2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Face detection based on Deep Convolutional Neural Networks exploiting incremental facial part learning

Danai Triantafyllidou
Artificial Intelligence and Information Analysis Lab
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
Email: danaimar@csd.auth.gr

Anastasios Tefas
Artificial Intelligence and Information Analysis Lab
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
Email: tefas@aiia.csd.auth.gr

***Abstract*—Deep learning methods are powerful approaches but often require expensive computations and lead to models of high complexity which need to be trained with large amounts of data. In this paper, we consider the problem of face detection and we propose a light-weight deep convolutional neural network that achieves a state-of-the-art recall rate at the challenging FDDB dataset. Our model is designed with a view to minimize both training and run time and outperforms the convolutional network used in [1] for the same task. Our model consists only of 113.864 free parameters whereas the previously proposed CNN for face detection had 60 million parameters. We propose a new training method that gradually increases the difficulty of both negative and positive examples and has proved to drastically improve training speed and accuracy. Our second approach, involves training a separate deep network to detect individual facial features whilst creating a model that combines the outputs of two different networks. Both methods are able to detect faces under severe occlusion and unconstrained pose variation and meet the difficulties and the large variations of real-world face detection.**

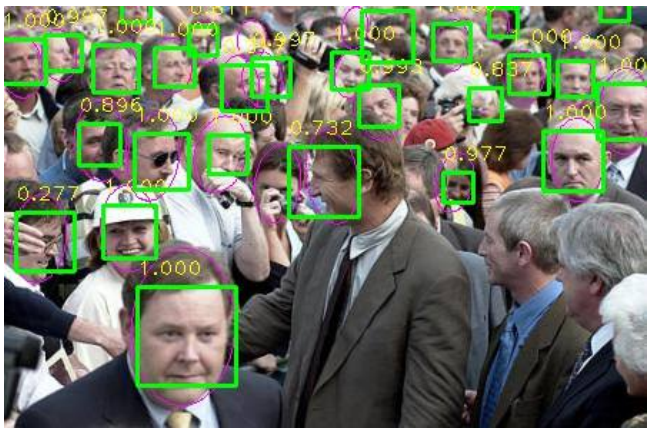


Figure 1: An example of face detection in various poses and occlusions. The bounding boxes and scores show output of the trained CNN.

1. Introduction

Face detection has been an active research area in the computer vision field for more than two decades mainly due to the countless number of applications that require face detection as a first step [30,31,32,33]. Many non neural network methods have been proposed and deployed in various commercial products like digital cameras or smartphones. The seminal work of Viola and Jones [2] made it possible to detect faces in real-time and later on inspired many cascade-based methods. Since then, research in face detection has made remarkable progress as a result of the availability of data in unconstrained capture conditions, the development of publicly available benchmarks and the fast growth in computational and processing power of modern computers.

The original Viola-Jones detector used Haar-like features and is fast to evaluate but fails in detecting faces from different angles. Some methods such as parallel cascade [5] and pyramid cascade [6] address this issue by using one classifier cascade for each specific facial view, while in [4] a decision tree is used for pose estimation and then the corresponding cascade is used to verify the detection. However, these approaches require pose/orientation annotations while complex cascade structures increase the computational cost. Numerous methods also focused on alternatives to Haar-features while others succeeded in improving the detection performance by using more powerful learning algorithms. [8]. The main line of research in this direction was based on the combination of robust descriptors with classifiers [7,29]. Among the variants, Headhunter [9] provided an improved performance by deploying the integral channel features method along with 22 cascades. The last method was extended by [28] where it was proposed to use sub-sampled channel features to learn a cascade of classifiers. Finally, a joint cascade-based method proposed in [3] achieved state-of-the-art results by introducing an alignment step in the cascade structure.

