Improving Multiview Face Detection with Multi-Task Deep Convolutional Neural Networks

Cha Zhang and Zhengyou Zhang
Microsoft Research
One Microsoft Way, Redmond WA 98052

Abstract

Multiview face detection is a challenging problem due to dramatic appearance changes under various pose, illumination and expression conditions. In this paper, we present a multi-task deep learning scheme to enhance the detection performance. More specifically, we build a deep convolutional neural network that can simultaneously learn the face/nonface decision, the face pose estimation problem, and the facial landmark localization problem. We show that such a multi-task learning scheme can further improve the classifier's accuracy. On the challenging FDDB data set, our detector achieves over 3% *improvement in detection rate at the same false positive rate compared with other state-of-the-art methods.*

1. Introduction

Face detection has been one of the fundamental technologies to enable natural human-computer interaction. In recent years, significant progress has been made on this topic [32][34]. While frontal face detection has been largely considered a solved problem thanks to the seminal work by Viola and Jones [29], multiview face detection remains a challenging task due to dramatic appearance changes under various pose, illumination and expression conditions. The performance of modern face detection solutions on multiview face data set is still unsatisfactory, as shown on the recently published FDDB benchmark [8].

The classic strategy for multiview face detection has been "divide and conquer". Namely, we divide face images into multiple categories, e.g., frontal, half profile, profile, etc. Different classifiers can then be trained for different subcategories. Research publications along this direction include[13][30][6][33], etc. The benefit of such a scheme is that the detection can be performed at a relatively fast speed, in particular when certain hierarchical structure is adopted [6][33] during detection. On the other hand, the performance of these detectors are not state-of-the-art, mostly due to the simple Haar features that are almost universally adopted due to detection speed concerns.

One direction to improve the situation is to apply more complex features in the detector, e.g., LBP features [12], generic linear features [17], SURF features [16], etc. Since these complex features are often slow to compute, a common practice is to construct post filters with these features. For instance, Rong *et al*. [31] applied a support vector machine (SVM) post-filter with wavelet features, and demonstrated improvement in the detector's precision.

In this paper, we propose to apply deep convolutional neural networks (DCNN) as the post filter, which is known to be able to extract effective features automatically during learning. Furthermore, we learn the post filter in a multi-task learning (MTL) framework, where we jointly train the face/non-face decision, the facial pose estimation, and the facial landmark localization problem simultaneously. Our experimental results show that MTL can improve the accuracy of the learned classifier, and the final DCNN classifier can achieve state-of-the-art performance on the challenging FDDB data set.

The rest of the paper is organized as follows. Related works are discussed in Section 2. An overview of our detection system is presented in Section 3. The multi-task DCNN is introduced in Section 4. Experimental results and conclusions are given in Section 5 and 6, respectively.

2. Related Works

Our work is certainly not the first that adopts neural networks on the face detection problem. In fact, before the Viola and Jones [29] detector was published, neural network had been a very popular approach [24][23][4] and achieved state-of-the-art performance at that time. More recently, the convolutional neural architecture was used for face detection in [5][19]. While no multi-task learning is involved, Osadchy *et al*.'s scheme [19] trained a convolutional neural network to map face images to points on a low dimensional face manifold parameterized by facial pose, and non-face images to points far away from the manifold. The detector was fast and achieved satisfactory performance, on par with the boosting based detectors such as [29].

In the past few years, deep neural networks (DNN) have seen a surge in research interest, thanks to the ground-breaking performance improvement on various applications such as speech recognition [22] and image classification [14]. A few research works have been reported to apply DNN on face related problems. For instance, Luo *et al*. applied DNN on face parsing [21]; Huang *et al*. applied convolutional deep belief networks on face verification [7]; Sun *et al*. applied deep convolutional cascade on facial landmark detection [28]; and Sermanet *et al*. applied DCNN on pedestrian detection [25]. To the authors's best knowledge, the latest deep learning algorithms have not been applied to face detection yet.

