correspondence matrix. Interpreted from [131].

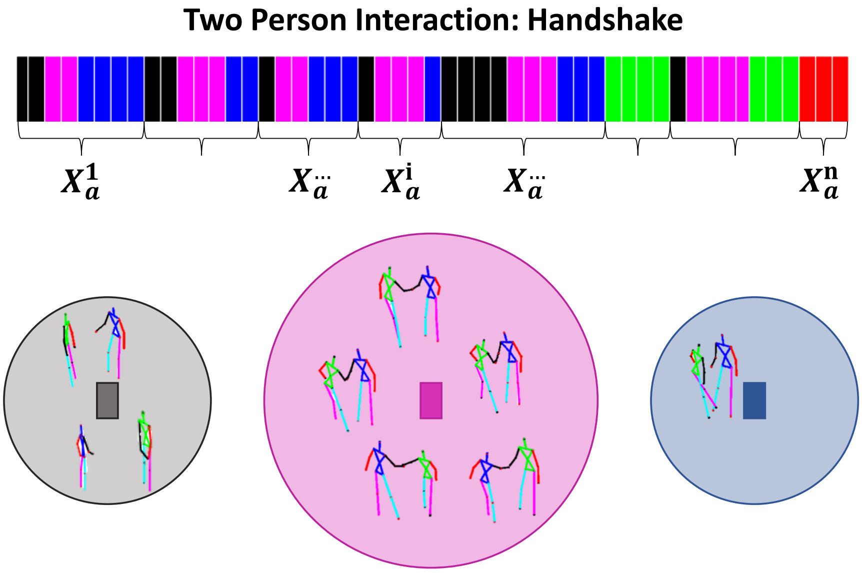


Figure 4.2: Examples of a learned sequence segmentation via aligned clustering. The training examples for a given class, X_a^i, represented by individual genome are segmented into clusters, exhibiting a repeating series of primitive gestures. Single frames of temporal gestures are shown for clarity. Overview of gesture identified: black) Hands coming together, magenta) Hands shaking, blue) Hands coming apart. Green and red segments represented gestures in observations where skeletal tracking was noisy.

poses are extracted by k-medoids from each gesture, identifying the poses which contribute to the primitive gestures. The generation of key poses from the segmented gestures is intended to reduce noise introduced by spatial variation between frames, whilst providing a sampling distribution that considers the gestures that comprise a given action. Using DTW as a nearest neighbor distance metric for classification, we are able to reduce the effects of slight temporal fluctuation in execution rates common in HAR. A test sample is predicted to share the label of the training template with the shortest warping distance.

4.2.3 Parameter Optimization

In order to optimize our gesture-based key poses we expand upon the evolutionary programming explored in [116], allowing parameters for our models to be identified over each successive generation of a population. We first construct a population $P_{1:n}$ containing n individuals, with each individual represented via a genomic sequence (Fig. 4.3), $p_i = [g_1, ..., g_L]$; where each gene vector, g_l, represents a parameter of the model. The training instance vector, $i_{1:n} \in \{0, 1\}$ for n training samples, is a binary indicator of whether a given sample in the training set is used to generate the key poses. The vector is updated by random initialization as training instances are added to the system. This vector aims to optimize the selection of informative training samples. The parameter vector, $p_{1:a} \in \{\mathbb{N}\{1, ..., K\}\}$, where K is an upper limit constraint on the possible number of key poses with which to populate the bag. Smaller k results in coarser approximation of action class a. Should the system learn a new action, then A is increased by 1 and representative key poses are learned for the new class. The feature selection vector, $f_{1:s\times m} \in \{0, 1\}$ for m possible joints and s subjects, is a binary indicator denoting if a given feature is used to generate key poses. By treating the individual subjects in a scene separately, we are able to optimize which joints are informative to the overall class; this is beneficial when an interaction class has the same label, but the two subjects react differently across instances. In each generation all individuals are ranked on poses they produce, maximizing the correct predicted class labels obtained by sequence alignment classification outlined in Section 4.2.2.

Standard Evolutionary Algorithm (EA) operators are used for *reproduction*, *recombination*, *mutation*, and *ranking* of the population within each generation, Figure 4.4. Genes within the population are treated as independent parameters, subjected to the behaviors of the EA operators defined below. The genes of the training vector and parameter vector are handled independently, whilst the feature vector has a recombination scheme which considers the branching

	Training Vector $\in \{0,1\}$	Feature Vector $\in \{0,1\}$	Parameter Vector $\in \{1,...,K\}$
$P_i =$	1 n	1 $s \times m$	1 a

Figure 4.3: Genetic makeup of individuals in the population. The training vector denotes observations chosen to generate a bag of gesture poses. The feature vector denotes which joints are observed on the human skeletons of multiple subjects. The parameter vector describes the number of poses to generate from each of the action gestures.

structure of the human skeletal model, as in Figure 4.5. The mutation operator considers all genetic parameters as independent variables, as outlined in Figure 4.4 *Recombination* for i and p occurs as outlined within common practice of EA, utilizing discrete sexual recombination via single point crossover [401]. Recombination of f occurs via a domain specific crossover method, Figure 4.5, in which a joint and its dependent branch is substituted with the second parent. Recombination helps to defer convergence onto a homogeneous population by introducing variation of genes between parents and offspring. Once a new offspring is generated, *mutation* provides variation within the population gene-pool, attempting to avoid optimizing towards a local minimum by widening the search space. Each gene within vectors $i_{1:n}$ and $f_{1:s\times m}$ are subject to a binary flip based on their respective mutation rates; whilst genes in vector p are either sampled from a random distribution over all possible values, or from a Gaussian distribution over a localized range, each with equal chance. For observations of human interaction we must modify the operators to handle two individuals, thus the recombination operator has a domain specific crossover method that accounts for the semantics that describe each of the two observed individuals.

Taking an evolutionary approach to learn our value of k allows us to set a value of K, and the population will grow to select a k for each action class which minimizes the accuracy error during the evolutionary development. Selecting the value of K is a hyper-parameter, and choosing a sufficiently sized K will define an upper limit on the number of segment clusters, limiting the number of different gestures which can be identified. Similarly, the selection of maximum/minimum parameter constraints on the segment lengths affects the temporal scale of the gestures that can be extracted from the observations. The evolutionary approach could be extended to further incorporate the parameters we wish to tune into the genome of the population, however doing so can greatly increase the complexity of the search space [402, 403]. By utilizing an evolutionary approach we can apply constraints on traversal through the search space through the use of the genetic operators, a preferable method of reducing the

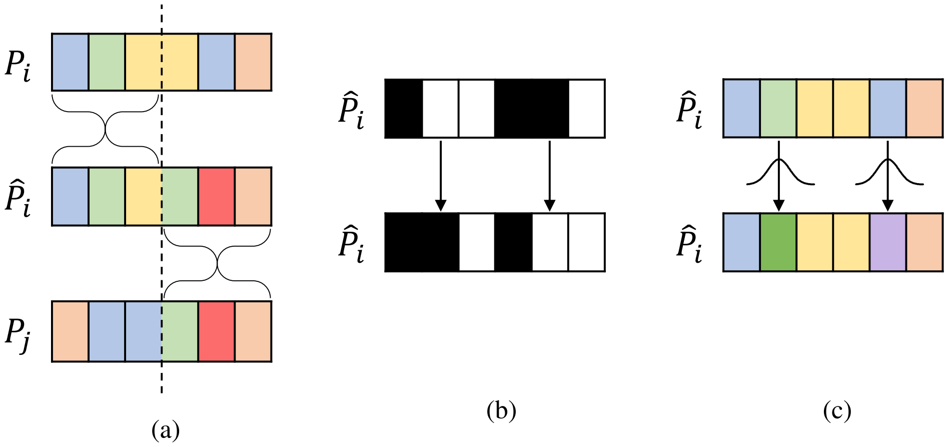


Figure 4.4: Evolutionary Operators. a) Single point crossover, two parents P_i and P_j produce an offspring $\hat{P}_i$ by recombination at a selected crossover point. b) Binary gene mutation as with the training and feature vectors. c) Gaussian based gene shift mutation as with the parameter vector.

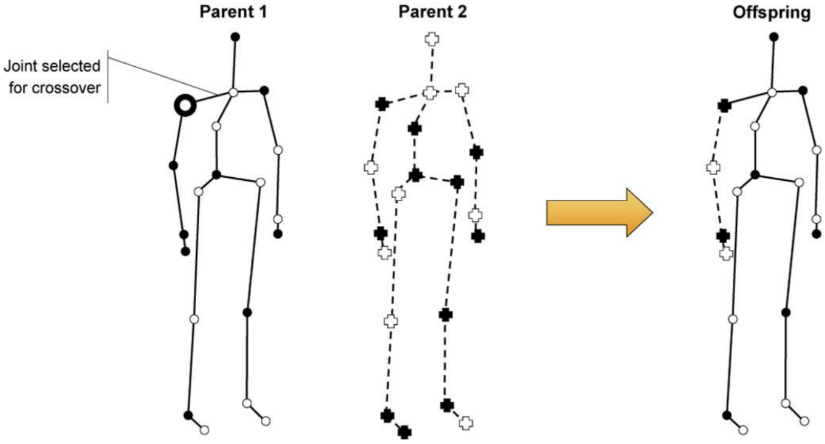


Figure 4.5: Domain specific single point crossover operator performed on the skeleton feature selection gene. When a given joint is chosen for crossover from parent 1, the branch of the skeleton below the selected joint is copied to the offspring. When expanded to multiple subjects, the branch is computed based for each skeleton model individually. Image from [116].

variability within random walks [404]. Although we utilize basic genetic algorithm operators for feature representation optimization, it is theoretically suitable to use other optimization schemes to select the joints of interest, the number of clusters per action and the samples to observe for training. Optimization strategies such as Particle Swarm [405], or Cuckoo Search [406, 407] could also be used to select the representation learning parameters.

4.3 Application

The method presented requires a number of initialization parameters to be selected before they are optimized using evolutionary programming. For all experiments, the mutation probability was dynamically selected via a random distribution from between 0.0 and 0.1 for instance vector i, and between 0.0 and 0.2 for vectors f and p. In our study, the Gaussian standard deviation for mutation of genes within parameter vector p was empirically set to $\sigma = 4$, producing a small localized mutation search space when using Gaussian gene mutation. For our evolutionary optimization, for comparability to [116], we selected an initial population size of $n = 10$, with 10 offspring created at each generation. For initial seeding of the population, genome vectors are randomly initialized. In both single action and interaction experiments we limited the number of generations per action to 50, as the populations began to converge, Figures 4.8 and 4.10; however evolutionary optimization can be repeated indefinitely to allow for the time restriction to be relaxed on the optimization. For k-means clustering, we limited the maximum value of k to 40 key poses; as it provided both a decrease in complexity, and marginal increase in accuracy on the 75 poses used by [116].

Single Action For single person HAR, the proposed method was evaluated on the 20 tracked joints of the Microsoft Research (MSR) Action3D dataset [102], Figure 4.6. This dataset is a commonly used standard for single person skeletal pose based HAR methods, and is comprised of 3 subsets containing various action classes; containing 20 action classes performed by 10 subjects, repeated up to 3 times. We evaluate our method on the AS2 subset, which is viewed as the most complex of the 3; utilizing the train/test split outlined by [102], producing a *leave-one-actor-out* cross-subject validation analysis. The AS2 set contains the 8 action classes listed in Table 4.1. Joint coordinates are utilized as features, with each gene of the feature vector f representing a joint marker, $f_{1:20}$. This subset was also used to evaluate the methods in [116], which we have also implemented here for cross-comparison. For ACA sequence segmentation on the MSR dataset, we empirically initialized the segmentation method to group the total-

Table 4.1: Action class sets used for evaluation, with generation in which they are introduced to the population. The first two actions are introduced simultaneously for initialization.

Generation	AS2 [102]	Stony Brook University (SBU) [108]
0	High arm wave	Approaching
0	Hand catch	Departing
50	Draw x	Pushing
100	Draw tick	Kicking
150	Draw circle	Punching
200	Two hand wave	Exchanging
250	Forward kick	Hugging
300	Side boxing	Handshake

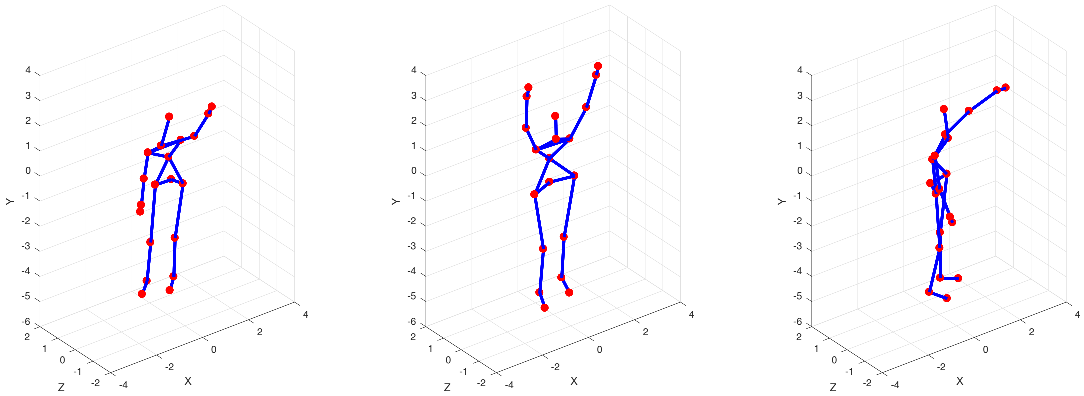


Figure 4.6: Example poses from the MSR Action3D dataset - High arm wave, Two hand wave and Side boxing.

instance sequence into $k' = 5$ sub-action gesture clusters, with a segment length limitation of between $nMin = 1$ and $nMax = 10$ frames.

Two Person Interactions To evaluate the method for the purpose of interaction recognition we utilize the SBU Kinect Interaction dataset [108], Figure 4.7; a dataset consisting of 8 interaction classes, Table 4.1. 21 pairs of subjects performed actions up to 3 times, and 15 joints were tracked via Kinect. Following the 5-fold cross validation split outlined in [108], with 4-5 interaction pairs per fold. Joint coordinates features are utilized, with the pairwise interaction encoded as a 30 dimensional temporal sequence, $f_{1:15}$ representing person A and $f_{16:30}$ person B. To indicate this in population genetics, the recombination of a given feature vector $f_{1:30}$

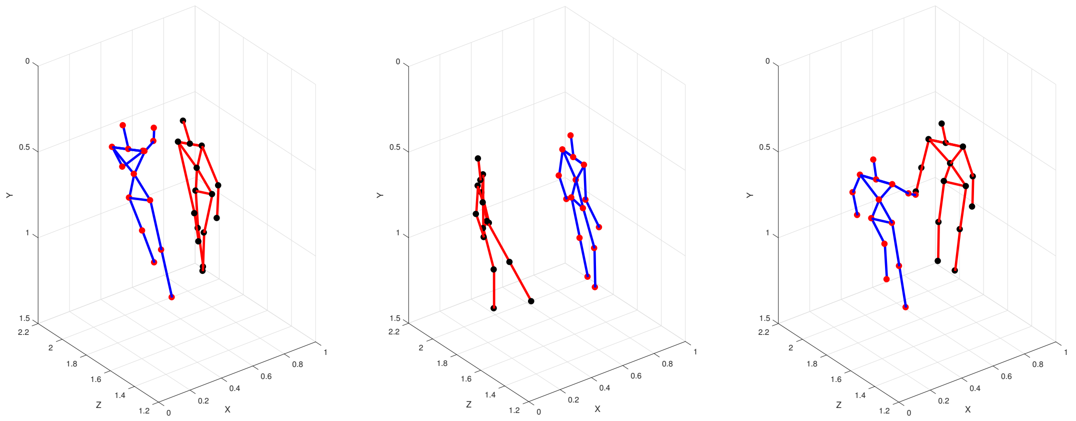


Figure 4.7: Example images from the SBU Two Person Interaction Dataset - Punching, Departing and Hand-shake.

occurred via a modified domain specific method; if the cross point fell between $f_{1:15}$, any dependent joints along the branch would be taken from the person A on the second parent, while cross points falling on genes in the range $f_{1:15}$ were selected from person B. By this method we have chosen to maintain domain specific recombination whilst applying it to handle the two individuals observed in the scene. To obtain ACA segmentation of the SBU dataset, we initialized the segmentation cluster value $k' = 5$ sub-action clusters, with gesture length between $nMin = 1$ and $nMax = 4$ frames as there are a lot of cyclic motions within the SBU class which have very short repetition rates.

4.4 Comparative Analysis and Results

We present the findings of our proposed method within Table 4.2 and Figures 4.8 and 4.10, utilizing ACA segmentation to generate the key poses used in recognizing observed action input sequences. The results shown are the averaging of cross-fold validation as detailed in Section 4.2, over 3 replicate runs.

4.4.1 Single Person Action

The proposed method of obtaining key poses from segmented gestures achieved a global accuracy that improves upon the comparable k-means method outlined by [116]. The prior seg-

Table 4.2: Global recognition rate (%) across all action classes at final generation of evolutionary optimization.

Dataset	[116]	k-means	Proposed gesture key poses
MSR Action3D	88.56	96.30 ± 5.30	**97.42** ± **3.44**
SBU Kinect Interaction	-	83.30 ± 4.57	**83.92** ± **4.58**

mentation of class training samples is able to extract informative gestures from the action class, with subsequent clustering of within-gesture poses identifying poses that are able to more comprehensively describe the action classes observed. As expected, introduction of a new action class does have negative effect on recognition rates of the currently optimized population. This initially results in decreased accuracy due to random initialization of the genetic representation of the new action class. However, evolutionary optimization returns the population to an acceptable level, as seen by the increase in accuracy in following generations, Figure 4.8.

Prior clustering of the action class into cross-subject gestures works well to produce key poses for sequence alignment based classification. The evolutionary method compliments this by adapting to the introduction of new action classes, optimizing towards the most informative set of parameters for the model. The MSR Action3D dataset is a common dataset within the pose estimation community, and recognition rates of 97.4% on perceivably its most complex subset are an indicator of the benefit to using gesture segmentation in the identification of key poses. From Figure 4.9 we can see that common error lies in partitioning between the classes '*high arm wave*' and '*side boxing*', where each individual in the population was unable to classify one of the testing samples. There was also a smaller level of confusion in predicting between '*hand catch*' and '*side boxing*' classes. Surprisingly there was little confusion in the recognition of the classes '*draw x*', '*draw tick*' and '*draw circle*', those which we would presume to contain the most subtle action gestures. The large fall in classifier accuracy at generation 50 is coupled with the introduction of the '*draw x*' class, with the subsequent '*draw ...*' classes providing a similar fall in population accuracy at generations 100 and 150. This is reasonably acceptable due to the complexity of the classes, and the small number of frames that their gestures are comprised of; however, within a few generations the population has optimized the parameters and returned previous accuracy levels. In the case of introducing '*two hand wave*' and '*forward kick*' there is an increase in population accuracy upon learning the new classes, this suggests that the training samples have then provided some benefit to partitioning the previously learned actions, boosting the recognition of these classes.

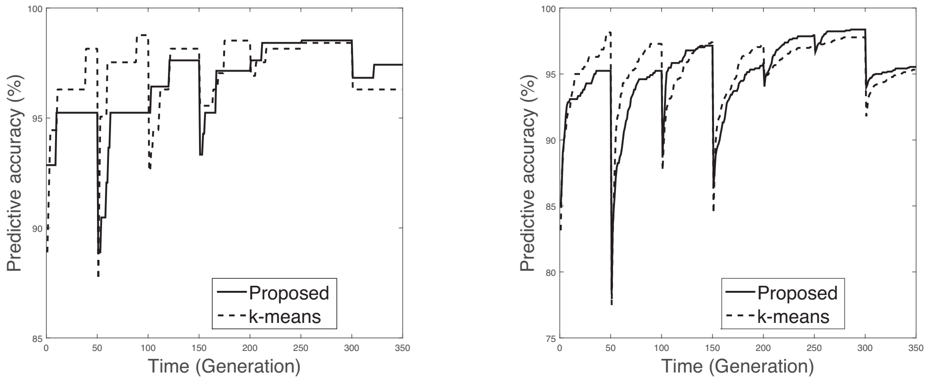
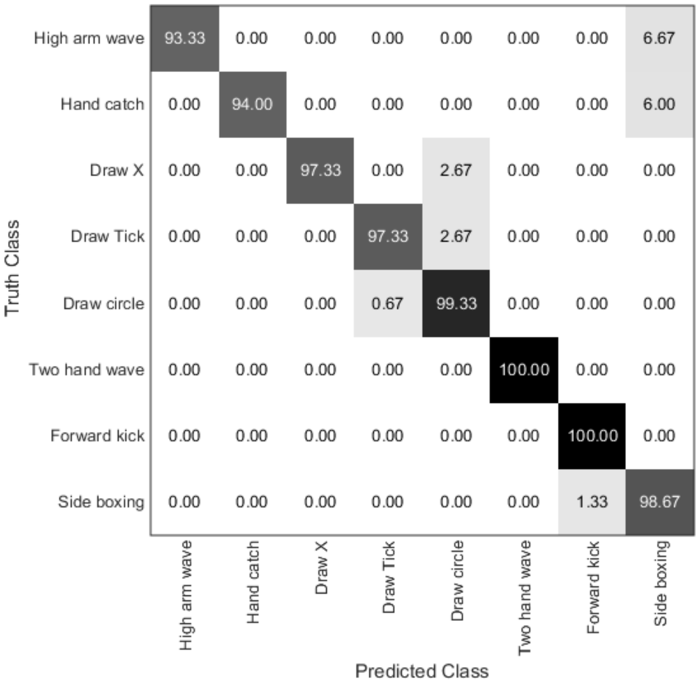


Figure 4.8: L-R: Maximum and Average predictive accuracy of population when classifying single person actions on the MSR AS2 dataset. A new class is introduced every 50 generations.

4.4.2 Two Person Interaction Recognition

Similar improvement over the use of standard key pose generation can be seen from the interaction recognition evaluation. Figure 4.10 shows that for the majority of the action classes observed the predictive accuracy is in excess of 95% when the bag of key poses has been generated using ACA. In both methods used, the recognition rate between the '*approaching*' and '*departing*' classes reached 100% within a small number of generations, if not immediately; this is believed to be due to the simple, almost polar opposite sequence of poses that are generated during the creation of the bag. Despite this issue being discussed in [108, 186, 387], we decided to keep these classes as part of recognition testing due to the need for adaptation with later introductions of unobserved classes. During the adaptation to new interaction classes, we observe a decrease in recognition accuracy as expected; however the drop in accuracy is not as noticeable as with the single action recognition. This may be due to the more simplistic classes provided by the SBU dataset, or due to the higher dimensional embedding of features. There was some difficulty for both methods to return to their previous level of accuracy once a new action class was introduced; although a small increase occurs within the allotted 50 generation time frame, the final prediction accuracy does not reach the standard it achieved before the introduction of the new class, as can be observed with the MSR action recognition. This could be due to the rate of mutation or the generation length being cut short. Despite this drop in accuracy we are still able to generate strong recognition accuracy on multiple complex pairwise interactions by first segmenting the action class into lower level gestures.



Truth Class \ Predicted Class	High arm wave	Hand catch	Draw X	Draw Tick	Draw circle	Two hand wave	Forward kick	Side boxing
High arm wave	93.33	0.00	0.00	0.00	0.00	0.00	0.00	6.67
Hand catch	0.00	94.00	0.00	0.00	0.00	0.00	0.00	6.00
Draw X	0.00	0.00	97.33	0.00	2.67	0.00	0.00	0.00
Draw Tick	0.00	0.00	0.00	97.33	2.67	0.00	0.00	0.00
Draw circle	0.00	0.00	0.00	0.67	99.33	0.00	0.00	0.00
Two hand wave	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00
Forward kick	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00
Side boxing	0.00	0.00	0.00	0.00	0.00	0.00	1.33	98.67

Figure 4.9: Confusion matrix of single person action class recognition. Values shown are predictive rates for the final generation of optimization.

4.5 Summary

This study has shown that key pose generation benefits from the initial segmentation of lower-level gestures from all observed training instances. This identifies key temporal sub-actions across instances of an action class, before then using these segments for the generation of the key poses. The use of aligned cluster analysis has allowed us to extract common gesture sequences from across all training observations by sequence alignment with Dynamic Time Warping. This segmentation has then in turn been utilized to create a bag of key poses that is able to accurately recognize action classes on both a single person, and two-person interaction level. Although this method generates significantly more key poses for the bag, it is these informative poses that are able to assist in classifying new observations in the scene by describing gestures that are repeatedly observed across numerous instances. The use of ACA segmentation to generate key pose representations has benefits in the recognition of pairwise interactions between two individuals, providing an increase in the correct prediction through

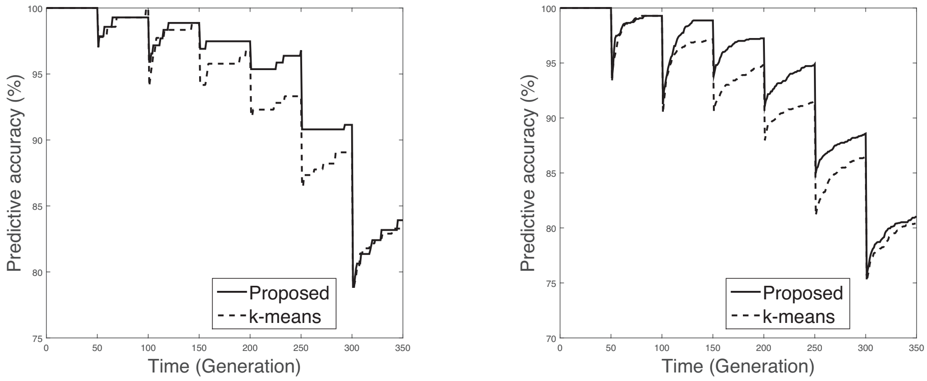


Figure 4.10: L-R: Maximum and Average predictive accuracy of population when classifying two person pairwise interactions. A new class is introduced every 50 generations.

use of key poses. Although the initial accuracy of the classifier is variable, the evolutionary optimization of the tuning parameters is able to increase the predictive accuracy over time.

The understanding of the underlying gestures are key to recognizing actions, as has been demonstrated by the use of key poses, sequences of key poses, bag of key pose, and sequence alignment techniques that have come to fruition over recent years. Further understanding of how an action execution can be comprised of gestures that are global across both subjects and observations will help to identify which portions of an event are beneficial to the partitioning of the action space.

In terms of performance; the number of key poses that this method creates is large, creating *k* key poses for each of the segmented gesture clusters. Therefore a reduction in the number of key poses that represent each segment cluster may be beneficial to the overall accuracy and speed of the system. Just as with the selection of training parameters, the use of evolutionary programming may guide the selection of an optimum ACA segmentation. The observed accuracies are acceptable for the HAR domain, and especially when considering the recognition of interactions between two individuals, in which level of variation in execution can vary on a large scale and the class labeling is broadly generalized.

The method also optimizes each of the classes as they are introduced to the system, and we can observe an initial drop in performance at the introduction of each new class. A population will optimize its representation sampling parameters for the current set of observed classes, and as a new class is introduced we observe a drop in accuracy, this is to be expected, given the need to classify a previously unobserved behavior. The population evolutionary operators allow for mutations to produce individuals that assist in searching the parameter space. The introduction of new classes leads to a new function space on which the population optimize their parameters, and the population genetics will drift accordingly, reducing the chance of stagnation in a population when faced with new classes and samples.

In the following chapters we look to more comprehensive methods of representation learning, focusing on the use of deep learning as a means to generating feature descriptors from observed data. In Chapter 5 we present a method for learning features in domains which exhibit an irregular spatial topology, such as the skeletal model explored here. Chapters 6 and 7 explore the use of such methods in learning features across multiple scales and from temporal observations, with Chapter 7 returning the problem of HAR to evaluate the use of irregular spatio-temporal learning in action classification.

Chapter 5

Deep Learning in Irregular Domains

Contents

5.1 Introduction

In the previous chapter we have looked at one method for representation learning in the context of Human Action Recognition, utilizing an evolutionary approach to training sample, feature set, and clustering parameter selection in order to segment actions into primitive gestures. In the following chapters we will move towards the use of deep learning as a representation learning approach, introducing the use of irregular domain based convolutional neural networks. During this chapter we will explore the development of generalized operators for convolution and pooling on a graph representation of the irregular spatial domain, providing evaluation on signal classification tasks, before moving on in later chapters to learn multi-scale features and temporal informations

In recent years, the machine learning and pattern recognition community has seen a resurgence in the use of neural network and deep learning architecture for the understanding of classification problems. Standard fully connected neural networks have been utilized for domain problems within the feature space with great effect, from text document analysis to genome characterization [408]. The introduction of the Convolutional Neural Network (CNN) provided a method for identifying locally aggregated features by utilizing kernel filter convolutions across the spatial dimensions of the input to extract feature maps [11]. Applications of CNNs have shown strong levels of recognition in problems from face detection [409], digit classification [410], and classification on a large number of classes [411].

The core CNN concept introduces the hidden convolution and pooling layers to identify spatially localized features via a set of receptive fields in kernel form. The convolution operator takes an input and convolves kernel filters across the spatial domain of the data provided some stride and padding parameters, returning feature maps that represent response to the filters. Given a multi-channel input, a feature map is the summation of the convolutions with separate kernels for each input channel. In CNN architecture, the pooling operator is utilized to compress the resolution of each feature map in the spatial dimensions, leaving the number of feature maps unchanged. Applying a pooling operator across a feature map enables the algorithm to handle a growing number of feature maps and generalizes the feature maps by resolution reduction. Common pooling operations are that of taking the average and max of receptive cells over the input map [336].

Due to the usage of convolutions for the extraction of partitioning features, CNNs require an assumption that the topology of the input dimensions provides some spatially regular sense of locality. Convolution on the regular grid is well documented and present in a variety of CNN implementations [412, 413], however when moving to domains that are not supported by the regular low-dimensional grid, convolution becomes an issue. Many application domains utilize irregular feature spaces [414], and in such domains it may not be possible to define a spatial kernel filter or identify a method of translating such a kernel across spatial domain. Methods of handling such an irregular space as an input include using standard neural networks, embedding the feature space onto a grid to allow convolution [29], identifying local patches on the irregular manifold to perform geodesic convolutions [359], or graph signal processing based convolutions on graph signal data [415]. The potential applications of a convolutional network in the spatially irregular domain are expansive, however the graph convolution and pooling is not trivial, with graph representations of data being the topic of on-going research [375, 416]. The use of graph representation of data for deep learning is introduced by [417], utilizing the Laplacian spectrum for feature mining from the irregular domain. This is further expanded upon in [415], providing derivative calculations for the backpropagation of errors during gradient descent. We formulate novel gradient equations that show more stable calculations in relation to both the input data and the tracked weights in the network.

In this methodology-focused study, we explore the use of graph-based signal-processing techniques for convolutional networks on irregular domain problems. We evaluate the effects of using interpolation in the spectral domain for identifying localized filters and we present the use of Algebraic Multigrid (AMG) node agglomeration for graph pooling. We have also identified an alternative to the gradient calculations of [415], by formulating the gradients in regards to the input data as the spectral convolution of the gradients of the output with the filters (Equation 5.3), and the gradients for the weights as the spectral convolution of the input and output gradients (Equation 5.4). These proposed gradient calculations show consistent stability over previous methods, which in turn benefit the gradient-based training of the network. Results are reported on the MNIST dataset on both the regular 2D grid, and an irregularly sampled variant of the grid.

The rest of the chapter is outlined as follows. Section 5.2 describes the generation of a graph-based CNN architecture, providing the convolution and pooling layers in the graph domain by use of signal-processing on the graph. Section 5.3 details the experimental evaluation of the proposed methods and a comparison against the current state of the art, with Section 5.4 reporting the results found and conclusions drawn in Section 5.5.

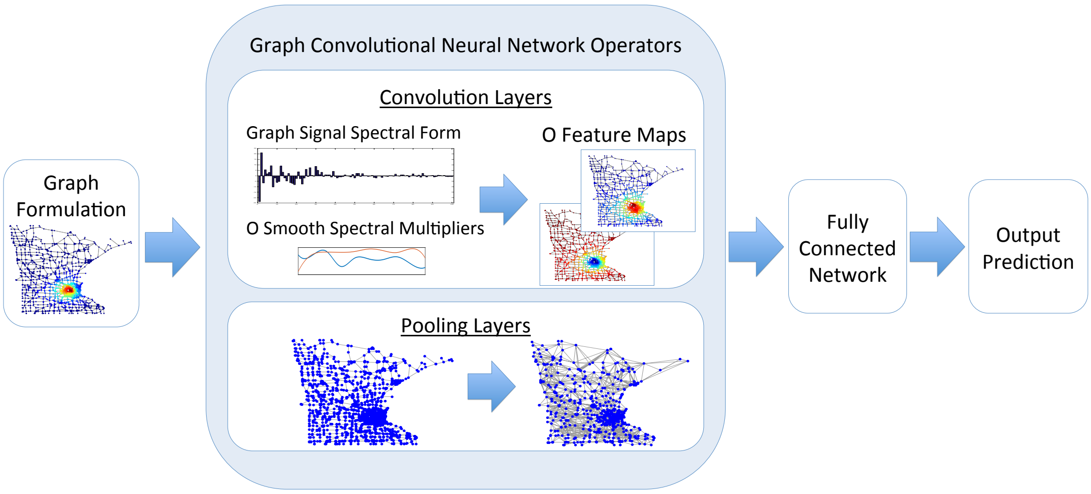


Figure 5.1: Graph based Convolutional Neural Network components. The Graph-CNN is designed from an architecture of graph convolution and pooling operator layers. Convolution layers generate O output feature maps dependent on the selected O for that layer. Graph pooling layers will coarsen the current graph and graph signal based on the selected vertex reduction method.