Another common family of face detection algorithms learn and deploy a Deformable Parts-based Model (DPM) [10] to model the information between facial parts. The DPM detectors are more robust to occlusion than cascade based methods but lack in computational efficiency and are prohibitive for real-time detection. A unified DPM framework for face detection, pose estimation, and landmark estimation was proposed in [11]. A general approach for making DPM based

978-1-5090-4846-5/16/$31.00 ©2016 IEEE

methods faster is to build a cascade of classifiers from DPMs [12]. A recent study based on a simple DPM, [13], provides excellent performance and outperforms more complex DPM variants. Finaly, a face detector called Deep Pyramid DPM, [14], significantly improves the detection accuracy. The last method, generates a deep feature pyramid and uses a linear SVM for classification.

The recent resurgence of interest in deep neural networks owes a certain debt to the availability of powerful GPUs which routinely speed up common operations such as large matrix computations. Deep convolutional neural networks have wide applications in language processing, object classification and recommendation systems. A deep network named Alexnet [15] which was trained on ILSRC 2012, outperformed all other methods used for large scale image classification and rekindled interest in convolutional neural networks. In particular, the R-CNN method proposed in [16] generates category-independent region proposals and uses a CNN to extract a feature vector from each region. Then it applies a set of class-specific linear SVMs to recognize the object category. In [1] a face detector called DDFD showed that a CNN can detect faces in a wide range of orientations using a single model.

In this study, we present a novel CNN for face detection that extends the work in [1]. More specifically, we trained two different CNNs that were combined in a single architecture. The first CNN was trained exclusively for the detection of facial features (eg eyes, nose, mouth) while the second CNN was trained for full face detection. This paper makes the following contributions:

1) We propose a novel light-weight model, consisting of only 113.864 free parameters, and we show that our method despite its minimum complexity can provide formidable results and is suitable for real-time detection with standard processing power as opposed to most neural network based detection techniques.
2) We introduce a new approach of handling occlusions and we show that the key to face detection is the information provided by local facial parts.
3) We present a new training methodology according to which the CNN is gradually supplied with training examples of scaling difficulty. We show that our method can drastically improve training speed and significantly reduce the number of false positives.
4) We propose adding a pooling layer to the output of the deep CNN to smoothen the produced heat map.

The proposed trained model is publicly available to the research Community[1]. Figure 1 shows examples of face detection.

2. Proposed method

2.1. CNN Architecture

At first, we trained a fully-convolutional CNN comprised of seven convolutional layers with images of size 32×32, which is shown in Figure 2 and Table 2. Secondly, we trained a network consisting of four convolutional layers interspersed by three dropout layers for the task of facial parts detection. The network was trained with images of size 16×16. The output of the network is comprised of four detection scores, each one corresponding to the four classes of the facial parts (e.g, mouth ,nose, eyes, irrelevant). The architecture of this CNN is also summarized in Figure 2 and Table 1. The first three layers of the facial parts CNN were connected in a parallel manner to the first three layers of the second CNN as shown in Figure 3. The output of the layers of the facial parts CNN, $11 \times 11 \times 24$ is concatenated with the output of the layers of the face detection CNN, $11 \times 11 \times 32$. The concatenation produces a volume of $11 \times 11 \times 56$ which is entered as input to the layers conv4-conv7 as shown in Figure 3. The combined model is trained with RGB images of size 32×32. The 16×16 images of facial parts occupy the $1/4$ of the 32×32 training face images. We believe that the information provided by the detection of local face parts is crucial for detecting face regions and should be part of the initial stages of the detection process.

TABLE 1: Face detection CNN

layer	kernel	# filters	input	output
convolution 1	3 x 3	24	32 x 32 x 3	30 x 30 x 24
convolution 2	4 x 4	24	30 x 30 x 24	14 x 14 x 24
convolution 3	4 x 4	32	14 x 14 x 24	11 x 11 x 32
convolution 4	4 x 4	48	11 x 11 x 32	8 x 8 x 48
convolution 5	4 x 4	32	8 x 8 x 48	5 x 5 x 32
convolution 6	3 x 3	16	5 x 5 x 32	3 x 3 x 16
convolution 7	3 x 3	2	3 x 3 x 16	1 x 1 x 2

