Combining Stacked Denoising Autoencoders and Random Forests for Face Detection

Jingjing Deng, Xianghua Xie*, and Michael Edwards

Department of Computer Science, Swansea University,
Singleton Park, Swansea SA2 8PP, United Kingdom
*x.xie@swansea.ac.uk
http://csvision.swan.ac.uk

Abstract. Detecting faces in the wild is a challenging problem due to large visual variations introduced by uncontrolled facial expressions, head pose, illumination and so on. Employing strong classifier and designing more discriminative visual features are two main approaches to overcoming such difficulties. Notably, Deep Neural Network (DNN) based methods have been found to outperform most traditional detectors in a multitude of studies, employing deep network structures and complex training procedures. In this work, we propose a novel method that uses stacked denoising autoencoders (SdA) for feature extraction and random forests (RF) for object-background classification in a classical cascading framework. This architecture allows much simpler neural network structures, resulting in efficient training and detection. The proposed face detector was evaluated on two publicly available datasets and produced promising results.

1 Introduction

Face detection has been an active research topic in computer vision for decades. View-specific face detection methods, such as VJ detector (Viola-Jones object detection framework [37]), have achieved great success and are used in various commercial products. Recently, researchers are focusing on more challenging face detection problems under uncontrolled environment, i.e. so-called faces in the wild, where factors such as large pose and facial expression variations, severe occlusion and clutter, and poor lighting scenario are taken into consideration. Zhang et al. [43], and Zafeiriou et al. [42] provide a comprehensive review on recently proposed face detection methods. However, faces in the wild remain a challenging problem.

Cascade based methods with strong classifiers and discriminative features are popular in tackling multi-view detection problems, utilizing methods such as aggregating channel features [39], multi-scale block local binary pattern [27], and normalized pixel difference feature [26]. The features are designed specifically for certain detection problems, which may not be universally applicable. Deformable Part Model (DPM) [11, 12, 16] defines an object as a pictorial configuration of its parts. For example, an human pose can be estimated using a star-structured model by finding a confident configuration of torso root and its displaced limb parts [6]. The benefit of having flexible representation of an object can greatly help to tackle the challenges of occlusion, pose and appearance variations, at the cost of considerably more complex computation requirements. There are some works on accelerating DPM-based detection, such as cascade part pruning [38]. Exemplar-based face detection methods [33, 24] formulate the problem as an image retrieval task, where a detection occurs when a successful matching is found in face exemplars for hypothesis by using visual feature descriptors, such as SIFT (Scale-Invariant Feature Transform). Matching with exemplars is time consuming, especially when a very large face dataset is required to cover huge amount of variations under uncontrolled environment, meaning that a sliding window method is not feasible. In these cases Hough voting based methods at multiple scales are used instead, to produce a confidence voting map to locate region candidates. The face can then be found by searching the peak regions on the ensemble confidence map. In order to achieve better performance, a large face exemplar dataset is critical in order to cover different variations in an uncontrolled environment. Meanwhile, detection speed will suffer due to exploring larger search space. Contour based object tracking method such as [5] can also be applied to face detection task.

