ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal
Regular Issue, Vol. 8 N. 3 (2019), 5-12
eISSN: 2255-2863
DOI: http://dx.doi.org/10.14201/ADCAIJ201983512

An Experimental Performance Comparison of Widely Used Face Detection Tools

Abdullah Talha Kabakus

Computer Engineering, Faculty of Engineering, Duzce University, Duzce, Turkey, 81620
talhakabakus@gmail.com

KEYWORD

face detection; face localization; computer vision

ABSTRACT

Face detection is the task of detecting faces on photos, videos as well as the streaming data such as a webcam. Face detection, which is a specific type of general-purpose object detection, is a key prerequisite for many other artificial intelligence tasks such as face verification, face tagging and retrieval, and face tracking. In addition to that, convolutional neural nowadays, face detection is commonly used in daily routines such as social media, and network camera software of smartphones. As a result of this necessity, several face detection tools have been proposed. In this study, an experimental performance comparison of well-known face detection tools in terms of (1) accuracy, and (2) elapsed time of detection, which has become even more critical criteria especially when the face detection mechanism is utilized for a real-time system, is proposed. As a result of this experimental study, it is aimed that shed light on the much-concerned query "which face detection tool provides the best performance?". In addition to that, this study succeeds in showing that convolutional neural networks achieve great accuracy for face detection.

1. Introduction

Object detection is the task of detecting objects, such as cars, people, animals and objects used in daily routine, in pictures or videos. Face detection, which is a specific type of general-purpose object detection, can be defined as determining whether or not if the given image contains a face, and if it contains, detects the location of each face (Yang *et al*., 2002). Face detection is a key task since it is a prerequisite for similar tasks such as face locala ization (Lam and Yan, 1994; Moghaddam and Pentland, 1997), face parsing (Moghaddam and Pentland, 1997; Li *et al*., 2014), face verification (Ma *et al*., 2015; Ou *et al*., 2014), face tagging and retrieval (Gao and Qi, 2005; Wu *et al*., 2010), face tracking (Essa and Pentland, 2002; Donato *et al*., 1999), and facial expression recognition (Crowley and Berard, 1997; Darrell *et al*., 2000; Edwards *et al*., 1998). Additionally, face detection has become more widespread after it has been used within the digital cameras and smartphones in order to detect faces automatically before capturing photos. Similarly, popular social networks, such as *Facebook*, provide a built-in face detection for the uploaded photos in order to let users tag the detected faces. Face detection has been widely studied for a decade or so but unfortunately most of the early works were not able to provide sufficient performance in term of accuracy due to the significant variations in scale, blur, and expressions (Zhang *et al*., 2019;

Zafeiriou *et al*., 2015) as well as orientation, pose, presence or absence of facial features (e.g., beards, mustaches, and glasses), occlusion, and image conditions (e.g., the effect of lighting, and camera characteristics) (Yang *et al*., 2002) as Fig. 1 presents some examples of these variations from the *CelebA* (Liu *et al*., 2015) dataset.

Figure 1: Some examples of the variations of face photos from the CelebA dataset (Liu et al., 2015).