A few novel face detection algorithms have also been presented recently. Notably, Zhu and Ramanan [35] presented a mixture of trees model with shared parts for face detection, pose estimation, and landmark estimation. Both shape and appearance parameters are learned discriminatively using a structured prediction framework. Shen *et al*. [26] presented a scheme to perform face detection and alignment by image retrieval. Their method contains a validation step that uses a similar voting scheme for face validation and landmark localization. These works indicate that the face detection problem is highly related to other facial tasks such as pose estimation and landmark detection. Therefore, in this paper, we adopt a multi-task learning framework [1] with DCNN to examine whether additional improvement on face detection could be achieved.

Multi-task learning is a machine learning approach that learns a problem together with other related problems at the same time using a shared representation. It often leads to a better model for a single task than learning it independently, because it allows the learner to use the commonality among tasks. It is very natural to apply multi-task learning on a neural network, since one can simply learn multiple targets and make them share the common lower layers. Multi-task learning with neural networks have been applied in natural language processing [2], traffic flow forecasting [11], speech recognition [3], etc.

3. System Overview

We use multi-task DCNN to build a post filter for a boosting based multiview face detector. The overall flow for predicting whether a given image patch is a face or not is shown in Fig. 1. The image patch first passes through a cascade-based multiview face detector. If the decision is true, we scale the patch to 32 × 32 pixels, and perform preprocessing for the patch. It is then sent to a DCNN for the final decision. Since the first stage multiview face detector is cascade-based, it is fast and can reject most negative patches. Consequently, the overall system can still run fast despite the relatively more complex DCNN classifier.

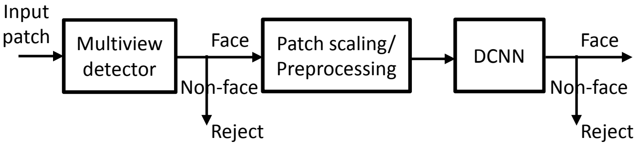

Figure 1. Algorithm flow for predicting whether an image patch is a face or not.

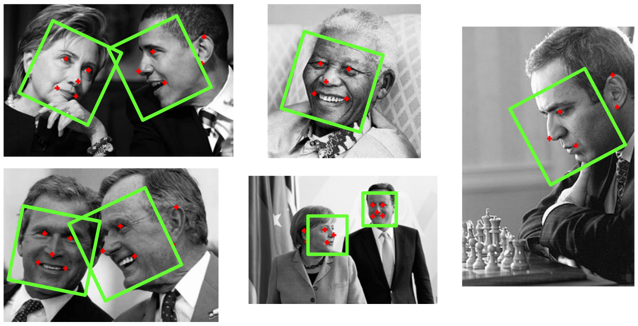

Figure 2. Example landmark labels for different face poses.

3.1. The Training Data

We collected face data from various sources, including images from the web, the Feret database [20], the BioID database [10], the PIE database [27], etc. The faces are categorized into 5 poses, namely, frontal, left/right half profile, left/right profile, each with 0 and ±30 degree in-plane rotations (15 subcategories in total). The data set includes about 49,000 frontal faces, 43,000 half profile faces, and 25,000 profile faces. Some example images and their labeled landmarks are shown in Fig. 2. For frontal and half profile faces, the left/right eye center, the nose tip and the left/right mouth corners are manually labeled. For profile faces, the visible eye's center, the nose tip, the visible mouth corner, ear top and bottom are labeled. The cropping windows of the faces are then computed from the labeled landmarks. The face patches are scaled to 32 × 32 pixels for training.

To increase the data variation, we apply numerous random lighting and geometry variations to the face patches, including flipping/shift/scale/rotation for the cropping region, shearing in horizontal and vertical directions, intensity variance scaling, intensity gamma nonlinearity, etc. In the end, the overall number of (artificially generated) face examples used for training is about 2 million. The negative image set is also collected from the web, which contains about 30 thousand images, or 10 billion non-face patches of size 32 × 32 pixels.