5.2 Proposed Approach

The familiar CNN architecture pipeline consists of an input layer, a collection of convolution and/or pooling layers followed by a fully connected neural network and an output prediction layer. One issue with CNNs is that the convolution of a filter across the spatial domain is non-trivial when considering domains in which there is no regular structure. One solution is to utilize the multiplication in the spectral graph domain to perform convolution in the spatial domain, obtaining the feature maps via graph signal processing techniques. The graph-based CNN follows a similar architecture to standard CNNs; with randomly initialized spectral multiplier based convolution learned in the spectral domain of the graph signal and graph coarsening based pooling layers, see Figure 5.1 for a pipeline. Training is compromised of a feed-forward pass through the network to obtain outputs, with loss propagated backwards through the network to update the randomly initialized weights.

The topic of utilizing graphs for the processing of signals is a recently emerging area in which the graph G forms a carrier for the data signal x [360]. The graph holds an underlying knowledge about the spatial relationship between the vertices and allows many common signal processing operators to be performed upon x via G, such as wavelet filtering, convolution, and

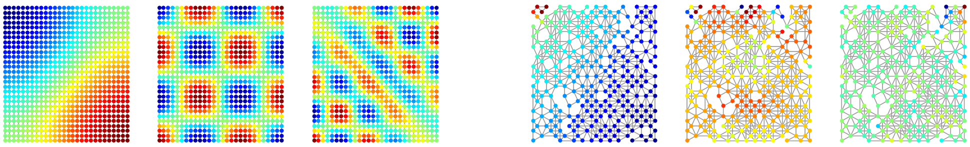


Figure 5.2: The 2nd, 20th, and 40th eigenvectors of the full 28×28 regular gird (left) and the irregularly sampled grid (right).

Fourier Transform [360, 364]. By representing the observed domain as a graph it is possible to perform the signal processing operators on the observed data as graph signals. Coupling these graph signal processing techniques with deep learning it is possible to learn within irregularly spaced domains, upon which conventional CNNs would be unable to convolve a regular kernel across. The proposed technique will therefore open the door for deep learning to be utilized by a wider collection of machine learning and pattern recognition domains with irregular, yet spatially related features.

5.2.1 Convolution on Graph

A graph $G = \{V, W\}$ consists of N vertices V and the weights W of the undirected, non-negative, non-selflooping edges between two vertices v_i and v_j. The unnormalized graph Laplacian matrix L is defined as $L = D - W$, where $d_{i,i} = \sum_{i=1}^{N} w_i$ forms a diagonal matrix containing the sum of all adjacent weights for a vertex. Given G, an observed data sample is a signal $x \in \mathbb{R}^N$ that resides on G, where x_i corresponds to the signal amplitude at vertex v_i. The normalized Laplacian $\tilde{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ is an alternative to the non-normalized Laplacian which enables normalization of the edge weights in A.

Convolution is one of the two key operations in the CNN architecture, allowing for locally receptive features to be highlighted in the input image [11]. A similar operator is presented in graph-based CNN, however due to the potentially irregular domain graph convolution makes use of the convolution theorem, where convolution in the spatial domain is approximated by element-wise multiplication in the frequency domain [365].

To project the graph signal into the frequency domain, the Laplacian L is decomposed into a full matrix of orthonormal eigenvectors $U = \{u_{l=0\ldots N-1}\}$, where u_l is a column of the matrix U, and the vector of associated eigenvalues $\lambda_{l=0\ldots N-1}$ [360], Figure 5.2. Using the matrix U, the graph Fourier transform is defined as $\tilde{x} = U^T x$, and the inverse as $f = U\tilde{x}$, where U^T is the transpose of the eigenvector matrix.

For forward convolution, a convolutional operator in the vertex domain can be composed as a multiplication in the Fourier space of the Laplacian operator [365]. Given the spectral form of our graph signal $\tilde{x} \in \mathbb{R}^N$ and the spectral multiplier $k \in \mathbb{R}^N$, the convolved output signal in the original spatial domain is the spectral multiplication, i.e. $y = U\tilde{x} \odot k$. It is possible to expand this for multiple input channels and multiple output feature maps:

$$y_{s,o} = U \sum_{i=1}^{I} U^T x_{s,i} \odot k_{i,o} \ , \tag{5.1}$$

where I is the number of input channels for x, s is a given batch sample, and o indexes an output feature map from O output maps.

One issue with the above formulation of the filter k is in the use of a spectral multiplier vector of length N, which provides a filter with an independent parameter for each Laplacian eigenvector of the graph. This not only provides a parameter complexity of $\mathscr{O}(n)$ per input map per output map per layer, but it also does not guarantee a localization of filters in the spatial domain. Localized regions in the spatial domain are defined by the kernel receptive field in CNNs, and for graph-based CNNs the spatial vertex domain localization is given by a smoothness within the spectral domain, given by the Parseval Identity,

$$\int_{-\infty}^{+\infty} |x|^{2k} |f(x)|^2 dx = \int_{-\infty}^{+\infty} \left| \frac{\partial^k \hat{f}(\omega)}{\partial \omega^k} \right|^2 d\omega \tag{5.2}$$

Therefore to identify local features within the spatial domain, the spectral multipliers used for spectral convolution are identified by tracking a subsampled set of filter weights $\hat{k}_{i,o}k \in \mathbb{R}^{<<N}$ which are interpolated up to a full filter via a smoothing kernel Φ such as cubic splines: $k_{i,o} = \Phi \hat{k}_{i,o}$. This has the added benefit of reducing the number of tracked weights, reducing the complexity to $\mathscr{O}(k << n)$ as with CNNs. Reducing the number of tracked weights increases the smoothness of the final interpolated filter, and as such provides the localized filtering required in the graph spatial domain.

5.2.2 Backpropagation on Graph

Backpropagation of errors is a pivotal component of deep learning, providing updates of weights and bias for the networks towards the target function with gradient descent. This requires obtaining derivatives in regards to the input and weights used to generate the output, in the case of graph-based CNN convolution the gradients are formulated in regards to the graph signal x and the spectral multipliers k. The gradients for an input feature map channel

$x_{s,i}$ is given as the convolution of the gradients for the output ∇y and the spectral multipliers in the spectral domain via

$$\nabla f_{s,i} = U \sum_{o=1}^{O} U^T \nabla y_{s,o} \odot k_{i,o} \tag{5.3}$$

for a provided batch of S graph signals. Gradients for the full set of interpolated spectral multipliers is formulated as the convolution of the gradients for the output ∇y with the input $x_{s,i}$ via

$$\nabla k_{i,o} = \sum_{s=1}^{N} U^T \nabla y_{s,o} \odot U^T x_{s,i}. \tag{5.4}$$

As the filters are spectral domain multipliers, we do not project this spectral convolution back through the graph Fourier transform. The smooth multiplier weights ∇k can then be projected back to the subsampled set of tracked weights by the multiplication with the inversed smoothing kernel $\nabla \hat{k}_{i,o} = \Phi^T \nabla k_{i,o}$.

5.2.3 Pooling on Graph

The pooling layer is the second key component in conventional CNNs, reducing the resolution of the input feature map in both an attempt to generalize the features identified and to manage the memory complexity when using numerous filters [336]. Such pooling operations stride across the regular spatial domain of the input feature map with an appropriate max or mean operator filtering the underlying receptive cell to produce a coarsened resolution map as output. Such pooling operations provide two main benefits, firstly the memory and computational complexity for convolution is reduced for smaller sized feature maps, secondly the learned features are generalized by compression of feature map resolution [336]. The standard CNN pooling operator maintains the spatial regularity of the domain, taking a Cartesian grid as input and returning a Cartesian grid feature map as output. During graph-based convolutions there is no reduction in size between the input signal and the output feature map due to the elementwise multiplication of the $\mathbb{R}^N$ filter with the $\mathbb{R}^N$ spectral signal. As such, each layer of a deep graph CNN would possess a graph with N vertices and numerous feature maps of length N, leading to scaling inefficiencies without a pooling operation. Since the graph convolution requires a fixed Fourier basis for the Graph Fourier Transform (GFT) it is possible to pre-compute the required graphs for the architecture before training and look them up for convenience, however the complexity of the forward and reverse Graph Fourier Transforms is directly linked to the number of vertices within the graph. If pooling is utilized, there is benefit gained from feature map generalization and reduction in complexity of the graph

Fourier transforms as each layer's vertex count N is lowered. To pool local features together on the graph, it is required to perform graph coarsening and project the input feature maps to reside on the reduced size graph. Coarsening $G = \{V, W\}$ to $\hat{G} = \{\hat{V}, \hat{W}\}$ not only requires the reduction of vertex counts, but also a handling of edges between the remaining $\hat{N}$ vertices and the observed graph signals. Common methods of generating $\hat{V}$ are to either select a subset of V to carry forward to $\hat{G}$ [384] or to form completely new set of nodes $\hat{V}$ from some aggregation of related nodes within V [371].

Selecting a collection of vertices to keep in the coarsened graph can take several forms, including a selection criteria based on the polarity of the eigenvector associated with the largest eigenvalue, $\hat{V} = \{U_{N,i}\}; U_{N,i} >= 0$, or the use of spectral clustering of the vertices via k-medoids over the eigenvectors. By utilizing the eigenvectors corresponding to the k largest eigenvalues we can group the spectral representation of the graph into k clusters, identifying k nodes in V to select for $\hat{V}$.

5.2.3.1 Kron's reduction

Kron's reduction, [381, 418], is the generation of a new coarsened graph $\hat{G}$ from G via the original finer graph Laplacian, some selected vertices $\hat{V}$, and their complement $\hat{V}^c$. Kron's reduction of the Laplacian

$$\hat{L} = L_{\hat{V},\hat{V}} L_{\hat{V},\hat{V}^c} - L_{\hat{V}^c,\hat{V}^c} L_{\hat{V}^c,\hat{V}} \tag{5.5}$$

provides a means to then reconstruct the reduced node weight matrix $\hat{W}$ from the removal of the discarded vertices from the rows and columns of L

$$\hat{W}_{n,m} \begin{cases} -\hat{L}_{n,m} & \text{for} \quad n \neq m \\ 0 & \text{for} \quad i = j \end{cases} \tag{5.6}$$

A selection of $\hat{V}$ is made by identifying the largest eigenvalue λ_N and splitting V into two subsets based on the polarity of the associated eigenvector U_N [382]. We can therefore define

$$\hat{V} = \{u_N, i\}; u_N, i <= 0 \tag{5.7}$$

and its complement

$$\hat{V}^c = \{u_N, i\}; u_N, i > 0 \tag{5.8}$$

These selections for the pooled vertices are then used in (5.5) to construct $\hat{G}$, although by reversing the selection for the polarity to keep it is just as understandable to choose $\hat{V}^c$ for

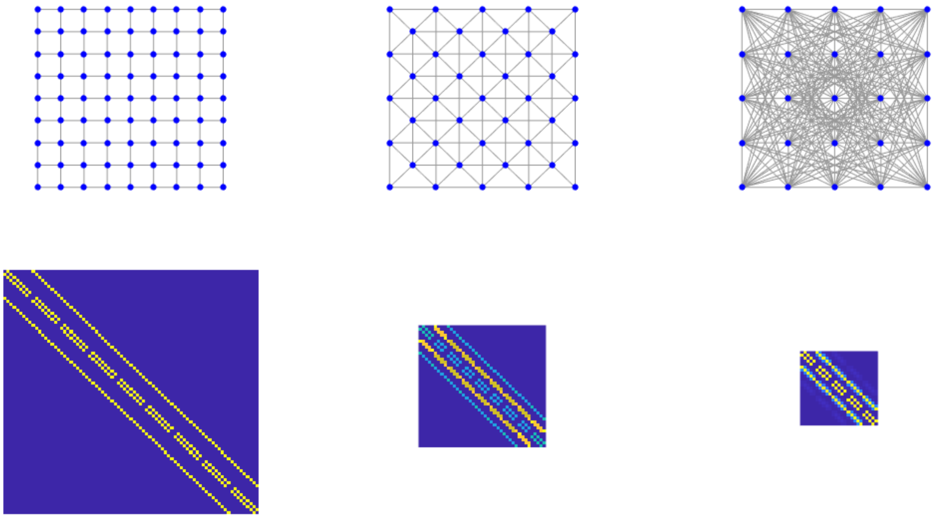


Figure 5.3: Left to right: Increasing levels of Kron's reduction of the 2D Grid via (5.5) on the Laplacian. Note the impact reducing the Laplacian has on the spatial structure of the original graph, reducing the vertex count but increasing the edge connectivity.

construction of $\hat{G}$. Kron's reduction has the effect of increasing the number of edge connections present in the graph, and as such it is often necessary to sparsify the connectivity in the graph by way of spectral sparsification [384, 419]. Given the original graph G and the edge selection tuning parameter Q we initialise the weight matrix $\hat{W}$ for the subgraph $\hat{G}$ to 0 for all vertex pairings $w_{n,m}$. Then we select Q random edges $e(n,m) \in \varepsilon$ from the original weight matrix W given a probability

$$p_e = \frac{\delta(n,m)W_{n,m}}{\sum_{e=(\alpha,\beta)\in\varepsilon}^{\varepsilon} \delta(\alpha,\beta)W_{\alpha,\beta}} \tag{5.9}$$

which is then accumulated into the new graph's weight matrix

$$\hat{W}_{n,m} = \hat{W}_{n,m} + \frac{W_{n,m}}{Qp_e} \tag{5.10}$$

for the coarser graph layer. This has been shown by [419] to be a useful coarsening method on larger graphs, maintaining localized structure relationships from the original graph.

With a coarser graph structure $\hat{G}$ it is required to then down-sample the graph signal $f_{1:N}$ into a new signal $\hat{f}_{1:\hat{N}}$ that is able to reside on $\hat{G}$. We down-sample $f \in \mathbb{R}^N$ on G to $\hat{f} \in \mathbb{R}^{\hat{N}}$ on $\hat{G}$ by pyramid analysis interpolation. Kron's pyramid utilizes a linear application of Green's functions derived from the Laplacian to interpolate the signal about a given vertex v_n in the

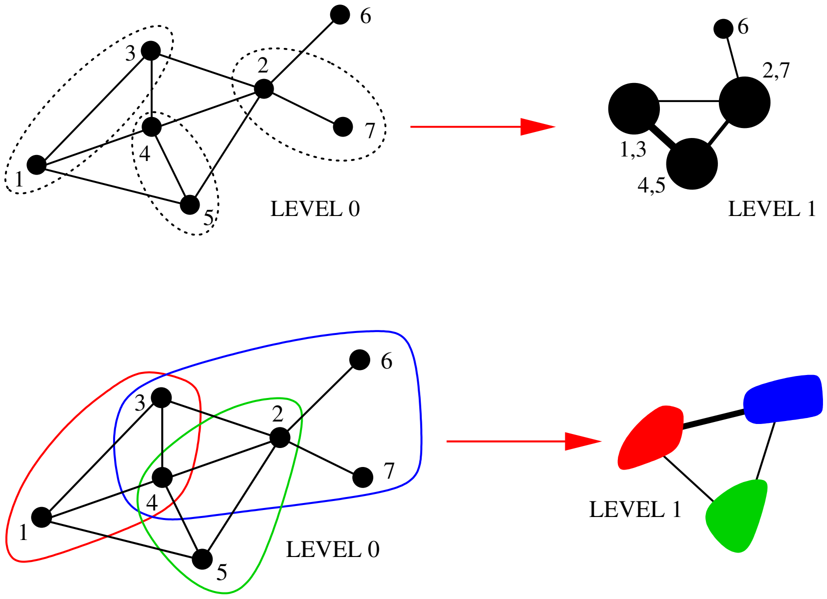


Figure 5.4: Two agglomerative pooling approaches: Strict Aggregation (SAG, top) and Weighted Aggregation (WAG, bottom). SAG merges disjoint pairs of nodes into a single node in the coarser graph. WAG can utilize non-disjoint subsets of vertices, allowing a vertex in a finer representation to be divided amongst several coarser level nodes. Image from [371].

spatial domain [384]. This allows a Kron's Pyramid to project our samples from fine to coarse resolutions during forward passes through the network, and from coarse to fine scale during the backpropagation learning.

5.2.3.2 AMG

An alternative approach to graph coarsening is to utilize agglomerative methods which look to combine vertices in the finer graph into singular vertices in the coarser approximation by contraction in the spatial domain. In this study we utilize AMG for graph coarsening, a method of projecting signals to a coarser graph representation obtained via greedy selection of vertices [371, 420]. Aggregation takes a subset of vertices on V and generates a singular vertex in the new set of coarsened nodes $\hat{V}$ in the output graph.

With a coarser graph structure $\hat{G}$ it is required to down-sample the graph signal $x_{1:N}$ into a new signal $\hat{x}_{1:n}$ that is able to reside on $\hat{G}$. AMG provides a set of matrices for the interpolation

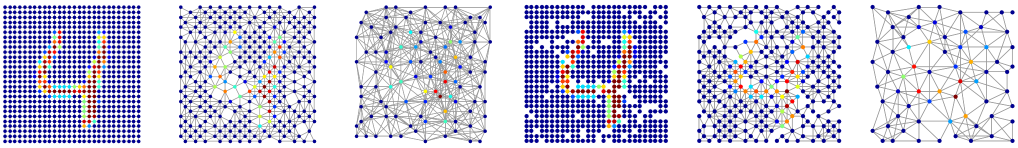


Figure 5.5: Two levels of graph pooling operation on regular and irregular grid with MNIST signal. From left: Regular grid, AMG level 1, AMG level 2, Irregular grid, AMG level 1, AMG level 2.

of the input signal f; the restriction matrix R and the projection matrix P. Down-sampling $x \in \mathbb{R}^N$ on G to $\hat{x} \in \mathbb{R}^{\hat{N}}$ on $\hat{G}$ is achieved by the multiplication of the signal with the restriction matrix, $\hat{x}_{s,i} = Rx_{s,i}$, whilst the reverse pooling required for backpropagation is achieved via multiplication with the projection matrix, $x_{s,i} = P\hat{x}_{s,i}$.

5.3 Application

Although we utilize forms of the 2D grid, both regular and irregularly sampled, the graph CNN is generalizable to more irregular domain problems; such as sensor networks, mesh signals, text corpora, human skeleton graphs and more. These domains quite often contain irregular spatial geometries, upon which it is non-trivial to define a filter kernel for convolution. In this study we evaluate the performance of the proposed graph CNN with an implementation on both the standard regular 28 grid and the irregularly sampled 2D grid.

The 2D grid is the graph representation of the Von Neumann neighborhood of vertices in a regular domain, most commonly applied to that of pixel relationships in images. Such a spatial domain utilizes an underlying correlation within localized pixel neighborhoods and is a staple of the CNN methodology, with an assumption being made that the input domain resides upon a grid domain across which a fixed size kernel can be convolved and optimized. Often the input domain is represented as multi-dimensional array or tensor [320, 412], however the grid can of course be formulated as a graph, where each pixel is represented by a vertex on G. Edge weights are taken as the Euclidean distance between the nodes in the Von Neumann neighborhood, connecting neighboring vertices. The intensities at a given pixel form a channel amplitude for each vertex, forming the graph signal x. In this form a graph represents the spatial relationship amongst the elements within the observed signal, based on some graph construction approach. To evaluate the performance of graph CNN on the 2D grid we utilize the MNIST dataset, consisting of 60,000 examples of handwritten numerical digits in 28×28 grayscale pixel images. The edge weights for G are the binary presence of an edge between v_i and v_j on the 4-way adjacency, with $V \in \mathbb{R}^{784}$.

To evaluate the suitability of the Graph-CNN approach and it's ability to learn localized filters on an irregular input domain we artificially subsample the 2D grid domain for the MNIST dataset. A conventional CNN cannot convolve across an irregular spatial geometry, and by removing pixels from the grid we are able to produce a situation in which we have spatial information, but without the ability to utilize CNNs with embedding or artificially resampling the input space. We subsampled the 28×28 grid by selecting a number of random vertices to exclude from the grid, in this implementation we drop 84 vertices to create a irregular spatial domain. Upon removing the selected vertices and their corresponding edges from the graph, we then subsample the MNIST dataset with the respective signals such that $x \in \mathbb{R}^{700}$. This irregular spatial domain now requires the graph-based CNN operators above to form a convolved output feature map. The choice of vertices to drop, both their number and location, is arbitrary and only serves to create a space in which spatial distribution of information is still relevant but shows an irregular form. Reduced spatial sampling rates, whether regular or irregular, just serves to more coarsely approximate the underlying domain, and removing more or less pixels from the grid affects the underlying representation of the domain. Once we have a toy example with an irregular domain we are inhibited from using CNNs.

The architecture of the graph CNN was set to $C^{20}PC^{50}PRF$; where C^{κ} defines a convolutional layer with 60 tracked weights and κ output feature maps, P defines an AMG pooling with a coarsening factor of $\beta = 0.05$ and 2 levels, R defines a rectified linear unit layer, and finally F describes fully connected layers providing output class predictions. Networks were trained for 500 epochs, with the full 10,000 validation samples being classified at each epoch to track the predictive performance of the network. A fixed learning rate of 10^{-3} was used in combination with a mini-batch gradient descent optimization approach, with a batch size of 32 and updates made using the average gradient of the mini-batch [328]. Although an empirical search of the hyper-parameter is possible to optimize the network architectures, the aim of this study is to explore the use of graph-based convolutional neural network operators.

To perform derivative checking, the calculation of the gradients for ∇x, ∇k and $\nabla \hat{k}$ were evaluated using random perturbations of errors on the scale of 10^{-4}. Derivatives for $\nabla \hat{k}$ were checked for interpolation over varying numbers of tracked weights in the network, including the full set $\hat{N} \in \mathbb{R}^{N}$. The experiment was repeated 100 times and the average percentage error

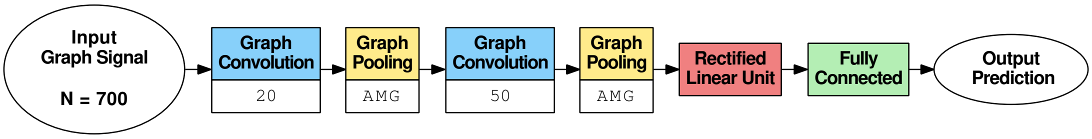


Figure 5.6: Graph-CNN architecture for classification of irregularly sampled MNIST images.

of the calculated gradient versus the empirically obtained gradient is reported in Figure 5.9.

The graph-based CNN architecture was implemented within MATLAB using GPU enabled operators.

5.4 Comparative Analysis and Results

The graph CNN method was evaluated on both the regular 28 × 28 grid and randomly sampled grid. We report the predictive accuracy of the network at each epoch of training using both the proposed graph CNN method and the method proposed by [415]. We also show the effects of smoothed spectral multiplier filters on the convolution output and the derivative errors we obtained for gradient calculations. In summary we found that by increasing the smoothness of the spectra filters we were able to increase the local relationship of features in the spatial domain, however this also resulted in higher error being introduced by the interpolation when calculating the gradients of the tracked weights $\hat{k}$. Overall we found that the proposed calculations for derivatives in respect to $\hat{k}$ introduced little error during backpropagation. The accuracy observed when testing unobserved samples is very promising, exceeding 94% on both the regular and irregular geometry domains.

5.4.1 Convolution and filter smoothness

Reducing the number of tracked filter weights produces smoother spectral multipliers after interpolation up to $k_{i,o} = \Phi\hat{k}_{i,o}$. Figure 5.7 shows the effect of interpolating weights from various lengths of $\hat{k}$ as applied to the 2D graph with the Cameraman.tif graph signal residing on it. As the number of tracked weights is reduced the spatial locality of the features learned is reduced, providing sharper features, whilst as the number of tracked weights approaches N the spatial localization of the feature map is lost.

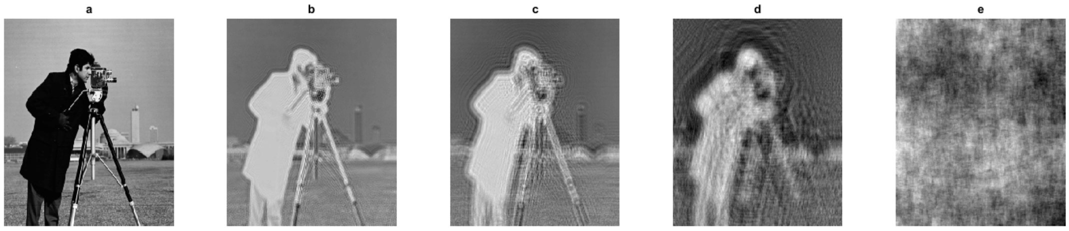
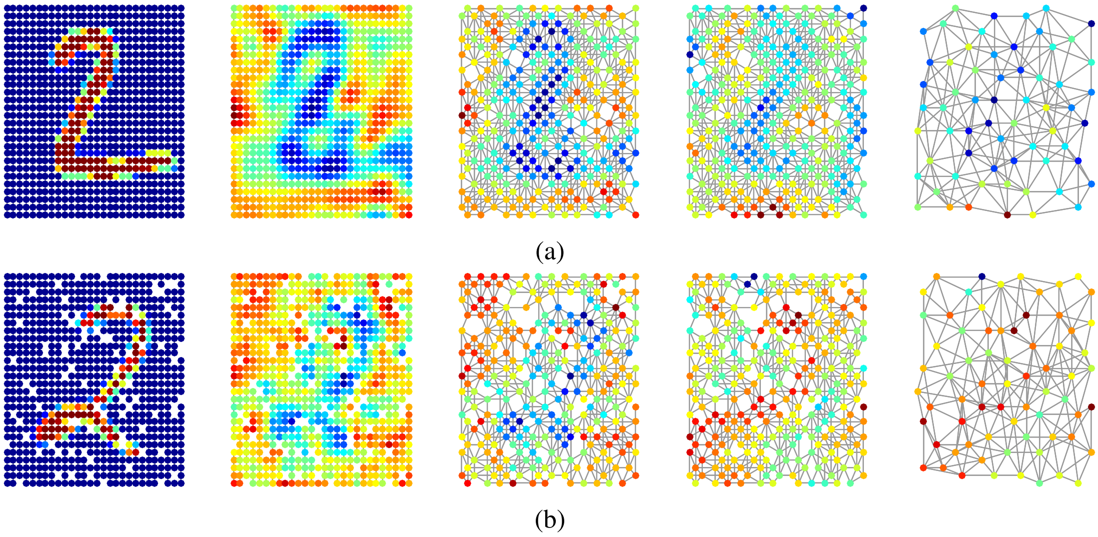


Figure 5.7: Effect of spline interpolation of tracked spectral weights on filter smoothness during graph-based convolution. a) Original image, b) $\hat{k} = \mathbb{R}^{\text{ceil}(\sqrt[4]{N})}$, c) $\hat{k} = \mathbb{R}^{\text{ceil}(\sqrt[3]{N})}$, d) $\hat{k} = \mathbb{R}^{\text{ceil}(\sqrt[2]{N})}$, e) $\hat{k} = \mathbb{R}^{N}$.


(a)

(b)

Figure 5.8: Feature maps formed by a feed-forward pass of the irregular domain. From left: Original image, Convolution round 1, Pooling round 1, Convolution round 2, Pooling round 2. a) Regular 2D grid, b) Irregularly sampled 2D grid.

5.4.2 Localized feature maps

By interpolating smooth spectral multipliers from the 60 tracked weights we were able to convolve over the irregular domain to produce feature maps in the spatial domain with spatially localized features. Figure 5.8 visualizes output for each layer of the Graph CNN convolution and pooling layers for both the regular and irregular domain graphs.

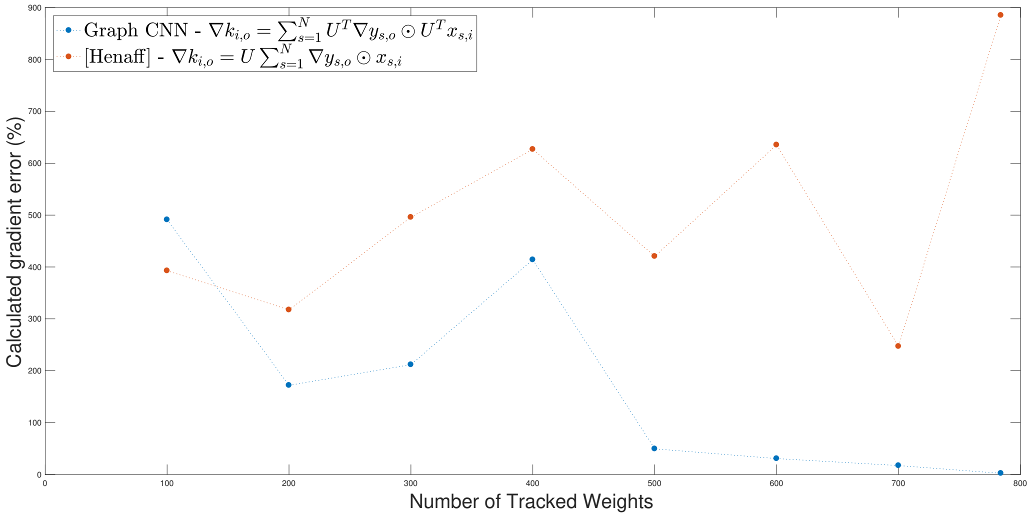


Figure 5.9: Effect of interpolation on calculation of gradients for tracked weights in spectral filters.

5.4.3 Backpropagation derivative checks

The proposed method gave an average of $1.41\%(\pm 4.00\%)$ error in the calculation of the gradients for the input feature map. In comparison, by not first applying a graph Fourier transform to $\nabla y_{s,o}$ in the calculation for $\nabla x_{s,i}$, as in [415], we obtain errors of $376.50\%(\pm 1020.79\%)$. Similarly the proposed method of obtaining the spectral forms of $\nabla y_{s,o}$ and $x_{s,i}$ in the calculation of $\nabla k_{i,o}$ gave errors of $3.81\%(\pm 16.11\%)$. By not projecting to the spectral forms of these inputs, errors of $826.08\%(\pm 4153.32\%)$ are obtained. Figure 5.9 shows the average percentage derivative calculation error for $\nabla \hat{k}$ of varying numbers of tracked weights over 100 runs. The proposed method of gradient calculation shows lower errors than the compared method gradient calculation of ∇k when $\hat{k} \in \mathbb{R}^N$ and all but the lowest number of tracked weights of $\hat{k} \in \mathbb{R}^{100}$. The introduction of interpolation leads to a higher introduction of error into the calculated gradient errors, especially in the presence of a low number of tracked weights.

5.4.4 Testing performance

Classification performance on the MNIST dataset is reported in Table 5.1, with progression of testing accuracy over epochs given in Figure 5.10 comparing between the proposed gradient calculations and those of [415]. The proposed graph CNN method does not obtain the

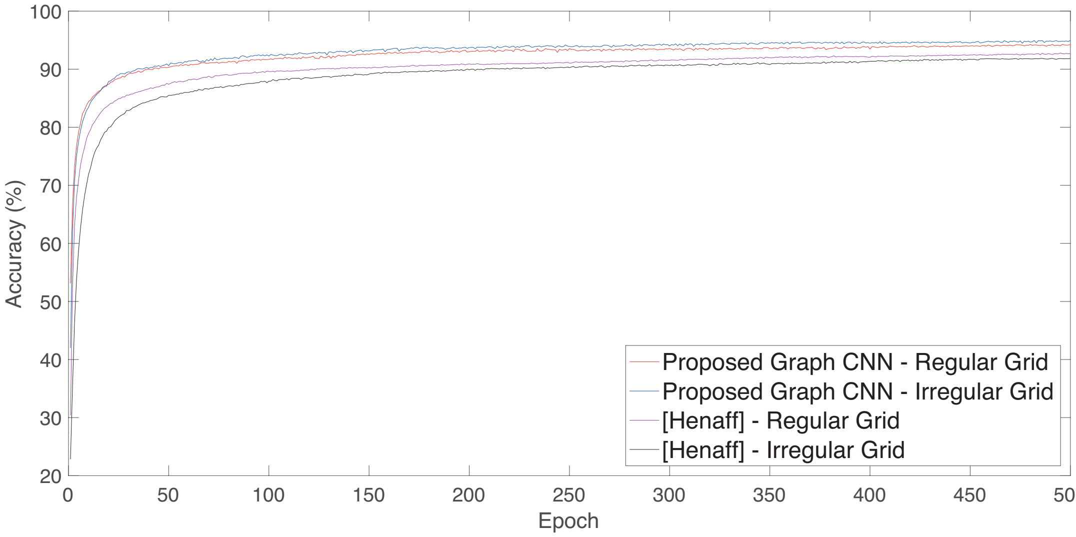


Figure 5.10: Test Set accuracy on the MNIST dataset on the regular and irregular 2D grid. An increasing in testing accuracy is observed when utilizing the proposed gradient calculations from equations 5.3 and 5.4.

Table 5.1: Testing set accuracy of network (%)

Dataset	CNNs [410]	[415]	Proposed Graph CNN
Regular grid MNIST	99.77	92.69	**94.23**
Irregularly sampled grid MNIST	-	91.84	**94.96**

99.77% accuracy rates of the state of the art CNN architecture presented by [410] on the full 28×28 grid. This is understandable, as standard CNNs are designed to operate in the regular Cartesian space, giving it a strong performance in the image classification problem. The main benefit of the graph CNN is in its ability to handle the irregular spatial domain presented by the subsampled MNIST grid by use of convolution in the graph spectral domain.

5.5 Summary

This study proposes a novel method of performing deep convolutional learning on the irregular graph by coupling standard graph signal processing techniques and backpropagation based neural network design. Convolutions are performed in the spectral domain of the graph Laplacian and allow for the learning of spatially localized features whilst handling the non-trivial irregular kernel design. Results are provided on both a regular and irregular domain classification problem and show the ability to learn localized feature maps across multiple layers of a network. A graph pooling method is provided that agglomerates vertices in the spatial domain to reduce complexity and generalize the features learned. GPU performance of the algorithm improves upon training and testing speed, however further optimization is needed. Although the results on the regular grid are outperformed by standard CNN architecture this is understandable due to the direct use of a local kernel in the spatial domain. The major contribution over standard CNNs is the ability to function on the irregular graph is not to be underestimated. Graph based CNN requires costly forward and inverse graph Fourier transforms, and this requires some work to enhance usability in the community. On-going study into graph construction and reduction techniques is required to encourage uptake by a wider range of problem domains.

