Optimal Score Fusion via a Shallow Neural Network to Improve the Performance of Classical Open Source Face Detectors

Moumen T. El-Melegy, Hesham A. M. Haridi, Samia A. Ali and Mostafa A. Abdelrahman
Department of Electrical Engineering, Assiut University, Assiut, Egypt

Keywords: OpenCV, Dlib, Shallow Neural Network, Skin Detector, HOG-based Face Detector, Classical OpenCV Face Detector, CNN-based Face Detectors.

Abstract: Face detection exemplifies an essential stage in most of the applications that are interested in visual understanding of human faces. Recently, face detection witnesses a huge improvement in performance as a result of dependence on convolution neural networks. On the other hand, classical face detectors in many renowned open source libraries for computer vision like OpenCV and Dlib may suffer in performance, yet they are still used in many industrial applications. In this paper, we try to boost the performance of these classical detectors and suggest a fusion method to combine the face detectors in OpenCV and Dlib libraries. The OpenCV face detector using the frontal and profile models as well as the Dlib HOG-based face detector are run in parallel on the image of interest, followed by a skin detector that is used to detect skin regions on the detected faces. To figure out the aggregation method for these detectors in an optimal way, we employ a shallow neural network. Our approach is implemented and tested on the popular FDDB and WIDER face datasets, and it shows an improvement in the performance compared to the classical open source face detectors.

1 INTRODUCTION

Face detection is one of the most broadly explored topics in computer vision and pattern recognition, which represents the initial and vital stage of many application pipelines, such as: face verification (Tu et al., 2017), face tracking (Kim et al., 2008), face clustering (Cao et al., 2015), and face identification (Parkhi et al., 2015). From many literature surveys like (Yang et al., 2002; Zafeiriou et al., 2015), we observe that face detection has sighted considerable breakthroughs since the revival of deep learning once again in 2006 (Wang and Raj, 2017). Since that time, many well-established face detectors depending on that technique are provided in literature like CNN-based face detectors (Li et al., 2016; Hu and Ramanan, 2017; Tang et al., 2018). However, there are many industrial applications (Shaikh et al., 2016; Frejlichowski et al., 2016; Zheng et al., 2016; Puttemans et al., 2016a; Puttemans et al., 2016b) still utilize the classical detectors existing in OpenCV (Bradski, 2000) and Dlib (King, 2018) libraries. OpenCV library (Bradski, 2000) has a face detector that relies on the seminal work of (Viola and Jones, 2001) and depends on a cascade of classifiers using Haar-like features. As another example, Dlib library (King, 2018) also includes a face detector that counts on SVM as a classifier using HOG (Histogram Oriented of Gradient) features (Dalal and Triggs, 2005). These open source face detectors are unable to give a higher performance on the well-known public datasets like FDDB (Jain and Learned-Miller, 2010) and WIDER FACE (Yang et al., 2016) compared to CNN-based detectors. Several reasons created this situation (Yang et al., 2002; Zafeiriou et al., 2015), such as: These classical detectors work effectively in detecting frontal faces and fail at extreme in-plane and out-plane rotations. In addition, they lack of robustness in detecting faces under extreme lighting conditions. Moreover, these detectors tend to fail in discovering tiny and occluded faces. *Thus, any endeavors to enhance their performance will have an effective impact on the applications that count on them.*

There are many methods, such as RSFFD1 (Robust Score Fusion Face Detection) (Rara et al., 2010), RSFFD2 (El-Barkouky et al., 2012), and IterativeHardPositives+ (Puttemans et al., 2017), that try to boost the performance of classical OpenCV face detector. We follow the same direction trying to improve the performance of classical face detectors in

El-Melegy, M., Haridi, H., Ali, S. and Abdelrahman, M.
Optimal Score Fusion via a Shallow Neural Network to Improve the Performance of Classical Open Source Face Detectors.
DOI: 10.5220/0007691206600667
In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019)*, pages 660-667
ISBN: 978-989-758-354-4