The application of neural networks methods to detection problems goes back to 1980s [18, 40], but its place was quickly taken by support vector machine and boosting based methods due to the limitations on computational cost. With the developments of better unsupervised initialization methods, availability of large amount of labeled data, and hardware improvements, especially the GPGPU (General-Purpose Graphics Processing Unit), training a deep neural network becomes a routine [17, 23]. The success of Convolutionary Neural Networks (CNNs) on large scale object detection and recognition [22, 32, 34, 15] shows that searching deep representative structures with a learned convolutionary feature space is vitally important for high-level visual recognition tasks, whereas it is indeed computationally expensive for relatively simple tasks. For instance, traditional object detection problem where over hundreds of thousands of sub-windows are required to evaluate for just one image, is not affordable in real-time application. However, the power of representative feature learning of deep neural network should not be overlooked. Region-based CNNs (R-CNNs) [13, 14, 30] are a series of CNN-based methods for multi-object detection and semantic segmentation. In order to avoid densely scanning an image, R-CNN methods use selective search methods [35] to generate a relatively small amount of object region proposals. The features of the candidate regions are extracted by CNN, and a linear SVM is used to classify them into object categories. Its CNN consists of five convolutional layers and two fully connected layers, and it was pre-trained on a large image dataset, LSVRC2012 (Large Scale Visual Recognition Challenge [31]) discriminatively, and then followed by domain-specific fine tuning. Precisely locating an object is still a challenging problem to R-CNNs even with the help of bounding box refinement regressor. As R-CNN methods have been reported with relatively weak performance [10], Farfade et al. proposed DDFD

(Deep Dense Face Detector) by densely scanning through the image with sliding windows, and then performing a binary classification directly using the output of a fine-tuned AlexNet [22]. The idea of using CNN as a classifier component was further extended to DPM [29, 41] for face detection. The most relevant work to our SdA-RF detector is [25], which constructed a 3-stages cascade detector using 3 detection nets, and 3 calibration nets. Within each stage, the detection net separates face and background, and then calibration net refines the locations of retained windows. The next stage processes the refined windows with the same procedure but using double resolution and deeper CNN models.

In this paper, we propose a general cascade object detection method, SdA-RF which embeds SdA and RF into a cascade framework. Two main differences compared to the deeper models make it unique. First, SdA-RF uses a rather simple neural network, 1-layer SdA, to extract the features, and RF to perform classification, which makes densely scanning an image possible for better localization without any refinement procedure. Second, SdA-RF does not rely on any pre-trained model. Unsupervised pre-training and supervised fine-tuning can all be done using the same dataset. The paper is organized as follows: In Sec. 2, we introduce SdA-RF model, and then present how an object detection cascade is built via combining individual SdA-RF stage classifiers; In Sec. 3, we show the experimental results on two public datasets, and discuss the findings; We conclude our work in Sec. 4.

2 Proposed Method

2.1 Stage Classifier

For detection problems, an ideal cascade detector requires individual stages trained with high recall rate and low fallout rate, which enable the cascade to eliminate negative sub-windows as early as possible, meanwhile preserve most of the positives. However, there is no free lunch to train such an ideal classifier. For example, Viola-Jones method [37] trained Adaboost classifiers via searching the feature space exhaustingly to meet the recall rate requirement. In that way, the classifiers constructed normally have very high recall but also have relatively high fallout, therefore increasing the number of stages is necessary to filter out most negatives. The upper bound of detection rate within a Viola-Jones cascade $Upper(R_c)$ depends on the number of stages N, and the recall rate of individual stage R_{s_i}, where $Upper(R_c) = \prod_i^N R_{s_i}$. Bourdev et al. [2] proposed a so-called soft cascade method, where the sub-windows are eliminated based on votes of multiple stages with importance weights, instead of one stage in a traditional cascade method. However, such a accumulative elimination scheme involves duplicative computational cost for each sub-windows. We seek to train each stage with a stronger classifier, which can greatly reduce the number of stages but also retain considerable high recall rate and low fallout rate.

Y. Bengio et al. [1] proposed a deep representation learning method, namely Stacked Autoencoder (SA). Individual layer of SA is a latent model trained iteratively using two phases, encoding and decoding in an unsupervised fashion.