In the coming chapters we will explore the use of Graph-based CNNs in learning features from multiple scales and from temporal information on irregular spatial domains. A multiscale sampling approach results in a singular irregularly spaced embedding of local and global features, upon which a standard Convolutional Neural Network would not be possible. Local and global information is used in a detection problem to accurately and stably identify small-scale structures in the wider context of placement in a larger volume. A Graph-CNN is then defined to learn features on spatio-temporal motion information of the human skeletal model for the purpose of human action recognition.

Chapter 6

Graph Convolutional Neural Network for Multi-Scale Feature Learning

Contents

6.1 Introduction

The use of contextual and local information is common in numerous domains, from understanding scene context in images to modeling sentence structure within speech [421,422]. The idea of a scale-space is introduced by Koenderink [423] and discussed by Lindeberg, [424,425], in which a multi-resolution decomposition of an input signal is an ordered set of signals at increasingly coarser representations, reducing the finer scale features of an input domain and providing an increasingly generalized representation of the data [426]. Given that an observed dataset may describe a sampling of a problem domain in which the spatial scale of the target may be unknown, it can be beneficial to represent the observation across multiple scales. Lindeberg [424] discusses that the use of feature descriptors are often dependent on the relationship between the size of points of interest within the data and the size of the operators which are to be applied to them. The lack of *a priori* knowledge about the target scale of the application domain can result in the hindrance of descriptors producing an interpretable response for observations with largely varying scales. Developing feature extractors that are able to provide information from various levels of scale has been an important area of research in vision communities. A similar concept of contextual focus and scaling is seen in the behavior of visual acuity and peripheral drop off in biological vision systems [427,428], as shown by the topology of the photoreceptors within the human eye in Figure 6.1. The density of sensory structures within the eye provides a region of focal acuity, whilst the reduction in density towards the outer field of view leads to a reduced resolution. Rosenholtz [429] explains that the loss of acuity in peripheral vision should not result in the perception of a blurred scene, as appears within Figure 6.2, despite the drop in resolution as angular distance increases from the fovea and the center of focus. Instead the brain is able to utilize the low frequency information as context to the central area. The understanding of scale and the utilization of contextual information is an important task in computer vision and its use in pattern recognition, and as such methodologies have been explored which look to handle changes in scale and the relation between an object and its wider context.

Scale invariant methods, such as Scale Invariant Feature Transform (SIFT) [414] and Speeded-Up robust Features (SURF) [430], aim to focus extractors on key locations within the observed image; upon which scale, translation, and rotation invariant features are extracted. Cascade based methods, such as those presented in the Viola-Jones cascade detector, [431], aim to speed up detection by detecting on contextual features before moving on to local information, utilizing the fast computation of descriptors to discard regions which do not

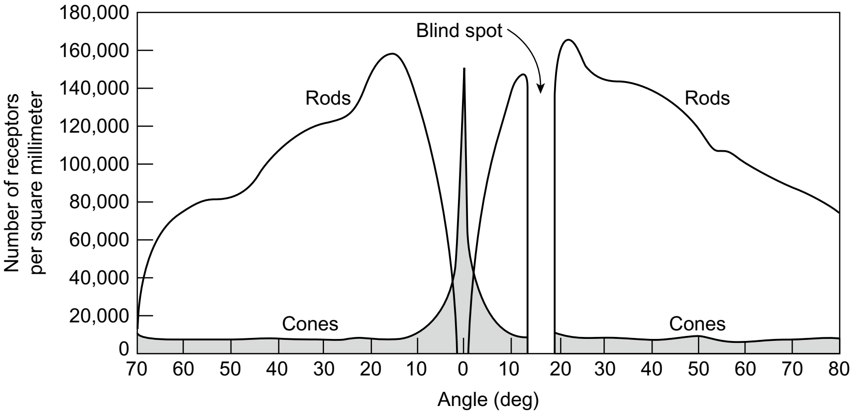


Figure 6.1: Spatial resolution of visual sensory receptors within the human eye. The resolution of peripheral vision gradually reduces as the angle of observation deviates from the foveal region (0 degrees). Image from [427].

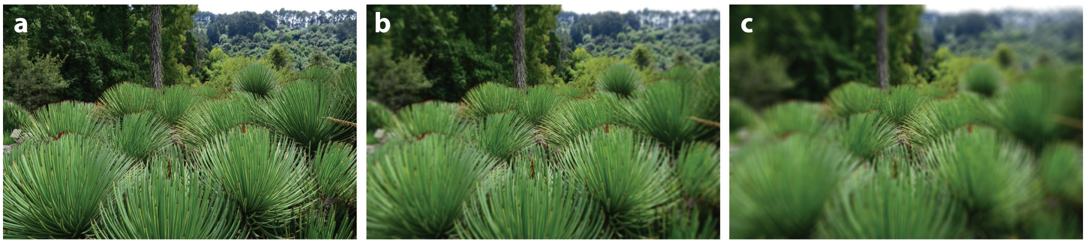


Figure 6.2: Effect of drop in acuity within the peripheral visual field. a) Original image. b) and c) increasing reduction in spatial acuity, leading to a loss of high resolutions within the peripheral sampling. "Blurring" is exaggerated in order to adequately display the effect. Image from [429].

match the learned context. Combination methods, such as multi-scale Convolutional Neural Networks (CNNs) attempt to learn both high and low level features, combining the wider context with localized information [35, 40, 432, 433]. Providing an understanding of wider contextual information and the relationship with high resolution localized information can be beneficial in detection problems, learning the placement of objects within a scene or structures within the body.

Many methods, such as SIFT and scale cascades have been developed to utilize hand-crafted descriptor sets in previous years; including the Haar, tilted-Haar, and steerable Haar features seen in many image based object detectors [431,434,435]. In more recent years however, the use of deep learning algorithms have become a popular alternative, capable of learning feature descriptors by combining inputs and adjusting their related importance weighting [309]. Standard neural networks have been shown to perform well in domains which exhibit no assumption of spatial relation between input features, and recently the usage of CNNs in domains residing on a regular Cartesian grid, such as 2D images and 3D volumes, has shown that spatially localized features can present beneficial descriptors for problems such as object recognition and detection [11, 95, 145]. Given that the appearance of local structure can significantly vary, contextual structures are often just as important for detection as local details. Using CNNs on large enough 3D patches to capture both local and contextual features is computationally impractical, often requiring complicated networks to capture information at various scales [34]. Kamnitsas *et al.* formulated a CNN architecture with multiple branches, one for each resolution, learning spatial features at different resolutions for brain lesion segmentation [35]. Each branch contains its own collection of filters and the learning of high- and low-resolution features are disjoint between the multiple branches. A similar branching scheme is seen in [36], with multiple resolutions being kept separate along different tracts of the architecture before being combined as input to a fully connected architecture. He [436] proposes a spatial pyramid pooling layer, maintaining local spatial information and removing the need to fix input sizes to a CNN when computing a fixed-length output vector. Ren combines a object proposal scheme with a CNN classifier to learn spatial information through a region-of-interest pyramid of reference boxes, with a 'Region Proposal Network' identifying key areas for the Fast R-CNN classifier to focus its attention [33]. Figure 6.3 gives an overview of current approaches to multi-scale deep learning. Although these presented methods aim to address the scaling issues within the recognition problem, the approach to detection through multi-scale descriptors is the same, in which features at differing scales are learned in order to refine performance.

In order to explore the use of spatial representation learning on the irregular domain of incorporated contextual and local features, we present a generalized methodology for constructing a multi-resolution graph using an irregularly spaced patch sampling method. By using a novel multi-resolution pooling method to create a relatively small patch which contains both local and contextual structural information, we are able to learn features from raw intensities; avoiding the inefficiencies of large patches and the need to train numerous CNN models for each scale. Due to the irregular spatial domain that is provided by the multi-resolution

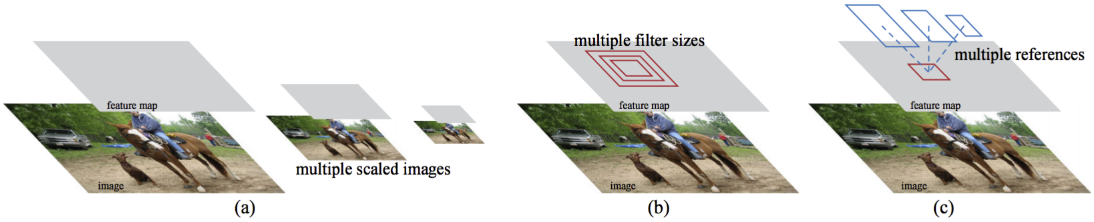


Figure 6.3: Current multi-scale schemes in feature representation learning in deep learning approaches. a) Pyramids of images and filters (branching), b) Pyramids of filters, c) Pyramids of reference boxes. Adapted from [33].

pooling, it is non-trivial to apply standard CNN operators to the patches [2, 415, 417]. We therefore propose a graph formulation of the multi-resolution spatial domain to apply Graph-based Convolutional Neural Network (Graph-CNN) operations for feature learning. A preliminary work explores the benefit of learning spatially related features within an irregular domain with Graph-CNN architectures in [2]. This Graph-CNN network acts as a detection classifier for Marginal Space Learning (MSL), which not only eliminates the burden of defining hand-crafted local and contextual features during training, but also significantly reduces the number of potential search hypotheses at the testing stage.

The chapter is structured as follows. In Section 6.2 we introduce an application problem of medical segmentation, on which we evaluate the use of multi-resolution sampling and deep learning on the irregular spatial domain of the non-uniformly sampled grid. In Section 6.3 we outline the proposed pipeline for automated segmentation using deep learning on the irregular domain. In Section 6.4 we present multi-resolution deep feature learning to drive MSL for position-orientation pose parameter estimation, and deformable model segmentation is utilized to obtain accurate regularized meshes. In Section 6.5 we evaluate the proposed pipeline on the case study of aortic root detection and segmentation on Computerised Tomography (CT) scans of the human torso, providing qualitative and quantitative analysis in Section 6.6. We draw conclusions on results found and the methodology proposed in Section 6.7.

6.2 Application Domain

The use of multi-scale learning is beneficial in many domains, and the proposed usage of multi-resolution sampling and deep learning on the irregular domain is generalizable to the overarching problem of multi-scale representation learning. In this study we provide evaluation of

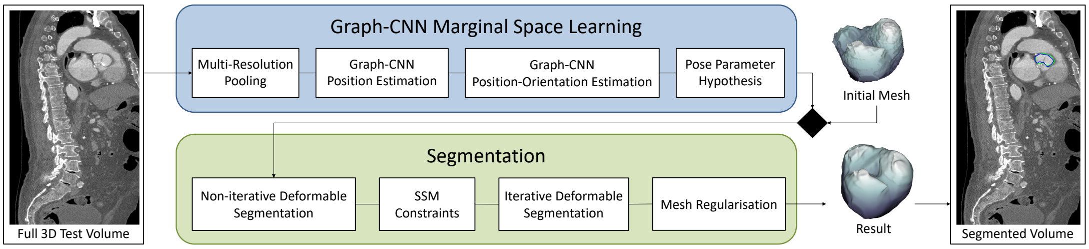

Full 3D Test Volume
Graph-CNN Marginal Space Learning
Multi-Resolution Pooling
Graph-CNN Position Estimation
Graph-CNN Position-Orientation Estimation
Pose Parameter Hypothesis
Initial Mesh
Segmentation
Non-iterative Deformable Segmentation
SSM Constraints
Iterative Deformable Segmentation
Mesh Regularisation
Result
Segmented Volume

Figure 6.4: Our testing pipeline for the proposed segmentation.

the proposed multi-resolution Graph-CNN on the domain application of medical segmentation, in which the understanding of detailed localized information and the general wider context of the human anatomy is beneficial in detecting small-scale anatomical structures with accuracy. Recently, there has been tremendous work in the application of neural network methods to medical image analysis [437], and in particular CNNs for anatomical organ detection [35,438] and unsupervised learning [439]. Segmentation is a key area in image analysis, and especially in the understanding of 3D volumes. Many applications make use of segmentation methods to process a volume into meaningful parts, especially medical volume analysis. Such methods often attempt to label each voxel of a volume into a given class of interest, utilizing appearance information or some structural features extracted from the volume. Many such applications can benefit from a combination of information about the local area to a voxel and wider contextual knowledge of the surrounding volume. One such application of medical volume understanding is the segmentation of the aortic valve. Aortic valve stenosis is a common heart disease affecting 3% of the global population, with many cases requiring surgical treatment. 3D segmentation of the aortic root is beneficial for patient selection, procedural planning and post-evaluation. It is therefore vital to reliably and accurately identify aortic root structure within a patient.

Due to image noise and other ambiguities, non-model based approaches are often unable to detect subtle boundaries between classes in a volume, e.g. those between the valve and left ventricular output tract [440]. However, given an initial shape, deformable models are able to identify this boundary, and have successfully been used for segmentation of the root structure [440–442]. Structure generalization and application of priors are often key in methods that perform detection and segmentation of medical imaging data.

Supervised automatic 3D deformable modeling is not only computationally demanding during the testing stage, but it is also labor intensive in preparing training data, e.g. in establishing correspondence for smooth 3D structures. Parameters for effective model regularization as well as useful feature extraction are chosen carefully depending on the application, which can be extremely time consuming. For example, model regularization regularly requires building a statistical model which often demands additional manual labeling [443]. Similarly, choosing optimized discriminative features for both object and boundary detection can be an excessively lengthy process. In this work, we aim to alleviate the burden of feature crafting, as well as implementing an efficient segmentation method using a bottom-up approach with prior regularization.

Appropriate automatic solutions to building statistical models are not well reported in the literature. Notably however, Frangi *et al.* [444] proposed finding mesh correspondences based on image data rather than the meshes themselves. The meshes were locally transformed to an atlas and anatomical points were propagated across the set. The transformation was estimated with intensity-based mutual information which is not suitable for noisy images with relatively low contrast between soft tissues, such as cardiac CT data. As such, we propose estimating the transformation using a mesh-based similarity metric and learned correspondences between training samples. The proposed method eliminates the process of manual landmark labeling, enabling a larger set of fiducial points per shape and providing a reduction in overall time taken to construct a shape model.

To initialize deformable models, they need to first be automatically aligned with the test image by performing object detection. Exhaustive pose parameter search in 3D is highly impractical due to possible position, orientation and scale permutations. Alternatively, MSL has been proposed for efficient 3D organ detection [440–442] by incrementally searching position, position-orientation and position-orientation-scale spaces. Choosing appropriate features for classifiers is challenging, as feature type, orientation and scale must be considered, and pathological structures often look significantly different between observations. We argue that a feature learning based approach should be adopted, such as those obtained through deep learning architectures.

The development of an end-to-end segmentation pipelines have only seen very recent study [445,446], and even fewer have been applied to volumetric data due to the complexity of dense segmentation of large volumes. Numerous dense segmentation methods exists, utilizing the fully convolutional neural network approach to provide a segmentation of input images [447]. [448] go one stage further and present a fully convolutional neural network for volumetric segmentation of medical images, providing a dense segmentation model which takes 48 hours to train on the $128 \times 128 \times 64$ volumes. This has since been expanded by [449] to utilize the Region Proposal Network structure from the Faster R-CNN approach to localize and focus the convolutional attention [33]. Both methods are dependent on the use of a fixed sampling resolution, limiting the ability to consider local and contextual features without increasing complexity with a branched multi-scale network architecture.

Overall, we present a novel pipeline method of deep learning on the graph representation of an irregular multi-resolution spatial domain for identifying target position and orientation hypotheses in aortic root detection. Raw local and contextual intensity features are used in a novel Graph-CNN architecture to mine spatially related features on an irregular spatial topology, avoiding relying upon hand-crafted features or an increased overhead from large patches. A marginal space learning approach is taken to reduce the search space complexity of the large 3D parameter space for segmentation initialization. An initial shape model is learned in an automated fashion by detecting a set of landmark features across the training meshes; reducing the manual effort of labeling fiducial landmarks on each mesh and allowing for a larger set of landmarks to be identified. A deformable segmentation framework is proposed that does not rely heavily on top-down constraints, instead presenting a non-iterative deformation and shape model regularization step for the initial segmentation of the volume. This is then followed by an iterative refinement of the mesh with local deformations and mesh-based regularization based on a strong boundary detection network. The use of Statistical-Shape-Modelling (SSM) shape constraints and mesh regularization utilizes prior information regarding learned shape context in order to produce a data-driven segmentation with reduced mesh entanglement and user guidance. Results on the proposed method show strong performance benefits in both aortic root pose estimation for the purpose of marginal space learning, and an accurate segmentation of the aortic root structure. Evaluation of the proposed approach is given in the medical segmentation domain.

6.3 Proposed Approach

We propose an efficient 3D segmentation method, that is fully automatic and is able to compute spatially related features on an underlying graph representation of the input domain. A novel technique to compute correspondences between 3D training meshes is proposed to reduce manual labeling in global model generalization. We combine MSL with multi-layer deep learning networks to avoid exhaustive search in a high dimensional parameter space, and avoid complex hand-crafting features by introducing novel multi-resolution pooling of raw intensities. We propose a graph-based convolutional neural network model to handle the irregularity in multi-resolution spatial domains while preserving the spatial relationships between the raw input features, training both position and position-orientation Graph-CNN estimators for MSL. Boundary features are also learned using neural network feature learning so that the method can be easily adapted to different modalities and problems. The shape regularization consists of both bottom-up local, and top-down global constraints. However, we take the view that overly relying on strong top-down constraints can be too restrictive, and as such we apply top-down non-iterative regularization only at the initial segmentation stage. Local deformation iteratively refines the segmentation, with reliable boundary detection using Neural Network (NN).

The proposed testing stage is shown within Figure 6.4. Localization and alignment parameters for the initial mesh, a median mesh from the training set, is efficiently carried out using a novel Graph-CNN-Based Marginal Space Learning (Graph-CNN-MSL) approach. Deformable segmentation is composed of boundary detection and 3D mesh regularization. This allows large-scale movements by setting long search paths at the boundary detector stage, and as a result shape constraints are applied to avoid mesh and shape irregularities. Local refinement is then performed using deformable segmentation in an iterative fashion, where each iteration is capable of small movements, followed by generic mesh processing.

6.4 Method

The proposed method consists of two major parts; the use of Graph-CNN-MSL to reduce the complexity of parameter search space, and SSM supported segmentation to generate an accurate and regularized mesh of the aortic root, Figure 6.4. The following section outlines the proposed pipeline components.

6.4.1 Graph CNNs for Marginal Space Learning

The estimation of pose parameters is often necessary for 3D object detection, for example there may be 3 optimal parameters each for position (x,y,z), orientation (ω,ϕ,θ), and local scale (S_x,S_y,S_z). Detection can often be formulated as a classification problem; however to exhaustively represent or search all pose combinations in a single high-dimensional space, Ψ, is computationally impractical. Most anatomical structures have some natural alignment (i.e. the aortic root is always near the left ventricle) and therefore it is observed that the probability distribution is clustered in a small localized region of Ψ. The idea of MSL is that the full simi-

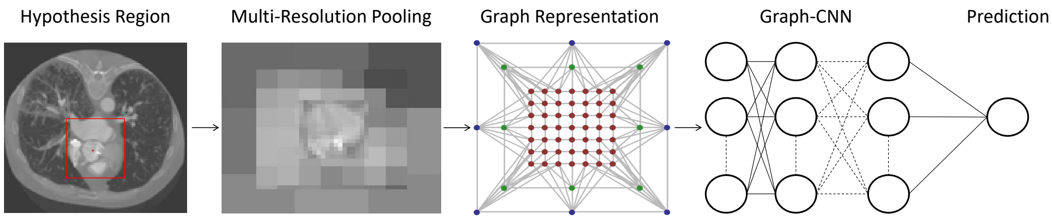


Figure 6.5: Illustration of proposed Marginal Space Learning classifier. Multi-resolution pooling reduces a large patch into a contextually and locally informative graph signal for use in a Graph-CNN architecture. A Graph-CNN architecture is then constructed using graph convolution and pooling layers to obtain a prediction on position or position-orientation of the observed patch.

larity search space can be marginalized in an attempt to reduce complexity for each increasing level of pose estimation:

$$\Psi_a \subset \Psi_{ab} \subset \Psi_{abc} = \Psi, \tag{6.1}$$

where Ψ_a is the position search space, Ψ_{ab} is the position-orientation space, and Ψ_{abc} is the position-orientation-scale space. It is assumed that the optimal hypothesis Π is contained within the highest probability hypotheses of all marginal spaces, such that

$$\Pi = \Pi_{abc} \subset \Pi_{ab} \subset \Pi_a. \tag{6.2}$$

Given three marginal spaces in (6.1), three classifiers C_a, C_{ab} and C_{abc} can be trained. At the testing stage, C_a can eliminate the vast majority of false hypotheses in Ψ_a, leaving high probability hypotheses Π_a. These are then passed through C_{ab} to leave Π_{ab}, which are subsequently passed through C_{abc} to leave $\Pi_{abc} = \Pi$. MSL therefore dramatically alleviates the high computation needed for exhaustive search and has been shown to reduce the number of test hypotheses significantly for applications in 3D volumes [440, 442].

We train two classifiers in our implementation which act as detectors for a positive position and orientation with the search space. MSL is formulated as a detection problem in which detections from one search space are used to constrain the subsequent search space [450]. A position estimator, C_a, tests all position hypotheses in the volume. The N highest scoring (x, y, z) hypotheses Π_a are then passed through a position-orientation estimator, C_{ab}. The highest scoring $(x, y, z, \omega, \phi, \theta)$ hypothesis is then taken as the position-orientation prediction result. Finally, for simplicity, we use the mean local scale of the training meshes to yield a 9-element pose estimation vector $(x, y, z, \omega, \phi, \theta, S_x, S_y, S_z)$. The use of mean scale incorporates scale information, simplifying the process over creating an appropriate scale search space [450]. Formulation of the problem is made as a binary classification task over a regression approach in order to obtain probabilities for selection of a set of hypothesis regions, decreasing the search space without confining the search too much. Deep learning approaches to pose estimation from medical volumes have seen a recent advance [451], however MSL is approached as a binary hypothesis detection problem. It is possible to utilize a logistic regression for the binary classification, however the non-linear activations of a network architecture are able to model more complex boundary decisions and as such can often more accurately reflect more complex problem domains when avoiding local minima traps [452].

When training both position and orientation estimator models, a positive hypothesis must satisfy the condition that

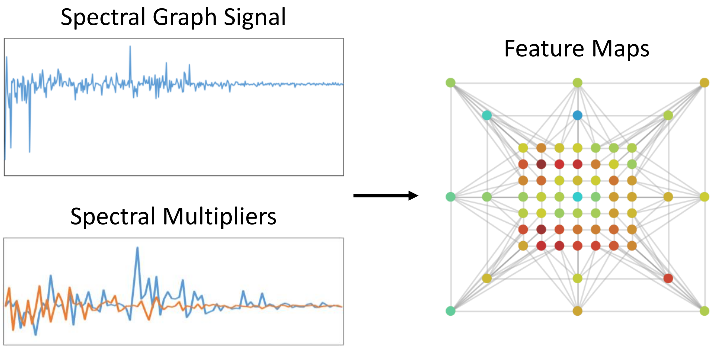
$$|P_k - P_k^t|/S_k \ \leq 1 \quad \forall k, \tag{6.3}$$

where P_k is a single pose hypothesis, P_k^t is its ground truth, and S_k is the corresponding parameter search step. For the position estimator, $P = (x,y,z)$, $P^t = (x_t,y_t,z_t)$, $S = (1,1,1)$ voxels, and the input layer features are the intensities from our globally aligned pooling. For the position-orientation estimator, $P = (x,y,z,\omega,\phi,\theta)$, $P^t = (x_t,y_t,z_t,\omega_t,\phi_t,\theta_t)$, $S = (1,1,1,10,10,10)$, and the patches are aligned with the orientation hypothesis. An example volume, with $512 \times 512 \times 512$ voxels and full 360 orientation space about the X, Y, and Z axes, would result in over 6.26×10^{15} pose parameter hypotheses. MSL allows us to first search 1.34×10^8 position parameters, select the top 10 probable position hypotheses, and then search roughly 4.66×10^8 position-orientation parameters. This is an overall reduction on the order of 10^7 over exhaustive search of the pose space.

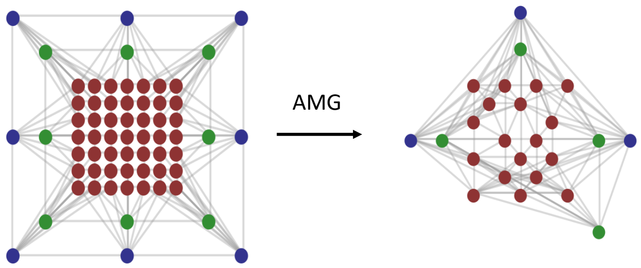
Given that pathological anatomical structures can significantly vary in appearance, defining hand-crafted features suitable for detection can be difficult. We choose deep learning algorithms to train our MSL classifiers due to their strong performance in feature-mining capabilities. The process of extracting large patches able to represent contextual information at full resolution can lead to large feature vectors, making the process computationally expensive for many machine learning methods. Sampling at lower resolutions eases complexity, at the cost of losing local descriptors highlighting structures of interest in the first instance. Recent multi-scale feature extractors make use of multi-scale neural network variants, [453–456]. Such multi-scale approaches can vary greatly in their use of contextual information within a scene; from utilizing cascading filtering for the incremental localization of attention [453], to the use of a branched network with a different scale input on each branch [456, 457].

To avoid trade-off between fine local information and wider contextual features, we introduce multi-resolution pooling on an irregular grid of varying resolutions, Figure 6.5. Given a large patch that represents a wider area of the full resolution space, our multi-resolution pooling decreases in resolution from the center out. This produces a representation that provides a dense local information patch at the center, and sparse contextual information towards the perimeter. Such a representation reduces the number of samples within a patch covering a large area of observation, but yields an irregular spatial grid, making standard convolutional kernel operations, and therefore CNNs, infeasible. To cover such an area with a regular patch sampling would create a much denser input volume, and would require the receptive field of the network to be increased, either through wider kernels or higher capacity networks. As discussed in Section 3.5, the embedding of irregular domains into a regular space has some issues, including the mapping behavior, and the impact padding or embedding has on the learned features. Padding the input space with zeros to make it conform to an array structuring is changing the underlying properties of the observed data, and may have an impact on the features learned by a given approach. Due to the irregular topology of the multi-resolution space, we can use fully-connected NN models to learn features for our marginal space classifiers. However, such architectures can often fail to learn spatially localized features from the input space, as would be learned by CNN classifiers on the regular Cartesian domain. Such CNN operators, as discussed in Chapter 5, are ill-defined for use on the irregular spatial domain. To make use of spatial relationships between the input features in an irregular domain we can formulate the multi-resolution topology as a graph G, with the input intensities forming a graph signal x that resides on G. By utilizing Graph-CNN operators, Figure 6.6, spatially localized features are able to be learned on the irregular spatial topology of the graph via spectral filtering techniques developed in the field of signal processing on graphs [2, 360, 415]. This allows end-to-end learning of features on the irregular space that allow our model to simultaneously observe local features and low-resolution wider contextual information without the overhead of learning a different CNN model for each scale.

Generation of the underlying graph structure is non-trivial, and research is on-going into a plethora of graph construction techniques for a variety of domain applications. In our multi-resolution pooling of the 3D Cartesian grid patch, we can formulate a graph that exhibits the dense central region of high resolution with a sparse region of lower sampling towards the extremities of the patch by using the multi-resolution coordinate points for each resolution. For each selected resolution level, l, we generate a set of Cartesian coordinates, P^l, sampled at the given resolution rate. We then remove points from P^l that are spanned by $P^{1:l-1}$, discarding



(a)



(b)

Figure 6.6: Graph Convolutional Neural Network operators. a) Graph Convolution. Spectral graph signals are multiplied with spectral multiplier filter weights. An inverse Graph Fourier Transform returns the signal to the spatial domain. b) AMG Pooling, fine-scale nodes are aggregated into coarser nodes in the pooled graph.

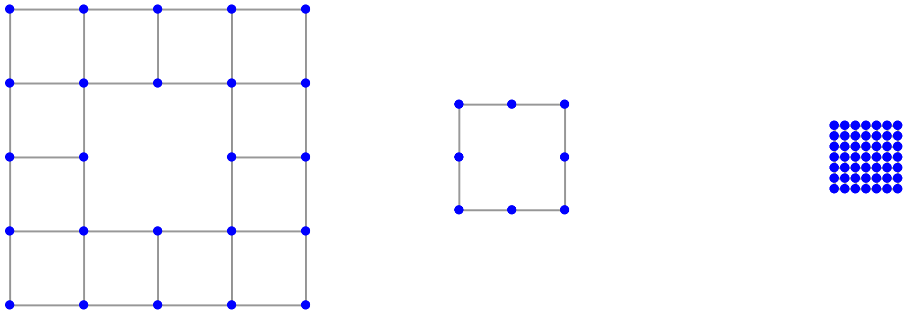


Figure 6.7: Exploded view of an example 2D multi-resolution graph. Note the empty center of each successive outer layer, and the irregular sampling distances between layers. These combining factors make such a multi-resolution sampling domain unsuitable for current convolutional neural network approaches.

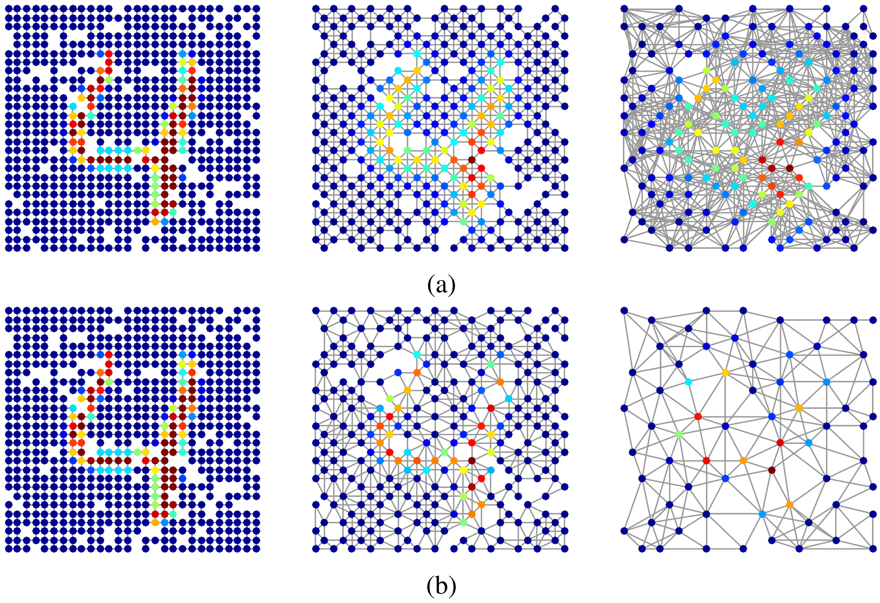
observed regions of overlap. This graph construction provides a graph G of V vertices, with a vertex for each coordinate point in the multi-resolution space, representing the concept of peripheral vision and the utilization of surrounding contextual information [428,429,458]. From the graph generation procedure we obtain the edge weighting and diagonalized adjacency matrices, $W \in \mathbb{R}^{N\times N}$ and $A \in \mathbb{R}^{N\times N}$ respectively. This allows us to construct the non-normalized Laplacian matrix representation of the graph structure, $L \in \mathbb{R}^{N\times N}$, by $L = D - W$ as previously discussed. Intra-layer edge weighting is calculated as

$$w_{(i,j)} = e^{-\frac{||v_i - v_j||^2}{\sigma}} \tag{6.4}$$

on an epsilon nearest neighborhood of vertex v_i, with a search radius of $\varepsilon = l$, the current sampling resolution, where $||v_i - v_j||$ is the L^2 distance between the vertices v_i and v_j and $\sigma = \frac{\varepsilon^2}{2}$. This returns the 4-way Von Neumann neighborhood relationships of the vertices within a resolution layer. Inter-layer edges connect the lower-resolution layer vertices to the higher-resolution core via the l nearest neighbor vertices, relating wider contextual features to the high-resolution region of interest within the patch. Weighting for inter-layer edges are defined by scaling Eq. 6.4 down by the current resolution factor, with v_i representing a vertex in the current layer, and v_j a vertex in the high-resolution core.

Given a complete Laplacian decomposition we can formulate a Fourier basis for the graph spectral domain and utilize the Graph-CNN operators presented in Chapter 5. Optimizing the weightings of spectral multipliers via back-propagation allows the training of a self-learning feature mining architecture, rather than arduously defining hand-crafted features for a complex multi-resolution space. To ensure that localized features are learned in the spatial domain, we can utilize the property of smoothness in the frequency domain providing spatial locality on the spatial domain [2,415]; thus the network tracks $n < N$ weights for each filter, interpolating them up to N with a smoothing kernel for use in graph convolution.