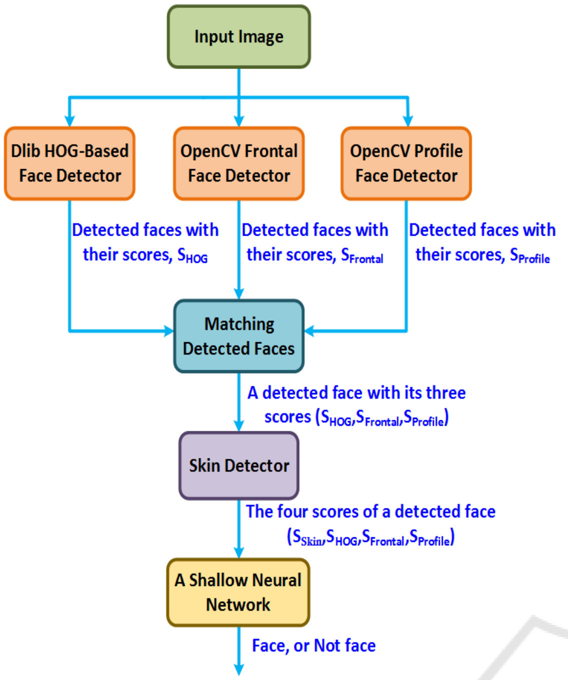


Figure 1: Our fusion method: In the matching block, the output consists of three scores for each detected face. If any detected face obtained by one detector has no matches with the other detected faces of the other detectors, the other scores of that face will be zeros.

OpenCV and Dlib libraries by:

- Running OpenCV Haar-based face detector on the image of interest using frontal and profile models.
- Running Dlib HOG-based face detector in parallel with the OpenCV detectors.
- Using a skin detector to detect skin regions in each face rectangle obtained from the above two points targeting to reduce the number of false positive faces (the obtained faces that are not truly faces) by taking into account skin color as an important feature for human faces.
- Employing a shallow neural network to learn the best method for aggregating the confidence scores of each face rectangle obtained from frontal, profile, HOG-based, and skin detectors, for more details see Figure 1.

The rest of this paper is organized as follows. Section 2 provides the related work achieved in the same direction of our target, while section 3 describes our approach for boosting the performance of classical open source face detectors. The experimental results are provided in section 4 followed by a conclusion and potential future work in section 5.

2 RELATED WORK

OpenCV (Bradski, 2000) and Dlib (King, 2018) libraries are the most renowned libraries employed in developing computer vision applications. They are updated from time to time with new algorithms from the community of academic researchers and industrial partners to help industrial users to accurately build their own working applications. (Viola and Jones, 2001) is one of the algorithms that is included in OpenCV and used extensively for object detection, especially face detection. They used AdaBoost algorithm to learn a cascade of classifiers to distinguish between faces and non-faces using Haar-Like features. The authors in (Dalal and Triggs, 2005) counted on HOG (Histogram of Oriented Gradients) features and SVM as a classifier. The accuracy of all these detectors still suffers when applied to public datasets, such as FDDB (Jain and Learned-Miller, 2010) and WIDER FACE (Yang et al., 2016). These recent public datasets have many challenges such as occlusion, illumination, and very tiny faces, however the open source face detectors are originally designed to detect frontal faces only. This situation motivates many researchers to improve the performance of these detectors.

The designers of Dlib library pursued the same direction, and they enhanced the face detector inspired from (Dalal and Triggs, 2005) by creating five HOG filters for the sliding window that is used to search about frontal and semi-frontal faces in an image, and they incorporate the updated detector in the library, but its accuracy still needs to go up. In (Li and Zhang, 2013), the authors adopted SURF (Speeded Up Robust Features) features (Bay et al., 2008) instead of using Haar-Like features of the original OpenCV face detector, and they used logistic regression to learn the best features that differentiate between faces and non-faces instead of using AdaBoost algorithm, aiming to raise the accuracy of the detector. In IterativeHardPositives+ face detector (Puttemans et al., 2017), the authors improved the negative training sample collection method, and they used an active learning scheme to iteratively append hard positive (positive rectangles categorized as negatives in the preceding iteration) and hard negative (negative rectangles labelled as positives in the former iteration) samples to the training process of the OpenCV detector. Also, they made a new annotation file for FDDB dataset, but despite their efforts, the accuracy of their detector became worse than the original OpenCV face detector but faster than it. As we can see, despite the attempts to boost the performance of open source face detectors, the accuracy of the detectors in (Bay et al., 2008; Li and Zhang, 2013; Puttemans et al., 2017; King, 2018) still suffers. In addition, some of them added more complexity and computational cost on the original detector such as the number of filters in (King, 2018). Furthermore, the detectors that depends on SURF or SIFT (Li and Zhang, 2013) features confront another problem because these feature descriptors are patent protected; they cannot be used for commercial purpose except with a permission from the original inventors.