Given the observation $v \in \mathbb{R}^{D_v}$, where D_v is the number of dimensions of visible variable, firstly the encoder maps (upwards) v into latent representation $h \in \mathbb{R}^{D_h}$, where D_h is the number of dimensions of the latent variable. Between visible nodes and latent nodes, a fully connected network is constructed to represent the mapping functions, but intra-connection between the same type of node is not allowed in order to keep the complexity of the model itself relatively simple. The decoder works in an opposite way to the encoder, it maps (downwards) the latent representation v back to the so-called reconstructed observation $\bar{x}$. The mapping functions are rametrized using a continuous-value extension of Restricted Boltzmann Machine (RBM) [17] with $\theta = (W, W', b, b')$, where the upwards and downwards mappings are formulated as $h = \mathbb{S}(Wx + b)$, and $\bar{x} = \mathbb{S}(W'h + b')$ respectively. Typically, the sigmoid is used as activation function $\mathbb{S}$, and tied weights constrain, $W' = W^T$, is applied to the model. The estimator of the model can be obtained by minimising the squared reconstruction error, $Loss(\bar{x}, x) = ||\bar{x} - x||^2$. Greedy layer-wise training is applied, where the latent variable of the previous layer is fed into the current layer as input. Trained layers are stacked hierarchically and followed by a fine-tuning procedure, which also can be done in a supervised way by stacking a Softmax layer on the top. Furthermore, P. Vincent [36] proposed SdA model, which introduces an artificial input corruption scheme into the layer-wise training procedure in order to avoid identity learning and improve robustness. For an input x, SdA stochastically forces a certain amount of input channels to 0 in order to generate a corrupted version $\tilde{x}$, and then trains a normal autoencoder using $\tilde{x}$. This training strategy shows that sensibility to small irrelevant changes in input can be significantly reduced.

SdA offers a layer-wise unsupervised representation learning method, which can be used as a general feature extractor for various object detections. In the case that ground truth labels are available, supervised fine-tuning will help to improve classification performance even further. However, it is notable that the classification power of SdA with Softmax layer is relatively weak compared to the-state-of-art discriminative models. For example, random forests (RF) was used for human interaction recognition [7], which is one of challenge problems in computer vision field. In order to address such shortcomings, in this work, RF is trained using encoded representations learned by SdA to classify subwindows into positives and negatives. RF [4] grows a number of decision trees independently using the bagging subsets [3] which are randomly sampled from the complete training set with replacement. Individual decision tree consists of a set of tests (non-leaf node) and predictors (leaf node), where either Gini impurity or information gain is used to find the best split. During the prediction stage, the testing samples traverse through each decision tree by evaluating its properties at non-leaf node, and finally reaches a leaf node at the bottom, which votes the class with largest proposition of training samples it holds. The random forests combine all voting results from individual decision trees, and assigns the most voted class to the testing sample. The training procedure for stage classifier combining SdA feature learning, and RF classification is described in **Algorithm** 1.

Algorithm 1: Train a Stage Classifier for Binary Classification using SdA and RF.

```
function M = trainSdARF $(I, L)$;
Input  : $I$ is a set of training images containing positives and negatives.
Input  : $L$ is a set of ground truth labels for $I$, where $L_i \in \{0, 1\}$,
         corresponding to negative and positive respectively.
Output: $M$ is stage classifier, which consists of one SdA model and one RF
         model.
$nLayers \leftarrow$ set the number of hidden layer used for training SdA;
$nHiddens \leftarrow$ set the numbers of hidden nodes of each layers;
$nTrees \leftarrow$ set the number of decision tree used for training RF;
$rLayers(0) \leftarrow$ initialise the encoded feature with original image $I$ for the first layer training;
for $j \leftarrow 1$ to $nLayers$ do
    $sdaLayers(j) \leftarrow$ train an SdA model using input feature $rLayers(j-1)$ in an unsupervised fashion with $nHiddens(j)$ hidden nodes;
    $rLayers(j) \leftarrow$ encode output feature using trained model $sdaLayers(j)$ and input feature $rLayers(j-1)$;
end
$smLayer \leftarrow$ train a Softmax layer using encoded feature $rLayers(nLayers)$ given by the top level SdA model, and ground truth label $L$ in a supervised fashion;
$Network \leftarrow$ stack the pre-trained $sdaLayers$ and $smLayer$ layer-wise to form a fully connected neural network;
$Network \leftarrow$ fine-tune $Network$ using input image $I$ and label $L$ in a supervised fashion;
$Representation \leftarrow$ collect the features encoded using $Network$;
$Forests \leftarrow$ train an RF model using $Representation$ and label $L$ with $nTrees$ trees;
$M \leftarrow$ assemble the neural networks encoder $Network$ and RF $Forests$;
return $M$;
```