For this application the selected graph pooling operation is an AMG graph coarsening, selecting vertices in the finer graph resolution for aggregation into coarser vertices within the pooled graph and avoiding an explosion of edges in the coarsened graph when compared to the use of Kron's reduction [371], Figure 6.8. Aggregation takes a spatially localized subsets of V from G, and generates a singular vertex for each subset in the new set of vertices $\hat{V}$ in the output graph $\hat{G}$. The graph signal, $f_{1:N}$, associated with G is then down-sampled to reside on $\hat{G}$ as the coarser graph signal $\hat{f}_{1:n}$, where N and n are the original number of vertices and the pooled number of vertices respectively. The AMG operation produces a set of interpolation



(a)

(b)

Figure 6.8: Two levels of graph pooling operation on an irregularly sampled 2D grid. a) Kron's reduction, b) AMG. Note that both methods retain overall spatial structural distribution, however the edge connectivity of Kron's reduction results in an explosion in edge count.

matrices, restriction matrix R and projection matrix P, for down-sampling and up-sampling transform of the graph signals between the finer and coarser graph levels.

The Graph-CNN models are built as follows. 1) A graph representation of the multi-resolution volume space is generated; 2) Intensities from large patches correspond to the network inputs; 3) Multi-resolution pooling yields a significantly reduced representation on the irregular grid graph; 4) Pooled patch values are fed into the Graph-CNN, undergoing graph spectral convolutions and graph pooling operators as defined, Figure 6.6 ; 5) An output detection prediction is given for each observed hypothesis. An example architecture pipeline for Graph-CNN-MSL is shown in Figure 6.5.

6.4.2 Deformable Segmentation

The proposed segmentation stage first uses the predicted hypothesis pose parameter vector $(x, y, z, \omega, \phi, \theta, S_x, S_y, S_z)$ to align the initial shape model, a median shape from the training set, to the volume. A boundary detector is then used to guide a non-iterative local deformation that is then constrained via a shape regularization step. Mesh refinement is then obtained via an iterative application of local deformations.

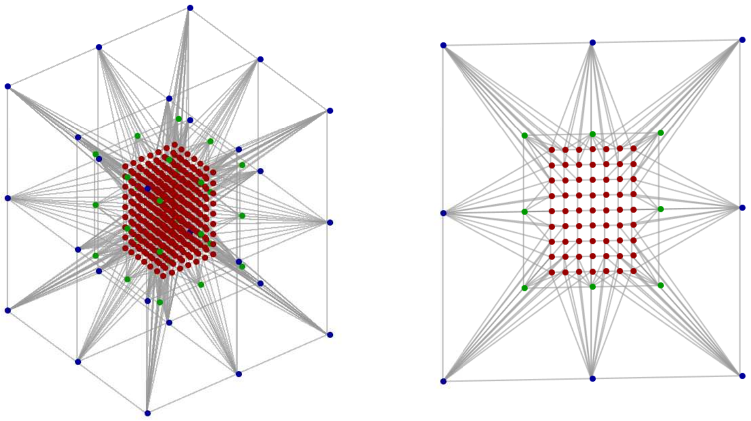


Figure 6.9: a) 3D multi-resolution volume graph for orientation estimation. b) 2D example, note removed nodes in regions of overlap. The high-resolution core is a cuboid structure, extending along the Z-axis to capture further information about the ascending aorta. Node coloring represents variable resolution sampling, from low resolution outer layer to the high resolution inner core. Best viewed in color.

Due to large deformations after the initial deformation stage, it is necessary to constrain the shape space so that the deformed shape is consistent with the training set, by using SSM for instance. However, rather than manually labeling corresponding points to generate the SSM, we propose automatically finding a subset of corresponding vertices across a set of shapes by locally transforming meshes to a reference and propagating points across the set.

Shape constraints are identified in an automated fashion, avoiding the scaling inefficiencies of labeling landmark points by hand. The automated landmark detection allows for a larger set of landmarks to be identified with little impact on the user. To construct the initial shape model, a target mesh $M_t = (V_t, E_t, F_t)$ with $|V_t| = n$ vertices is randomly selected from the training set, and a subset of m fiducial point vertices are labeled such that $P_t \subset V_t$, and $m << n$. All other meshes in the set are regarded as source meshes, such that $M_s = (V_s, E_s, F_s)$ where $|V_s| = p$, and $n \neq p$. The aim is to find a subset of m vertices $P_s \subset V_s$, that are correspondent with P_t. We work on the assumption that finding correspondences between two shapes becomes much easier if the shapes are similar to each other. Therefore we apply a transformation $T(x,y,z) : M_s \mapsto M_t$, consisting of global $T_g(x,y,z)$ and local $T_l(x,y,z)$ transformations.

T_g globally aligns both meshes and is formulated as an affine transformation from ground truth vectors. T_l then takes into account the local differences in shape, and is estimated using

(6.6), (6.7), and a similarity metric

$$E(V_t, V_s') = \sqrt{\sum_i^n (V_t - V_s'')^2} \tag{6.5}$$

where V_s'' are the corresponding nearest neighbor vertices in V_s' with respect to V_t. For every point in P_t, its nearest neighbor based on Euclidean distance is found in V_s', resulting in P_s'. Finally, applying T_l^{-1} to P_s' yields P_s.

After alignment of the initial mesh with pose parameters identified by MSL, the non-iterative deformation stage utilizes appearance features to adjust the vertex set by defining a search path along the normal direction. A boundary detector is trained to find the path coordinate with the optimal boundary response, and landmark vertices are deformed to these positions. In order to avoid hand-crafting features, we again utilize feature learning. For boundary detection we use a shallow fully connected NN, learning low-level features from a small set of intensities on a local patch, centered at the search path coordinate and aligned with the path direction. The small area of observation ensures iterative refinement of the mesh is based upon response to localized boundary features, with little interference from wider appearance. A 3×3 patch is extracted from each point along a boundary search path, vectorized, and input to a single-layer neural network.

Boundary detection now results in new vertex positions V', however there is potential for mesh entanglement amongst the new set of vertices. To counter this we use B-spline based 3D mesh regularization, [442], where a non-rigid transformation $T(x,y,z)$ is estimated between V and V' before performing a free-form-deformation (FFD) on V to fit V'. To estimate $T(x,y,z)$, control points $\phi^r_{i,j,k}$ separated by δ, are moved which warp an underlying 3D voxel lattice. Given a set of control points, the transformation is formulated as follows,

$$T(x,y,z) = \sum_{l=0}^{3} \sum_{m=0}^{3} \sum_{n=0}^{3} B_l(u) B_m(v) B_n(w) \phi_{i+l,j+m,k+n}, \tag{6.6}$$

where B_l represents the l^{th} basis function of the B-spline, $[i,j,k]$ are the voxel positions, and $[u,v,w]$ are the fractional positions. The positions of $\phi^r_{i,j,k}$ are optimized using gradient descent consisting of a smoothness cost and a sum-of-squared-difference similarity metric between V (after warping) and V'. The final transformation is estimated at multiple resolutions R, similar to FFD registration [459],

$$T^R(x,y,z) = \sum_{r=1}^{R} T^r(x,y,z). \tag{6.7}$$

For our purpose, $R = 3$, and at each mesh resolution the control point spacing is $\delta_r = \delta_0/2^r$, which controls the FFD degrees-of-freedom. After applying SSM constraints during segmentation, only corresponding fiducial points are regularized. Thin plate spline warping is used to interpolate remaining vertices, resulting in ~ 8000 final corresponding vertices.

The next stage of the pipeline is to take the regularized mesh and perform iterative refinement of the mesh boundary by applying repeated rounds of mesh deformation with the NN boundary detector. This avoids a heavy top-down constraint on the segmentation, instead having a single round of top-down shape constraint followed by an iterative data-driven refinement stage. Vertices are iteratively deformed by identifying boundaries along the normal direction as above. A final generic mesh-processing step rounds out the pipeline to regularize and smooth the mesh for output.

6.5 Implementation

For evaluation of fully automated Graph-CNN-MSL and segmentation, we provide an example application on aortic valve segmentation from 3D-CT scans. We perform 3-fold cross-validation via segmentation on 36 3D-CT volumes of size $512 \times 512 \times (500 \sim 800)$ and voxel size $0.48mm \times 0.48mm \times 0.62mm$. Benefits of utilizing Graph-CNN architectures to learn spatially related features for MSL are evaluated in comparison to use of standard NN and hand-crafted feature classifiers. Comparison of the proposed segmentation stage is given against a state-of-the-art method and traditional statistical shape model based segmentation.

6.5.1 Marginal Space Learning

Our multi-resolution position estimator consists of a patch comprised of 3 resolutions; an inner core of $1 \times 1 \times 1$, a middle region of $2 \times 2 \times 2$, and an outer region of $3 \times 3 \times 3$ times the mean local scale. Multi-resolution layers were pooled at resolutions of $\frac{1}{8}$, $\frac{1}{40}$, and $\frac{1}{56}$, resulting in a graph with 395 vertices. Position-orientation inputs were taken from a patch at 3 resolutions; an inner core of $1 \times 1 \times 1.2$, a middle region of $2 \times 2 \times 2$, and an outer region of $4 \times 4 \times 4$ times the mean local scale. Regions were pooled at resolutions of $[\frac{1}{8}, \frac{1}{40}, \frac{1}{56}]$ respectively, resulting in a graph with 591 vertices. The inner high-resolution core of this patch is cuboid in shape, extending along the Z-axis to provide further high detail information about the ascending aorta to assist with orientation estimation. Coordinates generated from this multi-resolution setup

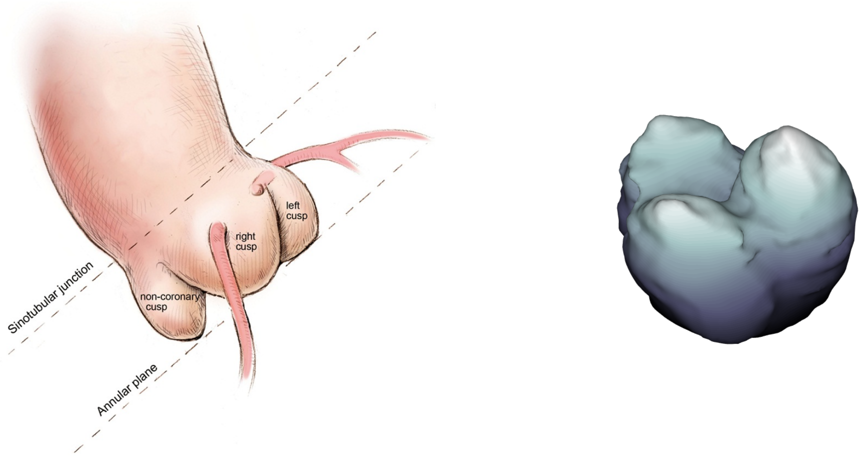


Figure 6.10: Structure of the aortic valve. Left: Diagrammatic representation of the aortic root and valve, detailing the three cusps. Image from [460]. Right: Ground truth mesh from the dataset, oriented vertically. Note that ground truth is labeled up to the sinotubular junction.

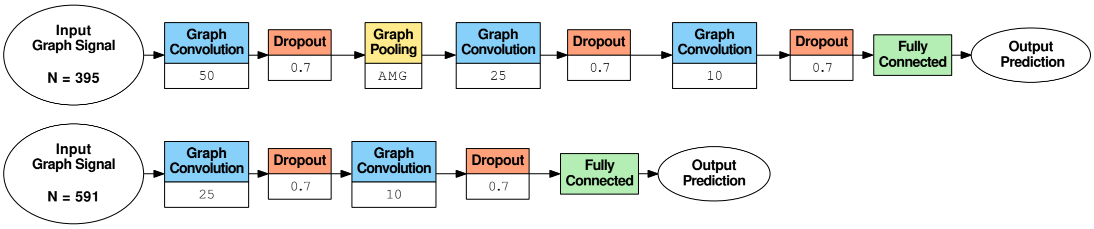


Figure 6.11: Graph-CNN MSL network architectures. Top: Graph-CNN position estimator. Bottom: Graph-CNN orientation estimator.

were then used to generate the graph structure for the Graph-CNN operators as defined in 6.4. Figure 6.9 shows the resulting graph structure for both classifiers utilized in Graph-CNN-MSL.

Two separate Graph-CNN architectures, Figure 6.11, are utilized to estimate position and position-orientation parameters for shape model alignment. The aortic root position estimation architecture was empirically defined as $C^{50}PC^{25}C^{10}$, where C^{o} is a graph convolutional layer with o output feature maps and P is an AMG graph pooling layer. Each graph convolutional layer is followed by Rectified Linear Unit (ReLU) activation, batch normalization, and dropout to further support generalization of features and reduce model overfitting. A soft-max and logistic loss provides output detection prediction labels and back-propagation target for training. Networks were trained using an ADAGRAD optimization strategy [329], with an initial learning rate of 10^{-3} and batch size of 32. Training samples were selected with an object/non-object ratio of $\frac{1}{50}$. The orientation estimator architecture was $C^{25}C^{10}$, and utilized an object/non-object ratio of $\frac{1}{25}$. For orientation estimation we found that it was best to avoid compression of features describing the Z-axis roll of the tubular structure observed during pooling operations. Graph-CNN implementation applied extracted multi-resolution intensities to respective nodes in the graph, with edge weighting described in 6.4.

In order to identify the benefit of utilizing a localized feature extraction constraint provided by the Graph-CNN architecture, a fully connected neural network was constructed where C^{o} layers are replaced with fully connected layers of the same size as in architectures above. These fully connected networks had no intrinsic representation of spatial relationships between features, essentially representing a fully connected and equally edge-weighted graph, as represented in Figure 3.6. Training hyper-parameters of the neural network implementations were kept the same as with the Graph-CNN, utilizing ADAGRAD optimization with a learning rate of 10^{-3} and batch size of 32.

Our boundary detector was trained with an equal boundary/non-boundary ratio using intensities from a 3×3 local patch. Patches were fed through shallow fully connected network in order to learn low-level boundary features. A comparison hand-crafted feature based approach utilized steerable features and a boosted tree ensemble classifier, as per [440].

6.5.2 Segmentation

To generate the required initial shape model for deformable segmentation, we label 70 fiducial points on a single target mesh, which were propagated across the remaining training set as set out in Section 6.4. We compared the proposed segmentation pipeline with two competing methods; a modified Active-Shape-Modelling (ASM) implementation, and another deformable modeling method [440]. Zheng et al [440] consisted of a boundary detector trained with steerable features, followed by Taubin mesh smoothing in an iterative fashion for mesh refinement. For fair comparison, we included a 3D mesh regularization stage in our implementation of ASM.

6.6 Comparative Analysis and Results

We evaluate our proposed methods utilizing the implementation outlined in 6.5, reporting averaged performance across 3-fold cross-validation. We report evaluation on both MSL and deformable segmentation portions of the pipeline, outlining contribution of Graph-CNN fea-

Table 6.1: Predictive accuracy of the Marginal Space Learning approaches. The addition of a locally receptive filtering operation within the Graph-CNN approach provides an improvement over the standard Neural Network method, lowering both the position and the orientation error of the predicted pose parameters.

MSL Method	Position (Voxels ± SEM)	Orientation (Degrees ± SEM)
Hand-crafted [440]	9.10 ± 0.57	14.69 ± 1.28
Fully Connected NN	3.79 ± 0.47	12.38 ± 1.24
GCNN	1.46 ± 0.36	6.78 ± 1.01

ture learning for aortic root position and orientation parameter estimation, and non-iterative shape deformation with regularization for segmentation.

6.6.1 Marginal Space Learning

A comparison of classifier methods utilized for MSL is presented in Table 6.1. The self-learning feature mining methods of NN and Graph-CNN outperform use of hand-crafted features for both position and position-orientation estimation, with Graph-CNN further improving over NN architecture. Being able to accurately and reliably provide hypothesis regions on which to initialize a segmentation algorithm is highly beneficial to following segmentation steps. The Graph-CNN position detector's sensitivity and specificity were 91.46% and 99.95%, respectively. Sensitivity and specificity of the position-orientation estimator was 89.84% and 84.16%. Given the huge parameter search space, strong specificity results are invaluable to reliably reduce parameter search spaces and greatly increase efficiency. Our implementation only considers the top 10 position predictions, with the Graph-CNN-MSL model implementation maintaining predictive accuracy and greatly reducing the position-orientation search space. The proposed method provides a significant increase in accuracy over both the hand-crafted and fully connected neural network implementations. Results showed a significant difference in position estimation accuracy between Graph-CNN and hand-crafted features, $t(35) = -11.76$, $p < .001$. Graph-CNN also provides benefit in orientation estimation over the hand-crafted feature approach, $t(35) = -9.37$, $p < .001$. The NN method also outperformed the hand-crafted feature approach, as shown in Table 6.1; the trained estimators provided a significant improvement over hand-crafted features for both position and orientation, $t(35) = -7.74$, $p < 0.001$ and $t(35) = -7.35$, $p < 0.001$ respectively. When comparing the spatially localized feature learning of the Graph-CNN architecture against the fully connected neural network approach, Table 6.1 shows that both the position and orientation estimator see marked improvements, with $t(35) = -4.31$, $p < .001$ and $t(35) = -4.89$, $p < .001$ respectively.

The Graph-CNN architectures provide a large gain in accuracy over the other methods, utilizing spatial relationships between the high and low resolution spaces in a single feature. One benefit of utilizing the underlying spatial topology of the problem domain is the ability to visualize the learned descriptors [337, 461], and the same can be utilized in Graph-CNN methodologies to identify the localized responses within the network's filters. Figure 6.12 shows some example feature maps produced by graph convolutions of spectral filters in the Graph-CNN model trained for the position classifier. The feature maps describe the activations of 3 filters (a, b, c) on multi-resolution patches centered at the ground truth of 3 different volumes to show the response to the aortic root structure. The visualization plots a slice through the center of the multi-resolution volume with the topology as in Figure 6.9. From these visualizations we are able to identify activation responses to the local and contextual information within the multi-resolution patch, with the outer layers responding to lower resolution features from the wider contextual region surrounding the patch center. We can see in the map that the pixel density at the center of the patch is higher, and reduces according to the defined layer resolutions towards the edges. Figure 6.12a shows activations which have highlighted consistent features across high and low resolutions. Note the strong responses in the top right and lower left area of the low resolution and the bottom and right edges of the higher resolution core. Figure 6.12b displays features which are consistently found in the high resolution region, in this example a diagonalized response across the center. Figure 6.12c shows features found in the mid-resolution layer of the graph, with some information found in the top left of the layer. All of these features can be observed to generalize across the three different testing volumes. The utilization of the proposed Graph-CNN architecture allows for spatial relationships between the multiple resolutions to be learned within a single filter, whereas the use of multi-branched CNNs would require numerous filters to learn individual features for each scale independently, increasing complexity of the network. The use of dilated CNNs filters, [37], to extend the field of view may allow contextual information to be captured by a smaller kernel, but the weights are shared for local and contextual convolution, learning the same feature at varying scales. In contrast to both of these the proposed Graph-CNN is able to learn relationships between the local and contextual features without filters dedicated to a given scale.

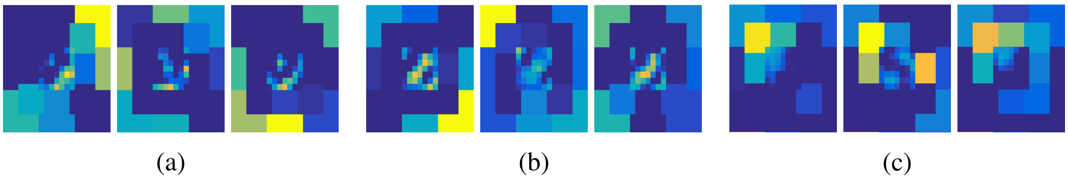


Figure 6.12: Example feature maps from positive patches from 3 separate test volumes. Feature-maps from filter responses to a) local and contextual features, b) local features c) non-local features.

Table 6.2: Comparison of segmentation approach accuracy.

Method	Mesh Error (mm ± STD)	Hausdorff Distance (mm ± STD)
Regularized ASM	2.01 ± 0.63	9.13 ± 2.58
Zheng et al. [440]	1.44 ± 0.59	10.29 ± 2.93
Proposed	1.27 ± 0.23	6.04 ± 1.50

6.6.2 Segmentation

Segmentation performance is evaluated in terms of the symmetrical point-to-mesh error and symmetrical Hausdorff distance. Mesh error provides an indication of average error in the predicted segmentation, however the Hausdorff distance gives insight into the presence of outlying regions on the predicted mesh. Overall the proposed pipeline showed notable improvements in segmentation accuracy compared to the comparison methods, with an average Mesh Error of 1.27 ± 0.23mm and a Symmetrical Hausdorff Distance of 6.04 ± 1.50mm (Table 6.2). The benefit of regularization for suppressing mesh entanglement can be seen by the lower symmetrical Hausdorff distances of the regularized ASM and proposed methods. The use of deformable segmentation refinement helps to drive the mesh error lower, iteratively bringing points closer to the appearance boundaries identified by the shallow network. For both error metrics the proposed method shows lower standard deviation, with the pipeline providing consistently accurate and reliable segmentation. The proposed method provides a significant improvement over the ASM approach for both mesh error and Hausdorff distance, with $t(35) = -7.17$, $p < .001$ and $t(35) = -3.45$, $p = .0015$ respectively. Compared to the method provided by Zheng, only the Hausdorff provided a significant difference in performance with $t(35) = -7.68$, $p < .001$. There was no significant difference between the proposed method and that of Zheng in regards to their mesh error, with $t(35) = 1.25$ and $p = .22$

Table 6.3: Comparison of MSL initialization methods on final segmentation performance.

Segmentation method	MSL method	Mesh Error (mm ± STD)	Hausdorff Distance (mm ± STD)
Regularized ASM	Hand-Crafted	2.01 ± 0.63	9.13 ± 2.58
	NN	2.00 ± 0.78	8.42 ± 3.28
	Graph-CNN	1.66 ± 0.45	6.92 ± 2.05
Zheng et al. [440]	Hand-Crafted	1.44 ± 0.59	10.29 ± 2.93
	NN	1.51 ± 0.66	10.59 ± 3.41
	Graph-CNN	1.23 ± 0.27	9.10 ± 2.26
Proposed	Hand-Crafted	1.50 ± 0.56	7.72 ± 3.20
	NN	1.49 ± 0.52	7.85 ± 3.24
	Graph-CNN	1.27 ± 0.23	6.04 ± 1.50

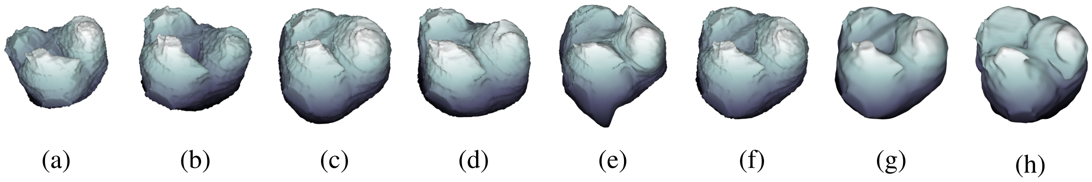


Figure 6.13: Output at each stage of segmentation. a) Initial shape model, b) pose alignment, c) non-iterative deformation, d) SSM constraint, e) iterative deformation, f) final mesh regularization, g) post-segmentation smooth, h) ground truth.

Table 6.3 highlights findings comparing hand-crafted features against the two deep learning methods. First, proposed use of Graph-CNN for mesh pose initialization provides a consistent benefit to the segmentation portion of our pipeline. Second, proposed segmentation steps are able to produce meshes with low Hausdorff Distance to the ground truth, a benefit of regularization for controlling mesh entanglement. It can also be seen that Graph-CNN methods provide low standard deviation across numerous test volumes, indicating that accurate pose parameter hypotheses from Graph-CNN-MSL are beneficial to the following segmentation steps.

Output from each stage of the segmentation pipeline can be seen in Figure 6.13, detailing the alignment of an initial mesh to pose parameters identified via Graph-CNN-MSL, non-iterative deformation, application of the SSM constraint, and the iterative deformation stage. Given the difference in pose between ground truth and median initial shape, it is important to identify accurate pose parameters for shape alignment.

Example cropped slices of our segmentation results are shown in Figure 6.14, including

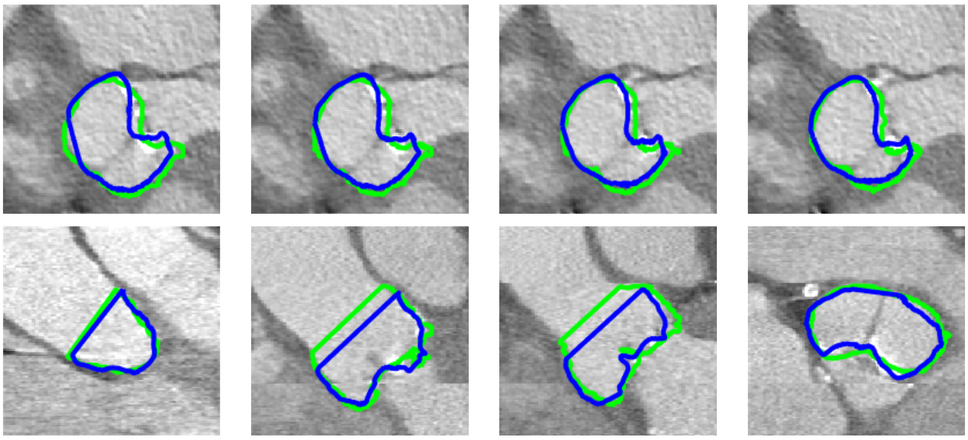


Figure 6.14: Segmentation shown on cropped image slices for illustration. Green contours show ground truth, blue contours show result of proposed method.

different axial views, and Figure 6.15 compares segmented slices from each evaluated method. Entanglement is observed in slices implementing a top-down approach with no regularization [440], Figure 6.15c, whilst the modified ASM fails to expand and meet the boundary contour due to the heavy shape constraint. The proposed method shows it is able to maintain a smooth regularized mesh whilst also deforming towards the boundaries in all spatial directions. Figure 6.14 shows the final segmentation provided by the pipeline, where the segmentation matches the overall shape of the underlying true segmentation, however some under-segmentation can be observed where the mesh does not fully expand to align with the structure boundary. The boundary detection phase of the proposed pipeline is utilized to identify boundary regions in the underlying data by searching along a normal vector and finding the maximal response to a trained boundary detector. In this study the use of a shallow neural network with a small patch sampling area may produce suboptimal identification of the structure boundaries. Increasing the search path length may also allow the shape deformation step to handle being initialized further from the true boundaries in the volume.

Although the point-to-mesh error of [440] is marginally lower (~ 0.04mm) than the proposed method when initialized with Graph-CNN-MSL, these meshes lack regularization, resulting in the higher symmetrical Hausdorff distance error. It is also worth noting that our method is automated at the training stage, whereas [440] requires extensive manual preparation time to produce suitable hand-crafted feature vector representations and identify landmark points across training meshes. By labeling landmark points in an automated fashion we are able to greatly reduce the pre-processing time required to start model training. Local transformation

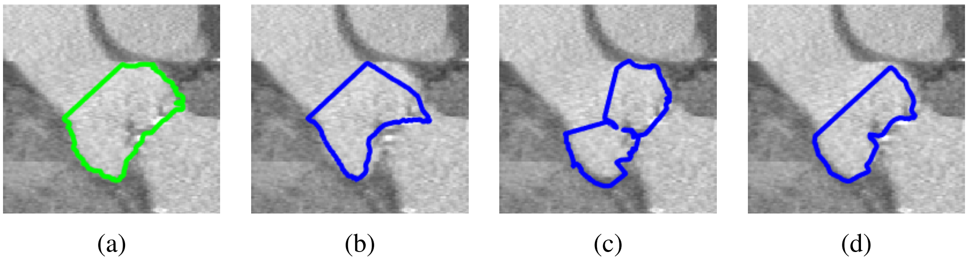


Figure 6.15: Segmentation comparison for three pipeline methods. a) Ground Truth, b) Modified ASM, c) Zheng [440], d) Proposed.

to identify a corresponding subset of vertices across training meshes allows scaling of identified fiducial landmark points in our shape model without drastic increase in pre-processing effort, as seen by the proposal to identify 70 landmark points compared to the 8 within [440].

Mesh comparisons in Figure 6.16 show that some shape constraint is beneficial for generating ordered mesh surfaces. The meshes produced by [440] are significantly entangled compared to both the proposed method and the modified ASM, however the modified ASM produced high point-to-mesh errors due to the lack of shape deformation towards the structural boundaries. This shows that strong shape generalization can be too restrictive, and it is critical not to overly rely on top-down constraints. We applied the Taubin smooth as a final mesh smoothing operation to both our proposed method and the comparison method from Zheng. We have also explored the effect of increasing the smoothing effect on the predicted meshes. As can be seen in Figure 6.17, the repeated smoothing does not correct the mesh entanglement but can initially reduce surface variance. Figure 6.18 shows the effect of over-smoothing, with the Hausdorff distance between the entangled predicted mesh and the ground truth reducing slightly before diverging once the mesh is over compressed. The observation here show that reliance on smoothing as a method for repairing the mesh surface is not an optimal approach, and instead an integrated mesh regularization approach which avoids mesh entanglement provides an initial prediction of a well-structured mesh surface which then be improved slightly with smoothing.

The results also show that although our Graph-CNN-MSL and boundary detector features were not hand-crafted , the use of deep learning methods as feature mining components lead to good initialization of position and orientation, greatly reducing search space complexity and providing good segmentation performance. Our learned features provide sufficient discriminative power while significantly speeding up the training process, indicating that deep learning algorithms are suitable for both object and boundary detection in deformable modeling.

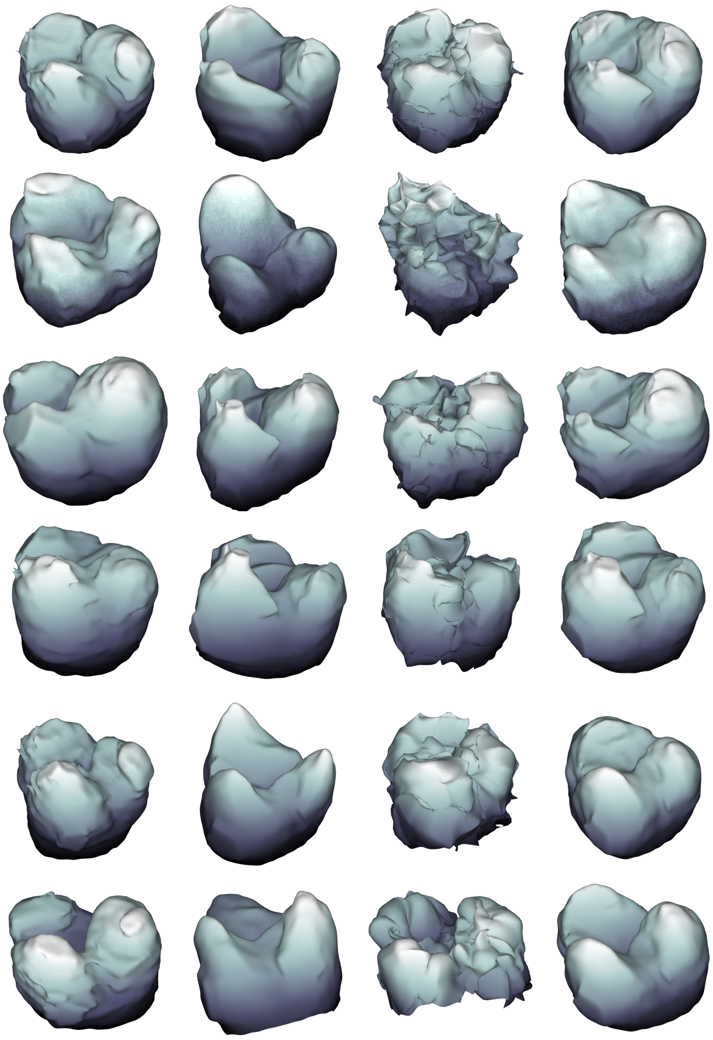


Figure 6.16: Segmentation comparison, where each row is the result of a different test image. Columns: 1) Ground truth; 2) modified ASM; 3) Zheng et al. [440]; 4) Proposed method.

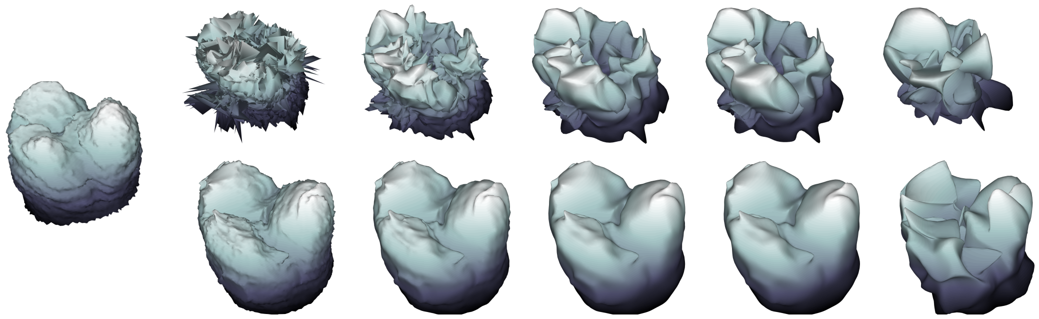


Figure 6.17: Repeated Taubin smoothing of meshes. Starting with an unregularized mesh (top) and a regularized mesh (bottom). Left to right: Ground truth, Applications of smoothing (iteration): 0, 1, 10, 20, 200. The entanglement of the mesh remains through the application of smoothing and the excessive smoothing results in meshes eventually beginning to diverge from the ground truth. Note that ground truth labeling is not locally smooth due to labeling process.