3.2. The Boosting-Based Multiview Face Detector

We follow the algorithm in [33] to build the boosting-based multiview face detector. It uses a winner-take-all approach to adaptively relabel the faces in order to achieve

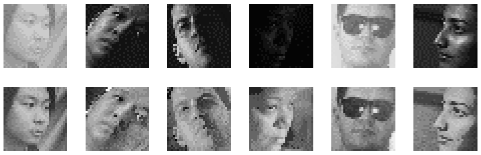

Figure 3. Patch preprocessing before being sent to the DCNN. Top row: before preprocessing; bottom row: after preprocessing.

better performance. Since practically any multiview face detector can be applied before our DCNN post filter, we skip the details of the boosting classifier in this paper.

It is important to set the final threshold of the boosting-based multiview detector appropriately. Setting the threshold too high may cause many positive examples be rejected even before entering the DCNN post filter, and setting it too low may lead to too many pass-through patches that needs to be classified further by the DCNN, slowing down the overall detection process. In this paper, we set the threshold of the boosting-based detector to successfully detect about 94% of the training face patches, which leads to about 0.057% false positive rate, or 10 to 100 positive patches for a typical image. We pass all the training patches through the detector, which leaves about 1.88 million positive patches and 570 thousand negative patches to be used to learn the post filter.

3.3. Patch Preprocessing

The image patches are preprocessed before sending to the DCNN for classification. The preprocessing consists of three steps: histogram equalization, linear lighting removal, and intensity normalization. In the first step, we perform standard histogram equalization to enhance the contrast of the image patch. We then fit a linear plane onto the image intensity, namely, let:

$$ax + by + c = I, \tag{1}$$

where (x, y) is the pixel location, and I is the corresponding pixel intensity. We use least square fitting, which has a close form solution in this case, to find the best fitting parameters a, b and c, and keep the value $I - ax - by - c$ as the pixel value at (x, y). The last step is to normalize the pixel values to unit variance. Fig. 3 shows the results of such a three-step preprocessing, which is very effective in improving image contrast and removing shadows.

4. Multi-Task DCNN

4.1. Network Architecture

We adopt a multi-task DCNN as shown in Fig. 4. Given the input 32×32 grayscale image patch, the first layer is a convolutional layer with 32 kernels of size $5 \times 5 \times 1$, followed by 2×2 max pooling. The second convolutional layer takes as input the output of the first layer, and filters it with 32 kernels of size $3 \times 3 \times 32$. The third convolutional layer has 24 kernels of size $3 \times 3 \times 32$, which is again followed by 2×2 max pooling. The fourth layer is a fully connected layer that has 512 neurons. Afterwards, the network is split into three branches:

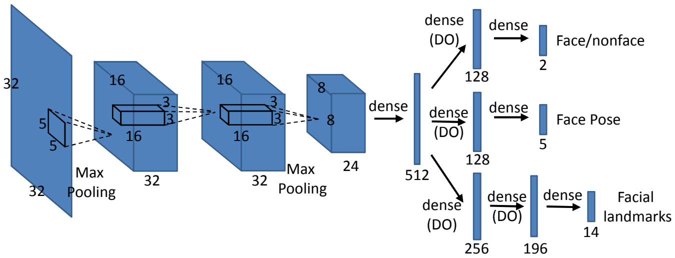

Figure 4. The multi-task DCNN network adopted in this paper. Here "DO" indicates that the densely connected layer has dropout enabled.

- The first branch learns the face/nonface decision, and it contains 2 additional fully connected layers, with size 128 and 2, respectively. Dropout [14] is enabled at the first dense layer. The output is fed to a 2-way softmax, and we minimizes the cross-entropy loss:

$$L_1 = -\sum_{i=1}^{2} t_i \log y_i, \tag{2}$$

where $(t_1, t_2) = (1, 0)$ for face patches and $(t_1, t_2) = (0, 1)$ for nonface patches, and (y_1, y_2) is the softmax output.