The recent works have benefitted the advantages of the latest software technologies as well as the rapid progress in processing power (e.g., CPU and GPU) and the storage capacity of the computers. Since face detection has been generally handled as a machine learning task, the more the data, more accurate results can be obtained. Thus, one of the most critical tasks to provide a high accuracy face detection mechanism is the construction of huge labeled datasets that contains faces with the aforementioned significant variations. To this end, some huge datasets have been constructed by researchers such as *PASCAL* (Everingham *et al*., 2010), *LFW* (Huang *et al*., 2007), *FDDB* (Jain and Learned-Miller, 2010), and *CelebA* (Liu *et al*., 2015). The more recent studies proposed more advanced techniques for face detection such as Support Vector Machines (SVM) (Osuna *et al*., 2002), neural network architectures (Krizhevsky *et al*., 2012), and Convolutional Neural Network (CNN) architectures (Zhang and Zhang, 2014; Chi *et al*., 2017; Li *et al*., 2015) which are a kind of forward propagation neural network with learnable weights and predictions, and primarily used on two-dimensional data such as images and videos (Kayikci, 2018). In addition to that, some widely used open-source tools such as *OpenCV* (ref, 2019) have been proposed. *Gunay and Ensari* (Gunay and Ensari, 2017) evaluated the performance of the face recognition algorithms namely k-Nearest Neighbors (k-NN), eigenfaces, Principle Component Analysis, and K-Means. According to the conducted experiments on a face dataset, they reported that k-NN and eigenfaces were found as the most successful algorithms. *Jung et al.* (Jung *et al*., 2018) evaluated the performance of four widely used face detection tools namely *Face++*, *IBM Bluemix Visual Recognition*, *AWS Rekognition*, and *Microsoft Azure Face API*. According to the experimental result which was conducted within that study, except than *IBM Bluemix Visual Recognition*, all tools were found capable of determining gender with accuracy rates greater than 90%. When it comes to inferring race, only one of the aforementioned four tools, *Face++*, was found capable. Inferring age was reported as a challenging task as all four tools were found incapable.

Within the scope of this study, three well-known face detection tools have been evaluated through the conducted experiments in order to shed light on the much-concerned query *"which face detection tool provides the best performance?"*. Additionally, a performance comparison schema in order to assess the performances of face detection tools is also proposed.

2. Material and Method

The performances of three well-known face detection tools namely *OpenCV* (ref, 2019), *YOLOFace* (Nguyen, 2019), and *face_recognition* (Geitgey, 2019) were evaluated through the conducted experiments which were implemented using the Python programming language. Each face detection tool is briefly described in the following subsections.

2.1. OpenCV

OpenCV is a widely used open-source image and video analysis tool which provides more than 2,500 optimized algorithms and has more than 40K users in the user group (Culjak *et al*., 2012). *OpenCV* was originally introduced by *Intel* which was implemented using C++ (Culjak *et al*., 2012). In order to conduct the experiments for *OpenCV*, the official Python wrapper for OpenCV namely *opencv-python* (Heinisuo, 2019) was utilized within the implemented Python script. *OpenCV* provides several built-in cascades which are used for specific tasks such as eye detection, hand detection, and smile detection. For the face detection, *OpenCV* provides a built-in haar feature-based cascade classifier namely *"haarcascade_frontalface_default.xml"* in order to detect faces in photos.

2.2. YOLOFace

YOLOFace is a *GitHub* repository that provides pre-trained *YOLOv3* weights file which was obtained after training the *WIDER FACE* dataset (Yang *et al*., 2016) on the state-of-the-art *YOLOv3* algorithm (Redmon and Farhadi, 2018). The *WIDER FACE* dataset consists of 393,703 labeled faces with a high degree of variability in scale, pose, and occlusion (Yang *et al*., 2016). Then, the *CNN* architecture proposed by the *YOLOv3* algorithm was constructed using this pre-trained *YOLOv3* weights file thanks to the utility functions of the *OpenCV* which are capable of reading neural networks from various widely used neural network libraries, such as *PyTorch*, *Tensorflow*, and *Caffe*, let developers construct the neural networks from pre-trained models.

2.3. face_recognition

face_recognition is an open-source Python library which wraps around the facial recognition functionality of the *dlib* library (King, 2019), a C++ library that is reported being achieved an accuracy of 99.38% on the *LFW* dataset (Momtahina *et al*., 2019). The model used within the face_recognition library is a *ResNet* network with 29 convolutional layers (King, 2017) which makes it a customized version of the *ResNet-34* network (Momtakhina *et al*., 2019; He *et al*., 2016).