There are other methods that follow the same direction, but they use the original simple building blocks that already exist in OpenCV library such as skin detection, Viola-Jones face detector, and Viola-Jones facial part detector, see for example RSFFD1 (Rara et al., 2010) and RSFFD2 (El-Barkouky et al., 2012). RSFFD1 was one of the best performers in the competition done by (Parris et al., 2011). RSFFD2 is a modified version of it, where saliency and skin information are added, and it consists of four-step pipeline. The first step is used to generate three scales for the image under interest. Then, OpenCV face detector is run on each scale, and each detected rectangle is assigned a score from 1 to 3 depending on how many times this rectangle appears in the three scales. In the second step, the same detector is applied on each detected rectangle from the first step to detect facial parts (two eyes and mouth), and each rectangle is given a score from 0 to 3 relying on how many parts found. The third step is used to run a skin detection algorithm on each rectangle given by the first step, and depending on how many skin pixels found, each rectangle is allocated a discrete score from 0 to 3. The fourth or the last step is employed to calculate the saliency map for each rectangle detected by the first step and depending on the saliency pixels acquired, each rectangle is assigned a discrete score from 0 to 3. At the end, each candidate face has 4 different scores one for each step. After that these scores are added up to give a value from 1 to 12 to each rectangle, where the higher score rectangles are more likely to be true faces counting on the threshold value that will be used.

3 OUR APPROACH

The techniques used in (Viola and Jones, 2001; ElBarkouky et al., 2012; Puttemans et al., 2017; King, 2018) suffer from some drawbacks, such as:

- They have a problem in detecting tiny, non-frontal, and occluded faces.
- The different scores of the information sources (RSFFD2 steps) in (El-Barkouky et al., 2012) are summed directly to give a final discrete score from 1 to 12, but other aggregation methods can provide better results.
- The running time of saliency map algorithm used in the fourth step in (El-Barkouky et al., 2012) is very long making the time of the entire pipeline exceeds 2 seconds on any device with limited hardware.
- The confidence score of OpenCV face detector in each scale in (El-Barkouky et al., 2012) is not taken into account. This information is very important which can be used to improve the confidence and the accuracy of the results.
- The new annotation file for FDDB dataset (Puttemans and Goedeme, 2017) used in (Puttemans et al., 2017) is not accurate, and we did not get a good reason for that from the authors when we emailed them.

Due to all the above limitations, each one from these techniques has an accuracy less than 80% (the maximum detected true faces to the total ground-truth faces). The basic idea of our approach is to deal with these limitations to increase the performance of detection. Our model consists of four steps, for more details see Figure 1. In the first step, we run in parallel on the image the three detectors (OpenCV frontal Haar-based, OpenCV profile Haar-based, and Dlib HOG-based face detectors). The output from this step is the detected faces with their scores. It is well-known that OpenCV has two types of face detector, one counts on Haar-Like features and the other relies on LBP (Local Binary Pattern) features, and the accuracy of the former outperforms the accuracy of the latter. This is the reason why we depend on Haar-based face detectors. The second step is to match the obtained faces of each detector with the others. The output from this step is the detected faces with three scores for each one without relying on the number of detectors that discover it. That is to say, if any detector fails to discover a specific detected face of the other detectors, its score will be zero for that face. The third step is to use the skin detector in (Brancati et al., 2016) to detect skin color in each obtained rectangle from the previous step. The score of this detector is represented by the ratio of the total skin pixels in each rectangle to its total area. In the fourth step, to calculate the final score of our model, Instead of summing the scores from various sources of information as in (El-Barkouky et al., 2012), our approach employs a shallow neural network to learn the optimal aggregation method for these scores. The network consists of three layers: input, hidden, and

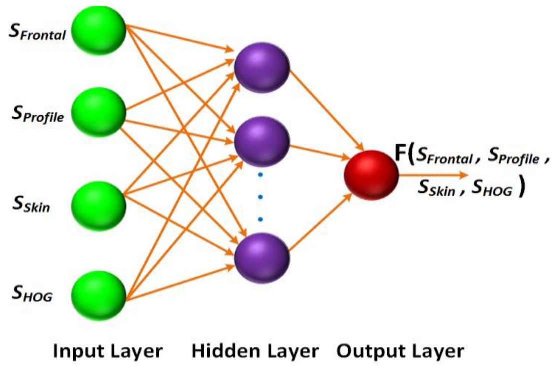


Figure 2: The proposed shallow neural network used to represent the function $F(S_{Frontal}, S_{Profile}, S_{Skin}, S_{HOG})$. The inputs are related to a candidate detected face and each one represents a score with a value in the range [0; 1]. $S_{Frontal}$ is the confidence score of a detected rectangle obtained by OpenCV face detector using Haar-based frontal model. $S_{Profile}$ is the score given by OpenCV face detector using Haar-based profile model. S_{Skin} is the score that represents the ratio of the total skin pixels in the detected rectangle to its total area, and S_{HOG} is the score of the same rectangle detected by Dlib HOG-Based face detector.