- The second branch learns the facial pose (frontal, left/right half profile, left/right profile), which also contains 2 additional fully connected layers of size 128 and 5. Again dropout is enabled at the first dense layer. The output is fed to a 5-way softmax, and we again minimizes the cross-entropy loss:

$$L_2 = -\sum_{i=1}^{5} t_i \log y_i, \tag{3}$$

where $\mathbf{t} = (t_1, \cdots, t_5)$ is a vector with the element corresponding to the ground truth pose set to 1, and 0 otherwise.

- The third branch learns the locations of the facial landmarks. There are 7 landmarks in total (Fig. 2), thus there are 14 output units. The branch has 3 fully connect layers, with size 256, 196 and 14, respectively. The first two dense layers have dropout turned on. The output directly predicts the landmark 2D coordinates, and we minimize a weighted mean square error:

$$L_3 = \frac{1}{2} \frac{\sum_i w_i (z_i - y_i)^2}{\sum_i w_i}, \tag{4}$$

where z_i is the ground truth coordinate value, and w_i is a weight defined on each output. Since during DCNN training we do not distinguish the input face poses, for frontal and half profile faces the ear top and ear bottom coordinate ground truth will be missing; and for profile faces one of the eyes and mouth corners will be missing. We thus set the corresponding weights w_i to 0 during training for the missing labels.

In all convolutional and fully connected layers the ReLU non-linearity [14] is applied, except for the ones in the facial landmark prediction branch, where the hyperbolic tangent nonlinearity is used.

When multi-task learning is performed, we minimize the following linear combination of losses:

$$L = \sum_{i=1}^{3} \alpha_i L_i, \tag{5}$$

where α_i are linear weights, currently set as all equal to 1.0. Note if we set the second and third weights to zero, we return to the traditional single task learning.

4.2. Implementation Details

We train our models using stochastic gradient decent with a batch size of 128 examples, momentum of 0.0, and weight decay of 0.0005. The learning rate is initialized as 0.01 and adapted during training. More specifically, we monitor the overall loss function. If the loss is not reduced for 5 epoches in a row, the learning rate is dropped by 50%. We deem the network converged if the learning rate has dropped below 0.0001.

For our face detection task, as mentioned earlier, the overall number of training examples for the post filter is about 1.88 million positive patches and 570 thousand negative patches. The data are randomly shuffled before sending to the network for training. Each epoch of training takes about 10 minutes on a Nvidia GTX Titan GPU with our inhouse implementation, and the network usually converges in 80-100 epoches.

5. Experimental Results

We tested the detector on the publicly available FDDB data set [8]. The data set contains 5171 faces in 2845 images. In the first experiment, we compare five different approaches to demonstrate the benefit of deep convolution neural networks, and multi-task learning:

- The first approach is directly using the boosting-based multiview detector. No post filter is applied. This is our baseline for this experiment.

- The second approach adopts an SVM based post filter. For each subcategory of the multiview detector, we collect all patches that successfully pass through the detector. We then extract LBP features [18] for the image patch, and train a linear SVM classifier for face/nonface decision in each subcategory. This approach is similar to what was presented in Rong *et al*. [31], except that LBP features are more robust to lighting variations than wavelet features.

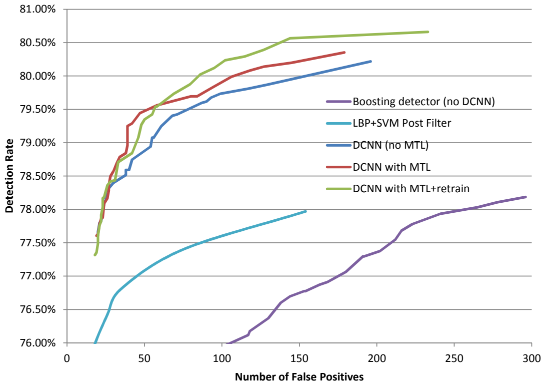

Figure 5. Comparison of five approaches for face detection on the FDDB data set.