TABLE 2: Part-based CNN

layer	kernel	# filters	input	output
convolution 1	3 x 3	16	16 x 16 x 3	14 x 14 x 16
convolution 2	4 x 4	24	14 x 14 x 16	6 x 6 x 24
convolution 3	4 x 4	32	6 x 6 x 24	3 x 3 x 32
convolution 4	3 x 3	4	3 x 3 x 32	1 x 1 x 4

Our work follows the pipeline presented in [1], in a sense that our method does not require any extra module (e.g SVM) for classification as the CNN's output is describing enough for the task of face detection. As the model is fully-convolutional it accepts images of arbitrary size and produces a heat map of the face classifier. We trained the model presented in Figure 3 with images of size 32×32. In addition, no pooling layer was used since there was no need or room to further decrease the size of data volume flowing through the network. As stated in [21], the use of the Parametric Rectified Linear Unit (PReLU) function had a positive impact regarding the detection accuracy of the CNN.

2.2. The dataset

The CNN was trained with positive examples extracted from the AFLW [17] and the MTFL [22] datasets. The first consists of 21K images with 24K face annotations while the second consists of 12K face annotations. Both datasets include real world images with expression, pose, gender, age and ethnicity variations. For AFLW we used the provided face rectangle annotations. For MTFL we used the given facial landmark annotations to produce face rectangles in a similar manner with AFLW examples regarding the positioning of faces.

1. https://github.com/danaitri/Face-detection-cnn

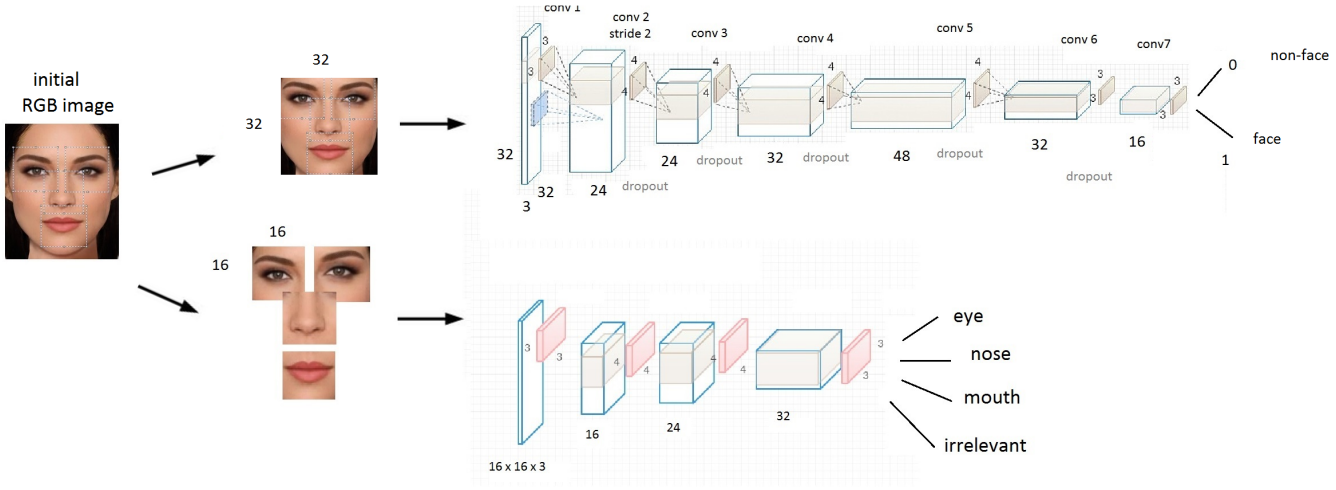


Figure 2: Up: The CNN trained for the task of full face detection. Down: The CNN trained for the task of facial parts detection