output layer, as shown in Figure 2. The input layer has four neurons: one for the confidence score coming from the frontal model ($S_{Frontal}$), the second for the confidence score of the profile model ($S_{Profile}$), the third for the score of skin detector (S_{Skin}, the ratio of the total skin pixels found in a detected rectangle to its total area), and the fourth for the confidence score coming from Dlib detector (S_{HOG}). The output layer consists of one neuron for the final score of our model, $F(S_{Frontal}, S_{Profile}, S_{Skin}, S_{HOG})$. The best number of hidden neurons is determined experimentally using cross-validation on a subset of WIDER FACE dataset, and it is found to be 10 neurons. In addition, the activation function of each neuron in the hidden and output layer is $logsig$ function; its values range from 0 to 1. Furthermore, our network is implemented using neural network toolbox of Matlab 2017a and trained using gradient descent with momentum. To prepare the training data for our network, we use WIDER FACE dataset. We use 12800 images from this dataset grouped as follows: 8000 for training, 2400 for validation and 2400 for testing. Each image may contain from a single face to more than 500 face. For training and validations we need to prepare good representative examples for faces and non-faces. For that sake, we rely on the concept of IOU (Intersection Over Union) (Jain and Learned-Miller, 2010) that is given by the equation:

$$IOU = \frac{area(d_i) \cap area(g_i)}{area(d_i) \cup area(g_i)} \quad (1)$$

Where d_i is the detected face rectangle and g_i is the ground-truth rectangle. This concept measures the overlapping between detected rectangles and ground-truth. Its values range from 0 to 1. In our model, any candidate face with IOU larger than 0.6 with any ground-truth face, its four scores will be considered a good positive example (a true face) with label 1, and anyone with IOU less than 0.4, its four scores are treated as a good negative example (a non-face) with label 0; anyone having IOU in the interval [0.4, 0.6] is ignored completely in training. So, by using these positive and negative examples with their labels, a neural network can learn the best aggregation method for these four scores to give one score for each detected rectangle. At the test stage, all we need to do is to apply the detectors of our approach on the image, then we feed the four scores of each detected rectangle to the trained shallow neural network to give one confidence score for it.

4 EXPERIMENTAL RESULTS

In this section, we evaluate the proposed method on WIDER FACE and FDDB datasets using several criteria in comparison with other methods.

4.1 Choosing the Best Detectors

To adopt the best detectors in our approach, the detectors that give a better accuracy (a larger number of true faces), a less number of false faces, and a less detection time, we study the effect of changing different detectors' parameters on the accuracy and detection time, for more details see Figure 3. For Dlib HOG-based face detector, it is favourable to run it on an image scale of 1.5 and a threshold value for its confidence score equals -1. That is to say, any detected rectangle has a score larger than or equals -1, the detector will consider it as a face. For the OpenCV frontal and profile detector, it is preferable to run them on an image of scale 1 and scale factor equals 1.1. The scale factor judges how the detector changes its sliding window size when searching an image for faces.

4.2 WIDER FACE Dataset

In this experiment, we use the test set of WIDER FACE (2400 images) that is formed as described before, and we adopt AUC (Area Under Curve)