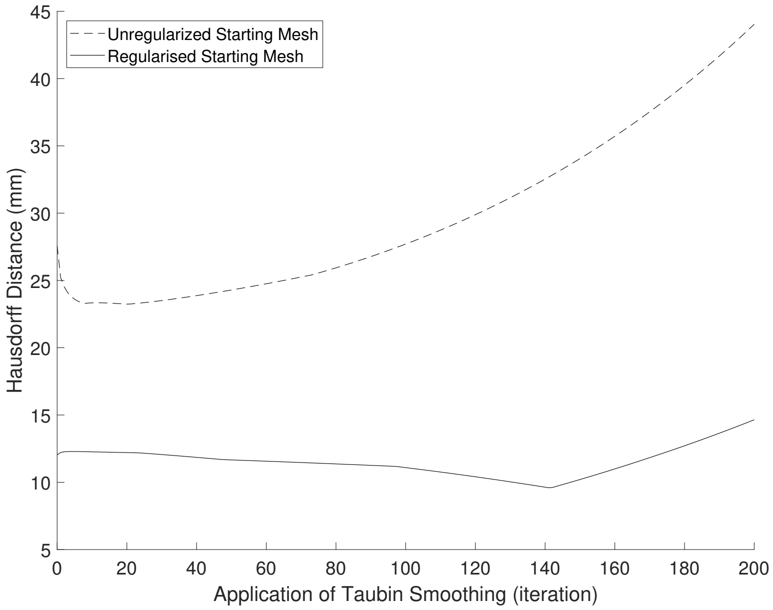


Figure 6.18: The effect of applying Taubin smoothing on mesh Hausdorff distance to ground truth. An initial increase in accuracy is observed, however both methods eventually suffer from the effects of meshes being over-smoothed. This over-smoothing effect can be seen in Figure 6.17

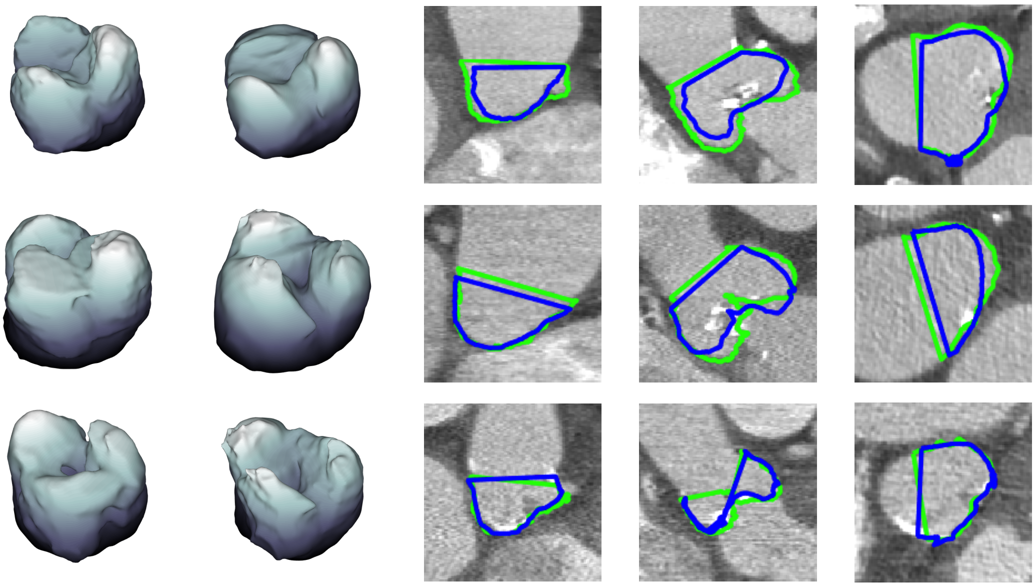


Figure 6.19: Example failure cases of the proposed pipeline. Left to right: Ground truth mesh, predicted mesh, yz-slice, xz-slice and xy-slice through segmented volume. Green contours show ground truth, blue contours show result of proposed method.

Failure cases for the proposed algorithm are shown in Figure 6.19. Visualization of the predicted mesh shows that the overall shape is often reasonable, with the overall model shape, including root and cusps, being present and well formed, however there are some issues with orientation alignment (Figure 6.19: second row and fourth column, bottom row and last column). The contours show that the failure cases give under-segmentation, often falling inside the boundary of the tissue. This suggests that either the search path is unable to localize the boundary, or the boundary detector can be improved to more robustly classify boundaries along that search path. The use of a shallow, fully connected boundary detector could be replaced with a Graph-CNN architecture which allows localized information from the small patches extracted along the search path to be learned.

6.6.3 Complexity of Marginal Space Learning classifier

Graph convolutions, as defined in Section 6.4, are an element-wise multiplication of the spectral graph signal and a spectral filter, resulting in $K^{l-1}N$ trainable weights and biases per output feature map, where N is the number of vertices in the graph and K^{l-1} is the number of input feature maps. For our Graph-CNN implementation, we utilize the property that smooth spectral filters provide localized filtering in the spatial domain. Such a formulation provides the benefit of spatially localized features, and a reduction in the number of tracked weights for our network to optimize. By tracking only $n < N$ weights and interpolating the filter up to N via a smoothing kernel, we are able to reduce the number of tunable parameters in an output feature map to $K^{l-1}n$. A smaller n provides more localization, but also introduces noise in the gradient steps during back-propagation optimization [2]. This parametrization helps improve parameter complexity of the graph convolution for a given filter from $\mathcal{O}(n)$ to $\mathcal{O}(K)$, given a constant tracked number of weights. For NNs, full connection provides $\mathcal{O}(n)$ complexity with a separate weighting for each input feature. If utilizing standard CNNs architectures, the ability to integrate local and contextual features comes with increased complexity from a multi-branch approach [35, 36] with a full set of weights for each branch, or from weight sharing through dilated kernels [462] which learns multiple scales of the same feature. With multi-resolution patches and Graph-CNNs we are able to learn spatial features between the high and low resolution input features without tracking multiple branches for each resolution.

6.7 Summary

In this chapter we have presented a novel method for deep learning in the irregular domain of the non-uniformly sampled grid. A patch-sampling mechanic generates a single spatial domain comprised of numerous layers at differing resolutions. Through the proposed Graph-CNN operators and architecture, we are able to learn features across multiple resolutions, utilizing the intrinsic spatial relationships between features at both local and wider scales. The use of conventional CNNs in such a domain is unfeasible due to the irregular sampling used, which does not satisfy the array structured input required for regular convolutional operations. The sampling method reduces the number of input features and does not require multiple branches to a pyramid of filters or inputs, reducing the complexity of the network architecture.

In evaluating the proposed method, we present a fully automatic, deformable modeling framework for 3D aortic root segmentation in CT images. The multi-resolution sampling strategy is generalized to 3D data, forming an irregularly spaced volume sampling method. The novel segmentation pipeline method significantly reduced the time taken for training by automatically finding shape correspondence across the training set and utilizing deep learning for discriminative feature extraction, rather than hand-crafting features for the task. The testing stage benefits from using Graph-CNN-MSL for aortic root detection, consecutively reducing the search space of pose parameters. The MSL search space optimization benefited from a novel implementation of the multi-resolution space for Graph-CNN based learning of features, learning spatially related features within an irregular spatial domain. Both qualitative and quantitative results justified our proposed segmentation pipeline over a top-down approach.

In the following chapter we explore the use of Graph-CNN architectures in representation learning on temporal feature in domains with an irregular domain. We return to the problem of Human Action Recognition and the human skeleton model, learning spatio-temporal descriptors from motion information of the skeletal joints.

Chapter 7

Graph Convolutional Neural Network for Temporal Feature Learning

Contents

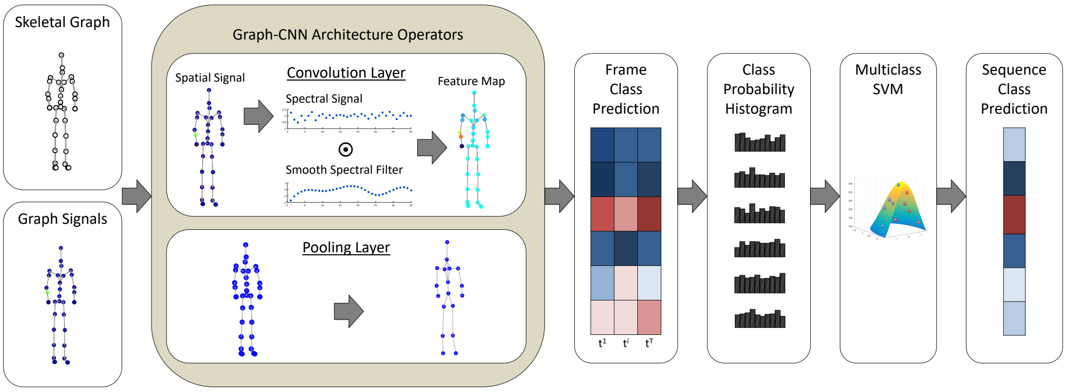


Figure 7.1: Graph based Convolutional Neural Network components. The Graph-CNN is designed from an architecture of graph convolution and pooling operator layers. Output prediction probabilities from classification on frames $t^{1:T}$ are histogram binned and passed into a multi-class SVM for sequence classification.

7.1 Introduction

Deep learning has been a prominent feature in data mining and pattern recognition in recent years, especially in problems such as classification and detection. Fully connected neural networks have shown promising usage in feature space learning in domains including text document analysis and genome characterization [408], with numerous architectures being designed that are able to self-tune features to the problem under investigation [13, 145]. By providing low level or raw input features, deep learning methods have been shown to learn high level descriptive features for various structures within the data [461, 463]. Such methods exhibit strong performances in various testing scenarios [145] and show promise for further data mining problems [464].

Convolutional Neural Networks (CNNs) expanded upon the concept of neural networks, learning localized features by convolving kernel filters with the input space to generate output feature maps [11]. With localization of features came a great increase in the ability of networks to learn descriptors in image mining problems [465, 466], and CNNs have shown promising applications in a wide range of image based data learning problems; including digit classification [410], face detection [409], and classification on a large number of classes [411]. CNN architectures presented two key operators, convolution and pooling, to learn the kernel receptive fields weightings. Convolution layers take input channels and output feature maps via the spatial convolution of the kernel weights across the spatial domain of input channels.

A pooling layer reduces the resolution of each feature map in the spatial dimension, simultaneously lowering complexity of the system and generalizing feature maps. Common pooling operations include taking an average or maximum of feature intensity within receptive cells over the input map [336].

In this study, we utilize graph-based signal processing techniques to generate a Graph-CNN architecture on the human skeletal model, providing an irregularly sampling patch. Recent study has shown that graph-based signal processing techniques can learn on irregular domains present in a wide range of applications [2, 415, 417]. The concept of employing the graph Laplacian to undertake signal processing based kernel learning on geometrically irregular space was first introduced in [417], while [415] goes on to explore use of smooth filters to identify localized regions in the spatial domain. The presented Graph-CNN operators are utilized to construct deep learning architectures for problem domains beyond image processing and the regular CNNs. We provide evaluation of Graph-CNN on the 3D pose Human Action Recognition (HAR) problem by developing an architecture that classifies human action from 3D skeletal representation, omitting any appearance information. The pipeline is achieved by training a Graph-CNN to identify individual frames before classifying whole sequences via an SVMs trained on the output probabilities of the Graph-CNN.

This study shows the first usage of the Graph-CNN architecture for the HAR from 3D pose problem, implementing deep learning in the natural spatial domain of the skeletal model. The proposed Graph-CNN avoids hand-tuning features and the spatial embedding utilized by current methods to adapt the 3D pose information into the regular CNN framework. By using very low-level features of motion, the network is able to learn spatial relationships on the irregular domain of the graph by the proposed convolution and pooling operations.

The rest of the chapter is as follows. Section 7.2 describes Graph-CNN architecture, providing convolution and pooling operators in the graph domain by use of graph-based signal-processing. A domain specific application is then presented in the context of human action recognition in Section 7.3, with results presented in Section 7.4. Conclusions are then drawn in Section 7.5.

7.2 Proposed Approach

We propose to learn localized information across a graph representation of the human skeleton, as defined by the placement of Motion Capture (MoCap) markers on the body, by utilizing the Graph-CNN operators outlined in Chapter 5. For this study we look at using skeletal pose

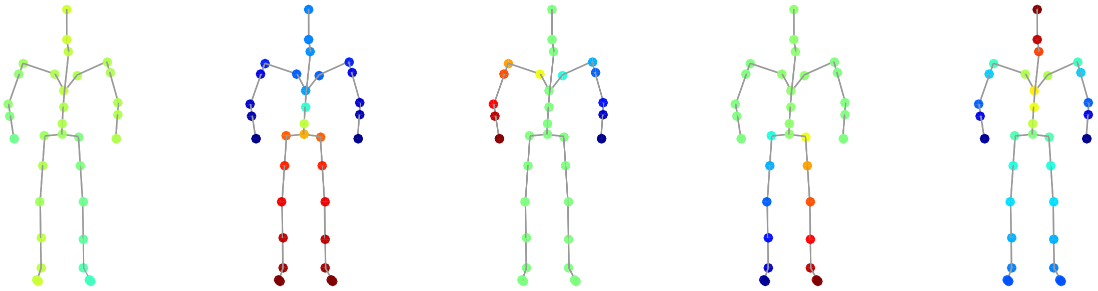


Figure 7.2: The first five eigenvectors of the human skeletal graph, $U_{1:5}$. These low frequency eigenvectors display strong relation to key structures in the human skeleton. By utilizing U we can compute the Graph Fourier Transform of input graph signals, allowing for spectral filtering in the Graph-CNN convolution operator.

information directly, however the use of pose estimation techniques can be utilized to obtain predicted joint positions within a 3D space [467, 468]. We first look at MoCap information as it shows a marked improvement in pose accuracy over current estimation methods [126]. The skeletal model domain contains the explicit spatial relationship between the markers and details the layout of structures such as the joints and bones making up the skeleton. Such a graph can be seen as a natural representation of the problem domain, upon which localized information can be learned via an architecture that utilizes the graph convolutional neural network operators. We will present the construction of a graph representation of a given motion capture marker domain, upon which we will project 3D pose and motion features to learn a representation via a deep Graph-CNN architecture. These methods are then implemented and evaluated for sequence-wise HAR classification in Section 7.3. See Figure 7.1 for an overview of the general proposed Graph-CNN architecture pipeline for human action recognition.

As discussed in Chapter 2 there are many methods which utilize appearance based information to estimate the 3D pose of a human skeletal model from still images [469–471]. The proposed method can make use of such skeletal models obtained via pose estimation, however the accuracy of the pose obtained from such techniques is often less accurate than the positioning achieved by professional motion capture sequences [126]. A disjoint pipeline approach, or even an end-to-end pose-estimation and Graph-CNN approach, going from input images to learning features on the extracted skeleton is possible. In order to evaluate the feasibility in using Graph-CNNs architectures to learn information on the skeleton model, we first utilize skeletal sequences obtained from MoCap techniques.

Table 7.1: Action classes of the Multi-Modal Human Action Database (MHAD).

MHAD		
Jump in place	Jumping jacks	Bending
Punch	Wave two hands	Wave one hand
Clap	Throw ball	Stand up
Sit down	Sit down then stand up	

7.3 Application: Recognition of Human Action from 3D Pose

Graph-CNN is generalizable to numerous irregular domain problems; including sensor networks, mesh signals, and text corpora. This study focuses on its use within Human Action Recognition of 3D skeletal pose, with classification on the Berkeley Multimodal Human Action Database (MHAD) dataset [84]. To the best of our knowledge this is the first such study that formulates the HAR problem with the use of Graph-CNN. A common method in pose-based HAR is to represent the body in terms of the 3D coordinate points that represent each of N tracked joints on a given individual across time t. In recent CNN based HAR methods, the joints have been hand-selected and subjected to hand-crafted feature extraction. This is used to create a 2D representation of the motion of the joints that can be easily passed as input to a standard CNN construction for classification of each action class [29, 30]. In this representation there is a risk of relying on the suitable selection of features and their orientation when embedded in the 2D space to obtain the optimal performance. Use of such highly tuned feature extraction, and forced spatial embedding, can easily lead to over-fitting as they fail to handle a wider array of action classes. By converting the human skeletal model to a graph-based representation, we are able to utilize our Graph-CNN method without arbitrarily defining a set of hand-crafted high-level features, or projecting the data into a regular space just to suit standard CNNs. We instead allow the network to mine the features that it requires to suitably generalize the training set observations, without making assumptions on the spatial relationship of hand-crafted features.

MHAD contains 11 action classes (Table 7.1), performed by 12 subjects, and captured via an array of modalities. We utilize the 3D motion capture information to represent the pose of the human body during an observation, omitting appearance information in this instance. Although appearance and depth fusion has shown benefit in HAR, the problem of action recognition from pose is of interest to this study. The motion capture data provides 35 tracked points on the body, captured at a very high frame rate of 480Hz. We normalize the data on a sequence-by-sequence basis in both orientation to camera and scale, as per the normalizing algorithm presented by [41]. Due to the high frame capture rate, we subsampled the sequences down from 480Hz to 30Hz, bringing it in line with commercial pose capturing sensors such as the Microsoft Kinect and Kinect V2.

7.3.1 Graph Construction for Human Action Recognition from Skeletal Pose

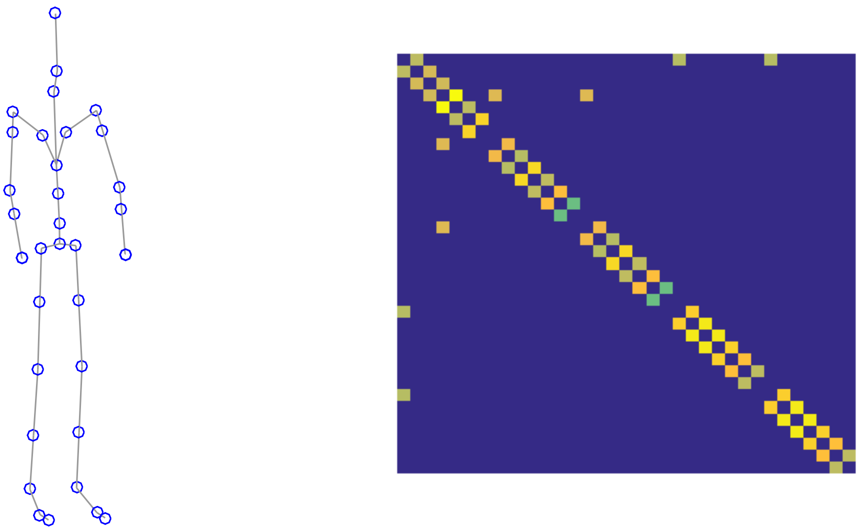
The problem of human pose has a well-defined structure of connectivity to formulate into a graph. Tracked 3D points in MoCap data constitute the graph vertices on the graph of the human body, and the adjacency between these points (bones) can be defined by the human skeleton in binary adjacency matrix A. From this prior knowledge of the domain we can define connected vertices for the human skeleton as in Figure 7.3a. Given S training observations and adjacency matrix A, we can generate weight matrix W for the edge between adjacent vertices n and m as

$$w_{n,m} = A_{n,m} + \exp\left(- \frac{dist(v_n, v_m)}{2\sigma^2} \right) \tag{7.1}$$

where σ is the average bone length, and $\text{dist}(v_{s,n}, v_{s,m})$ is the average squared Euclidean distance between adjacent vertices n and m across S observations

$$dist(v_n, v_m) = \frac{1}{S} \sum_{s=1}^{S} ||v_{s,n} - v_{s,m}||^2 \tag{7.2}$$

By adding the adjacency matrix into the similarity weighting matrix we are able to define weighted edges for data points that occupy the same physical space. Such phenomena can be common in 3D pose data for smaller digits such as fingers and toes where confidence of tracking is low. The final form of W provides zeros for non-edge connection, ones for an edge which occupies the same XYZ locations on its two end points, and a value larger than one for all edges with a distance based weight, shown in Figure 7.3b. Due to their prominent use in pose based action recognition we wish to learn on low level joint motion features from each frame of the observation, [3, 25, 120]. We extract the XYZ coordinates, along with multi-scale motion features of velocity and acceleration for all tracked markers, returning an $V \times I \times X$ matrix of X frames with $I = 123$ channel graph signals residing on $V = 35$ vertices. We extract the features for each of the 3 spatial dimensions X, Y, and Z attributed to $v_n^{\text{X,Y,Z}}$, calculating the velocity as $vel(v_n^{\text{X}}) = \frac{\Delta v_n^{\text{X}}}{\Delta t}$ Velocity is calculated in relation to directional vectors of upper back to left shoulder, upper back to right shoulder, horizontal shoulder to shoulder, and vertical



(a) Edge connectivity.

(b) Weight matrix W.

Figure 7.3: Human skeletal graph for the MHAD MoCap data. a) Connective adjacency of the MoCap markers. b) Example edge weighting matrix as given by (7.1).

upper back to lower back. Acceleration is then given as $acc(v_n^{\mathrm{X}}) = \frac{\Delta vel(v_n^{\mathrm{X}})}{\Delta t}$, where t defines the time step and v_n^{X} is a given spatial dimension X, Y and Z of the tracked point v_n. Velocities are extracted for sub-second frame steps, calculating motion on the scales of $\frac{1}{30}^{\text{th}}$, $\frac{1}{10}^{\text{th}}$, $\frac{1}{5}^{\text{th}}$, $\frac{1}{4}^{\text{th}}$, and $\frac{1}{2}^{\text{th}}$ of a second (rounded up to the nearest frame), resulting in short-scale frame motion information.

7.3.2 Pooling on the Human Skeleton

As identified in Chapter 5, there are several methods which aim to coarsen graphs and the signals which reside upon them, providing an analogue to the pooling layers of more conventional CNNs architectures. Such operations aim to reduce the number of vertices within the graph in order to lower both computational complexity and memory requirements of architectures, but also generalize learned features within a localized spatial region. The use of Kron's reduction and Algebraic Multigrid (AMG) are highlighted in Chapter 5, and the results of both pooling methods on the human MoCap skeleton graph constructed within Section 7.3.1 is provided in Figure 7.4.

For this domain problem we utilize Kron's reduction [381], reducing G via a subset of

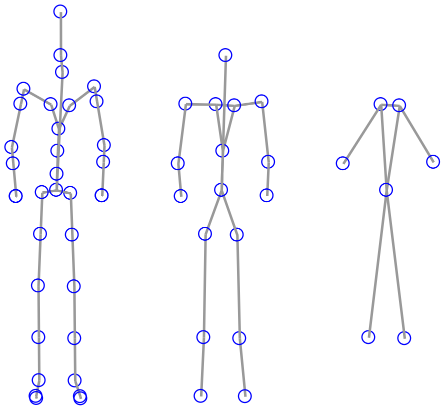


Figure 7.4: Two levels of Kron's Reduction pooling on the human MoCap skeleton.

vertices to keep $\hat{V}$ and the original graph Laplacian L by

$$\hat{L} = L_{\hat{V},\hat{V}} L_{\hat{V},\hat{V}^c} - L_{\hat{V}^c,\hat{V}^c} L_{\hat{V}^c,\hat{V}} \tag{7.3}$$

where $\hat{V}^c$ is the complement of $\hat{V}$. Kron's provides a means to reconstruct the reduced node weight matrix $\hat{W}$, via the removal of the discarded vertices from the rows and columns of the original graph Laplacian L

$$\hat{W}_{n,m} \begin{cases} -\hat{L}_{n,m} & \text{for} \quad n \neq m \\ 0 & \text{for} \quad i = j \end{cases} \tag{7.4}$$

A selection of $\hat{V}$ is made by identifying the largest eigenvalue in the sorted eigenvector matrix U, which coincides to the last eigenvalue, λ_N. Kron's then splits selections of V into two subsets based on the polarity of the associated eigenvector U_N [382]. We can therefore select vertices to retain as

$$\hat{V} = \{V_i\}; u_{N,i} <= 0 \tag{7.5}$$

and its complement, the vertices to remove, as

$$\hat{V}^c = \{V_i\}; u_{N,i} > 0 \tag{7.6}$$

These selected vertices are then used in (7.3) to construct $\hat{G}$. Kron's reduction has the effect of increasing the number of edge connections present in the graph, and as such it is often necessary to sparsify the connectivity in the graph by way of spectral sparsification [384, 419]. Given the original graph G and the edge selection tuning parameter Q, the weight matrix $\hat{W}$ for the subgraph $\hat{G}$ is initialized to 0 for all vertex pairings $w_{n,m}$. Sparsification selects Q random

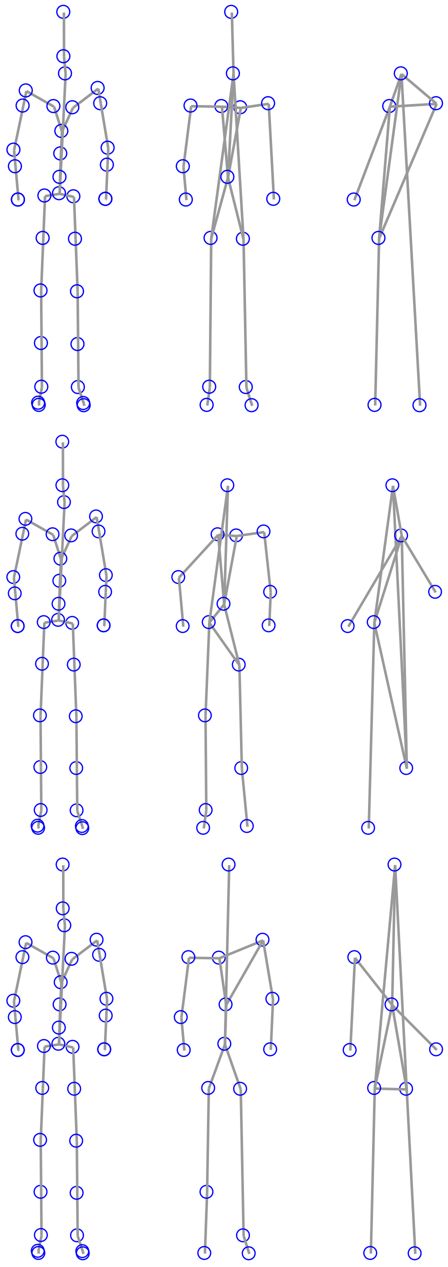


Figure 7.5: Two levels of AMG pooling on the human MoCap skeleton. Note that due to the random seeding of the AMG algorithm the resulting coarsened graph will vary with each run. Due to the requirement for a fixed Fourier basis, we perform the graph pooling in advance and use the collection of graphs for all observations.

edges $e(n,m) \in \varepsilon$ from the original weight matrix W, given a probability

$$p_e = \frac{\delta(n,m)W_{n,m}}{\sum_{e=(\alpha,\beta)\in\varepsilon}^{\varepsilon} \delta(\alpha,\beta)W_{\alpha,\beta}} \tag{7.7}$$

This selected edge weight is then accumulated into the new subgraph's weight matrix for the coarser graph layer with

$$\hat{W}_{n,m} = \hat{W}_{n,m} + \frac{W_{n,m}}{Qp_e} \tag{7.8}$$

With a coarser graph structure, $\hat{G}$, it is necessary to down-sample the graph signal $x_{1:N}$ into a new signal $\hat{x}_{1:\hat{N}}$ that is able to reside on $\hat{G}$. We down-sample $x \in \mathbb{R}^N$ on G to $\hat{x} \in \mathbb{R}^{\hat{N}}$ on $\hat{G}$ by pyramid analysis interpolation. Kron's pyramid utilizes a linear application of Green's functions, derived from the Laplacian, to interpolate the signal about a given vertex v_n in the spatial domain [384]. This allows us to project our samples from fine to coarse resolutions during forward passes through the network, and from coarse to fine scale during the backpropagation of errors.

Evaluation on the MHAD dataset has been carried out in several ways by previous studies. The initial paper reports a 7 vs. 5 approach, training on the observations of subjects 1 to 7, and testing on subjects 8 to 12 [84]. Baselines are reported for this evaluation in Table 7.2. Other methods have utilized a k-fold cross validation strategy, performed in MHAD as 5-fold validation. This validation trains a model on 4 randomized folds and tests on the 5th, before rotating through the folds to obtain an overview of method stability [29,30]. These results are provided in Table 7.3. K-fold cross validation is problematic however, as it enables observations from an individual subject to be present in both the training and testing data. This is often not the case for real-world usage of HAR systems. Instead, it is common for a model to be tested on subjects that are never observed in the training examples. The validation of models to this form of testing is known as Leave-One-Subject-Out (LOSO), where all samples of a single actor are omitted from the training set. This is often seen as a harder problem, as the inter- and intra-subject variations of action executions can vary substantially [389]. Previous LOSO results for MHAD are provided in Table 7.4. In addition to the original 7 vs. 5 validation, we evaluate our HAR based Graph-CNN model on both 5-fold and LOSO validation. This ensures that the testing subjects are completely novel to the trained model. LOSO evaluation will give us confidence that our proposed Graph-CNN model is able to learn subject general features that are able to act as informative classification descriptors when classifying unseen test subjects.

The overall HAR Graph-CNN sequence classifier is as follows. First a Graph-CNN is trained to classify frames in a sequence-wise fashion. A histogram is taken of the returned

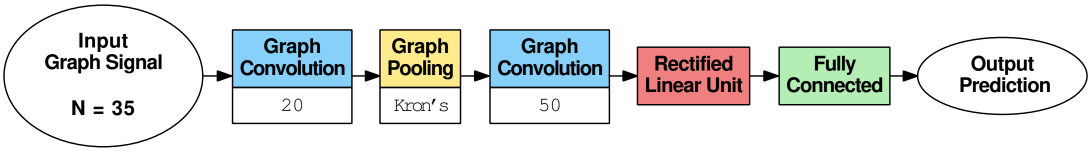
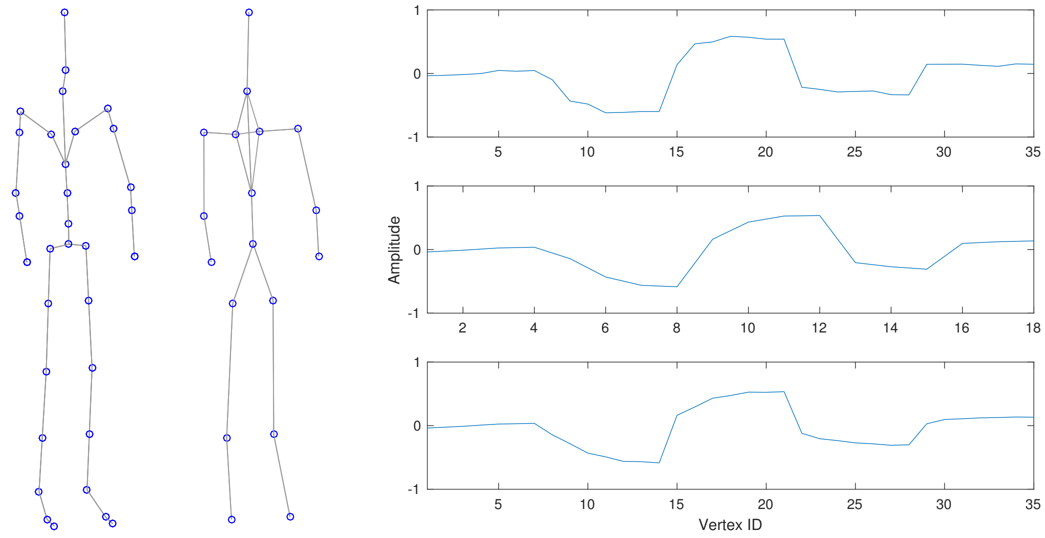


Figure 7.6: Graph-CNN architecture for classification of human actions on the human skeletal model.

probabilities for each class across the sequence, acting to remove the temporal variation from the sequence lengths and return a distribution of class predictions across all frames of the sequence, compressing the temporal dimension of the observations into a fixed length feature vector. For this study we utilized a histogram with 10 equispaced bins between 0 and 1, grouping the temporal probabilities in a coarse representation. It is possible to utilize a different bin spacing, and increasing the bin resolution or non-uniformly binning may produce a vector which is able to provide a finer distribution of the prediction probabilities. This vector is then used to train a multi-class SVM to classify whole sequences into a single action class for the action recognition task. For testing sequences, each frame is fed forward through the Graph-CNN and their class probabilities are then compressed via histogram binning to fit into the pre-trained SVM. The SVM returns a prediction on the class label for the entire sequence.

The architecture of the Graph-CNN, Figure 7.6, is defined as $C^{20}PC^{50}RF$; where C^{κ} defines a graph convolutional layer with 5 knots and κ output feature maps, P defines a graph coarsening, R defines a rectified linear unit layer, and finally F describes fully connected layers providing output class predictions. Graph pooling was achieved via Kron's reduction and spectral sparsification. The two human skeleton graphs used in the architecture can be seen in Figure 7.7a, and the signal pooling can be seen in Figure 7.7b. Networks were trained for 100 epochs via ADAGRAD optimization. An initial learning rate of 10^{-3} was used. Mini-batch sizes were set to 32. After the Graph-CNN was trained we performed a forward pass with the training data and a histogram was taken of the output predictions from the fully connected neural network layer, returning a fixed length feature vector for each sequence. These sequences were then used to train a multi-class SVM classifier. The test set was then fed forward through the Graph-CNN in the same manner, and the histogram representation of the testing sequence probabilities were then classified using the pre-trained SVM. We report on the final sequence-wise classification accuracy for the original 7 vs. 5, 5-fold and LOSO evaluations in Tables 7.2, 7.3, and 7.4 respectively.



(a) Vertex pooling.

(b) Signal pooling.

Figure 7.7: Kron's reduction on the MHAD MoCap skeleton graph and graph signal. a) Vertex coarsening in the spatial domain. b) Graph signal pooling: from top to bottom: original signal, pooled signal, up-sampled signal. Detailed in Section 7.3.2.

7.4 Comparative Analysis and Results

The proposed Graph-CNN shows improved results over the state-of-the-art methods reported on the MHAD dataset. The Graph-CNN architecture achieves 99.40% accuracy on sequence classification in the 5-fold cross validation evaluation scenario. It is important to note that the closest rival methods for 5-fold validation utilize hand-crafted features for classification, focusing on only using 3 out of the possible 35 MoCap markers and embedding the hand tuned features into an image space [29, 30]. Such model tuning may not translate when applied to other HAR datasets apart from MHAD motion capture data. This is especially problematic when observing action classes that use alternative joints, such as the legs, and would require re-tuning of the extracted features. Table 7.2 shows that the proposed method improves over the baselines reported by [84], and also over newer methods presented by [472].