3. Experimental Result and Discussion

The first 100K photos from the *CelebA* dataset, which consists of 202,599 face photos in the dimensions of 178x218, was selected in order to conduct experiments on the aforementioned face detection tools. The *CelebA* dataset consists of faces in significant variations which makes the dataset an ideal data source in order to reveal the performance differences of the face detection tools which reflect the performance in practice. Each face photo from the *CelebA* dataset was given as the input of the aforementioned tools, and the face detection result is stored in the CSV files. The detections for each tool are stored in the respective CSV file in the format of *"<image_id>,<x>,<y>,<width>,<width>"* in order to keep the record of each detection with the information of (1) the id of the analyzed photo (which is the file name of the photo), (2) the *x* location of the detection, (3) the *y* location of the detection, (4) the width of the detection, and (5) the height of the detection, respectively. If a tool was not able to detect any faces in the analyzed photo, the respective image was simply ignored, and as a natural consequence, no rows were added into the respective CSV file. When multiple faces were detected, a row was inserted per each detected face. Each photo in the *CelebA* dataset contains a single face. Hence, if a tool is not able to detect any faces, that means it misses the detection and is called as a *miss* in this paper. Similarly, if a tool detects faces more than one, that means at least one misdetection has occurred. This situation is called as *misdetection* in this paper. Both of *misses* and *misdetections* for each tool were calculated. An overview of the proposed architecture for conducting the experiments for each tool is presented in Fig. 2.

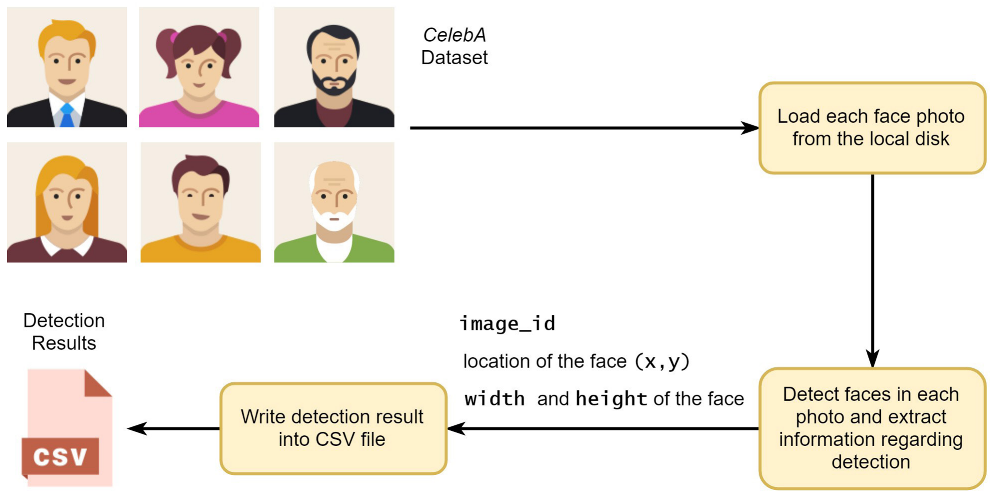

Figure 2: An overview of the proposed architecture for conducting the experiments.

According to the experimental result, that evaluates each tool by the number of *misses* and *misdetections* as it is presented in Fig. 3, while *YOLOFace* was found as the tool that provides at least misses, *face_recognition* was found as the tool provides the best performance in term of *misdetections*. *OpenCV* provides the worst performance in terms of *misses* and *misdetections* among others.

Figure 3: The performance comparison of the face detection tools in terms of misses and misdetections.