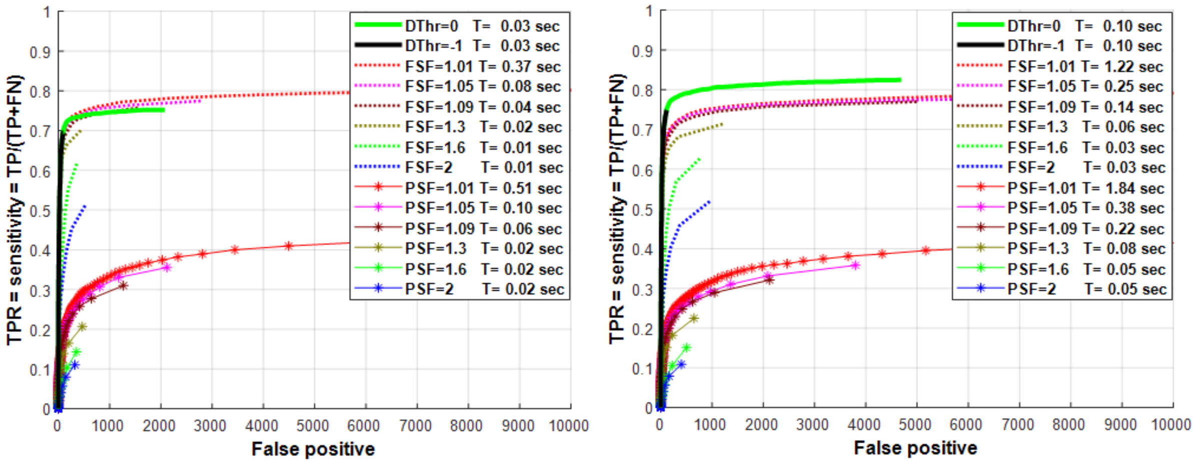


Figure 3: The effect of changing image scale, scale factors (FSF and PSF of OpenCV frontal and profile face detectors, respectively), and the threshold value (DThr) of the confidence score of HOG-based face detector on the performance and time of detection: Image scale 1 on the left and image scale 2 on the right.

concept for ROC curve (Sensitivity vs. 1-Specificity) as an assessment criterion. As shown in Figure 4, the AUC of our approach is indeed better than those of the RSFFD2 and OpenCV face detector.

4.3 FDDB Dataset

In this experiment, we evaluate our approach compared to OpenCV Haar-based face detectors with their default parameter values (image scale equals 1 and scale factor is 1.1), Dlib HOG-based face detector with its default parameter values (image scale equals 1 and the threshold value for its confidence score is 0), RSFFD2, and IterativeHardPositives+ on the complete FDDB dataset. Note, our shallow neural network is trained only on the WIDER FACE dataset. Figure 5 demonstrates an enhancement in performance of our model compared to the other detectors. Our model has a maximum true positive rate of 84.5% which exceeds the rates of the other detectors in the comparison. In addition, the number of true faces detected by our approach at any operating point selected on ROC curves exceeds its counterpart for the other detectors. All the algorithms included in the evaluation are implemented and tested using C++ running on Intel Core I7-6700k CPU, 4 GHz. One key aspect of the proposed approach is that the additional computational overhead is limited as demonstrated in Table 1. The table gives the detection time of five methods in comparison. Furthermore, we use the same evaluation criterion used by FDDB to examine the impact of our approach on the gap between OpenCV and CCN-based face detectors. As a result of their notability, we use Conv3D (Li et al., 2016) and HR-ER (Hu and Ramanan, 2017) as examples of CNN-based face detector. Figure 6 obviously shows that our technique indeed takes a notable step towards narrowing the gap between the two types of detectors.

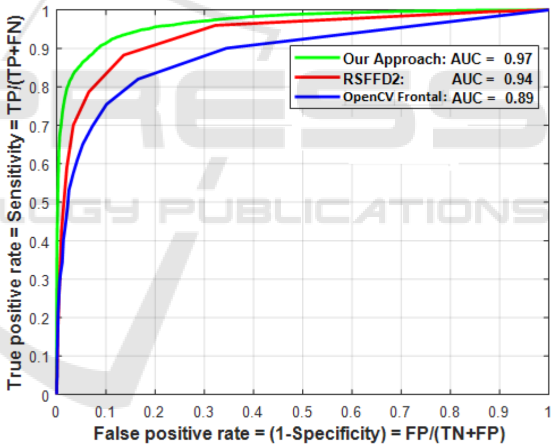


Figure 4: Comparison between our approach, RSFFD2, and OpenCV frontal face detector on a test set from WIDER FACE dataset using the AUC (area under the curve) with the highest AUC (0.97) for our approach.

4.4 Visual Comparison

Figure 7 offers some results for our model and OpenCV frontal face detector on some images from FDDB dataset. In all images, our model outperforms this detector via detecting the same candidate faces (green rectangles) in addition to new candidate faces (blue rectangles). Figure 8 presents some results for our approach and IterativeHardPositives+ face detector on some images from FDDB dataset. Although, the two detectors suffer from a few false

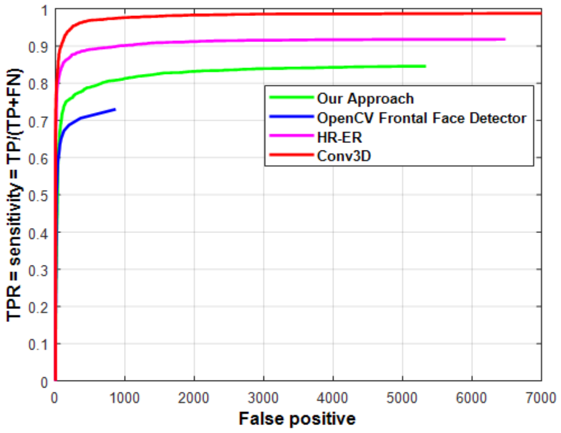


Figure 5: Comparison between our approach, RSFFD2, OpenCV face detector using frontal and profile models, and Dlib HOG-based face detector running on FDDB dataset using discrete score.

candidates (red rectangles), our approach beats this detector via ascertaining the same candidate faces (green rectangles) in addition to new candidate faces (blue rectangles).