The 100% accuracy reported by [473] is an obvious issue, suggesting that the problem is solved for the 7 vs. 5 evaluation scenario. The benefit our proposed Graph-CNN gives over the method provided by [472] is that utilization of very low-level features, whereas [472] extract a large number of temporal scales across many hierarchical clusters of the human body. Table 7.4 shows that the proposed Graph-CNN is able to perform well when tested on an individual subject which it has never observed before in the LOSO scenario.

In all compared approaches, the use of LOSO cross validation provides a lower classification accuracy than with 5-fold cross validation. It is expected that classification of a novel individual's performance of an action should be a challenging task, with a given subject not being present within the observed training data. In 5-fold cross validation we are holding out a sample of sequences, where the training data may contain observations of a given individual, and as such the trained model has chance to learn some features on a given subjects behavior. Human behavior can be very personal and individualized [474, 475] and Leave-One-Subject-Out cross validation can provide insight into a model's generalization of information across different individuals, often showing a lower predictive accuracy in comparison to methods which contain previous observations of an individual within the training data.

In all of the closest state of the art results the use of hand-crafted features is evident. Although these features can provide strong performances on a given dataset it is difficult to apply them on a new HAR scenario due to their selection of informative joints and feature extractors. Using heavily hand-crafted features are at odds with the self-learning feature extractors of common deep learning methods such as CNNs, AutoEncoder (AE), and the proposed Graph-CNN. Graph-CNNs are able to optimizing towards informative features via gradient descent, obtaining an understanding of the initial observations based on very low-level or even raw data input. We are able to train Graph-CNN with very low-level motion and spatial information regarding each of the joints on the human skeleton, and from here the algorithm is able to learn generalized features for frame-wise classification.

Comparison between Figure 7.4 and Figure 7.5 shows the importance in selection of graph coarsening approach for a given application domain. Whereas Chapters 5 and 6 benefited from the use of AMG pooling, due to the reduction in nodes and edges, the human skeletal model is highly sensitive to the initialization of the random seeding of the agglomerative method. Overall the proposed Graph-CNN has shown strong performance in the domain of 3D pose based HAR. The graph convolution operator presented is able to generate feature maps on the spatially irregular graph of the human skeleton, acting as a learned feature extractor when trained within a deep learning framework. The graph coarsening operator allows us to reduce the graph resolution in order to generalize feature maps and reduce complexity. We have shown favorable classification accuracies in both 5-fold and Leave-One-Subject-Out evaluation. The performance on the 7 vs. 5 evaluation is also promising, given that the rival methods all utilize sets of user tuned features.

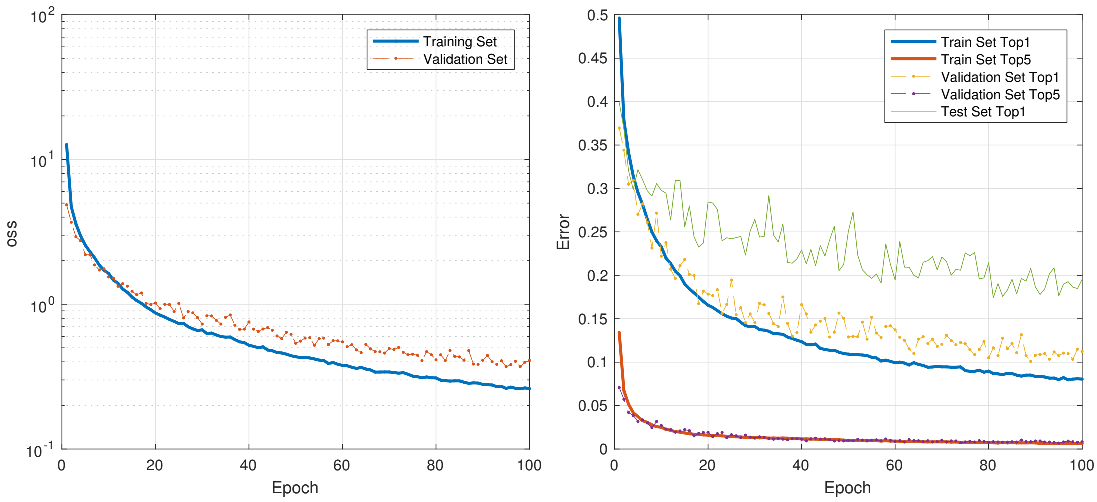


Figure 7.8: Average training curves during the frame-wise Graph-CNN feature embedding using Leave-One-Subject-Out cross-validation. Left: Objective loss. Right: Error rates for training, validation, and testing sets.

Table 7.2: Classification results on the Berkeley MHAD dataset using train on 7 subjects, test on 5 subjects.

Method	Classification Accuracy (%)
1 Nearest Neighbor [84]	74.82
3 Nearest Neighbor [84]	75.55
K-SVM [84]	79.93
Multi-factor [472]	87.83
Single-factor [472]	89.85
Hierarchy of LDSs [473]	**100.00**
Proposed Graph CNN + SVM	93.82

Table 7.3: Classification results on the Berkeley MHAD dataset using 5-fold cross validation.

Method	Classification Accuracy (%)
Hand-crafted CNN [29]	98.38
Hand-crafted Fuzzy CNN [30]	99.25
Graph CNN [415]	98.94
Proposed Graph CNN + SVM	**99.40**

Table 7.4: Classification results on the Berkeley MHAD dataset using Leave One Subject Out cross validation.

Method	Leave One Subject Out Classification Accuracy (%)
SVM [476]	96.06
Meta-Cognitive RBF Network [477]	97.58
Graph CNN [415] + SVM	94.54
Proposed Graph CNN + SVM	**98.33**

7.5 Summary

This study has proposed a method for the end-to-end mining of localized features in domains with irregular geometry. The combination of graph signal processing techniques and deep learning architecture design has allowed for features to be learned on low-level data in an end-to-end fashion. The local features are learned using spectral domain convolution of graph signals and spectral multipliers, in architecture similar to that seen in regular usage within standard CNNs. Convolutions are performed in the spectral domain of the graph Laplacian and allow for the learning of spatially localized features via the gradient calculations provided. Results are provided on the domain of HAR, although the scope for further application is much wider. Evaluation on HAR in a range of cross validation scenarios shows the ability of Graph-CNN to learn localized feature maps for frame-wise classification.

The use of a recurrent networks and LSTMs will move further towards utilizing the temporal dimension of the sequences. Development of a Recurrent Graph-CNN approach may benefit from the feedback of signals across time, whilst a Temporal-Graph-CNN can embed the temporal information within the graph representation, learning across fixed length time clips. Exploration of these approaches is required and is worthwhile to be generalized to the graph representation domain, given the development of recurrent networks and LSTM modules in the field of time series learning and in appearance based spatio-temporal analysis of videos via deep learning.

The major contribution of Graph-CNN over standard CNNs is the ability to function in the irregular geometric domain, with self-taught features mined from the observed data being used for function learning. On-going study into graph construction and reduction techniques is required to encourage uptake by a wider range of problem domains.

As discussed, it is possible to utilize pose estimation approaches to predict 3D locations of joints from appearance data [467, 468], such methods could be used to generate the skeletal information which is fed to the Graph-CNN architecture. The benefits of doing so would require evaluation, as it would require providing some improvement over the current state of spatio-temporal CNN methods. The addition of pose information and mined features may provide benefit to the use of appearance information in a fusion model, but we leave such exploration for future investigation.

In the following chapter we will summarize on the presented work, drawing conclusions on the use of deep learning and its generalization to irregular domain problems. We will look forward to the future of the field in irregular domain deep learning and will highlight potential avenues of research.

Chapter 8

Conclusions and Future Work

Contents

8.1 Conclusions

The presented work investigated the use of representation learning methods for identifying spatially related features on domains which do not exhibit the regular Cartesian spatial topology of image domain problems. We initially present an unsupervised clustering mechanism of identifying key primitive gestures for human action recognition, based on the underlying structure of the human skeleton. From this study we identify the need to adapt current spatial feature learning techniques present in the deep learning community beyond their current use in the image processing domain, providing a Graph-based Convolutional Neural Network approach able to generalize to more complex spatial topologies. The use of the Graph-CNN is proposed as a method for learning features at varying scales, introducing a multi-resolution volume sampling in which high and low resolution information is used simultaneously for detection of small-scale anatomical structures. We then explore the use of Graph-CNN in learning features from temporal information in the domain of human action recognition on the human skeleton model.

Representation learning has seen significant focus in recent years, with machine learning approaches utilizing ever lower level features in an attempt to develop algorithms which are able to learn an appropriate representation of the underlying data relationships. In Chapter 4 we present a method for learning a bag of words representation in which the distribution of primitive gestures are used as a discriminative tool for classifying human actions and interactions. An unsupervised Dynamic Time Warping (DTW) clustering mechanism drives the production of low level primitive gestures from observed sequences in a given dataset. An evolutionary approach is then used to optimize the selection of informative joints upon which we are able to identify gestures, effectively removing noisy information from the bag of gestures. Evaluation of this approach in both action and interaction recognition shows that dynamic time warping enables informative gestures to be identified, which in turn benefits the classification process. Such representation learning however still requires a two step approach, in comparison to the end-to-end learning provided by deep learning methods.

The use of deep learning has been observed to provide strong representation learning performance in a number of domains. This is especially notable in the image-processing domain, in which convolutional neural network architectures are able to identify spatially localized features by optimizing the weights of filter kernels. Such kernel filters are well defined for the regular Cartesian grid of image domain problems, however such convolution and pooling operators do no generalize to domains that do not reside on the grid. These domains have often resorted to ignoring spatial information within their input features by using standard fully connected neural network approaches, or by forcing a spatial embedding of the input domain in order to exploit the use of Convolutional Neural Network (CNN) operators. Such spatial embeddings can make assumptions on the spatial relationships between features that are not representative of the actual topology of the domain. By utilizing a graph representation of the input domain, we are able to encode the spatial structure of the domain as edge weightings between nodes on a graph, upon which we can exploit signal-processing techniques to provide analogous operations to convolution and pooling. The generalized approach to spatial feature learning in Chapter 5 provides a method of extracting localized features on signals which reside on the graph, with the optimization of filter weights provided via the deep learning framework. Such an approach will hopefully expand the use of locally informative deep representation learning to domains beyond its current use in image processing applications.

In Chapter 7 we evaluated the ability for a Graph-CNN architecture to learn features from temporal information on the human skeleton, classifying human behavior from skeletal motion. The 3D position of joints on the skeleton model provided an irregular space on which we wished to learn informative features that contained some spatial relationships. Rather than taking the approach of embedding observations on an image grid to utilize conventional CNN methodologies, we instead represented the skeleton as a graph, in which vertices represent joints and edges represent the bones connecting two joints. The Graph-CNN operators enabled us to define a deep learning architecture that takes the connectivity and relation between joints and develop learned filters for feature extraction. The classification results reported show that the proposed Graph-CNN architecture is able to accurately classify behaviors from the skeletal motion, improving on numerous hand-crafted feature descriptors for the task. Although the method did not achieve the 100% accuracy reported by one previous method, it utilized significantly less human intervention to develop a feature set, instead learning the representation from the observed training data.

In Chapter 6 we introduce the use of Graph-CNNs for mining features from multiple scales in a single observation, using a multi-resolution sampling scheme represented as a graph domain. The use of multi-scale features allows for wider context to be related to highly detailed localized information, identifying relationships between local and global structure. The development of the multi-resolution sampling introduces a lack of regularity in the spatial relationship between sampled points, and as such the use of standard CNNs is ill-defined. Representing the multi-resolution space as a graph composed of layers of varying density sample points, we are able to utilize the Graph-CNN pipeline proposed in 5 for representation learning at multiple scales in a single observation. This is evaluated in Chapter 6 for the purpose of aortic root segmentation from CT scans of the human body. The multi-resolution Graph-CNN is able to accurately and reliably localize the root structure by learning features on both the local shape of the root and its position relative to other structures in the chest cavity. Strong localization of the aortic structure is obtained, providing favorable usage in the marginal space learning scheme to reduce the search space complexity of estimating the pose parameters of the initial shape used for segmentation. In all segmentation methods evaluated the use of Graph-CNN as the marginal space learning classifier was observed to be superior, provided added benefit to the following segmentation stage of the overall pipeline.

8.2 Contributions

The main contributions can be summarized as follows.

- **A gesture learning scheme for a bag-of-gestures approach to action recognition.**
  We present a method for representation learning via a learned genome denoting training sample and feature set selection, clustering observed sequences of an action class into low-level primitive gestures. Gestures are then used in a bag-of-words approach to human action recognition. We dub the method 'bag-of-gestures', due to the use of spatio-temporal gestures in a bag-of-words style approach for the classification of higher-level behaviors from motion of the skeletal model. Observed sequences are selected based on the population's identification of informative joints and informative samples, learning generalized representations of action classes.

- **A deep learning approach to learning localized features on irregular spatial domains.**
  An approach to deep learning of localized feature representations is presented which generalizes conventional spatial filtering operations to irregular domains. Graph-based Convolutional Neural Network operators are introduced with a stable weight update scheme. The presented method provides a smoother weight optimization and an increased sta-

bility. An evaluation on a proof-of-concept domain is given, utilizing an irregularly sampled 2D grid upon which regular convolutional operators cannot function.

- **Local and global feature learning via a multi-resolution Graph-CNN.**
  We present a method for representation learning which incorporates local and global scale features into a single irregularly sampled spatial domain. The introduced method combines local and global information into a single filtering operation, maintaining spatial relationships between features whilst avoiding creating a complex branching networks or filters for separate scales, such as those seen in [35–39]. A case study is provided in the context of medical segmentation, utilizing multi-resolution Graph-CNNs for segmentation of the aortic root in human body scans.

- **Learning temporal features on domains with an irregular spatial topology.**
  Utilizing presented Graph-CNN operators to develop an architecture for learning features from temporal information. Spectral filters are learned via a Graph-CNN architecture that relates a given frame to a class based on multi-scale temporal information from preceding frames. Irregular domain operators enable localized feature descriptors to be constructed via a deep learning approach on the irregular spatial topology of the human skeleton without resorting to an assumption of a regular Cartesian embedding as seen in [29, 30]. An evaluation is provided on the skeletal model graph for the purpose of action recognition, learning features based on observed localized motion of bodily joints.

8.3 Future Work

The adaptation of deep learning approaches which utilize spatially related feature descriptors to a more generalized form has large scope. Deep learning methods that use convolutional operators on the Cartesian grid may provide benefit to further application domains by being formulated on the graph domain. By generalizing the representation learning scheme it would be possible to enable more application domains to access the significant gains observed in self-learning feature descriptor approaches. The number of application domains is vast, from social networks to distributed weather monitoring stations. It will be interesting to see how such applications are able to make use of deep learning methods, which previously were unable to make use of the explicit spatial relations between nodes on the input domain.

The development of methodologies within the field of irregular domain deep learning is still relatively young. Understanding of the learned filters requires study, including study of the optimization and visualization of learned filters. Visualizing feature maps on the spatial domain is possible in certain domains, due to the defined spatial structure, however an optimized strategy for exploring the learned descriptors like those seen in CNN filter visualization is not currently provided. Advances in deep learning in the area of ConvNets, such as those identified in Section 3.4 are obvious targets for generalization to the irregular domain approaches provided within this work. The field of regular domain deep learning has expanded to present the use of residual information with ResNets [12], AutoEncoders [478–480], Inception modules [13], and Generative Adversarial Networks [344]. All of these methods currently make use of the regular spatial domain input of images and volumes, however their utilization could theoretically be applied to the graph domain.

The selection of applicable graph construction and pooling methods for a given problem domain is still an area of active study, and opening collaborations with the graph domain community will help to develop an understanding of the choice of approaches for a given Graph-CNN architecture. By generalizing the convolution and pooling operations the ability to use a deep representation learning approach with a localized filtering behavior is more readily available to numerous application domains. A drawback of this relaxation is that applying domain-targeted constraints on how a architecture is able to utilize the input data can provide some benefits to applications within the specific domain; much as the application of the array-based input constraint provided significant gains in performance for the image domain community. Efforts to develop intuition regarding the application of graph based deep learning is required to help spread the adoption of the methods but may take time, especially given the on-going development of novel methods and understanding in the deep learning community as a whole. The graph domain community have already established methods for clustering graph types [481] and for the clustering of similar vertices [482], and making use of this knowledge will help to facilitate the development of Graph Convolutional Neural Network methodology.

Applying the Graph-CNN approach to the human action recognition problem appears promising, avoiding the need to hand-craft features or embed the information into an image space. Given the aim of understanding more complex behaviors and interactions, one area for further study is in developing a graph which models the relationship between multiple individuals. The production of a graph representation of multiple skeletons may enable learning of features for both participants in an interaction. Such a scheme may represent each actor as a skeletal graph with features learned in two potential ways. Firstly, we could produce a branched network in which we produce a branch for each actor in the scene, learning a selection of features for each stream. For example, in a two person interaction we would learn a feature extractor branch for person *A* and *B*, each describing their behavior within the interaction; i.e. describing person *A* as a 'kicker' and person *B* as 'being kicked'. Branches would merge at some point further in the pipeline in order to classify the overall interaction. An alternative approach may generate a singular graph relating multiple human skeletons to one another, providing a tighter coupling between the features describing individuals in the scene. Such a formulation would learn the interaction as a whole, rather than descriptors of roles within the interaction.

The learning of temporal features via Graph-CNN would benefit from expanding current recurrent neural network methods to the generalized graph domain. The Long Short-Term Memory (LSTM) modules provided for learning interactions of features over extended time scales would allow raw observations on the graph to generate temporally informative features. Such an approach would bring the learning of temporal information closer to the goal of representation learning, with motion information being learned via the gated memory mechanism of the LSTM. This ties back to the possibility in evaluating previously established deep learning approaches to find areas in which a generalized spatial sampling method will provide benefits to domains beyond image processing. Such an approach could incorporate the multi-scale feature learning shown in Chapter 6, learning long and short distance features in the temporal dimension. Recurrent neural networks and LSTMs learn such relationships by feedback loops in the network structure. A graph approach may define temporal information as a multi-resolution temporal window, with the detail of a signal degrading over successive time frames. Such temporal information is used in numerous time series domains; including natural language processing and sequence synthesis.

Overall the potential for deep learning in irregular domains is vast, the field has numerous open problems to explore and understand. The development of this emerging community will help to bring deep learning approaches to a wider collection of applications, benefiting both the machine learning community and those application domains able to make use of the representation learning technique.

Bibliography

[1] M. Edwards and X. Xie, “Graph-based CNN for human action recognition from 3d pose,” in Proceedings of the *British Machine Vision Conference Workshop: Deep Learning on Irregular Domains*, 2017.

[2] M. Edwards and X. Xie, “Graph based convolutional neural networks,” in Proceedings of the *British Machine Vision Conference*, 2016, pp. 114.1–114.11.

[3] M. Edwards, J. Deng, and X. Xie, “From pose to activity: Surveying datasets and introducing CONVERSE,” *Computer Vision and Image Understanding*, vol. 144, pp. 73–105, 2016.

[4] M. Edwards and X. Xie, “Generating local temporal poses from gestures with aligned cluster analysis for human action recognition,” in Proceedings of the *British Machine Vision Workshop*, 2015.

[5] M. Edwards, J. Deng, and X. Xie, “Labelling subtle conversational interactions,” in Proceedings of the *Annotation of User Data for Ubiquitous Systems Workshop*, 2017.

[6] J. Deng, X. Xie, and M. Edwards, “Combining stacked denoising autoencoders and random forests for face detection,” in Proceedings of the *International Conference on Advanced Concepts for Intelligent Vision Systems*, 2016, pp. 349–360.

[7] M. Edwards and X. Xie, “Footstep pressure signal analysis for human identification,” in Proceedings of the *International Conference on Biomedical Engineering and Informatics*, 2014, pp. 307–312.

[8] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[9] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer, 2001, vol. 1.

[10] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer, "cuDNN: Efficient primitives for deep learning," *CoRR*, vol. abs/1410.0759, 2014. [Online]. Available: http://arxiv.org/abs/1410.0759

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[14] Y. Zhao, H. Di, J. Zhang, Y. Lu, and F. Lv, "Recognizing human actions from low-resolution videos by region-based mixture models," in Proceedings of the *IEEE International Conference on Multimedia and Expo*, 2016, pp. 1–6.

[15] A. T. Nghiem, F. Bremond, M. Thonnat, and V. Valentin, "ETISEO, performance evaluation for video surveillance systems," in Proceedings of the *IEEE Conference on Advanced Video and Signal-Based Surveillance*, 2007, pp. 476–481.

[16] G. Zen, L. Porzi, E. Sangineto, E. Ricci, and N. Sebe, "Learning personalized models for facial expression analysis and gesture recognition," *IEEE Transactions on Multimedia*, vol. 18, no. 4, pp. 775–788, 2016.

[17] S. w. Leigh, H. Agrawal, and P. Maes, "Robotic symbionts: Interweaving human and machine actions," *IEEE Pervasive Computing*, vol. 17, no. 2, pp. 34–43, 2018.

[18] A. Gaidon, Z. Harchaoui, and C. Schmid, "Temporal localization of actions with actoms," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 35, no. 11, pp. 2782–95, 2013.

[19] Y. Zhang, L. Cheng, J. Wu, J. Cai, M. N. Do, and J. Lu, "Action recognition in still images with minimum annotation efforts," *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5479–5490, 2016.

[20] B. Fernando, S. Shirazi, and S. Gould, “Unsupervised human action detection by action matching,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 1604–1612.

[21] Y. Zhou, J. Deng, and S. Zafeiriou, “Improve accurate pose alignment and action localization by dense pose estimation,” in Proceedings of the *IEEE Conference on Automatic Face and Gesture Recognition*, 2018, pp. 480–484.

[22] R. Li, Z. Liu, and J. Tan, “Human motion segmentation using collaborative representations of 3d skeletal sequences,” *IET Computer Vision*, vol. 12, no. 4, pp. 434–442, 2018.

[23] S. Liu, L. Feng, Y. Liu, H. Qiao, J. Wu, and W. Wang, “Manifold warp segmentation of human action,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1414–1426, 2018.

[24] I. Laptev, “On space-time interest points,” *International Journal of Computer Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[25] A. Yao, J. Gall, G. Fanelli, and L. V. Gool, “Does human action recognition benefit from pose estimation?” in Proceedings of the *British Machine Vision Conference*, 2011, pp. 67.1–67.11.

[26] D. Wu, N. Sharma, and M. Blumenstein, “Recent advances in video-based human action recognition using deep learning: A review,” in Proceedings of the *International Joint Conference on Neural Networks*, 2017, pp. 2865–2872.

[27] V. Veeriah, N. Zhuang, and G. J. Qi, “Differential recurrent neural networks for action recognition,” in Proceedings of the *International Conference on Computer Vision*, 2015, pp. 4041–4049.

[28] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie, “Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks,” in Proceedings of the *AAAI Conference on Artificial Intelligence*, 2016, pp. 3697–3703.

[29] E. P. Ijjina and C. K. Mohan, “Human action recognition based on MOCAP information using convolution neural networks,” in Proceedings of the *International Conference on Machine Learning and Applications*, 2014, pp. 159–164.

[30] E. P. Ijjina and C. K. Mohan, "Human action recognition based on motion capture information using fuzzy convolution neural networks," in Proceedings of the *International Conference on Advances in Pattern Recognition*, 2015, pp. 1–6.

[31] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, *Sequential Deep Learning for Human Action Recognition*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 29–39.

[32] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[33] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[34] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," *CoRR*, vol. abs/1607.07155, 2016. [Online]. Available: http://arxiv.org/abs/1607.07155

[35] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker, "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation," *Medical Image Analysis*, vol. 36, pp. 61 – 78, 2017.

[36] J. Kawahara and G. Hamarneh, "Multi-resolution-tract CNN with hybrid pretrained and skin-lesion trained layers," in Proceedings of the *Machine Learning in Medical Imaging*, 2016, pp. 164–171.

[37] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in Proceedings of the *International Conference on Learning Representations*, 2016.

[38] W. Yang, Y. Chen, Y. Liu, L. Zhong, G. Qin, Z. Lu, Q. Feng, and W. Chen, "Cascade of multi-scale convolutional neural networks for bone suppression of chest radiographs in gradient domain," *Medical Image Analysis*, vol. 35, no. Supplement C, pp. 421 – 433, 2017.

[39] L. Zeng, X. Xu, B. Cai, S. Qiu, and T. Zhang, “Multi-scale convolutional neural networks for crowd counting,” *CoRR*, vol. abs/1702.02359, 2017. [Online]. Available: http://arxiv.org/abs/1702.02359

[40] Z. Cui, W. Chen, and Y. Chen, “Multi-scale convolutional neural networks for time series classification,” *CoRR*, vol. abs/1603.06995, 2016. [Online]. Available: http://arxiv.org/abs/1603.06995

[41] A. A. Chaaraoui, J. R. Padilla-López, P. Climent-Pérez, and F. Flórez-Revuelta, “Evolutionary joint selection to improve human action recognition with RGB-D devices,” *Expert Systems with Applications*, vol. 41, no. 3, pp. 786–794, 2014.

[42] P. K. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, “Machine recognition of human activities: A survey,” *IEEE Transactions on Circuits Systems for Video Technology*, vol. 18, no. 11, pp. 1473–1488, 2008.

[43] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, “Actions as space-time shapes,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 29, no. 12, pp. 2247–53, 2007.

[44] M. Müller, T. Röder, and M. Clausen, “Efficient content-based retrieval of motion capture data,” *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 677–685, 2005.

[45] J. Sung, C. Ponce, B. Selman, and A. Saxena, “Unstructured human activity detection from rgbd images,” in Proceedings of the *International Conference on Robotics and Automation*, 2012.

[46] J. Nascimento, M. Figueiredo, and J. Marques, “Segmentation and classification of human activities,” in Proceedings of the *Interntional Workshop on Human Activity Recognition and Modelling*, 2005.

[47] E. Spriggs, F. De La Torre, and M. Hebert, “Temporal segmentation and activity classification from first-person sensing,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition Workshops*. Ieee, 2009, pp. 17–24.

[48] J. M. Carmona and J. Climent, “Temporal segmentation of human actions in video sequences,” in Proceedings of the *Intelligent Systems Conference*, 2017, pp. 786–790.

[49] G. Johansson, “Visual perception of biological motion and a model for its analysis,” *Perception & Psychophysics*, vol. 14, no. 2, pp. 201–211, 1973.

[50] G. Johansson, “Visual motion perception,” *Scientific American*, vol. 232, pp. 76–88, 1975.

[51] D. Marr and H. K. Nishihara, “Representation and recognition of the spatial organization of three-dimensional shapes,” in Proceedings of the *Royal Society of London. Series B, Containing papers of a Biological character.*, vol. 200, no. 1140, 1978, pp. 269–94.

[52] R. Rashid, “Towards a system for the interpretation of moving light displays,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 2, no. 6, pp. 574–581, 1980.

[53] D. Hogg, “Model-based vision: A program to see a walking person,” *Image and Visual Computing*, vol. 1, pp. 5–20, 1983.

[54] H. Lee and Z. Chen, “Determination of 3D human body postures from a single view,” *Computer Vision, Graphics and Image Processing*, vol. 30, no. 2, pp. 148–168, 1984.

[55] Z. Chen and H. Lee, “Knowledge-guided visual perception of 3-D human gait from a single image sequence,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 2, pp. 263–267, 1992.

[56] K. Rohr, “Towards model-based recognition of human movements in image sequences,” *CVGIP: Image Understanding*, vol. 59, no. 1, pp. 94 – 115, 1994.

[57] J. Aggarwal, Q. Cai, W. Liao, and B. Sabata, “Articulated and elastic non-rigid motion: A review,” in Proceedings of the *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, 1994, pp. 2–14.

[58] L. Campbell and A. Bobick, “Recognition of human body motion using phase space constraints,” *International Journal of Computer Vision*, pp. 624–630, 1995.

[59] J. Aggarwal and Q. Cai, “Human motion analysis: A review,” *Computer Vision and Image Understanding*, vol. 73, no. 3, pp. 428–440, 1999.

[60] R. Polana and R. Nelson, “Low level recognition of human motion (or how to get your man without finding his body parts),” in Proceedings of the *IEEE Workshop on Motion of Non-rigid and Articulated Objects*, 1994, pp. 77–82.

[61] S. Osaka, "Recognition of human body motions by robots," in Proceedings of the *Intelligent Robots and Systems*, 1992, pp. 2139–2146.

[62] M. Rossi and A. Bozzoli, "Tracking and counting moving people," *IEEE Transactions on Image Processing*, pp. 212–216, 1994.

[63] A. Azarbayejani and A. Pentland, "Real-time self-calibrating stereo person tracking using 3-D shape estimation from blob features," in Proceedings of the *International Conference on Pattern Recognition*, vol. 3, 1996, pp. 627–632.

[64] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[65] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden Markov model," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 1992.

[66] O. Chomat and J. Crowley, "Probabilistic recognition of activity using local appearance," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 1999, pp. 0–5.

[67] A. Bobick and J. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 23, no. 3, pp. 257–267, 2001.

[68] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[69] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan, "Fast human detection using a cascade of histograms of oriented gradients," *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[70] I. Laptev and T. Lindeberg, "Space-time interest points," in Proceedings of the *International Conference on Computer Vision*, 2003, pp. 432–439 vol.1.

[71] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in Proceedings of the *International Conference on Pattern Recognition*, vol. 3, 2004, pp. 32–36.

[72] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, “Behavior recognition via sparse spatio-temporal features,” in Proceedings of the *IEEE Interntional Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005, pp. 65–72.

[73] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, “Actions as space-time shapes,” in Proceedings of the *International Conference on Computer Vision*, 2005, pp. 1395–1402.

[74] G. Cheng, Y. Wan, A. N. Saudagar, K. Namuduri, and B. P. Buckles, “Advances in human action recognition: A survey,” *CoRR*, vol. abs/1501.05964, 2015.

[75] W. Choi, S. Khuram, and S. Savarese, “What are they doing? : Collective Activity Classification Using Spatio-Temporal Relationship Among People,” in Proceedings of the *International Conference on Computer Vision Workshops*, 2009, pp. 1282–1289.

[76] J. M. Chaquet, E. J. Carmona, and A. Fernández-Caballero, “A survey of video datasets for human action and activity recognition,” *Computer Vision and Image Understanding*, vol. 117, no. 6, pp. 633–659, 2013.

[77] L. Xia, C. Chen, and J. Aggarwal, “View Invariant Human Action Recognition Using Histograms of 3D Joints,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 20–27.

[78] O. Oreifej and Z. Liu, “HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 716–723.

[79] E. Borzeshi, O. Perez Concha, R. Xu, and M. Piccardi, “Joint action segmentation and classification by an extended hidden Markov model,” *IEEE Signal Processing Letters*, vol. 20, no. 12, pp. 1207–1210, 2013.

[80] W. Bian, D. Tao, and Y. Rui, “Cross-domain human action recognition,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 298–307, 2012.

[81] S. Stein and S. J. McKenna, “Combining embedded accelerometers with computer vision for recognizing food preparation activities,” in Proceedings of the *International Joint Conference on Pervasive and Ubiquitous Computing*, 2013.

[82] I. Ar and Y. Akgul, “Action recognition using random forest prediction with combined pose-based and motion-based features,” in Proceedings of the *IEEE International Conference on Electrical and Electronics Engineering*, 2013, pp. 315–319.

[83] C. Chen, R. Jafari, and N. Kehtarnavaz, “Improving Human Action Recognition Using Fusion of Depth Camera and Inertial Sensors,” *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 1, pp. 51–61, 2015.

[84] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, “Berkeley MHAD: A comprehensive multimodal human action database,” in Proceedings of the *IEEE Workshop on Applications of Computer Vision*, 2013, pp. 53–60.

[85] Q. Wu, Z. Wang, F. Deng, Z. Chi, and D. D. Feng, “Realistic Human Action Recognition With Multimodal Feature Selection and Fusion,” *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 43, no. 4, pp. 875–885, 2013.

[86] I. Lefter, G. J. Burghouts, and L. J. M. Rothkrantz, “An audio-visual dataset of human-human interactions in stressful situations,” *Journal on Multimodal User Interfaces*, vol. 8, no. 1, pp. 29–41, 2014.

[87] L. Liu, S. Ma, and Q. Fu, “Human action recognition based on locality constrained linear coding and two-dimensional spatial-temporal templates,” in Proceedings of the *Chinese Automation Congress*, 2017, pp. 1879–1883.

[88] I. El-Henawy, K. Ahmed, and H. Mahmoud, “Action recognition using fast hog3d of integral videos and smith-waterman partial matching,” *IET Image Processing*, vol. 12, no. 6, pp. 896–908, 2018.

[89] A. Ulhaq, X. Yin, J. He, and Y. Zhang, “On space-time filtering framework for matching human actions across different viewpoints,” *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1230–1242, 2018.

[90] M. Bregonzio, S. Gong, and T. Xiang, “Recognising action as clouds of space-time interest points,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1948–1955.

[91] B. Chakraborty, M. B. Holte, T. B. Moeslund, and J. Gonzàlez, “Selective spatio-temporal interest points,” *Computer Vision and Image Understanding*, vol. 116, no. 3, pp. 396–410, 2012.

[92] D. Weinland, R. Ronfard, and E. Boyer, “A survey of vision-based methods for action representation, segmentation and recognition,” *Computer Vision and Image Understanding*, vol. 115, no. 2, pp. 224–241, 2011.

[93] P. V. K. Borges, N. Conci, and A. Cavallaro, “Video-based human behavior understanding: A survey,” *IEEE Transactions on Circuits Systems for Video Technology*, vol. 23, no. 11, pp. 1993–2008, 2013.