2.2 Detection Cascade

For object detection, as the number of positives is far less than negatives, cascade-based methods, which often bias towards negatives, are relatively more efficient. However, as discussed in Sec. 2.1, adding more stages is required to reduce false positive rate, at the expense of reducing true positive rates. The proposed stage classifier **Algorithm** 1 addresses this contradiction by introducing better feature learning methods and more discriminative models. To train each classifier stage a sliding window method is used to generate negative sub-windows, which then pass through the previous stage's classifier. Only those predicted as positives are retained and used for training the current stage. It is notable that the classification problem becomes more challenging with increasing stage depth, as retained sub-windows are collected from more different images. With the number of stage growth, the number of tree in RF is progressively increased to overcome the difficulties introduced by the larger diversity present in the negative set. The cascade training procedure is described in **Algorithm** 2.

Algorithm 2: Train an Object Detection Cascade.

```
function C = trainCascade ($Pos, Neg$);
Input  : $Pos$ is a set of positive training images all of which have the same
         size. $h$, $w$, $nPos$ are height, width, and total number of positive
         images respectively.
Input  : $Neg$ is a set of negative training images with no target object, where
         $nNeg$ is the total number of negative images.
Output : $C$ is object detection cascade, which consists of multiple stage
         classifiers.
$maxStages \leftarrow$ set the maximum number of stages;
$minRecall \leftarrow$ set the minimum overall recall rate;
$maxFallout \leftarrow$ set the maximum overall fallout rate;
$ratioNegPos \leftarrow$ set the number ratio of training samples, negatives over
positives;
$nTrees \leftarrow$ set the number decision trees used for training stage classifier;
for $j \leftarrow 1$ to $maxStages$ do
    $trnWindows \leftarrow$ create $nPos \times ratioNegPos$ negative samples of size $(h, w)$
    using sliding window methods from negative images $Neg$, where only those
    ones pass through $C(1 : j - 1)$ are retained, and then combine with positive
    sample $Pos$;
    $trnLabels \leftarrow$ label the training windows as 0 for negative, and 1 for positive;
    $Ctemp \leftarrow$ train a stage classifier with $nTrees$ using $trnWindows$ and
    $trnLabels$;
    $nTrees \leftarrow$ increase the number of decision trees for next stage training;
    $(oaRecall, oaFallout) \leftarrow$ compute the overall recall rate, and fallout rate;
    if $(oaRecall < minRecall) \parallel (oaFallout > maxFallout)$ then
        break the stage training loop;
    end
    $C(j) \leftarrow Ctemp$ assign stage classifier to collection;
end
return $C$;
```

3 Experiments and Discussion

We used AFLW (Annotated Facial Landmarks in the Wild [21]) dataset to train a face detector. The dataset contains 22,712 labelled faces out of 21,123 images. The positive face windows were further augmented by applying 5 random perturbations to the location of face window within the range of 5% of its size, and also collecting all flipped face windows. In total, 227,120 faces are used in the training procedure, and some examples of positive samples are shown in Fig. 1 (a). The negative images should contain no face. To bootstrap non-face images the AFLW dataset was used, where the labeled face windows were replaced with no face patches randomly cropped from PASCAL VOC dataset [9, 8] (person subset was excluded). In total, 19,458 negative images were generated using this bootstrapping approach. As considerable amount of images of AFLW dataset are not well labeled with face bounding box, we further applied face detection on the negative images using Koestinger's VJ-LBP model (Viola-Jones detector with Local Binary Patterns feature) [20]. After removing those that have true positive response, the negative image set contains 18,089 images.