Table 1: Timing results (in secs) of OpenCV face detectors, Dlib HOG-based detector, our model, and RSFFD2 running on FDDB dataset.

Method	Whole FDDB	Per Image
OpenCV Frontal	113.8	0.04
OpenCV Profile	125.2	0.04
HOG-Based	142.25	0.05
Our Approach	256.05	0.09
RSFFD2	6771.1	2.38

5 CONCLUSIONS

In this paper, we have proposed a fusion method to combine OpenCV and Dlib face detectors in one detector with the target of enhancing their performance. It is constructed from simple models already existing in OpenCV and Dlib libraries, such as: OpenCV frontal and profile face detector, skin detector, and Dlib HOG-based face detector. Furthermore, it employs a shallow neural network to optimally learn the best aggregation method to combine all these information sources. We have examined our approach on the FDDB and WIDER FACE datasets, and the results have shown that our adaptations have produced a reasonable increase in performance. We believe that our approach has taken a notable step towards narrowing the gap between classical open source and CNN-based face detectors, but we are not there yet. As a future task, it would be

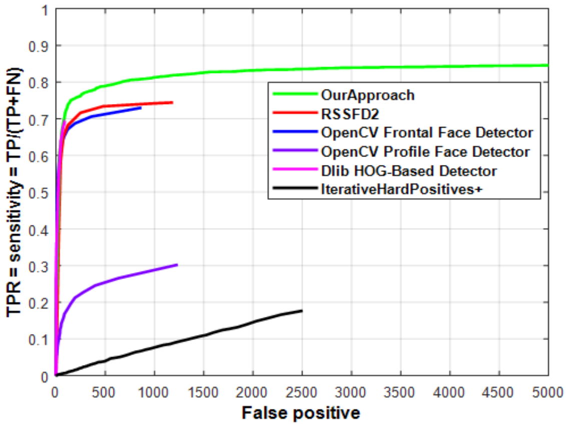


Figure 6: Comparison between our approach, OpenCV face detector, and deep learning-based approaches (Conv3D (Li et al., 2016) and HR-ER (Hu and Ramanan, 2017)) running on FDDB using discrete score.

interesting to tighten this gap even further. We think that there is still some room to increase the number of detected faces. Our model is only evaluated on frontal and profile faces, but it could be modified to detect faces with severe pose variations. Also, our model could be tested on the other state-of-the-art face detectors.

ACKNOWLEDGEMENTS

This research is supported by Information Technology Industry Development Agency of Egypt (grant # CFP130).

REFERENCES

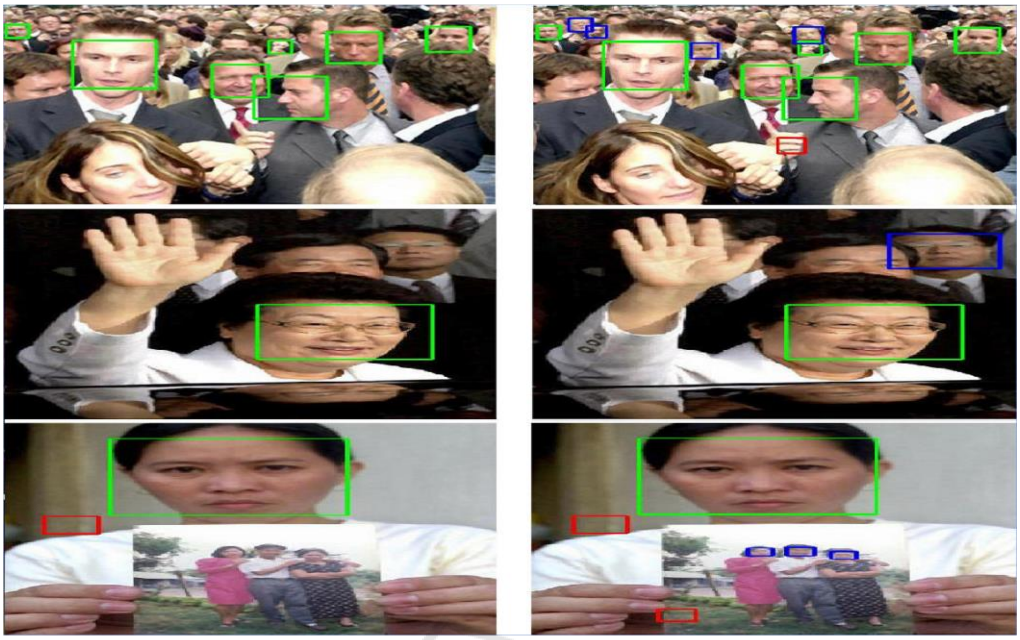
Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V. (2008). Speeded-up robust features (surf). In *Journal Computer Vision and Image Understanding*, pages 346 – 359. Science Direct.