- The third approach trains a DCNN on the post filter training data (1.88 million positive examples and 570 thousand negative patches) directly, without involving multi-task learning. We basically set $(\alpha_1, \alpha_2, \alpha_3) = (1.0, 0.0, 0.0)$ in the loss function of Eq. (5), and train a DCNN.

- The fourth approach trains a multi-task DCNN on the same data. This time we set $(\alpha_1, \alpha_2, \alpha_3) = (1.0, 1.0, 1.0)$, although in the end we only verify the face detector's performance in this paper.

- The fifth approach takes the trained multi-task DCNN from the previous approach, freezes all parameters in the second and third branches, and retrain the detector based on the same training data. We hypothesize that the benefit of multi-task DCNN comes from better tuning of the lower layers (thanks to multi-task learning), thus the network is less likely to over-fit. On the other hand, forcing the network to also perform pose estimation and landmark detection may to some degree impact the overall face/nonface decision accuracy. We expect that the retraining will start from a good set of initial network parameters, and thus will converge quickly to a classifier that performs even better.

The results are shown in Fig. 5. We limit the plot to 300 false detections on the whole data set, as we believe detectors with higher false positive rates are not practically useful. It can be seen that in general the three DCNN post filters outperform the baseline detector significantly: at the same false positive rate, the detection rate improves by over

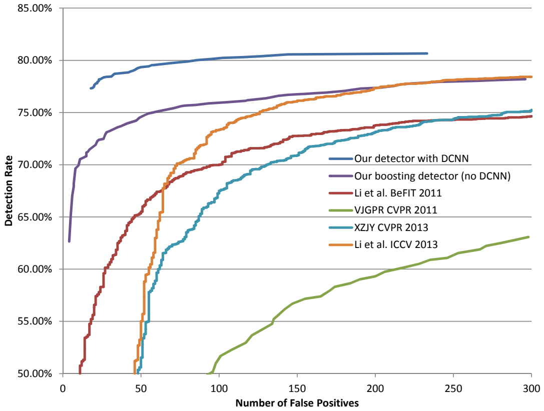

Figure 6. Comparison of our face detector against existing state-of-the-art approaches.

3%. Compared with the linear SVM based post filter, the improvement is also significant: over 2% in detection rate at the same false positive rate.

The three DCNN based post-filters are relatively close to each other in performance. Multi-task learning provides improvement over the vanilla DCNN by $0.2-0.5\%$, while retraining gains another 0.2%. The fully connected layers in our network are mostly equipped with dropout, which is known to be very robust to over-fitting. We think that could be the reason that the improvement was less than what we originally expected. However, a change of 0.5% in detection rate still corresponds to 25 more correct detections, which is good to have.

To give a perspective about how our detector performs against some of the state-of-the-art approaches, we compare our method with four recent publications, including the VJGPR detector by Jain *et al.* in [9], the SURF cascade detector by Li *et al.* [16], the XZJY detector by Shen *et al.* [26] and the adaptation detector by Li *et al.* [15]. The results are shown in Fig. 6. The baseline boosting detector without DCNN and the DCNN post filter with multi-task learning and retraining are also included. It can be seen that our detectors' performance is much better than the existing methods.

One interesting issue we observe is that when we set the DCNN's final threshold to 0.999 (the maximum softmax output is 1), we achieve 77.32% detection rate, with 18 false positives. We were curious how these false positive patches pass through the DCNN with such high confidence. In Fig. 7 we show all the 18 "false" detections made by our detector. At a glance, all of them appear to be legitimate detections. We then compared these detection results with the ground truth, and noted two major cases. For image 1, 3, 4, 8, 9, 11, 12, 15 and 16, these "false" detections are basically missed faces in the ground truth file. In other words, these are indeed faces but were somehow not labeled in the ground truth. For image 2, 5, 6, 7, 10, 13, 14, 17 and 18, the problem appears to be caused by how a detected face is considered as a true detection in FDDB. As we mentioned, our cropping windows are computed from landmark points during training, and they are different from the FDDB labels. This is particularly clear on profile faces, where our computed cropping window tends to place the visible eye at the horizontal center column. Another factor is that our detector output square windows, and this tends to be miss-judged by FDDB if the ground truth face is an elongated ellipsoid. If better agreement can be made on how the ground truth faces are labeled, we expect our detector to perform even better on the same data set.