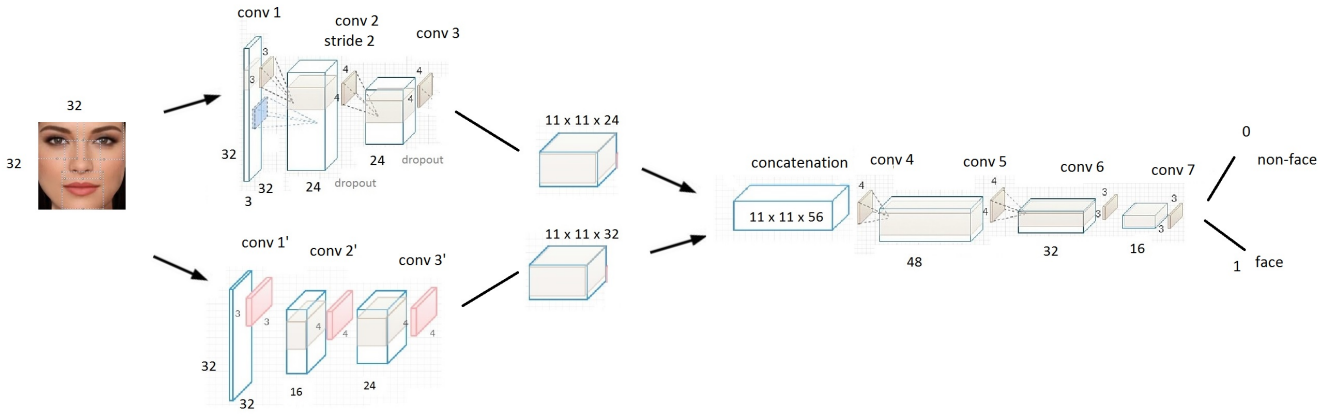


Figure 3: The combined model used for face detection during training and deployment.

The final training images of faces were resized to 32×32. This is a relatively small image size compared to image sizes typically used by AlexNet and other deep networks (e.g. in [1] AlexNet is trained with images 225×225). However, it has been proved that images of this size contain enough information to train the CNN. The relatively small image size allowed to reduce the image down to 1×1 (a face /no face result) without the use of pooling layers. We used only convolution and PReLU layers, each convolution producing a feature smaller than its input layer.

In order to increase the number of positive and negative examples we used horizontal mirroring (flip) of the images with a probability 0.5. This has been proved very effective as it increased the number of training examples and it also led to a practically unlimited number of combinations of images to be entered in each training batch, thus allowing better generalization of the CNN. We also tried augmenting the face dataset by horizontal and vertical displacement of face images by 1 to 3 pixels along the horizontal and vertical axes. This technique proved to be inefficient in experiments, it did not give better results while training and testing and was finally abandoned.

2.3. Proposed Training Methodology

The CNN was trained successively in a set of five different data sets. Let $\mathcal{N}_T$ be a collection of images that will serve as a pool of negative examples. Let $\mathcal{D}_0$ be the original training set consisting of the original set of positive examples $\mathcal{P}_0$ and the set of negative examples $\mathcal{N}_0 \in \mathcal{P}$:

$$\mathcal{D}_0 = \mathcal{T}_0 \cup \mathcal{N}_0 \tag{1}$$

Once the training process is complete, we run the network to the set of images $\mathcal{N}_T$ and we recollect a new set of false

positives $\mathcal{F}_1$ which is added to the original set of negative examples $\mathcal{N}_0$:

$$\mathcal{N}_1 = \mathcal{N}_0 \cup \mathcal{F}_1 \tag{2}$$

The set $\mathcal{F}_1$ is selected according to the network's score. During each training round, we sort the false positives according to their score and we select a predefined number of examples. In order to maintain the same ratio of positive to negative examples after each training round we increase the number of positive examples proportionally. A new set of images containing faces $\mathcal{T}_1$ is added to the original set of positive examples $\mathcal{P}_0$

$$\mathcal{T}_1 = \mathcal{T}_0 \cup \mathcal{P}_1 \tag{3}$$

The aforementioned process of training and increase of training examples is repeated after completion of training in the set D_i:

$$\mathcal{N}_{i+1} = \mathcal{N}_i \cup \mathcal{F}_{i+1} \tag{4}$$

$$\mathcal{D}_{i+1} = \mathcal{P} \cup \mathcal{N}_{i+1} \tag{5}$$

$$\mathcal{P}_{i+1} = \mathcal{P}_i \cup \mathcal{T}_{i+1} \tag{6}$$

Hence, the sets $\mathcal{N}_{i+1}$, $\mathcal{P}_{i+1}$ contain a larger number of negative examples than the sets $\mathcal{N}_i$, $\mathcal{P}_i$.

The increasing difficulty of the described process as well as the adaptation of the training set in the neural network's errors improved the network's performance in unknown data. The process of gradual training in i stages, as described previously, resolves a significantly important issue which was validated in practice. In the event of a training set being unequally distributed between the two classes a training batch may contain little to no actual examples of a class. As a result, the network may be deprived of the presence of examples of said class and the ability to identify between the two may be negatively impacted.

In all our experiments we trained the CNNs using Stohastic Gradient Descent (SGD). We start with a learning rate of 0.001 for the first 200.000 iterations and then we lower to 0.0001. The parallel layers of the combined model shown in Figure 3 were initially locked as they had a fixed learning rate of zero value. After training the layers conv4 to conv7 we unlocked the locked layers to finalize the model. The weights of the network were initialized according to the Xavier method [23]. Figure 4 shows the results of the described procedure for the FDDB dataset.

3. Experiments

3.1. System analysis

We implemented the proposed face detector using the Caffe library [18]. The output of the network shows the scores of the CNN for every 32×32 window with a stride of 2 pixels in the original image. In order to detect faces smaller or larger than 32×32 we scale up or down the original image respectively. We apply the non maximum suppression strategy according to which all bounding-boxes with a possibility lower than the score of the maximum window multiplied by a constant factor are removed. The system was able to detect faces in the FDDB dataset that were not annotated. These detections were removed as they would count for false positives and lead to a deteriorated performance.

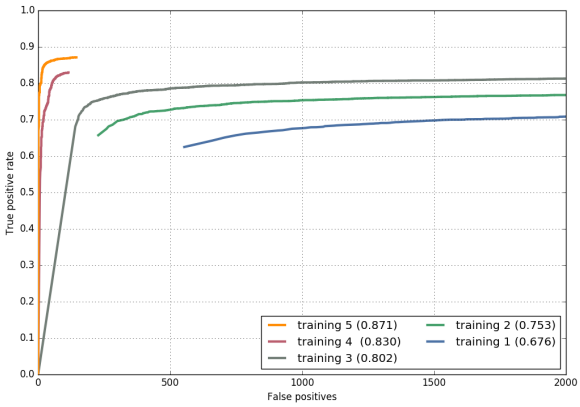


Figure 4: The results of the proposed training methodology on the FDDB dataset for the face detection CNN. A similar approach was used for the part-based CNN and the combined CNN.

During deployment of the CNN, we add an extra max pooling layer to the final output of the network. It has been verified that this layer reduces the number of false positives. The heatmap produced by the CNN is smoothened and only the pixel coordinates having values greater than a specified threshold are stored. Additionaly, the heatmap pixel coordinates having neighbouring coordinates with similar values are stored resulting in reduced false positives and improved performance.

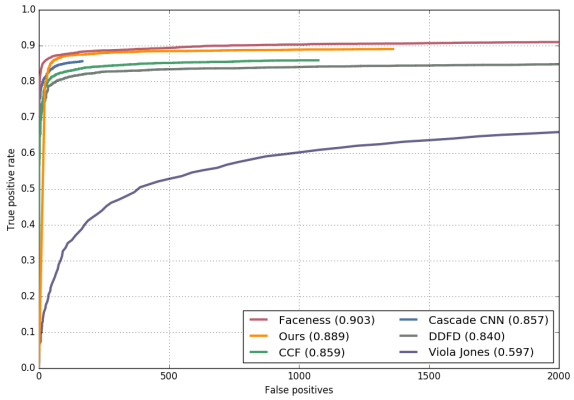
3.2. Comparison with the state of the art