Let m and w denote the number of *misses*, and the number of *misdetections*, respectively, the accuracy of each tool (*acc*) was calculated by getting the percentage of the value calculated through subtracting the sum of the *misses* (*m*) and *misdetections* (*w*) from the total number of photos in the dataset (*100,000*) as the following equation (Eq. 1) states:

$$acc = \frac{100,000 - (m + w)}{100,000} * 100 \tag{1}$$

Since this problem is a *face localization* task which means each face photo contains a single face, the aim is the detection of that face (Yang *et al*., 2002). This makes the problem one class (*a face exists* or *no faces exist*) binary classification problem. When it comes to accuracy, as the experimental result is presented in Fig. 4, *YOLOFace* has achieved the best performance with an accuracy as high as 98.96% which proves the effectiveness of *CNN* architectures for face detection (Zhang and Zhang, 2014; Chi *et al*., 2017; Li *et al*., 2015; Mehta *et al*., 2018; Zheng *et al*., 2016).

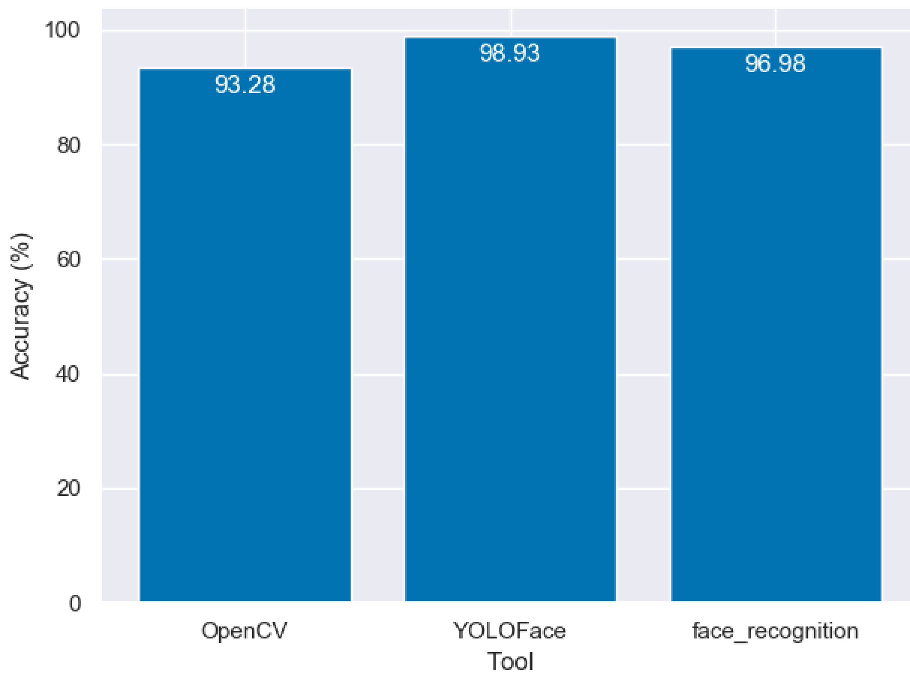

Figure 4: The performance comparison of the face detection tools in term of accuracy.

As it is mentioned in Introduction, face detection mechanism has already been integrated into the digital cameras and smartphones as well. As a natural consequence of this fact, the detection speed of the proposed face detection mechanism has become even more critical. Hence, the detection speeds of the three face detection tools have also investigated within the scope of this study. As the comparison of the elapsed time in minutes for analyzing the whole dataset (100K photos) is presented in Fig. 5, *OpenCV* was found as the fastest tool with completing the analysis of 100K photos in 37 minutes (which equals a photo analysis per 0.02 seconds) ahead of *face_recognition*, and *YOLOFace*, respectively. It is worth to mention that *YOLOFace* was found a way slower than the other two which is reasonable since *YOLOFace* utilizes the *YOLOv3* algorithm that is deeper than the network model of the algorithm *face_recognition* utilizes (*ResNet-34*).