6. Conclusions

In this paper, we applied multi-task deep convolutional neural networks to build a post filter to improve the accuracy of multiview face detection. The idea is to learn the face/nonface decision together with facial pose estimation and facial landmark localization. We achieved state-of-the-art performance on the challenging FDDB data set.

For future work, we would like to include more facial attributes into our multi-task learning framework. These attributes may include gender, age, facial expression, lighting, etc. It would also be interesting to vary the weights among different learning tasks, to examine which tasks are more closely related, and thus are more effective when applying multi-task learning.

References

[1] R. Caruana. *Multitask Learning*. PhD thesis, Carnegie Mellon University, 1997. 2

[2] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proc. of ICML*, 2008. 2

[3] G. E. Dahl, D. Yu, L. Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42, 2012. 2

[4] R. Féraud, O. J. Bernier, J.-E. Viallet, and M. Collobert. A fast and accurate face detector based on neural networks. *IEEE Trans. on PAMI*, 23(1):42–53, 2001. 1

[5] C. Garcia and M. Delakis. Convolutional face finder: A neural architecture for fast and robust face detection. *IEEE Trans. on PAMI*, 26(11):1408–1423, 2004. 1

[6] C. Huang, H. Ai, Y. Li, and S. Lao. Vector boosting for rotation invariant multi-view face detection. In *Proc. of ICCV*, 2005. 1

[7] G. B. Huang, H. Lee, and E. Learned-Miller. Learning hierarchical representations for face verification with convolutional deep belief networks. In *Proc. of CVPR*, 2012. 2

[8] V. Jain and E. Learned-Miller. FDDB: A benchmark for face detection in unconstrained settings. Technical report, University of Massachusetts, Amherst, UM-CS-2010-009, 2010. 1, 4

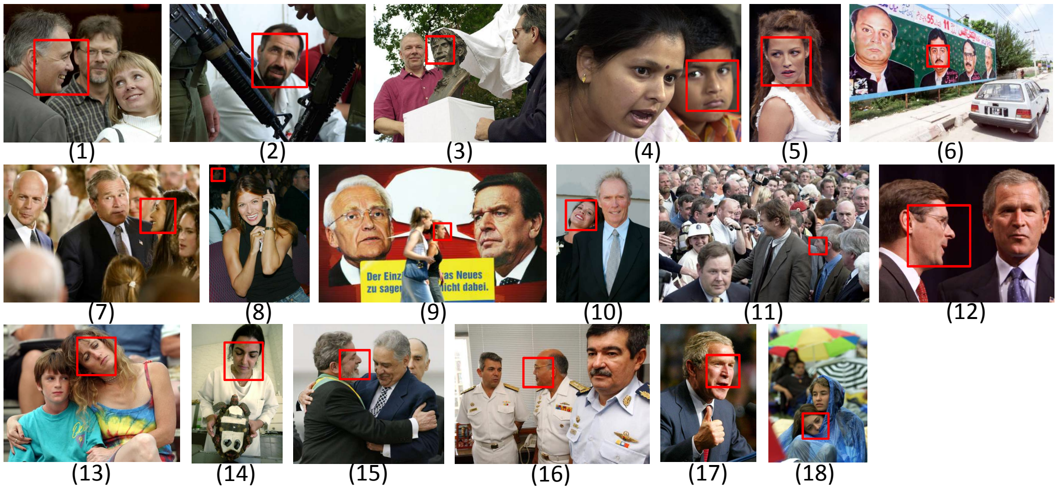

Figure 7. Some "false" detections by our detector.