Bradski, G. (2000). The opencv library. In *Doctor Dobbs Journal*, pages 120 – 126.

Brancati, N., Pietro, G. D., Frucci, M., and Gallo, L. (2016). Human skin detection through correlation rules between the ycb and ycr subspaces based on dynamic color clustering. In *Journal of Computer Vision and Image Understanding*. Science Direct.

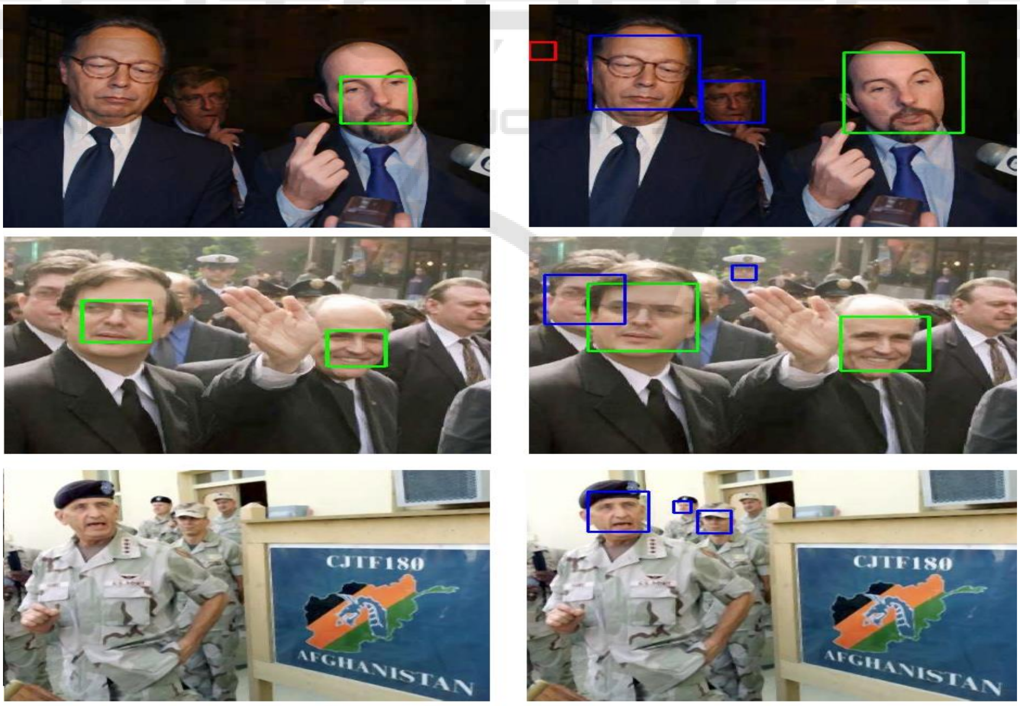
Cao, X., Wei, X., Han, Y., and Lin, D. (2015). Robust face clustering via tensor decomposition. In *IEEE Transactions on Cybernetics*, pages 2546 – 2557. IEEE.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *International Conference on Computer Vision and Pattern Recognition Workshops (CVPR)*, pages 886 – 893. IEEE.



a) b)

Figure 7: Detection results on FDDB: a) The output of OpenCV face detector, b) The output of our approach. Green rectangles exemplify true faces detected by the two algorithms. Blue rectangles symbolize the new true faces detected by our approach. Red rectangles represent false positive faces. In the images that contain no detected rectangles, the algorithm fails to detect any face at all.



a) b)

Figure 8: Detection results on FDDB: a) The output of IterativeHardPositives+ (Puttemans et al., 2017), b) The output of our approach. Green rectangles exemplify true faces detected by the two algorithms. Blue rectangles symbolize the new true faces

detected by our approach. Red rectangles represent false positive faces. In the images that contain no detected rectangles, the algorithm fails to detect any face at all.