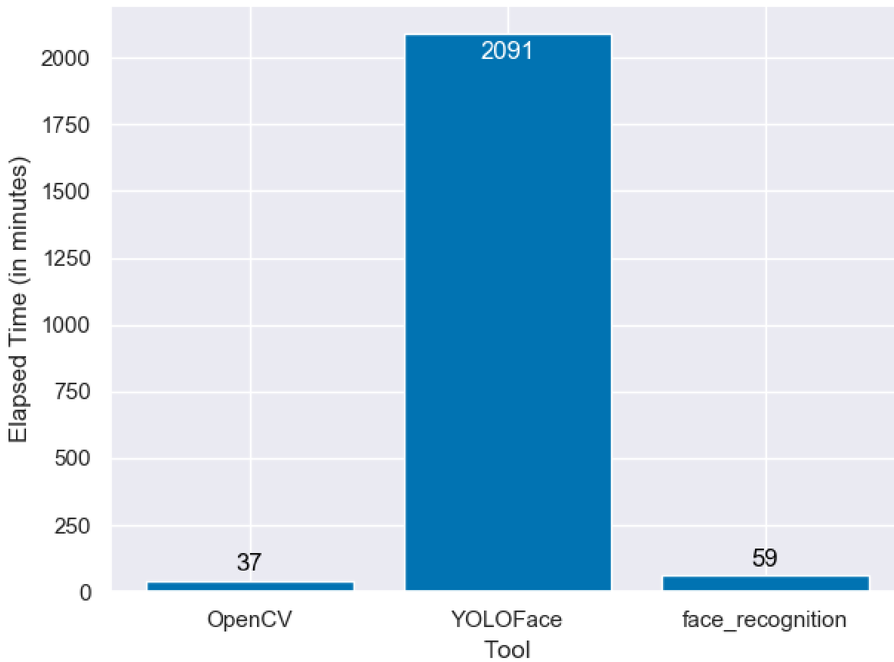

Figure 5: The performance comparison of the face detection tools in term of elapsed time in minutes.

4. Conclusions

Face detection has already become an important part of artificial intelligence due to it is commonly used in daily routines of people including but not limited to social media, digital cameras, and smart home automation. Several face detection tools, which are based on various face detection algorithms such as haar feature-based cascade classification algorithms, (deep) neural networks, and machine learning algorithms, have been proposed by researchers of computer vision. As a natural consequence, the performances of these tools in term of accuracy are needed to be shed light on. Hence, the performances of three well-known face detection tools, which utilize different face detection techniques, namely *OpenCV*, *YOLOFace*, and *face_recognition*, were evaluated through the conducted experiments within the scope of this study. The criteria in order to assess the performance were determined as (1) accuracy, and (2) elapsed time to complete the detection. According to the experimental result, *YOLOFace* was found as the best tool in term of accuracy with achieving an accuracy as high as 98.93% ahead of *face_recognition*, and *OpenCV*, respectively. This experimental result proves the effectiveness of *CNN* architectures for face detection. When it comes to elapsed time to complete the face detection task, *OpenCV* was found as the fastest tool ahead of *face_recognition*. *YOLOFace* was found as way slower than others.

As future work, more tools, which utilize various face detection techniques, will be included in order to propose a much more comprehensive analysis. In addition to that, more face photos will be included in variants in order to make the testbed to reflect the real world as much as possible.

5. References

Chi, L., Zhang, H., and Chen, M., 2017. End-To-End Face Detection and Recognition. *arXiv preprint*, 1703.10818:1-9.

Crowley, J. and Berard, F., 1997. Multi-modal tracking of faces for video communications. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 640-645. IEEE. doi:10.1109/cvpr.1997.609393.

Culjak, I., Abram, D., Pribanic, T., Dzapo, H., and Cifrek, M., 2012. A brief introduction to OpenCV. In *Proceedings of the 35th International Convention MIPRO (MIPRO 2012)*, pp. 1725-1730. IEEE. ISBN 9789532330724. 21-25 May 2012.

Darrell, T., Gordon, G., Harville, M., and Woodfill, J., 2000. Integrated person tracking using stereo, color, and pattern detection. *International Journal of Computer Vision*, 37(2):175-185. ISSN 09205691. doi: 10.1023/A:1008103604354.