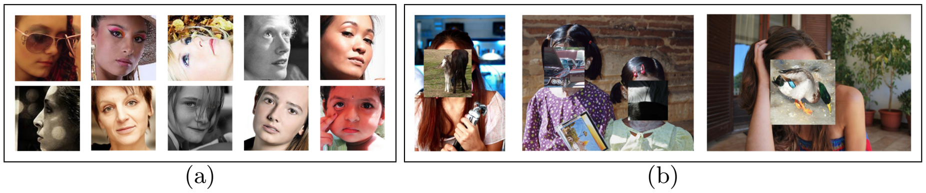
(a)
(b)

Fig. 1. Positive training face images (a), and negative images (b) from AFLW and PASCAL VOC datasets.

The size of the training image window is 24 × 24 pixels, to which all face windows were resized and converted into grey scale. There is no histogram equalization or any further image enhancement. To create negative training windows, we applied sliding window method to each negative image with scale factor $Sn = 1.2$, and stride $Sx = Sy = 2$ pixels. The generated negative windows were firstly sent to previous stage classifiers, only those ones passed through were retained for current training procedure. For SdA model, one hidden-layer with 12×12 nodes was used, which was trained in an unsupervised fashion with image intensity, this was then followed by a supervised fine-tuning. Fig. 2 shows the visualization of the partial weights given by hidden nodes of the first and last stage classifier before and after supervised fine-tuning. The weights are shown as a set of basis for reconstructing original image signal using the output of encoder, which capture the characteristics of face rather well. It is notable that the reconstruction basis of the last stage classifier (2nd row) is more informative compared to the first stage (1st row). When the cascade goes deeper, training a stage classifier becomes more challenging, because easy negative windows are filtered out by the previous stages and hard ones are retained. Also we observed that fine-tuning SdA by back-propagating the prediction error given by the Softmax layer makes the basis more specific for face detection. For example, two red bounding boxes in Fig. 2 show the weights from the same hidden node before and after supervised fine-tuning. The same phenomenon can be observed across all stages. The output of the encoder was used as a high-level representation to train an RF classifier. The minimum number of samples in each leaf node was set to 3 in order to avoid over-fitting. The number of decision trees of the first stage was set to 25, and it was progressively increased with 5 more trees every one stage deeper. The whole cascade training finished with 5 stages as no more negative windows can be generated given 18,089 non-face images. This is a significant reduction in terms of number of stages compared to traditional VJ detectors (20 stages used in[21]).

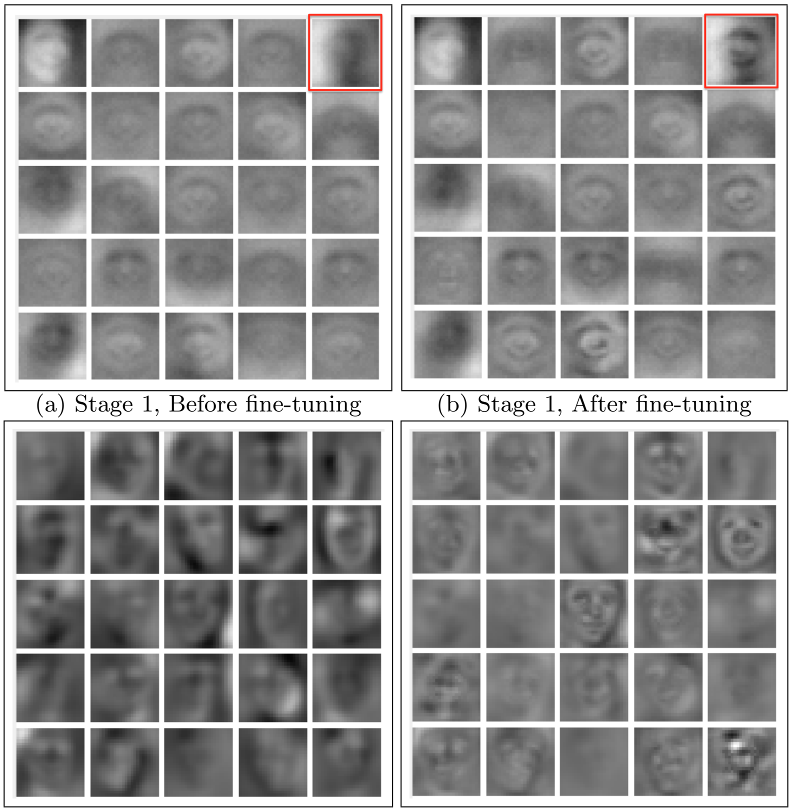
(a) Stage 1, Before fine-tuning
(b) Stage 1, After fine-tuning
(c) Stage 5, Before fine-tuning
(d) Stage 5, After fine-tuning