El-Barkouky, A., Rara, H., Farag, A., , and Womble, P. (2012). Face detection at a distance using saliency maps. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 31 – 36. IEEE.

Frejlichowski, D., Gosciewska, K., Forczma, P., Nowosielski, A., and Hofman, R. (2016). Applying image features and adaboost classification for vehicle detection in the sm4public system. In *Image Processing and Communications Challenges*. Springer.

Hu, P. and Ramanan, D. (2017). Finding tiny faces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

Jain, V. and Learned-Miller, E. (2010). Fddb: a benchmark for face detection in unconstrained settings. In *technical report, University of Massachusetts Amherst*.

Kim, M., Kumar, S., Pavlovic, V., and Rowley, H. (2008). Face tracking and recognition with visual constraints in real-world videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1 – 8. IEEE.

King, D. (2018). In http://dlib.net.

Li, J. and Zhang, Y. (2013). Learning surf cascade for fast and accurate object detection. In *International Conference on Computer Vision and Pattern Recognition Workshops (CVPR)*, pages 3468 – 3475. IEEE.

Li, Y., Sun, B., Wu, T., and Wang, Y. (2016). Face detection with end-to-end integration of a convnet and a 3d model. In *European Conference on Computer Vision (ECCV)*. Springer.

Parkhi, O., Vedaldi, A., and Zisserman, A. (2015). Deep face recognition. In *Proceedings of the British Machine Vision Conference*. BMVA Press.

Parris, O., Wilber, M., Heflin, B., Rara, H., El-barkouky, A., Farag, A., Movellan, J., Castriln-Santana, M., Lorenzo-Navarro, J., Teli, M., Marcel, S., Atanasoaei, C., and Boult, T. (2011). Face and eye detection on hard datasets. In *IEEE IAPR International Joint Conference on Biometrics (IJCB)*, pages 1 – 10. IEEE.

Puttemans, S., Ergun, C., and Goedeme, T. (2017). Improving open source face detection by combining an adapted cascade classification pipeline and active learning. In *TEMPLATE'06, 1st International Conference on Template Production*. VISAPP.

Puttemans, S. and Goedeme, T. (2017). https://iiw.kuleuven.be/onderzoek/eavise/facedetectiondataset/home/.

Puttemans, S., Van, W., Ranst, and Goedeme, T. (2016a). Detection of photovoltaic installations in rgb aerial imaging: a comparative study. In *GEOBIA*. GEOBIA.

Puttemans, S., Vanbrabant, Y., Tits, L., and Goedeme, T. (2016b). Automated visual fruit detection for harvest estimation and robotic harvesting. In *International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE.

Rara, H., Farag, A., Elhabian, S., Ali, A., Miller, W., Starr, T., and Davis, T. (2010). Face recognition ata-distance using texture and sparse-stereo reconstruction. In *Fourth IEEE International Conference on Biometrics: Theory Applications*, pages 1 – 7. IEEE.

Shaikh, F., Sharma, A., P.Gupta, and Khan, D. (2016). A driver drowsiness detection system using cascaded adaboost. In *Imperial Journal of Interdisciplinary Research*. IJIR.

Tang, X., Du, D., He, Z., and Liu, J. (2018). Pyramidbox:a context-assisted single shot face detector. In *European Conference on Computer Vision*. IEEE.

Tu, C., Lin, M., and Hsiao, S. (2017). Subspace learning for face verification. In *ICASI, International Conference on Applied System Innovation*, pages 582 – 585. IEEE.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages I–511 – I–518. IEEE.

Wang, H. and Raj, B. (2017). On the origin of deep learning. In *arXiv: 1702.07800v4*. arXiv.

Yang, M., Kriegman, D., and Ahuja, N. (2002). Detecting faces in images: A survey. In *IEEE Transactions on Pattern Analysis and Machine Intelligences*, pages 34 – 58. IEEE.

Yang, S., Luo, P., Loy, C., and Tang, X. (2016). Wider face: a face detection benchmark. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 5525 – 5533. IEEE.

Zafeiriou, S., Zhang, C., and Zhang, Z. (2015). A survey on face detection in the wild: past, present and future. In *Journal of Computer Vision and Image Understanding*, pages 1 – 24. Science Direct.

Zheng, Y., Yang, C., Merkulov, A., and Bandari, M. (2016). Early breast cancer detection with digital mammograms using haar-like features and adaboost algorithm. In *Sensing and Analysis Technologies for Biomedical and Cognitive Applications*. SPIE.