Donato, G., Bartlett, M., Hager, J., Ekman, P., and Sejnowski, T., 1999. Classifying facial actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):974-989. ISSN 01628828. doi: 10.1109/34.799905.

Edwards, G. J., Taylor, C. J., and Cootes, T. F., 1998. Learning to identify and track faces in image sequences. In *Proceedings of Sixth International Conference on Computer Vision*, pp. 317-322. IEEE. ISBN 0818683449. doi:10.1109/AFGR.1998.670958. 7 Jan. 1998.

Essa, I. and Pentland, A., 2002. Facial expression recognition using a dynamic model and motion energy. In *Proceedings of 5th IEEE International Conference on Computer Vision*, pp. 360-367. IEEE. doi: 10.1109/iccv.1995.466916. 20-23 June 1995.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A., 2010. The PASCAL Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303-338. ISSN 09205691. doi:10.1007/s11263-009-0275-4.

Gao, Y. and Qi, Y., 2005. Robust visual similarity retrieval in single model face databases. *Pattern Recognition*, 38(7):1009-1020. ISSN 00313203. doi:10.1016/j.patcog.2004.12.006.

Geitgey, A., 2019. ageitgey/face_recognition: The world's simplest facial recognition api for Python and the command line.

Gunay, M. and Ensari, T., 2017. Comparison of face recognition algorithms. In *2017 25th Signal Processing and Communications Applications Conference (SIU)*, pp. 1-4. IEEE. ISBN 978-1-5090-6494-6. doi: 10.1109/SIU.2017.7960469. 15-18 May 2017.

He, K., Zhang, X., Ren, S., and Sun, J., 2016. Deep residual learning for image recognition. In *Proceedings of the 2016 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778. IEEE. ISBN 9781467388504. ISSN 10636919. doi:10.1109/CVPR.2016.90. 27-30 June 2016.

Heinisuo, O.-P., 2019. opencv-python.

Huang, G. B., Ramesh, M., Berg, T., and Learned-Miller, E., 2007. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*.

Jain, V. and Learned-Miller, E., 2010. *FDDB: A Benchmark for Face Detection in Unconstrained Settings*.
Jung, S.-G., An, J., Kwak, H., Salminen, J., and Jansen, B. J., 2018. Assessing the Accuracy of Four Popular Face Recognition Tools for Inferring Gender, Age, and Race. In *Proceedings of the Twelfth International AAAI Conference on Web and Social Media (ICWSM-18)*, pp. 624-627. AAAI Press. June 25, 2018 - June 28, 2018.

Kayikci, S., 2018. A Deep Learning Method for Passing Completely Automated Public Turing Test. In *3rd International Conference on Computer Science and Engineering (UBMK 2018)*, pp. 41-44. IEEE. ISBN 9781538678930. doi:10.1109/UBMK.2018.8566318. 20-23 Sept. 2018.

King, D., 2017. dlib C++ Library: High Quality Face Recognition with Deep Metric Learning. King, D., 2019. dlib C++ Library.

Krizhevsky, A., Sutskever, I., and Hinton, G. E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'12)*, pp. 1097-1105. December 03 - 06, 2012.

Lam, K.-M. and Yan, H., 1994. Fast algorithm for locating head boundaries. *Journal of Electronic Imaging*, 3(4):351-359. ISSN 1017-9909. doi:10.1117/12.183806.

Li, H., Lin, Z., Shen, X., Brandt, J., and Hua, G., 2015. A Convolutional Neural Network Cascade for Face Detection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 5325-5334. IEEE. ISBN 978-1-4673-6964-0. doi:10.1109/CVPR.2015.7299170. 7-12 June 2015.

Li, Y., Cai, C., Qiu, G., and Lam, K. M., 2014. Face hallucination based on sparse local-pixel structure. *Pattern Recognition*, 47(3):1261-1270. ISSN 00313203. doi:10.1016/j.patcog.2013.09.012.