Fig. 2. The visualisation of SdA weights before and after supervised fine-tuning from the first and last stages.

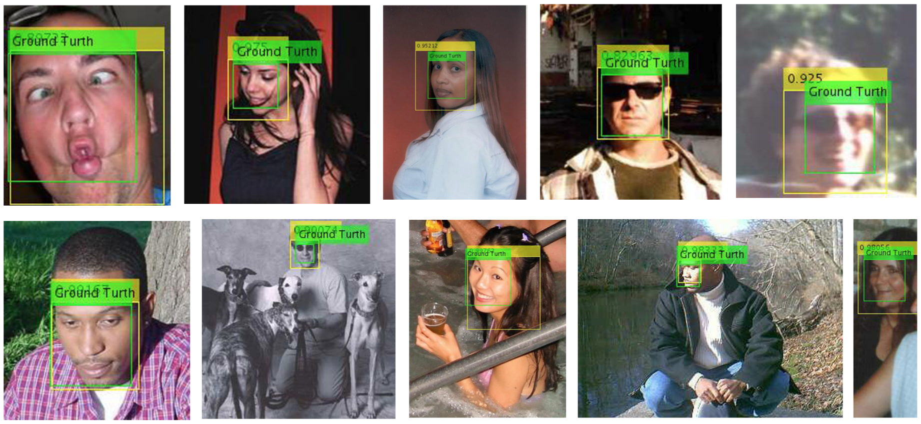

Fig. 3. Representative results of face detection on GENKI-SZSL dataset. Green bounding boxes are the ground truth, and yellow boxes are detection results given by SdA-RF detector.