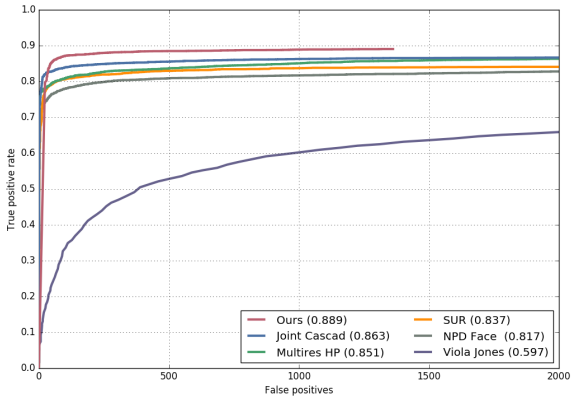
We evaluate the proposed detector on the challenging dataset Face Detection Data Set and Benchmark (FFDB) [20]. Some of the recently published methods compared in this section include: DP2MFD [24], DDFD, Faceness [25], Headhunter, JointCascade [26], SURF [29], ACF [28] and CCF [27]. For evaluation we use the toolbox provided by [19] which includes corrected annotations for the aforementioned benchmarks.

FDDB datase consists of 2845 images with 5171 face annotations collected from journalistic articles and is one of the most commonly used benchmark for face detection. It is a really challenging dataset mainly due to the fact that it is rich in occluded and out-of-focus cases. FDDB faces are annotated with elliptic regions. As stated in [19] changing the output format of detections to ellipses increases the overlap region between the detections and ground truth boxes. However, our detector achieves a high recall rate without this conversion.

Figures 6(a)-6(c) show the precision recall curves for the FDDB dataset. Our method achieves a recall rate of 88.9 %, outperforming all recent published face detection methods except Faceness and DP2MDF. Figure 5 shows some detection examples.

Figure 5: Examples of face detection in the FDDB dataset. The system is able to detect out-of-focus and occluded cases.





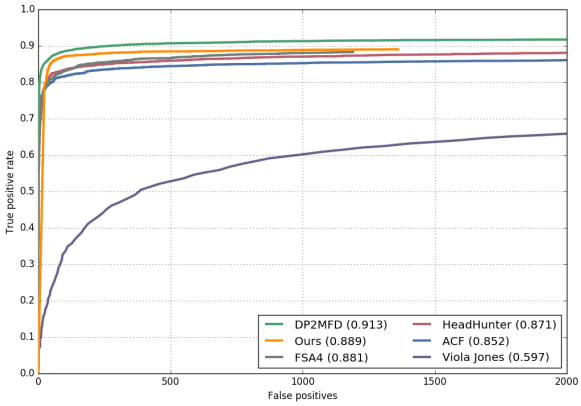


Figure 6: Comparison of different face detectors on FDDB dataset. Against deep architectures (a) Against other state-of-the-art approaches (b),(c)

3.3. Complexity

The complexity of the competitive algorithms is very large compared to the proposed network. Indeed, the proposed deep CNN has 113.864 free parameters whereas the previously proposed deep CNN [1] had 60 million parameters. This issue is very important during training and also during testing and deployment. The proposed lightweight model can be easily deployed to smart devices (e.g. smartphones, notepads, etc) or robotic systems (e.g. drones) that do not have expensive and energy consuming multiple GPUs installed. Additionally, the proposed approach proves that when we have to deal with a specific task (i.e., face detection), even if it is very complex, we can design and train smaller and efficient architectures that outperform deeper and larger networks in performance and in execution time.