[9] V. Jain and E. G. Learned-Miller. Online domain adaptation of a pre-trained cascade of classifiers. In *Proc. of CVPR*, 2011. 5

[10] O. Jesorsky, K. Kirchberg, and R. Frischholz. Robust face detection using the hausdorff distance. *Audio and Video based Person Authentication - AVBPA 2001*, pages 90–95, 2001. 2

[11] F. Jin and S. Sun. Neural network multitask learning for traffic flow forecasting. In *Proc. of IJCNN*, 2008. 2

[12] H. Jin, Q. Liu, H. Lu, and X. Tong. Face detection using improved lbp under bayesian framework. In *Third Intl. Conf. on Image and Grahics (ICIG)*, 2004. 1

[13] M. Jones and P. Viola. Fast multi-view face detection. Technical report, Mitsubishi Electric Research Laboratories, TR2003-96, 2003. 1

[14] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In *Proc. of NIPS*, 2012. 2, 3, 4

[15] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang. Probabilistic elastic part model for unsupervised face detector adaptation. In *Proc. of ICCV*, 2013. 5

[16] J. Li, T. Wang, and Y. Zhang. Face detection using SURF cascade. In *ICCV 2011 BeFIT workshop*, 2011. 1, 5

[17] C. Liu and H.-Y. Shum. Kullback-Leibler boosting. In *Proc. Of CVPR*, 2003. 1

[18] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. on PAMI*, 24:971–987, 2002. 4

[19] M. Osadchy, M. L. Miller, and Y. L. Cun. Synergistic face detection and pose estimation with energy-based models. In *Proc. of NIPS*, 2004. 1

[20] P. J. Phillips, H. Moon, P. J. Rauss, and S. Rizvi. The FERET evaluation methodology for face recognition algorithms. *IEEE Trans. on PAMI*, 22(10):1090–1104, 2000. 2

[21] X. W. Ping Luo and X. Tang. Hierarchical face parsing via deep learning. In *Proc. of IEEE CVPR*, 2012. 2

[22] A. r. Mohamed, D. Yu, and L. Deng. Investigation of fullsequence training of deep belief networks for speech recognition. In *INTERSPEECH*, 2010. 2

[23] D. Roth, M.-H. Yang, and N. Ahuja. A SNoW-based face detector. In *Proc. of NIPS*, 2000. 1

[24] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. In *Proc. of CVPR*, 1996. 1

[25] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun. Pedestrian detection with unsupervised multi-stage feature learning. In *Proc. of CVPR*, 2013. 2

[26] X. Shen, Z. Lin, J. Brandt, and Y. Wu. Detecting and aligning faces by image retrieval. In *Proc. of CVPR*, 2013. 2, 5

[27] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression database. *IEEE Trans. on PAMI*, 25(12):1615–1618, 2003. 2

[28] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *Proc. of CVPR*, 2013. 2

[29] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. of CVPR*, 2001. 1, 2

[30] B. Wu, H. Ai, C. Huang, and S. Lao. Fast rotation invariant multi-view face detection based on real adaboost. In *Proc. of IEEE Automatic Face and Gesture Recognition*, 2004. 1

[31] R. Xiao, M.-J. Li, and H.-J. Zhang. Robust multipose face detection in images. *IEEE Trans. on CSVT*, 14(1):31–41, 2004. 1, 4

[32] M.-H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Trans. on PAMI*, 24(1):34–58, 2002. 1

[33] C. Zhang and Z. Zhang. Winner-take-all multiple category boosting for multi-view face detection. In *ECCV Workshop on Face Detection: Where are we, and what next*, 2009. 1, 2

[34] C. Zhang and Z. Zhang. A survey of recent advances in face detection. Technical report, Microsoft Research MSR-TR-2010-66, 2010. 1

[35] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proc. of CVPR*, 2012. 2