[94] A. Grushin, D. D. Monner, J. A. Reggia, and A. Mishra, “Robust human action recognition via long short-term memory,” in Proceedings of the *International Joint Conference on Neural Networks*, 2013, pp. 1–8.

[95] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014.

[96] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” *CoRR*, vol. abs/1406.2199, 2014.

[97] L. L. Presti and M. L. Cascia, “3d skeleton-based human action classification: A survey,” *Pattern Recognition*, vol. 53, pp. 130–147, 2016.

[98] A. Agarwal and B. Triggs, “Recovering 3d human pose from monocular images,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 28, no. 1, pp. 44–58, 2006.

[99] M. Andersen, T. Jensen, P. Lisouski, A. Mortensen, M. Hansen, T. Gregersen, and P. Ahrendt, “Kinect depth sensor evaluation for computer vision applications,” Aarhus University, Department of Engineering, Tech. Rep., 2012.

[100] K. Berger, S. Meister, R. Nair, and D. Kondermann, “A state of the art report on kinect sensor setups in computer vision,” in *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, M. Grzegorzek, C. Theobalt, R. Koch, and A. Kolb, Eds. Springer, 2013, vol. 8200, pp. 257–272.

[101] J. Han, L. Shao, D. Xu, and J. Shotton, “Enhanced computer vision with Microsoft Kinect sensor: A review,” *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1318–1334, 2013.

[102] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3D points," in Proceedings of the *IEEE International Workshop on Computer Vision and Pattern Recognition for Human Communicative Behavior Analysis*, 2010.

[103] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[104] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Progressive search space reduction for human pose estimation," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[105] B. Daubney and X. Xie, "Estimating 3D pose via stochastic search and expectation maximization," in Proceedings of the *Articulated Motion and Deformable Objects*, 2010.

[106] B. Daubney and X. Xie, "Entropy driven hierarchical search for 3D human pose estimation," *British Machine Vision Conference*, 2011.

[107] B. Daubney and X. Xie, "Tracking 3D human pose with large root node uncertainty," *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[108] K. Yun, J. Honorio, D. Chattopadhyay, T. L. Berg, D. Samaras, and S. Brook, "Two-person interaction detection using body-pose features and multiple instance learning," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2012, pp. 28–35.

[109] L. Huynh, T. Ho, Q. Tran, T. B. Dinh, and T. Dinh, "Robust classification of human actions from 3D data," in Proceedings of the *IEEE Interntional Symposium on Signal Processing and Information Technology*, 2012, pp. 263–268.

[110] Y. Zhu, W. Chen, and G. , "Fusing spatiotemporal features and joints for 3d action recognition," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 486–491.

[111] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3D action recognition with random occupancy patterns," in Proceedings of the *European Conference on Computer Vision*, 2012, pp. 872–885.

[112] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, “NTU RGB+D: A large scale dataset for 3D human activity analysis,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[113] F. De la Torre, J. Hodgins, A. Bargteil, X. Martin, J. Macey, A. Collado, and P. Beltran, “Tech. report cmu-ri-tr-08-22: Guide to the carnegie mellon university multimodal activity database,” Robotics Institute, Carnegie Mellon University, Tech. Rep., 2008.

[114] C. Wang, Y. Wang, and A. L. Yuille, “An approach to pose-based action recognition,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 915–922.

[115] F. Zhou, F. De la Torre, and J. K. Hodgins, “Hierarchical aligned cluster analysis for temporal clustering of human motion,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 35, no. 3, pp. 582–596, 2013.

[116] A. A. Chaaraoui and F. Flórez-revuelta, “Adaptive human action recognition with an evolving bag of key poses,” *Autonomous Mental Development*, vol. 6, no. 2, pp. 139–152, 2014.

[117] B. Yao, X. Jiang, A. Khosla, A. Lin, L. Guibas, and L. Fei-Fei, “Human action recognition by learning bases of action attributes and parts.” in Proceedings of the *International Conference on Computer Vision*, 2011, pp. 1331–1338.

[118] J. Deng, X. Xie, B. Daubney, H. Fang, and P. W. Grant, “Recognizing conversational interaction based on 3D human pose,” in Proceedings of the *International Conference on Advanced Concepts for Intelligent Vision Systems*, 2013, pp. 138–149.

[119] J. Deng, X. Xie, and B. Daubney, “A bag of words approach to 3D human pose interaction classification with random decision forests,” in Proceedings of the *Computational Visual Media Conference*, 2013.

[120] J. Deng, X. Xie, and B. Daubney, “A bag of words approach to subject specific 3D human pose interaction classification with random decision forests,” *Graphical Models*, vol. 76, no. 3, pp. 162 – 171, 2014.

[121] J. Deng, X. Xie, and S. Zhou, “Conversational interaction recognition based on bodily and facial movement,” in Proceedings of the *International Conference on Image Analysis and Recognition*, 2014.

[122] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. D. Bimbo, "3-D human action recognition by shape analysis of motion trajectories on riemannian manifold," *IEEE Transactions on Cybernetics*, vol. 45, no. 7, pp. 1340–1352, 2015.

[123] R. Slama, H. Wannous, M. Daoudi, and A. Srivastava, "Accurate 3D action recognition using learning on the Grassmann manifold," *Pattern Recognition*, vol. 48, no. 2, pp. 556 – 567, 2015.

[124] D. C. Luvizon, H. Tabia, and D. Picard, "Learning features combination for human action recognition from skeleton sequences," *Pattern Recognition Letters*, vol. 99, pp. 13 – 20, 2017.

[125] E. Ceseracciu, Z. Sawacha, and C. Cobelli, "Comparison of markerless and marker-based motion capture technologies through simultaneous data collection during gait: Proof of concept," *PLoS One*, vol. 9, no. 3, p. e87640, 2014.

[126] M. J. Gómez, C. Castejón, J. C. García-Prada, G. Carbone, and M. Ceccarelli, "Analysis and comparison of motion capture systems for human walking," *Experimental Techniques*, vol. 40, no. 2, pp. 875–883, 2016.

[127] T. H. Yu, T. K. Kim, and R. Cipolla, "Unconstrained monocular 3d human pose estimation by action detection and cross-modality regression forest," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3642–3649.

[128] J. Song, L. Wang, L. V. Gool, and O. Hilliges, "Thin-slicing network: A deep structured model for pose estimation in videos," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5563–5572.

[129] A. A. Halim, C. Dartigues-Pallez, F. Precioso, M. Riveill, A. Benslimane, and S. Ghoneim, "Human action recognition based on 3d skeleton part-based pose estimation and temporal multi-resolution analysis," in Proceedings of the *IEEE International Conference on Image Processing*, 2016, pp. 3041–3045.

[130] W. Du, Y. Wang, and Y. Qiao, "Rpan: An end-to-end recurrent pose-attention network for action recognition in videos," in Proceedings of the *International Conference on Computer Vision*, 2017, pp. 3745–3754.

[131] F. Zhou, F. De la Torre, and J. K. Hodgins, "Aligned cluster analysis for temporal segmentation of human motion," Carnegie Mellon University, Tech. Rep., 2008.

[132] V. Parameswaran and R. Chellappa, “View invariants for human action recognition,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2003, pp. 613–619.

[133] A. Chaaraoui, P. Climent-Prez, and F. Flrez-Revuelta, “An efficient approach for multi-view human action recognition based on bag-of-key-poses,” in Proceedings of the *Human Behavior Understanding*, vol. 7559, 2012, pp. 29–40.

[134] A. A. Chaaraoui, P. Climent-Pérez, and F. Flórez-Revuelta, “Silhouette-based human action recognition using sequences of key poses,” *Pattern Recognition*, vol. 34, no. 15, pp. 1799–1807, 2013.

[135] H. Sakoe and S. Chiba, “Dynamic programming algorithm optimization for spoken word recognition,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[136] H. Li and M. Greenspan, “Multi-scale gesture recognition from time-varying contours,” in Proceedings of the *International Conference on Computer Vision*, vol. 1, 2005, pp. 236–243.

[137] Y. Chen, Q. Wu, and X. He, “Using dynamic programming to match human behavior sequences,” *Control, Automation, Robotics and Vision*, pp. 17–20, 2008.

[138] T. Vajda, “Action recognition based on fast dynamic-time warping method,” in Proceedings of the *International Conference on Intelligent Computer Communication and Processing*, 2009, pp. 127–131.

[139] Y. Shen and H. Foroosh, “View-invariant action recognition from point triplets,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 31, no. 10, pp. 1898–1905, 2009.

[140] S. Sempena, N. U. Maulidevi, and P. R. Aryan, “Human action recognition using dynamic time warping,” in Proceedings of the *International Conference on Electrical Engineering and Informatics*, 2011, pp. 1–5.

[141] D. Weinland and E. Boyer, “Action recognition using exemplar-based embedding,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–7.

[142] T. Wang, S. Wang, and X. Ding, "Learning a similarity metric discriminatively for pose exemplar based action recognition," in Proceedings of the *Interntional Congress on Image and Signal Processing*, 2011, pp. 404–408.

[143] T. Hassner, "A critical review of action recognition benchmarks," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 245–250.

[144] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[145] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Proceedings of the *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[146] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in Proceedings of the *International Conference on Computer Vision*, 2011, pp. 2556–2563.

[147] K. Soomro, A. R. Zamir, and M. Shah, "CRCV-TR-12-01: UCF101 : A dataset of 101 human actions classes from videos in the wild," University of Central Florida, Center for Research in Computer Vision, Tech. Rep., 2012.

[148] K. Papoutsakis, C. Panagiotakis, and A. A. Argyros, "Temporal action co-segmentation in 3D motion capture data and videos," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2146–2155.

[149] S. Stein and S. J. McKenna, "50 Salads dataset." [Online]. Available: http://cvip.computing.dundee.ac.uk/datasets/foodpreparation/50salads/

[150] S. Stein and S. J. McKenna, "User-adaptive models for recognizing food preparation activities," in Proceedings of the *Interntional Workshop on Multimedia for Cooking and Eating Activities*, 2013, pp. 39–44.

[151] S. J. Blunsden and R. B. Fisher, "Behave interactions test case scenarios." [Online]. Available: http://groups.inf.ed.ac.uk/vision/BEHAVEDATA/INTERACTIONS/

[152] S. J. Blunsden and R. B. Fisher, "The BEHAVE video dataset: Ground truthed video for multi-person behavior classification," *Annals of the BMVA*, no. 4, pp. 1–11, 2010.

[153] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Berkeley Multimodal Human Action Database." [Online]. Available: http://tele-immersion.citris-uc.org/berkeley_mhad

[154] Y. Kong, Y. Jia, and Y. Fu, "BIT-Interaction dataset." [Online]. Available: https://sites.google.com/site/alexkongy/software

[155] Y. Kong, Y. Jia, and Y. Fu, "Learning human interaction by interactive phrases," in Proceedings of the *European Conference on Computer Vision*, vol. 7572, 2012, pp. 300–313.

[156] Cornell University, "Cornell Activity Datasets CAD-60, CAD-120." [Online]. Available: http://pr.cs.cornell.edu/humanactivities/data.php

[157] H. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from RGB-D videos," *International Journal of Robotics Research*, vol. 32, no. 8, pp. 951–970, 2013.

[158] Institute of Automation Chinese Academy of Sciences, "CASIA action database for recognition." [Online]. Available: http://www.cbsr.ia.ac.cn/english/Action%20Databases%20EN.asp

[159] K. Huang, D. Tao, Y. Yuan, X. Li, and T. Tan, "View-independent behavior analysis." in Proceedings of the *IEEE Conference on Systems, Man and Cybernetics*, vol. 39, no. 4, 2009, pp. 1028–35.

[160] R. Fisher, "CAVIAR test case scenarios." [Online]. Available: http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/

[161] R. Fisher, "The PETS04 surveillance ground-truth data sets," in Proceedings of the *International Workshop on Performance Evaluation of Tracking and Surveillance*, 2004.

[162] F. De la Torre, J. Hodgins, J. Montano, S. Valcarcel, R. Forcadam, and J. Macey, "Quality of Life Technology Center. Grand Challenge Data Collection." [Online]. Available: http://kitchen.cs.cmu.edu/

[163] CMU Graphics Lab, “CMU graphics lab motion capture database.” [Online]. Available: http://mocap.cs.cmu.edu/

[164] Swansea University Computer Vision and Medical Image Analysis Group, “CONVERSE dataset.” [Online]. Available: http://csvision.swan.ac.uk/converse

[165] I. Laptev and P. Pérez, “Drinking and smoking action annotaion.” [Online]. Available: http://www.di.ens.fr/~laptev/download.html

[166] I. Laptev and P. Pérez, “Retrieving actions in movies,” in Proceedings of the *International Conference on Computer Vision*, 2007, pp. 1–8.

[167] Inria, “ETISEO: Video understanding evaluation.” [Online]. Available: http://www-sop.inria.fr/orion/ETISEO/download.htm

[168] V. Bloom, D. Makris, and V. Argyriou, “G3D gaming datasets.” [Online]. Available: http://dipersec.king.ac.uk/G3D/G3D.html

[169] V. Bloom, D. Makris, and V. Argyriou, “G3D: A gaming action dataset and real time action recognition evaluation framework,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2012, pp. 7–12.

[170] V. Bloom, V. Argyriou, and D. Makris, “G3Di gaming datasets.” [Online]. Available: http://dipersec.king.ac.uk/G3D/G3Di.html

[171] V. Bloom, V. Argyriou, and D. Makris, “G3Di: A gaming interaction dataset with a real time detection and evaluation framework,” in Proceedings of the *European Conference on Computer Vision*, 2014, pp. 698–712.

[172] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, “HMDB: A large human motion database.” [Online]. Available: http://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/

[173] I. Laptev, M. Marszaek, C. Schmid, and B. Rozenfeld, “Learning human actions from movies.” [Online]. Available: http://www.di.ens.fr/~laptev/actions/

[174] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, “Learning realistic human actions from movies,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[175] M. Marszałek, I. Laptev, and C. Schmid, "Human actions and scenes dataset." [Online]. Available: http://www.di.ens.fr/~laptev/actions/hollywood2/

[176] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2929 – 2936.

[177] S. Hadfield, "Hollywood3D." [Online]. Available: http://cvssp.org/Hollywood3D/

[178] S. Hadfield and R. Bowden, "Hollywood 3D: Recognizing actions in 3D natural scenes," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3398–3405.

[179] L. Sigal, A. O. Balan, and M. J. Black, "HumanEva dataset." [Online]. Available: http://humaneva.is.tue.mpg.de/

[180] L. Sigal, A. Balan, and M. J. Black, "HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *International Journal of Computer Vision*, vol. 87, pp. 4–27, 2010.

[181] D. Weinland, R. Ronfard, and E. Boyer, "INRIA Xmas Motion Acquisition Sequences." [Online]. Available: http://4drepository.inrialpes.fr/public/viewgroup/6

[182] D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *Computer Vision and Image Understanding*, vol. 104, no. 2-3, pp. 249–257, 2006.

[183] M. S. Ryoo and L. Matthies, "JPL first-person interaction dataset." [Online]. Available: http://michaelryoo.com/jpl-interaction.html

[184] M. S. Ryoo and L. Matthies, "First-Person activity recognition: What are they doing to me?" in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2730–2737.

[185] "K3HI dataset." [Online]. Available: http://www.lmars.whu.edu.cn/profweb/zhuxinyan/DataSetPublish/dataset.html

[186] T. Hu, X. Zhu, W. Guo, and K. Su, "Efficient interaction recognition through positive action representation," *Mathematical Problems in Engineering*, vol. 2013, pp. 1–11, 2013.

[187] I. Laptev and T. Lindeberg, “Recognition of human actions.” [Online]. Available: http://www.nada.kth.se/cvap/actions/

[188] C. Wolf, J. Mille, E. Lombardi, O. Celiktutan, M. Jiu, E. Dogan, G. Eren, M. Baccouche, E. Dellandrea, C.-E. Bichot, C. Garcia, and B. Sankur, “The LIRIS human activities dataset.” [Online]. Available: http://liris.cnrs.fr/voir/activities-dataset/

[189] C. Wolf, E. Lombardi, J. Mille, O. Celiktutan, M. Jiu, E. Dogan, G. Eren, M. Baccouche, E. Dellandrea, C.-E. Bichot, C. Garcia, and B. Sankur, “Evaluation of video activity localizations integrating quality and quantity measurements,” *Computer Vision and Image Understanding*, vol. 127, pp. 14 – 30, 2014.

[190] M. P. I. for Informatics, “MPI08 dataset.” [Online]. Available: http://www.tnt.uni-hannover.de/project/MPI08_Database/

[191] G. Pons-Moll, A. Baak, T. Helten, M. Müller, H.-P. Seidel, and B. Rosenhahn, “Multisensor-fusion for 3D full-body human motion capture,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 663–670.

[192] A. Baak, T. Helten, M. Müller, G. Pons-Moll, B. Rosenhahn, and H.-P. Seidel, “Analyzing and evaluating markerless motion tracking using inertial sensors,” in Proceedings of the *European Conference on Computer Vision*, 2010, pp. 139–152.

[193] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele, “MPII cooking activities dataset.” [Online]. Available: https://www.mpi-inf.mpg.de/departments/computer-vision-and-multimodal-computing/research/human-activity-recognition/mpii-cooking-activities-dataset/

[194] M. Rohrbach and S. Amin, “A database for fine grained activity detection of cooking activities,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1194–1201.

[195] M. Rohrbach, M. Regneri, M. Andriluka, S. Amin, M. Pinkal, and B. Schiele, “MPII cooking composite activities.” [Online]. Available: https://www.mpi-inf.mpg.de/departments/computer-vision-and-multimodal-computing/research/human-activity-recognition/mpii-cooking-composite-activities/

[196] M. Rohrbach, M. Regneri, M. Andriluka, S. Amin, M. Pinkal, and B. Schiele, "Script data for attribute-based recognition of composite activities," in Proceedings of the *European Conference on Computer Vision*, 2012, pp. 144–157.

[197] Microsoft Research, "MSR action recognition datasets and codes." [Online]. Available: http://research.microsoft.com/en-us/um/people/zliu/actionrecorsrc/

[198] J. Yuan, "Discriminative subvolume search for efficient action detection," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2442–2449.

[199] J. Yuan, Z. Liu, and Y. Wu, "Discriminative video pattern search for efficient action detection." *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 33, no. 10, pp. 1728–1743, 2011.

[200] A. Kurakin, Z. Zhang, and Z. Liu, "A real time system for dynamic hand gesture recognition with a depth sensor," in Proceedings of the *European Signal Processing Conference*, 2012, pp. 1975–1979.

[201] S. Singh, S. Velastin, and H. Ragheb, "MuHAVi: Multicamera Human Action Video dataset." [Online]. Available: dipersec.king.ac.uk/MuHAVi-MAS/

[202] S. Singh, S. A. Velastin, and H. Ragheb, "MuHAVi: A Multicamera Human Action Video dataset for the evaluation of action recognition methods," in Proceedings of the *Workshop on activity Monitoring by Multi-camera Surveillance Systems*, 2010, pp. 48—-55.

[203] J. C. Niebles, C.-W. Chen, and L. Fei-Fei, "Olympic Sports dataset." [Online]. Available: http://vision.stanford.edu/Datasets/OlympicSports/

[204] J. C. Niebles, C.-W. Chen, and L. Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," in Proceedings of the *European Conference on Computer Vision*, 2010, pp. 392–405.

[205] C. Wallraven, M. Schultze, B. Mohler, A. Vatakis, and K. Pastra, "POETICON Corpus." [Online]. Available: http://poeticoncorpus.kyb.mpg.de

[206] C. Wallraven and M. Schultze, "The POETICON enacted scenario corpus: A tool for human and computational experiments on action understanding," in Proceedings of the *IEEE Conference on Automatic Face and Gesture Recognition Workshops*, 2011.

[207] R. Messing, C. Pal, and H. Kautz, "University of Rochester Activities of Daily Living dataset." [Online]. Available: http://www.cs.rochester.edu/~rmessing/uradl/

[208] R. Messing, C. Pal, and H. Kautz, "Activity recognition using the velocity histories of tracked keypoints," in Proceedings of the *International Conference on Computer Vision*, 2009, pp. 104–111.

[209] K. Yun, J. Honorio, D. Chattopadhyay, T. L. Berg, and D. Samaras, "Two-person interaction detection using body-pose features and multiple instance learning." [Online]. Available: http://www3.cs.stonybrook.edu/~kyun/research/kinect_interaction/

[210] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. J. Guibas, and L. Fei-Fei, "Stanford 40 Actions dataset." [Online]. Available: http://vision.stanford.edu/Datasets/40actions.html

[211] M. Tenorth, J. Bandouch, and M. Beetz, "TUM Kitchen dataset." [Online]. Available: https://ias.cs.tum.edu/software/kitchen-activity-data

[212] M. Tenorth, J. Bandouch, and M. Beetz, "The TUM Kitchen Data Set of everyday manipulation activities for motion tracking and action recognition," in Proceedings of the *International Conference on Computer Vision Workshops*, 2009, pp. 1089–1096.

[213] K. Soomro, A. R. Zamir, and M. Shah, "UCF101 action recognition dataset." [Online]. Available: http://crcv.ucf.edu/data/UCF101.php

[214] J. Liu, J. Luo, and M. Shah, "UCF YouTube action dataset." [Online]. Available: http://crcv.ucf.edu/data/UCF_YouTube_Action.php

[215] J. Liu, "Recognizing realistic actions from videos in the wild," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1996–2003.

[216] K. K. Reddy and M. Shah, "UCF50 action recognition dataset." [Online]. Available: http://crcv.ucf.edu/data/UCF50.php

[217] K. K. Reddy and M. Shah, "Recognizing 50 human action categories of web videos," *Machine Vision and Applications*, vol. 24, no. 5, pp. 971–981, 2012.

[218] M. D. Rodriguez, J. Ahmed, and M. Shah, "UCF Sports action dataset." [Online]. Available: http://crcv.ucf.edu/data/UCF_Sports_Action.php

[219] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action MACH a spatio-temporal maximum average correlation height filter for action recognition," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[220] N. van der Aa, X. Luo, G. Giezeman, R. Tan, and R. Veltkamp, "Utrecht Multi-Person Motion Benchmark." [Online]. Available: http://www.projects.science.uu.nl/umpm/

[221] N. van der Aa, X. Luo, G. Giezeman, R. Tan, and R. Veltkamp, "Utrecht multi-person motion benchmark: A multi-person dataset with synchronized video and motion capture data for evaluation of articulated human motion and interaction," in Proceedings of the *Workshop on Human Interaction in Computer Vision*, 2011, pp. 1264–1269.

[222] M. S. Ryoo and J. K. Aggarwal, "UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities," 2010. [Online]. Available: http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html

[223] M. S. Ryoo and J. K. Aggarwal, "Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities," in Proceedings of the *International Conference on Computer Vision*, 2009, pp. 1593–1600.

[224] H. Ragheb, S. Velastin, P. Remagnino, and T. Ellis, "ViHASi: Virtual Human Action Silhouette data for the evaluation of silhouette-based action recognition methods." [Online]. Available: http://dipersec.king.ac.uk/VIHASI/

[225] H. Ragheb and S. Velastin, "ViHASi: Virtual Human Action Silhouette data for the performance evaluation of silhouette-based action recognition methods," in Proceedings of the *International Conference on Distributed Smart Cameras*, 2008, pp. 1–10.

[226] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. Aggarwal, H. Lee, L. Davis, E. Swears, X. Wang, Q. Ji, K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, B. Song, A. Fong, A. Roy-Chowdhury, and M. Desai, "VIRAT video dataset." [Online]. Available: http://www.viratdata.org/

[227] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. K. Aggarwal, H. Lee, L. Davis, E. Swears, X. Wang, Q. Ji, K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, B. Song, A. Fong, A. Roy-Chowdhury, and M. Desai, “A large-scale benchmark dataset for event recognition in surveillance video,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, no. 2, 2011, pp. 3153–3160.

[228] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, “Actions as space-time shapes.” [Online]. Available: http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html

[229] V. Kulathumani, “WVU Multi-View action recognition dataset.” [Online]. Available: http://csee.wvu.edu/~vkkulathumani/wvu-action.html

[230] S. Ramagiri, R. Kavi, and V. Kulathumani, “Real-time multi-view human action recognition using a wireless camera network,” in Proceedings of the *International Conference on Distributed Smart Cameras*, 2011, pp. 1–6.

[231] R. Kavi and V. Kulathumani, “Real-time recognition of action sequences using a distributed video sensor network,” *Journal of Sensor and Actuator Networks*, vol. 2, no. 3, pp. 486–508, 2013.

[232] J. A. Stork, L. Spinello, J. Silva, and K. O. Arras, “Audio-based human activity recognition using non-Markovian ensemble voting,” in Proceedings of the *IEEE International Symposium on Robot and Human Interactive Communication*, 2012, pp. 509–514.

[233] M. Ermes, J. Parkka, J. Mantyjarvi, and I. Korhonen, “Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, no. 1, pp. 20–26, 2008.

[234] C. McCall, K. K. Reddy, and M. Shah, “Macro-class selection for hierarchical K-NN classification of inertial sensor data,” in Proceedings of the *International Conference on Pervasive and Embedded Computing and Communication Systems*, 2012.

[235] B. Krawczyk, “Learning from imbalanced data: open challenges and future directions,” *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, 2016.

[236] M. Buda, A. Maki, and M. A. Mazurowski, “A systematic study of the class imbalance problem in convolutional neural networks,” *CoRR*, vol. abs/1710.05381, 2017. [Online]. Available: http://arxiv.org/abs/1710.05381

[237] L. Taylor and G. Nitschke, “Improving deep learning using generic data augmentation,” *CoRR*, vol. abs/1708.06020, 2017. [Online]. Available: http://arxiv.org/abs/1708.06020

[238] L. Perez and J. Wang, “The effectiveness of data augmentation in image classification using deep learning,” *CoRR*, vol. abs/1712.04621, 2017. [Online]. Available: http://arxiv.org/abs/1712.04621

[239] J. Ferryman, “PETS,” in Proceedings of the *International Workshop on Performance Evaluation of Tracking and Surveillance*, 2009.

[240] E. Andrade, S. Blunsden, and R. Fisher, “Modelling crowd scenes for event detection,” in Proceedings of the *International Conference on Pattern Recognition*, 2006, pp. 175–178.

[241] E. Andrade, R. Fisher, and S. Blunsden, “Detection of emergency events in crowded scenes,” in Proceedings of the *IEEE International Symposium on Imaging for Crime Detection and Prevention*, 2006, pp. 528–533.

[242] P. Dai, H. Di, L. Dong, L. Tao, and G. Xu, “Group interaction analysis in dynamic context,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 1, pp. 34–42, 2009.

[243] D. Münch, S. Becker, W. Hübner, and M. Arens, “Towards a real-time situational awareness system for surveillance applications in unconstrained environments,” *Future Security*, pp. 517–521, 2012.

[244] A.-L. Ellis and J. Ferryman, “Benchmark datasets for detection and tracking,” in *Visual Analysis of Humans*, 2011, pp. 109–128.

[245] M. Elhamod and M. D. Levine, “Real-time semantics-based detection of suspicious activities in public spaces,” in Proceedings of the *Conference on Computer and Robot Vision*, 2012, pp. 268–275.

[246] M. Elhamod and M. Levine, “Automated real-time detection of potentially suspicious behavior in public transport areas,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 688–699, 2013.

[247] W. M.-C. D. Moltisanti, M. Wray and D. Damen, “Trespassing the boundaries: Labelling temporal bounds for object interactions in egocentric video,” in Proceedings of the *International Conference on Computer Vision*, 2017, pp. 2905–2913.

[248] C. Gu, C. Sun, S. Vijayanarasimhan, C. Pantofaru, D. A. Ross, G. Toderici, Y. Li, S. Ricco, R. Sukthankar, C. Schmid, and J. Malik, “AVA: A video dataset of spatio-temporally localized atomic visual actions,” *CoRR*, vol. abs/1705.08421, 2017. [Online]. Available: http://arxiv.org/abs/1705.08421

[249] N. Hu, G. Englebienne, Z. Lou, and B. Krse, “Learning to recognize human activities using soft labels,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 10, pp. 1973–1984, 2017.

[250] W. Dai, K. Yoshigoe, and W. Parsley, “Improving data quality through deep learning and statistical models,” in Proceedings of the *Information Technology - New Generations*, 2018, pp. 515–522.

[251] J. S. Sánchez, R. Barandela, A. I. Marqués, R. Alejo, and J. Badenas, “Analysis of new techniques to obtain quality training sets,” *Pattern Recognition Letters*, vol. 24, no. 7, pp. 1015–1022, 2003.

[252] V. S. Sheng, F. Provost, and P. G. Ipeirotis, “Get another label? improving data quality and data mining using multiple, noisy labelers,” in Proceedings of the *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 614–622.

[253] B. Nicholson, V. S. Sheng, and J. Zhang, “Label noise correction and application in crowdsourcing,” *Expert Systems with Applications*, vol. 66, pp. 149–162, 2016.

[254] O. Alonso, “Challenges with label quality for supervised learning,” *Journal of Data and Information Quality*, vol. 6, no. 1, pp. 2:1–2:3, 2015.

[255] J. E. Olson, *Data Quality: The Accuracy Dimension*. Elsevier, 2003.

[256] D. George, X. Xie, Y.-K. Lai, and G. K. Tam, “A deep learning driven active framework for segmentation of large 3d shape collections,” *CoRR*, vol. abs/1807.06551, 2018. [Online]. Available: http://arxiv.org/abs/1807.06551

[257] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce, “Automatic annotation of human actions in video,” in Proceedings of the *International Conference on Computer Vision*, 2009, pp. 1491–1498.

[258] D. H. Nga and K. Yanai, “Automatic construction of an action video shot database using web videos,” in Proceedings of the *International Conference on Computer Vision*, 2011, pp. 527–534.

[259] O. Sener, A. R. Zamir, S. Savarese, and A. Saxena, “Unsupervised semantic parsing of video collections,” in Proceedings of the *International Conference on Computer Vision*, 2015, pp. 4480–4488.

[260] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles, “Dense-captioning events in videos,” in Proceedings of the *International Conference on Computer Vision*, 2017, pp. 706–715.

[261] O. Kliper-Gross, T. Hassner, and L. Wolf, “The action similarity labeling challenge,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 34, no. 3, pp. 615–621, 2012.

[262] S. Ondobaka, R. D. Newman-Norlund, F. P. de Lange, and H. Bekkering, “Action recognition depends on observer?s level of action control and social personality traits,” *PLoS One*, vol. 8, p. e81392, 11 2013.

[263] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon, “Efficient regression of general-activity human poses from depth images,” in Proceedings of the *International Conference on Computer Vision*, 2011, pp. 415–422.

[264] W. Shen, K. Deng, X. Bai, T. Leyvand, B. Guo, and Z. Tu, “Exemplar-based human action pose correction and tagging,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1784–1791.

[265] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon, “The vitruvian manifold : Inferring dense correspondences for one-shot human pose estimation,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 103–110.

[266] M. B. Holte, C. Tran, M. M. Trivedi, and T. B. Moeslund, "Human pose estimation and activity recognition from multi-view videos: Comparative explorations of recent developments," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 5, pp. 538 –552, 2012.

[267] E. A. Mosabbeb, K. Raahemifar, and M. Fathy, "Multi-view human activity recognition in distributed camera sensor networks." *Sensors*, vol. 13, no. 7, pp. 8750–8770, 2013.

[268] A. Iosifidis, A. Tefas, N. Nikolaidis, and I. Pitas, "Human action recognition in stereoscopic videos based on bag of features and disparity pyramids," in Proceedings of the *European Signal Processing Conference*, 2014, pp. 1317–1321.

[269] T. Shiratori, H. S. Park, L. Sigal, Y. Sheikh, and J. K. Hodgins, "Motion capture from body-mounted cameras," *ACM Transactions on Graphics*, vol. 30, no. 4, pp. 31:1–31:10, 2011.

[270] C. Tan, H. Goh, and V. Chandrasekhar, "Understanding the Nature of First-Person Videos: Characterization and Classification using Low-Level Features," *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 535–542, 2014.

[271] A. Betancourt, P. Morerio, C. S. Regazzoni, and M. Rauterberg, "An overview of first person vision and egocentric video analysis for personal mobile wearable devices," *Circuits and Systems for Video Technology*, pp. 744–760, 2014.

[272] S. Narayan and M. S. Kankanhalli, "Action and Interaction Recognition in First-person videos," *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 526 – 532, 2014.

[273] A. Fathi, "Social interactions: A first-person perspective," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1226–1233.

[274] S. Ryoo, M, J. Fuchs, T, L. Xia, K. Aggarwal, J, and L. Matthies, "Robot-centric activity prediction from first-person videos: What will they do to me?" in Proceedings of the *ACM/IEEE International Conference on Human-Robot Interaction*, 2015, pp. 295–302.

[275] D. Weinland, E. Boyer, and R. Ronfard, "Action recognition from arbitrary views using 3D exemplars," in Proceedings of the *International Conference on Computer Vision*, 2007, pp. 1–7.

[276] D. Weinland, M. Özuysal, and P. Fua, "Making action recognition robust to occlusions and viewpoint changes," in Proceedings of the *European Conference on Computer Vision*, 2010.

[277] S.-R. Ke, J.-N. Hwang, K.-M. Lan, and S.-Z. Wang, "View-invariant 3d human body pose reconstruction using a monocular video camera," in Proceedings of the *ACM/IEEE International Conference on Distributed Smart Cameras*, 2011, pp. 1–6.

[278] B. Li, O. I. Camps, and M. Sznaier, "Cross-view activity recognition using Hankelets," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1362–1369.

[279] J. Zheng and Z. Jiang, "Learning view-invariant sparse representations for cross-view action recognition," in Proceedings of the *International Conference on Computer Vision*, 2013, pp. 3176–3183.