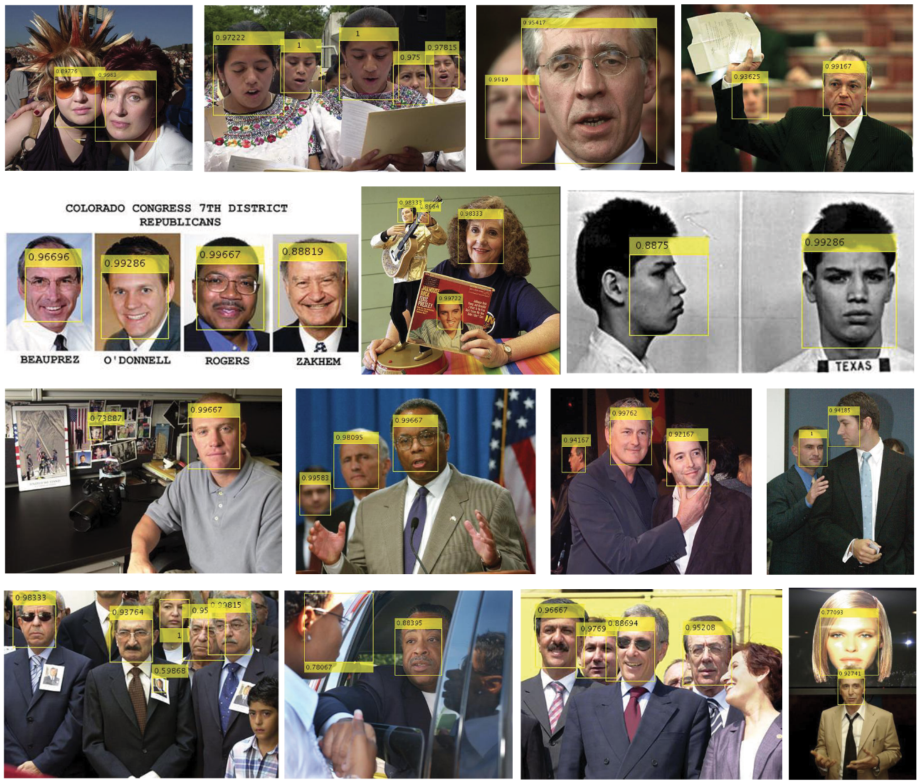

Fig. 4. Representative results of face detection on FDDB dataset.

The face detector was verified on two public datasets, GENKI [28] and FDDB (Face Detection Dataset and Benchmark [19]) and qualitative results are shown in Fig. 3 and Fig. 4 respectively. We evaluated our detector on SZSL, a subset of the GENKI database, which contain 3,500 images. Fig. 3 shows our detector can handle different face expressions, view angles, illumination conditions. FDDB contains 2,845 images with a total of 5,171 faces. It is extremely challenging dataset, for example, Fig. 4 shows some representative detection results on images with severe occlusion and blurring (see 3rd and 4th images of 1st row), and over 90 degree rotation (see 3rd image of the 2nd and 3rd rows, and 2nd image of 4th row).

4 Conclusion and Future Work

In this paper, we presents a general cascade-based object detection methods by employing SdA for feature extraction, and RF for object-background classification. It shows that by combining shallow neural networks and discriminative classifier it is possible to carry out binary object detection, and there is perhaps no need to introduce deeper models and complex training procedures. The preliminary results on two public datasets are promising. Quantitative analysis, code optimization with GPU implementation, and application on other detection problems such as pedestrian, are three main aspects for our future work.

References

1. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. Advances in Neural Information Processing Systems 19, 153–160 (2007)
2. Bourdev, L., Brandt, J.: Robust object detection via soft cascade. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. vol. 2, pp. 236–243 (June 2005)
3. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
4. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
5. Chiverton, J., Xie, X., Mirmehdi, M.: Automatic bootstrapping and tracking of object contours. IEEE Transactions on Image Processing 21(3), 1231–1245 (March 2012)
6. Daubney, B., Xie, X., Deng, J., Parthalin, N.M., Zwiggelaar, R.: Fixing the root node: Efficient tracking and detection of 3d human pose through local solutions. Image and Vision Computing 52, 73 – 87 (2016)
7. Deng, J., Xie, X., Daubney, B.: A bag of words approach to subject specific 3d human pose interaction classification with random decision forests. Graphical Models 76(3), 162 – 171 (2014)
8. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision 111(1), 98–136 (Jan 2015)
9. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes (VOC) challenge. International Journal of Computer Vision 88(2), 303–338 (2010)