4. Conclusion

In this paper, we presented a novel deep convolutional neural network for the task of face detection. Our experiments on publicly available benchmarks show the success of our method. Our detector is able to recognize faces in a wide range of orientations and expressions. Our detector does not require any extra modules usually used in deep learning methods such as SVM or bounding-box regression. Our work, extends the DDFD detector by using a light-weight model that improves run time and training speed. Our model combines outputs of two different networks trained for face detection and local facial parts, showing that the information provided by the latter is crucial for the specific task. It also outperforms the DDFD detector in the challenging FDDB dataset by a magnitude of 4%. We show that a properly trained smaller model is efficient and outperforms a more complex and large network used for the same task.

References

[1] S. S. Farfade, M. Saberian, and L.-J. Li. Multi-view face detection using deep convolutional neural networks. *ICMR*, 2015

[2] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 2004.

[3] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. *European Conference on Computer Vision*, 2014

[4] M. Viola and P. Viola. Fast multi-view face detection. textitProceedings of CVPR, 2003.

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

[6] S. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, H. Shu. Statistical learning of multi-view face detection, *Proceedings of ECCV*, 2002.

[7] B. Jun, I. Choi, D. Kim, Local transform features and hybridization for accurate face and human detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 2013

[8] L. Bourdev and J. Brandt. Robust object detection via soft cascade. *CVPR*, 2005

[9] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. *Proceedings of ECCV*, 2014

[10] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. *Proceedings of CVPR*, 2008.

[11] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, Cascade object detection with deformable part models. *Computer vision and pattern recognition*, 2010

[13] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. *European Conference on Computer Vision*,2014.

[14] R. Ranjan, V. M. Patel, and R. Chellappa. A deep pyramid deformable part model for face detection. *International Conference on Biometrics Theory, Applications and Systems*, 2015

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classi cation with deep convolutional neural networks. *Proceedings of NIPS*, 2012.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of CVPR*, 2014.

[17] P. M. R. Martin Koestinger, Paul Wohlhart and H. Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. *Proceedings of IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, 2011.

[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093, 2014.

[19] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. *Proceedings of ECCV*, 2014.

[20] V. Jain and E. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010.

[21] Shaoqing Ren Kaiming He, Xiangyu Zhang Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[22] Chen Change Loy Zhanpeng Zhang, Ping Luo Xiaoou Tang. Facial landmark detection by deep multi-task learning. *Proceedings of European Conference on Computer Vision (ECCV)*, 2014.

[23] X. Glorot Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. *International conference on artificial intelligence and statistics*, 2010.

[24] R. Ranjan, V. M. Patel, and R. Chellappa. A deep pyramid deformable part model for face detection. *International Conference on Biometrics Theory, Applications and Systems*, 2015

[25] S. Yang, P. Luo, C. C. Loy, and X. Tang. From facial parts responses to face detection: A deep learning approach. *IEEE International Conference on Computer Vision*, 2015.

[26]] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. *European Conference on Computer Vision*, 2014

[27] B. Yang, J. Yan, Z. Lei, and S. Z. Li. Convolutional channel features. *IEEE International Conference on Computer Vision*, 2015

[28] B. Yang, J. Yan, Z. Lei and S. Z. Li. Aggregate channel features for multi-view face detection. *International Joint Conference on Biometrics*, 2014.

[29] J. Li and Y. Zhang. Learning SURF cascade for fast and accurate object detection. *CVPR*, 2013.

[30] D. Triantafyllidou and A. Tefas. A Fast Deep Convolutional Neural Network for face detection in Big Visual Data. *INNS conference on Big Data, Thessaloniki*, 2016

[31] E. Marami and A. Tefas. Using Particle Swarm Optimization for Scaling and Rotation invariant Face Detection. *IEEE congress on Evolutionary Computation, World Congress on Computational Intelligence*, 2010

[32] E. Marami and A. Tefas. Face Detection using Particle Swarm Optimization and Support Vector Machines. *6th Hellenic Conference on Artificial Intelligence, Greece*, 2010

[33] C.Kotropoulos and I.Pitas. Rule-based face detection in frontal views. *Proc. of 1997 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'97), Germany*, 1997