[280] S. Herath, M. Harandi, and F. Porikli, "Going deeper into action recognition: A survey," *Image and Vision Computing*, vol. 60, pp. 4–21, 2017.

[281] M. Koohzadi and N. M. Charkari, "Survey on deep learning methods in human action recognition," *IET Computer Vision*, vol. 11, no. 8, pp. 623–632, 2017.

[282] Z. Zhang, X. Ma, R. Song, X. Rong, X. Tian, G. Tian, and Y. Li, "Deep learning based human action recognition: A survey," in Proceedings of the *Chinese Automation Congress*, 2017, pp. 3780–3785.

[283] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in Proceedings of the *International Conference on Computer Vision*, 2015, pp. 4489–4497.

[284] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi, "Human action recognition using factorized spatio-temporal convolutional networks," in Proceedings of the *International Conference on Computer Vision*, 2015, pp. 4597–4605.

[285] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[286] L. Wang, Y. Qiao, and X. Tang, “Action recognition with trajectory-pooled deep-convolutional descriptors,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4305–4314.

[287] S. Sharma, R. Kiros, and R. Salakhutdinov, “Action recognition using visual attention,” *CoRR*, vol. abs/1511.04119, 2015. [Online]. Available: http://arxiv.org/abs/1511.04119

[288] Q. Li, Z. Qiu, T. Yao, T. Mei, Y. Rui, and J. Luo, “Action recognition by learning deep multi-granular spatio-temporal video representation,” in Proceedings of the *International Conference on Multimedia Retrieval*, 2016, pp. 159–166.

[289] N. Srivastava, E. Mansimov, and R. Salakhutdinov, “Unsupervised learning of video representations using LSTMs,” in Proceedings of the *International Conference on Machine Learning*, 2015, pp. 843–852.

[290] S. Li, W. Zhang, and A. B. Chan, “Maximum-margin structured learning with deep networks for 3D human pose estimation,” in Proceedings of the *International Conference on Computer Vision*, 2015, pp. 2848–2856.

[291] W. Yang, W. Ouyang, H. Li, and X. Wang, “End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3073–3082.

[292] B. Ai, Y. Zhou, Y. Yu, and S. Du, “Human pose estimation using deep structure guided learning,” in Proceedings of the *IEEE Winter Conference on Applications of Computer Vision*, 2017, pp. 1224–1231.

[293] Y. Chen, C. Shen, X. S. Wei, L. Liu, and J. Yang, “Adversarial PoseNet: A structure-aware convolutional network for human pose estimation,” in Proceedings of the *International Conference on Computer Vision*, 2017, pp. 1221–1230.

[294] G. Chéron, I. Laptev, and C. Schmid, “P-CNN: Pose-based CNN Features for Action Recognition,” in Proceedings of the *International Conference on Computer Vision*, 2015.

[295] H. Rahmani, A. S. Mian, and M. Shah, “Learning a deep model for human action recognition from novel viewpoints,” *CoRR*, vol. abs/1602.00828, 2016. [Online]. Available: http://arxiv.org/abs/1602.00828

[296] J. Liu, N. Akhtar, and A. Mian, "Skepxels: Spatio-temporal image representation of human skeleton joints for action recognition," *CoRR*, vol. abs/1711.05941, 2017. [Online]. Available: http://arxiv.org/abs/1711.05941

[297] Z. Huang, C. Wan, T. Probst, and L. V. Gool, "Deep learning on Lie groups for skeleton-based action recognition," *CoRR*, vol. abs/1612.05877, 2016. [Online]. Available: http://arxiv.org/abs/1612.05877

[298] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1110–1118.

[299] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent network models for human dynamics," in Proceedings of the *International Conference on Computer Vision*, 2015, pp. 4346–4354.

[300] J. Liu, G. Wang, L. Y. Duan, K. Abdiyeva, and A. C. Kot, "Skeleton-based human action recognition with global context-aware attention LSTM networks," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1586–1599, 2018.

[301] Y. Hu, L. Cao, F. Lv, S. Yan, Y. Gong, and T. Huang, "Action detection in complex scenes with spatial and temporal ambiguities," in Proceedings of the *International Conference on Computer Vision*, 2009, pp. 128–135.

[302] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional SIFT descriptor and its application to action recognition," in Proceedings of the *ACM International Conference on Multimedia*, 2007, pp. 357–360.

[303] C. Bishop, *Pattern recognition and machine learning (Information science and statistics).* Springer, 2006.

[304] K. P. Murphy, *Machine learning: A probabilistic perspective.* MIT press, 2012.

[305] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukvuoglu, "WaveNet: A generative model for raw audio," *CoRR*, vol. abs/1609.03499, 2016. [Online]. Available: http://arxiv.org/abs/1609.03499

[306] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. K. andKeith Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016. [Online]. Available: http://arxiv.org/abs/1609.08144

[307] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 4, pp. 664–676, 2017.

[308] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[309] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[310] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[311] F. Rosenblatt, "The perceptron: A perceiving and recognizing automaton," Cornell Aeronautical Laboratory, Tech. Rep. 85-460-1, 1957.

[312] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002.

[313] K. O. Stanley, D. B. D'Ambrosio, and J. Gauci, "A hypercube-based encoding for evolving large-scale neural networks," *Artificial Life*, vol. 15, no. 2, pp. 185–212, 2009.

[314] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient transfer learning," *CoRR*, vol. abs/1611.06440, 2016. [Online]. Available: http://arxiv.org/abs/1611.06440

[315] M. Babaeizadeh, P. Smaragdis, and R. H. Campbell, "NoiseOut: A simple way to prune neural networks," *CoRR*, vol. abs/1611.06211, 2016. [Online]. Available: http://arxiv.org/abs/1611.06211

[316] P. J. Werbos, *Applications of advances in nonlinear sensitivity analysis*. Springer, 1982, pp. 762–770.

[317] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[318] K. Janocha and W. M. Czarnecki, "On loss functions for deep neural networks in classification," *CoRR*, vol. abs/1702.05659, 2017. [Online]. Available: http://arxiv.org/abs/1702.05659

[319] A. G. Baydin, B. A. Pearlmutter, and A. A. Radul, "Automatic differentiation in machine learning: A survey," *CoRR*, vol. abs/1502.05767, 2015. [Online]. Available: http://arxiv.org/abs/1502.05767

[320] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. J. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Józefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. G. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. A. Tucker, V. Vanhoucke, V. Vasudevan, F. B. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," *CoRR*, vol. abs/1603.04467, 2016. [Online]. Available: http://arxiv.org/abs/1603.04467

[321] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in Proceedings of the *International Conference on Machine Learning*, 2013, pp. 1310–1318.

[322] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[323] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[324] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[325] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in Proceedings of the *International Conference on Artificial Intelligence and Statistics*, vol. 9, 2010, pp. 249–256.

[326] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” *CoRR*, vol. abs/1502.01852, 2015. [Online]. Available: http://arxiv.org/abs/1502.01852

[327] Y. Bengio, “Practical recommendations for gradient-based training of deep architectures,” *CoRR*, vol. abs/1206.5533, 2012. [Online]. Available: http://arxiv.org/abs/1206.5533

[328] M. Li, T. Zhang, Y. Chen, and A. J. Smola, “Efficient mini-batch training for stochastic optimization,” in Proceedings of the *International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 661–670.

[329] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[330] M. D. Zeiler, “ADADELTA: An adaptive learning rate method,” *CoRR*, vol. abs/1212.5701, 2012. [Online]. Available: https://arxiv.org/abs/1212.5701

[331] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: https://arxiv.org/abs/1412.6980

[332] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in Proceedings of the *International Conference on Machine Learning*, ser. ICML’10. USA: Omnipress, 2010, pp. 807–814.

[333] B. Xu, N. Wang, T. Chen, and M. Li, “Empirical evaluation of rectified activations in convolutional network,” *CoRR*, vol. abs/1505.00853, 2015. [Online]. Available: http://arxiv.org/abs/1505.00853

[334] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, “What is the best multi-stage architecture for object recognition?” in Proceedings of the *International Conference on Computer Vision*, 2009, pp. 2146–2153.

[335] C. R. R. Molina and O. P. Vila, “Solving internal covariate shift in deep learning with linked neurons,” *CoRR*, vol. abs/1712.02609, 2017. [Online]. Available: https://arxiv.org/abs/1609.08144

[336] Y.-L. Boureau, J. Ponce, and Y. LeCun, “A theoretical analysis of feature pooling in visual recognition,” in Proceedings of the *International Conference on Machine Learning*, 2010, pp. 111–118.

[337] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in Proceedings of the *European Conference on Computer Vision*, 2014, pp. 818–833.

[338] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[339] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep face recognition,” in Proceedings of the *British Machine Vision Conference*, 2015, pp. 41.1–41.12.

[340] A. Graves, A.-r. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in Proceedings of the *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 38, 2013.

[341] W. Xiong, J. Droppo, X. Huang, F. Seide, M. L. Seltzer, A. Stolcke, D. Yu, and G. Zweig, “Toward human parity in conversational speech recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 12, pp. 2410–2423, 2017.

[342] Q. V. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. S. Corrado, J. Dean, and A. Y. Ng, “Building high-level features using large scale unsupervised learning,” in Proceedings of the *International Conference on Machine Learning*, 2012, pp. 507–514.

[343] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, “Stacked convolutional auto-encoders for hierarchical feature extraction,” in Proceedings of the *International Conference on Artificial Neural Networks and Machine Learning*, 2011, pp. 52–59.

[344] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Proceedings of the *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[345] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *CoRR*, vol. abs/1511.06434, 2015. [Online]. Available: http://arxiv.org/abs/1511.06434

[346] J. J. Zhao, M. Mathieu, and Y. LeCun, “Energy-based generative adversarial network,” *CoRR*, vol. abs/1609.03126, 2016. [Online]. Available: http://arxiv.org/abs/1609.03126

[347] Z. Qin, F. Yu, C. Liu, and X. Chen, “How convolutional neural networks see the world — a survey of convolutional neural network visualization methods,” *Mathematical Foundations of Computing*, vol. 1, pp. 149–180, 2018.

[348] D. Wei, B. Zhou, A. Torralba, and W. T. Freeman, “Understanding intra-class knowledge inside CNN,” *CoRR*, vol. abs/1507.02379, 2015. [Online]. Available: http://arxiv.org/abs/1507.02379

[349] A. Mahendran and A. Vedaldi, “Understanding deep image representations by inverting them,” *CoRR*, vol. abs/1412.0035, 2014. [Online]. Available: http://arxiv.org/abs/1412.0035

[350] L. Nanni, S. Ghidoni, and S. Brahnam, “Handcrafted vs. non-handcrafted features for computer vision classification,” *Pattern Recognition*, vol. 71, pp. 158–172, 2017.

[351] M. N. Kashif, S. E. A. Raza, K. Sirinukunwattana, M. Arif, and N. Rajpoot, “Handcrafted features with convolutional neural networks for detection of tumor cells in histology images,” in Proceedings of the *IEEE International Symposium on Biomedical Imaging*, 2016, pp. 1029–1032.

[352] T. Majtner, S. Yildirim-Yayilgan, and J. Y. Hardeberg, “Combining deep learning and hand-crafted features for skin lesion classification,” in Proceedings of the *International Conference on Image Processing Theory, Tools and Applications*, 2016, pp. 1–6.

[353] V. M. Khong and T. H. Tran, “Improving human action recognition with two-stream 3d convolutional neural network,” in Proceedings of the *International Conference on Multimedia Analysis and Pattern Recognition*, 2018, pp. 1–6.

[354] E. E. Hansley, M. P. Segundo, and S. Sarkar, “Employing fusion of learned and handcrafted features for unconstrained ear recognition,” *IET Biometrics*, vol. 7, no. 3, pp. 215–223, 2018.

[355] K. Guo, D. Zou, and X. Chen, “3d mesh labeling via deep convolutional neural networks,” *ACM Transactions on Graphics*, vol. 35, no. 1, pp. 3:1–3:12, 2015.

[356] A. Johnson, “Spin-Images: A representation for 3-D surface matching,” Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, 1997.

[357] A. Johnson and M. Hebert, “Surface matching for object recognition in complex three-dimensional scenes,” *Image and Vision Computing*, vol. 16, no. 9, pp. 635–651, 1998.

[358] D. George, X. Xie, and G. K. Tam, “3d mesh segmentation via multi-branch 1d convolutional neural networks,” *Graphical Models*, vol. 96, pp. 1–10, 2018.

[359] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst, “ShapeNet: Convolutional neural networks on non-euclidean manifolds,” *CoRR*, vol. abs/1501.06297, 2015. [Online]. Available: http://arxiv.org/abs/1501.06297

[360] D. Shuman, S. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[361] J. H. Metzen, “Learning graph-based representations for continuous reinforcement learning domains,” in Proceedings of the *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2013, pp. 81–96.

[362] J. D. Wendt, R. Wells, R. V. Field, and S. Soundarajan, “On data collection, graph construction, and sampling in twitter,” in Proceedings of the *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2016, pp. 985–992.

[363] R. Merris, “Laplacian graph eigenvectors,” *Linear Algebra and its Applications*, vol. 278, no. 1, pp. 221 – 236, 1998.

[364] D. K. Hammond, P. Vandergheynst, and R. Gribonval, “Wavelets on graphs via spectral graph theory,” *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129 – 150, 2011.

[365] R. Bracewell, *The Fourier Transform & Its Applications*. McGraw, 1999.

[366] A. Buades, B. Coll, and J. M. Morel, “A review of image denoising algorithms, with a new one,” *Multiscale Model Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[367] A. Gadde, S. K. Narang, and A. Ortega, “Bilateral filter: Graph spectral interpretation and extensions,” in Proceedings of the *IEEE International Conference on Image Processing*, 2013, pp. 1222–1226.

[368] A. Sandryhaila and J. M. F. Moura, “Discrete signal processing on graphs: Graph filters,” in Proceedings of the *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6163–6166.

[369] Y.-X. Wang, J. Sharpnack, A. J. Smola, and R. J. Tibshirani, “Trend filtering on graphs,” *Journal of Machine Learning Research*, vol. 17, no. 105, pp. 1–41, 2016.

[370] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition).* Prentice-Hall, Inc., 2006.

[371] I. Safro, “Comparison of coarsening schemes for multilevel graph partitioning,” in Proceedings of the *International Conference on Learning and Intelligent Optimization*, 2009, pp. 191–205.

[372] D. Ron, I. Safro, and A. Brandt, “Relaxation-based coarsening and multiscale graph organization,” *Multiscale Modeling & Simulation*, vol. 9, pp. 407–423, 2011.

[373] B. O. F. Auer and R. H. Bisseling, “Graph coarsening and clustering on the GPU,” in Proceedings of the *Graph Partitioning and Graph Clustering*, vol. 588, 2012, pp. 223–240.

[374] I. Safro, P. Sanders, and C. Schulz, *Proceedings of the Interntional Symposium on Experimental Algorithms.* Springer, 2012, ch. Advanced Coarsening Schemes for Graph Partitioning, pp. 369–380.

[375] C. Zhang, D. Florencio, and P. Chou, “Graph signal processing - a probabilistic framework,” Microsoft Research, Tech. Rep. MSR-TR-2015-31, 2015.

[376] G. Karypis and V. Kumar, “Analysis of multilevel graph partitioning,” in Proceedings of the *ACM/IEEE Conference on Supercomputing*, 1995.

[377] P. Sanders and C. Schulz, *European Symposium on Algorithms.* Springer, 2011, ch. Engineering Multilevel Graph Partitioning Algorithms, pp. 469–480.

[378] G. Karypis and V. Kumar, “A fast and high quality multilevel scheme for partitioning irregular graphs,” *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 359–392, 1998.

[379] D. Kushnir, M. Galun, and A. Brandt, “Fast multiscale clustering and manifold identification,” *Pattern Recognition*, vol. 39, no. 10, pp. 1876 – 1891, 2006, similarity-based Pattern Recognition.

[380] O. E. Livne and A. Brandt, “Lean algebraic multigrid (LAMG): fast graph laplacian linear solver,” *SIAM Journal of Scientific Computing*, vol. 34, no. 4, 2012.

[381] F. Dorfler and F. Bullo, “Kron reduction of graphs with applications to electrical networks,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 1, pp. 150–163, 2013.

[382] P. Liu, X. Wang, and Y. Gu, “Graph signal coarsening: Dimensionality reduction in irregular domain,” in Proceedings of the *IEEE Global Conference on Signal and Information Processing*, 2014, pp. 798–802.

[383] L. Wang, Y. Xiao, B. Shao, and H. Wang, “How to partition a billion-node graph,” in Proceedings of the *IEEE International Conference on Data Engineering*, 2014, pp. 568–579.

[384] D. I. Shuman, M. J. Faraji, and P. Vandergheynst, “A multiscale pyramid transform for graph signals,” *IEEE Transactions on Signal Processing*, vol. 64, no. 8, pp. 2119–2134, 2016.

[385] A. Loukas and P. Vandergheynst, “Spectrally approximating large graphs with smaller graphs,” *CoRR*, vol. abs/1802.07510, 2018. [Online]. Available: http://arxiv.org/abs/1802.07510

[386] J. K. Aggarwal and M. S. Ryoo, “Human activity analysis: A review,” *ACM Computing Survey*, vol. 43, no. 3, pp. 16:1–16:43, 2011.

[387] S. Park and J. K. Aggarwal, “A hierarchical Bayesian network for event recognition of human actions and interactions,” *Multimedia Systems*, vol. 10, no. 2, pp. 164–179, 2004.

[388] S. Cheema, A. Eweiwi, C. Thurau, C. Bauckhage, F. Iais, and S. Augustin, "Action recognition by learning discriminative key poses," in Proceedings of the *Computer Vision Workshops*, 2011, pp. 1302–1309.

[389] A. Veeraraghavan, A. Srivastava, A. K. Roy-Chowdhury, and R. Chellappa, "Rate-invariant recognition of humans and their activities." *IEEE Transactions on Image Processing*, vol. 18, no. 6, pp. 1326–39, 2009.

[390] F. Lv and R. Nevatia, "Single view human action recognition using key pose matching and Viterbi path searching," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[391] S. Baysal, M. C. Kurt, and P. Duygulu, "Recognizing human actions using key poses," in Proceedings of the *International Conference on Pattern Recognition*, 2010, pp. 1727–1730.

[392] A. Eweiwi, S. Cheema, C. Thurau, and C. Bauckhage, "Temporal key poses for human action recognition," in Proceedings of the *International Conference on Computer Vision Workshops*, 2011, pp. 1310–1317.

[393] D. Gong, G. Medioni, and X. Zhao, "Structured time series analysis for human action segmentation and recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 36, no. 7, pp. 1414–1427, 2014.

[394] D. Gong and G. Medioni, "Dynamic manifold warping for view invariant action recognition," in Proceedings of the *International Conference on Computer Vision*, no. 3, 2011, pp. 571–578.

[395] F. Zhou and F. De la Torre, "Canonical time warping for alignment of human behavior," in Proceedings of the *Advances in Neural Information Processing Systems*, 2009.

[396] W. Brendel and S. Todorovic, "Activities as time series of human postures," in Proceedings of the *European Conference on Computer Vision*, no. 1, 2010, pp. 1–14.

[397] M. Raptis, D. Kirovski, and H. Hoppe, "Real-time classification of dance gestures from skeleton animation," in Proceedings of the *SIGGRAPH/Eurographics Symposium on Computer Animation*, vol. 1, 2011, p. 147.

[398] H. Shimodaira, K. ichi Noma, M. Nakai, and S. Sagayama, “Dynamic time-alignment kernel in support vector machine,” in Proceedings of the *Advances in Neural Information Processing Systems*, 2002, pp. 921–928.

[399] D. J. Berndt and J. Clifford, “Using dynamic time warping to find patterns in time series,” in Proceedings of the *International Conference on Knowledge Discovery and Data Mining*, 1994, pp. 359–370.

[400] M. Mller, *Dynamic Time Warping*. Springer, 2007, pp. 69–84.

[401] T. Bäck, *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1996.

[402] J. McCall, “Genetic algorithms for modelling and optimisation,” *Journal of Computational and Applied Mathematics*, vol. 184, no. 1, pp. 205–222, 2005.

[403] C. y. Lee, “Variable length genomes for evolutionary algorithms,” in Proceedings of the *Proceedings of the Genetic and Evolutionary Computation Conference*, 2000, p. 806.

[404] K. Gallagher, M. Sambridge, and G. Drijkoningen, “Genetic algorithms: An evolution from monte carlo methods for strongly non-linear geophysical optimization problems,” *Geophysical Research Letters*, vol. 18, no. 12, pp. 2177–2180, 1991.

[405] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in Proceedings of the *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.

[406] S. Walton, O. Hassan, K. Morgan, and M. Brown, “Modified cuckoo search: A new gradient free optimisation algorithm,” *Chaos, Solitons and Fractals*, vol. 44, no. 9, pp. 710–718, 2011.

[407] A. Joshi, O. Kulkarni, G. Kakandikar, and V. Nandedkar, “Cuckoo search optimization-a review,” *Materials Today: Proceedings*, vol. 4, no. 8, pp. 7262–7269, 2017.

[408] M.-L. Zhang and Z.-H. Zhou, “Multilabel neural networks with applications to functional genomics and text categorization,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338–1351, 2006.

[409] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, “A convolutional neural network cascade for face detection,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5325–5334.

[410] D. C. Ciresan, U. Meier, and J. Schmidhuber, “Multi-column deep neural networks for image classification,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3642–3649.

[411] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, “Learning and transferring mid-level image representations using convolutional neural networks,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1717–1724.

[412] A. Vedaldi and K. Lenc, “MatConvNet – convolutional neural networks for MATLAB,” in Proceedings of the *ACM International Conference on Multimedia*, 2014.

[413] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” *CoRR*, vol. abs/1408.5093, 2014. [Online]. Available: https://arxiv.org/abs/1408.5093

[414] D. G. Lowe, “Object recognition from local scale-invariant features,” in Proceedings of the *International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157.

[415] M. Henaff, J. Bruna, and Y. LeCun, “Deep convolutional networks on graph-structured data,” *CoRR*, vol. abs/1506.05163, 2015. [Online]. Available: https://arxiv.org/abs/1506.05163

[416] L. Grady and J. R. Polimeni, *Discrete Calculus - Applied Analysis on Graphs for Computational Science.* Springer, 2010.

[417] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral networks and locally connected networks on graphs,” in Proceedings of the *International Conference on Learning Representations*, 2014.

[418] G. Kron, *Tensor analysis of networks*. MacDonald, 1939.

[419] D. A. Spielman and N. Srivastava, “Graph sparsification by effective resistances,” in Proceedings of the *ACM Symposium on Theory of Computing*, 2008, pp. 563–568.

[420] T. F. Chan, J. Xu, and L. Zikatanov, “An agglomeration multigrid method for unstructured grids,” *Contemporary Mathematics*, vol. 218, pp. 67–81, 1998.

[421] R. Yu, X. Chen, V. I. Morariu, and L. S. Davis, "The role of context selection in object detection," *CoRR*, vol. abs/1609.02948, 2016. [Online]. Available: http://arxiv.org/abs/1609.02948

[422] J. Oh, H.-I. Kim, and R.-H. Park, "Context-based abnormal object detection using the fully-connected conditional random fields," *Pattern Recognition Letters*, vol. 98, pp. 16–25, 2017.

[423] J. Koenderink, "The structure of images," *Biological Cybernetics*, vol. 50, no. 5, pp. 363–370, 1984.

[424] T. Lindeberg, "Scale-space: A framework for handling image structures at multiple scales," in Proceedings of the *CERN School of Computing*, no. 8, 1996, pp. 27–38.

[425] L. Tony, *Scale-Space*. John Wiley and Sons, 2008, pp. 2495–2504.

[426] L. Florack, B. ter Haar Romeny, M. Viergever, and J. Koenderink, "The gaussian scale-space paradigm and the multiscale local jet," *International Journal of Computer Vision*, vol. 18, no. 1, pp. 61–75, 1996.

[427] P. H. LINDSAY and D. A. NORMAN, *Human Information Processing*. Academic Press, 1977.

[428] C. A. Curcio, K. R. Sloan, R. E. Kalina, and A. E. Hendrickson, "Human photoreceptor topography," *Journal of Comparative Neurology*, vol. 292, no. 4, pp. 497–523, 1990.

[429] R. Rosenholtz, "Capabilities and limitations of peripheral vision," *Annual Review of Vision Science*, vol. 2, pp. 437–457, 2016.

[430] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[431] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. 511–518.

[432] R. Mottaghi, X. Chen, X. Liu, N. G. Cho, S. W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 891–898.

[433] N. Bergboer, E. Postma, and H. van den Herik, “Context-based object detection in still images,” *Image and Vision Computing*, vol. 24, no. 9, pp. 987–1000, 2006.

[434] W. T. Freeman and E. H. Adelson, “The design and use of steerable filters,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 13, no. 9, pp. 891–906, 1991.

[435] R. Lienhart and J. Maydt, “An extended set of Haar-like features for rapid object detection,” in Proceedings of the *IEEE International Conference on Image Processing*, vol. 1, 2002, pp. 900–903.

[436] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” in Proceedings of the *European Conference on Computer Vision*, 2014, pp. 346–361.

[437] J. Jiang, P. Trundle, and J. Ren, “Medical image analysis with artificial neural networks,” *Computerized Medical Imaging and Graphics*, vol. 34, no. 8, pp. 617–631, 2010.

[438] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, “Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning,” *IEEE Transaction Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

[439] H.-C. Shin, M. R. Orton, D. J. Collins, S. J. Doran, and M. O. Leach, “Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 35, no. 8, pp. 1930–1943, 2013.

[440] Y. Zheng, M. John, R. Liao, A. Nottling, J. Boese, J. Kempfert, T. Walther, G. Brockmann, and D. Comaniciu, “Automatic aorta segmentation and valve landmark detection in c-arm CT for transcatheter aortic valve implantation,” *IEEE Transaction Medical Imaging*, vol. 31, pp. 2307–2321, 2012.

[441] S. Grbić, R. Ionasec, D. Vitanovski, I. Voigt, B. Georgescu, N. Navab, and D. Comaniciu, “Complete valvular heart apparatus model from 4D cardiac CT,” *Medical Image Analysis*, vol. 16, pp. 1003–1014, 2012.

[442] R. Palmer, X. Xie, and G. Tam, “Automatic aortic root segmentation with shape constraints and mesh regularisation,” in Proceedings of the *British Machine Vision Conference*, 2015.

[443] T. F. Cootes, D. H. Cooper, and J. Graham, "Active shape models - Their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[444] A. F. Frangi, D. Rueckert, J. A. Schnabel, and W. J. Niessen, "Automatic construction of multiple-object three-dimensional statistical shape models: Application to cardiac modeling," *IEEE Transaction Medical Imaging*, vol. 21, no. 9, pp. 1151–1166, 2002.

[445] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *CoRR*, vol. abs/1411.4038, 2014. [Online]. Available: http://arxiv.org/abs/1411.4038

[446] H. Caesar, J. R. R. Uijlings, and V. Ferrari, "Region-based semantic segmentation with end-to-end training," *CoRR*, vol. abs/1607.07671, 2016. [Online]. Available: http://arxiv.org/abs/1607.07671

[447] Y. Xue, T. Xu, H. Zhang, L. R. Long, and X. Huang, "SegAN: Adversarial network with multi-scale L_1 loss for medical image segmentation," *CoRR*, vol. abs/1706.01805, 2017. [Online]. Available: http://arxiv.org/abs/1706.01805

[448] F. Milletari, N. Navab, and S. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," *CoRR*, vol. abs/1606.04797, 2016. [Online]. Available: http://arxiv.org/abs/1606.04797

[449] Z. V. Zhang, M. Tang, D. Cobzas, D. Zonoobi, M. Jägersand, and J. L. Jaremko, "End-to-end detection-segmentation network with ROI convolution," *CoRR*, vol. abs/1801.02722, 2018. [Online]. Available: http://arxiv.org/abs/1801.02722

[450] Y. Zheng and D. Comaniciu, *Marginal Space Learning for Medical Image Analysis*. Springer, 2014.

[451] N. Gessert, M. Schlüter, and A. Schlaefer, "A deep learning approach for pose estimation from volumetric OCT data," *CoRR*, vol. abs/1803.03852, 2018. [Online]. Available: http://arxiv.org/abs/1803.03852

[452] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[453] X. Li, F. Yang, H. Cheng, J. Chen, Y. Guo, and L. Chen, "Multi-scale cascade network for salient object detection," in Proceedings of the *ACM International Conference on Multimedia*, 2017, pp. 439–447.

[454] S. Yan, J. S. Smith, W. Lu, and B. Zhang, "Hierarchical multi-scale attention networks for action recognition," *CoRR*, vol. abs/1708.07590, 2017. [Online]. Available: http://arxiv.org/abs/1708.07590

[455] C. Gao, N. Sang, and R. Huang, "Biologically inspired scene context for object detection using a single instance," *PLoS One*, vol. 9, pp. 1–13, 2014.

[456] L. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," *CoRR*, vol. abs/1511.03339, 2015. [Online]. Available: http://arxiv.org/abs/1511.03339

[457] J. Zhang, Y. Dai, B. Li, and M. He, "Attention to the scale: Deep multi-scale salient object detection," in Proceedings of the *International Conference on Digital Image Computing: Techniques and Applications*, 2017, pp. 1–7.

[458] S. J. Anderson, K. T. Mullen, and R. F. Hess, "Human peripheral spatial resolution for achromatic and chromatic stimuli: limits imposed by optical and retinal factors," *Journal of Physiology*, vol. 442, pp. 47?–64, 1991.

[459] D. Rueckert, L. I. Sonoda, C. Hayes, D. Mill, O. Leach, and D. J. Hawkes, "Nonrigid registration using free-form deformations: Application to breast MR images," *IEEE Transaction Medical Imaging*, vol. 18, no. 8, pp. 712–721, 1999.

[460] A. Cheung and K. M. Lichtenstein, "Illustrated techniques for transapical aortic valve implantation," *Annals of Cardiothoracic Surgery*, vol. 1, no. 2, 2012.

[461] J. Yosinski, J. Clune, A. M. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," *CoRR*, vol. abs/1506.06579, 2015. [Online]. Available: http://arxiv.org/abs/1506.06579

[462] J. M. Wolterink, T. Leiner, M. A. Viergever, and I. Išgum, "Dilated convolutional neural networks for cardiovascular MR segmentation in congenital heart disease," in Proceedings of the *Joint International Workshop on Reconstruction and Analysis of Moving Body Organs, and Whole-Heart and Great Vessel Segmentation from 3D Cardiovascular MRI in Congenital Heart Disease*, 2017, pp. 95–102.

[463] K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep inside convolutional networks: Visualising image classification models and saliency maps,” *CoRR*, vol. abs/1312.6034, 2013. [Online]. Available: http://arxiv.org/abs/1312.6034

[464] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, “Deep learning applications and challenges in big data analytics,” *Journal of Big Data*, vol. 2, no. 1, pp. 1–21, 2015.

[465] Y. Li, L. Liu, C. Shen, and A. van den Hengel, “Mining mid-level visual patterns with deep CNN activations,” *CoRR*, vol. abs/1506.06343, 2015. [Online]. Available: http://arxiv.org/abs/1506.06343

[466] Y. Li, L. Liu, C. Shen, and A. van den Hengel, “Mid-level deep pattern mining,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 971–980.

[467] D. C. Luvizon, D. Picard, and H. Tabia, “2d/3d pose estimation and action recognition using multitask deep learning,” *CoRR*, vol. abs/1802.09232, 2018. [Online]. Available: http://arxiv.org/abs/1802.09232

[468] A. Grinciunaite, A. Gudi, H. E. Tasli, and M. den Uyl, “Human pose estimation in space and time using 3d CNN,” *CoRR*, vol. abs/1609.00036, 2016. [Online]. Available: http://arxiv.org/abs/1609.00036

[469] A. Toshev and C. Szegedy, “DeepPose: Human pose estimation via deep neural networks,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1653–1660.

[470] W. Ouyang, X. Chu, and X. Wang, “Multi-source deep learning for human pose estimation,” in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2337–2344.

[471] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei, “Towards 3d human pose estimation in the wild: A weakly-supervised approach,” in Proceedings of the *International Conference on Computer Vision*, 2017, pp. 398–407.

[472] M. S. Cheema, A. Eweiwi, and C. Bauckhage, “Human activity recognition by separating style and content,” *Pattern Recognition*, vol. 50, pp. 130 – 138, 2014.

[473] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy, and R. Vidal, "Bio-inspired dynamic 3D discriminative skeletal features for human action recognition," in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 471–478.

[474] A. Zunino, J. Cavazza, and V. Murino, "Revisiting human action recognition: Personalization vs. generalization," *CoRR*, vol. abs/1605.00392, 2016. [Online]. Available: http://arxiv.org/abs/1605.00392

[475] J. R. Padilla-López, A. A. Chaaraoui, and F. Flórez-Revuelta, "A discussion on the validation tests employed to compare human action recognition methods using the MSR action3d dataset," *CoRR*, vol. abs/1407.7390, 2014. [Online]. Available: http://arxiv.org/abs/1407.7390

[476] S. Vantigodi and R. V. Babu, "Real-time human action recognition from motion capture data," in Proceedings of the *National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*, 2013, pp. 1–4.

[477] S. Vantigodi and V. B. Radhakrishnan, "Action recognition from motion capture data using meta-cognitive RBF network classifier," in Proceedings of the *IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2014, pp. 1–6.

[478] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[479] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in Proceedings of the *International Conference on Machine Learning Workshop on Unsupervised and Transfer Learning*, vol. 27, 2012, pp. 37–49.

[480] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[481] K. Riesen and H. Bunke, *Graph Classification and Clustering Based on Vector Space Embedding*. World Scientific Publishing Co., Inc., 2010.

[482] S. E. Schaeffer, "Graph clustering," *Computer Science Review*, vol. 1, no. 1, pp. 27–64, 2007.