10. Farfade, S.S., Saberian, M.J., Li, L.J.: Multi-view face detection using deep convolutional neural networks. In: Proceedings of the ACM on International Conference on Multimedia Retrieval. pp. 643–650. ACM (2015)
11. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.: Cascade object detection with deformable part models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2241–2248 (2010)
12. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(9), 1627–1645 (2010)
13. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1440–1448 (2015)
14. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 580–587 (2014)
15. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 37(9), 1904–1916 (2015)
16. Heisele, B., Serre, T., Poggio, T.: A component-based framework for face detection and identification. International Journal of Computer Vision 74(2), 167–181 (2007)
17. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science 313(5786), 504–507 (2006)
18. Hjelmås, E., Low, B.K.: Face detection: A survey. Computer Vision and Image Understanding 83(3), 236–274 (2001)
19. Jain, V., Learned-Miller, E.: FDDB: A benchmark for face detection in unconstrained settings. Tech. Rep. UM-CS-2010-009, University of Massachusetts, Amherst (2010)
20. Koestinger, M.: Efficient Metric Learning for Real-World Face Recognition. Ph.D. thesis, Graz University of Technology, Faculty of Computer Science (2013)
21. Koestinger, M., Wohlhart, P., Roth, P.M., Bischof, H.: Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In: IEEE International Workshop on Benchmarking Facial Image Analysis Technologies (2011)
22. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. pp. 1097–1105 (2012)
23. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature 521(7553), 436–444 (2015)
24. Li, H., Lin, Z., Brandt, J., Shen, X., Hua, G.: Efficient boosted exemplar-based face detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1843–1850 (2014)
25. Li, H., Lin, Z., Shen, X., Brandt, J., Hua, G.: A convolutional neural network cascade for face detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5325–5334 (2015)
26. Liao, S., Jain, A.K., Li, S.Z.: A fast and accurate unconstrained face detector. IEEE Transactions on Pattern Analysis and Machine Intelligence 38(2), 211–223 (Feb 2016)
27. Liao, S., Zhu, X., Lei, Z., Zhang, L., Li, S.Z.: Learning multi-scale block local binary patterns for face recognition. In: Advances in Biometrics, pp. 828–837. Springer (2007)
28. MPLab, University of California, S.D.: The MPLab GENKI Database, GENKI-SZSL Subset (2009), http://mplab.ucsd.edu, accessed: 2016-05-12

29. Ouyang, W., Luo, P., Zeng, X., Qiu, S., Tian, Y., Li, H., Yang, S., Wang, Z., Xiong, Y., Qian, C., et al.: Deepid-net: multi-stage and deformable deep convolutional neural networks for object detection. arXiv preprint arXiv:1409.3505 (2014)
30. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems. pp. 91–99 (2015)
31. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) 115(3), 211–252 (2015)
32. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229 (2013)
33. Shen, X., Lin, Z., Brandt, J., Wu, Y.: Detecting and aligning faces by image retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3460–3467 (2013)
34. Szegedy, C., Toshev, A., Erhan, D.: Deep neural networks for object detection. In: Advances in Neural Information Processing Systems. pp. 2553–2561 (2013)
35. Uijlings, J.R., van de Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. International Journal of Computer Vision 104(2), 154–171 (2013)
36. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of International Conference on Machine Learning. pp. 1096–1103. ACM (2008)
37. Viola, P., Jones, M.J.: Robust real-time face detection. International Journal of Computer Vision 57(2), 137–154 (2004)
38. Yan, J., Lei, Z., Wen, L., Li, S.: The fastest deformable part model for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2497–2504 (2014)
39. Yang, B., Yan, J., Lei, Z., Li, S.Z.: Aggregate channel features for multi-view face detection. In: IEEE International Joint Conference on Biometrics. pp. 1–8 (2014)
40. Yang, M.H., Kriegman, D.J., Ahuja, N.: Detecting faces in images: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(1), 34–58 (2002)
41. Yang, S., Luo, P., Loy, C.C., Tang, X.: From facial parts responses to face detection: A deep learning approach. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3676–3684 (2015)
42. Zafeiriou, S., Zhang, C., Zhang, Z.: A survey on face detection in the wild: Past, present and future. Computer Vision and Image Understanding 138, 1–24 (2015)
43. Zhang, C., Zhang, Z.: A survey of recent advances in face detection. Tech. Rep. MSR-TR-2010-66, Microsoft Research (June 2010)