Liu, Z., Luo, P., Wang, X., and Tang, X., 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV '15)*, pp. 3730-3738. IEEE. ISBN 9781467383912. ISSN 15505499. doi:10.1109/ICCV.2015.425. December 07 - 13, 2015.

Ma, J., Zhao, J., Ma, Y., and Tian, J., 2015. Non-rigid visible and infrared face registration via regularized Gaussian fields criterion. *Pattern Recognition*, 48(3):772-784. ISSN 00313203. doi:10.1016/j.patcog.2014.09.005.

Mehta, J., Ramnani, E., and Singh, S., 2018. Face Detection and Tagging Using Deep Learning. In *2nd International Conference on Computer, Communication, and Signal Processing: Special Focus on Technology and Innovation for Smart Environment (ICCCSP 2018)*, pp. 1-6. IEEE. ISBN 9781538611418. doi:10.1109/ICCCSP.2018.8452853. FEB 22-23, 2018.

Moghaddam, B. and Pentland, A., 1997. Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):696-710. ISSN 01628828. doi:10.1109/34.598227.

Momtahina, Hossain, R., Rahman, M. M., and Tania, O. A., 2019. *Image Capturing and Automatic Face Recognition*. Ph.D. thesis, United International University.

Nguyen, T., 2019. sthanhng/yoloface: Deep learning-based Face detection using the YOLOv3 algorithm.

Osuna, E., Freund, R., and Girosit, F., 2002. Training support vector machines: an application to face detection. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE. doi:10.1109/cvpr.1997.609310. 17-19 June 1997.

Ou, W., You, X., Tao, D., Zhang, P., Tang, Y., and Zhu, Z., 2014. Robust face recognition via occlusion dictionary learning. *Pattern Recognition*, 47(4):1559-1572. ISSN 00313203. doi:10.1016/j.patcog.2013.10.017.

Redmon, J. and Farhadi, A., 2018. YOLOv3: An Incremental Improvement. *arXiv*, pp. 1-6.

Wu, Z., Ke, Q., Sun, J., and Shum, H.-Y., 2010. Scalable Face Image Retrieval with Identity-Based Quantization and Multireference Reranking. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3469-3476. IEEE. ISSN 0162-8828. doi:10.1109/tpami.2011.111. 13-18 June 2010.

Yang, M. H., Kriegman, D. J., and Ahuja, N., 2002. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34-58. ISSN 01628828. doi:10.1109/34.982883.

Yang, S., Luo, P., Loy, C. C., and Tang, X., 2016. WIDER FACE: A face detection benchmark. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1-9. IEEE. ISBN 9781467388504. ISSN 10636919. doi:10.1109/CVPR.2016.596. June 27-30 2016.

Zafeiriou, S., Zhang, C., and Zhang, Z., 2015. A survey on face detection in the wild: Past, present and future. *Computer Vision and Image Understanding*, 138:1-24. ISSN 1090235X. doi:10.1016/j.cviu.2015.03.015. Zhang, C. and Zhang, Z., 2014. Improving multiview face detection with multi-task deep convolutional neural networks. In *2014 IEEE Winter Conference on Applications of Computer Vision (WACV 2014)*, pp. 1036-1041. IEEE. ISBN 9781479949854. doi:10.1109/WACV.2014.6835990. 24-26 March 2014.

Zhang, Y., Ding, M., Bai, Y., and Ghanem, B., 2019. Detecting small faces in the wild based on generative adversarial network and contextual information. *Pattern Recognition*, 97:74-86. ISSN 00313203. doi: 10.1016/j.patcog.2019.05.023.

Zheng, Y., Zhu, C., Luu, K., Bhagavatula, C., Le, T. H. N., and Savvides, M., 2016. Towards a Deep Learning Framework for Unconstrained Face Detection. *arXiv preprint*, 1612.05322:1-8.