Importance of different facial parts for face detection networks

Philipp Hofer, Michael Roland, Philipp Schwarz, Martin Schwaighofer, and René Mayrhofer
Johannes Kepler University Linz
Institute of Networks and Security, Institute of Computational Perception, CDL Digidow
Linz, Austria
Email: { philipp.hofer, michael.roland, martin.schwaighofer, rene.mayrhofer }@ins.jku.at, philipp.schwarz@jku.at

2021 IEEE International Workshop on Biometrics and Forensics (IWBF) | 978-1-7281-9556-8/20/$31.00 ©2021 IEEE | DOI: 10.1109/IWBF50991.2021.9465087

***Abstract*—Most state-of-the-art face detection algorithms are usually trained with full-face pictures, without any occlusions. The first novel contribution of this paper is an analysis of the accuracy of three off-the-shelf face detection algorithms (MTCNN, Retinaface, and DLIB) on occluded faces. In order to determine the importance of different facial parts, the face detection accuracy is evaluated in two settings: Firstly, we automatically modify the CFP dataset and remove different areas of each face: We overlay a grid over each face and remove one cell at a time. Similarly, we overlay a rectangle over the main landmarks of a face – eye(s), nose and mouth. Furthermore, we resemble a face mask by overlaying a rectangle starting from the bottom of the face. Secondly, we test the performance of the algorithms on people with real-world face masks. The second contribution of this paper is the discovery of a previously unknown behaviour of the widely used MTCNN face detection algorithm – if there is a face inside another face, MTCNN does not detect the larger face.**

***Index Terms*—importance of eye region, influence of face masks, MTCNN, Retinaface, DLIB, face detection performance on real-world face masks, face-in-face malfunction**

I. Introduction

Face recognition systems are widely used and are still gaining popularity. Moreover, in various situations people (have to) cover their faces, for example by wearing face masks due to the current Covid-19 outbreak or medical staff wearing protective clothing during their work. This poses the question if and how the performance of state-of-the-art face detection algorithms is impacted by these occlusions.

Additionally, knowing if certain facial parts influence the accuracy of state-of-the-art face detection models more-than-average can increase privacy because it allows individuals to take action and specifically cover these important areas.

In this work we differentiate between three distinct tasks related to face recognition:

1) *Face detection:* This type of network receives an image as input and detects the bounding box for each face.
2) *Face mask detection:* These networks are similar to *face detection algorithms*, with the major difference that instead of a single class, two classes are detected: faces wearing masks and faces not wearing them. The output are bounding boxes for both classes.
3) *Face recognition:* The input is a cropped and further pre-processed image of a single face. The goal of face recognition algorithms is to identify people. In order to be able to compare two yet unseen people, the output of the network is an *embedding*. This embedding is a vector of numbers and should be constructed with low inter-class and high intra-class similarity.

The training dataset plays an important role in current face detection and recognition tools and it largely influences its accuracy. Face recognition tools are trained with millions of face images (ArcFace [3] uses 5.8 million images, FaceNet [4] 200 million images). Datasets which are used to train current state-of-the-art face recognition tools do not mention the usage of images of people with face masks, and thus we suspect that only a small fraction of the training images contain a person wearing a face mask. To support this claim, we ran an off-the-shelve face mask detection algorithm [23] (with a threshold value of 0.95) on the VGGFace2 dataset [21], a large dataset (3,141,890 images), which is commonly used to train face detection and recognition networks. The face mask detection algorithm classified only 12,671 (0.40%) images as *person with facemask*. Through manual verification of these proposals, we found 12,616 false positives. Only 34 show people wearing a medical face mask and 21 show people with mouth and nose covered with fabrics. Thus, only 0.0018% (55/3,141,890) of all images in the dataset [21] depict people with face masks.

Because of this discrepancy of models not seeing masked faces while training and people wearing face masks in real-world settings, this paper analyzes the performance of three off-the-shelf face detection algorithms (MTCNN [5], Retinaface [6], and DLIB [7]) in this setting. MTCNN uses a three stage pipeline to exploit the inherent correlations between face detection and face alignment using deep convolutional neural networks [26]. In contrast to MTCNNs multi-stage pipeline, Retinaface is a robust single-stage face detector, which employs only light-weight backbone networks while still achieving state-of-the-art accuracy. In order to analyze a completely different architecture, we included DLIBs face detection algorithm, which uses histogram of oriented gradients (HOG) to detect faces. These networks have been trained with a negligible amount – if any – of faces with masks and are evaluated against images where parts of the face are occluded, e.g. by putting on a face mask. The main research question is how different facial parts influence the accuracy of state-of-the-art face detection networks.

The code used for modifying the dataset and performing the evaluation is available at https://github.com/mobilesec/occluded-facedetection-performance.

II. Related Work

As Damer et al. [8] stated, the *detection of occluded faces is a well-studied issue in the computer vision domain.* Especially since the Covid-19 outbreak, a lot of research results have appeared in this area.

In order to increase face detection performance on occluded faces, Zhang et al. [16] propose a hard image mining strategy. This results in more emphasis on hard samples, which models reality more closely. Furthermore, in an effort to detect partially occluded faces, Zeng et al. [17] introduced the *triplet loss* training strategy.

In order to objectively analyze current performance of face detection algorithms on masked faces and to increase the accuracy for future face recognition algorithms, new datasets with masked faces have been proposed [9]. However, these datasets are still in the early stages as they feature only a 4-digit number of faces. Thus, they are between 3 (w.r.t. MS1M [22]) and 4 (w.r.t. FaceNet) orders of magnitude smaller than current face detection datasets without masks.

Due to the current increase in popularity of face masks, literature tries to improve performance despite having large parts of the face covered [20]. This is an ongoing research activity. Many current popular face detection algorithms are not yet specifically trained on occluded faces. Therefore, this paper studies the performance of these popular face detection algorithms.

Every face recognition algorithm depends on a face detection algorithm [24]. Thus, if the face detection algorithm does not detect a face, any face recognition algorithm is rendered useless. A study for face recognition algorithms has been performed by NIST, where they published an evaluation of the performance of current state-of-the-art face recognition algorithms [13], without being fine-tuned for masks. This is a reasonable assumption since it holds for most of the currently used face recognition systems. NIST also plans to perform a similar experiment with algorithms specifically tuned to recognize people wearing masks [25]. Similar to this, our experiments evaluate the performance of popular face detection algorithms, consequently focusing on the preprocessing stage that images need to pass in order to even be considered for later face recognition.

III. Datasets

In this work we use two different datasets:

1) The Celebrities in Frontal-Profile (CFP) [12] data set, which consists of 500 individuals with 10 frontal images per person. The dataset also contains 4 additional profile images per person, which will not be used in this evaluation.
2) A real-world mask dataset [9] which has been crawled from various data sources. This collection contains 525 different people with 2203 masked faces.

In order to check which part of the face is most important for face detection (further discussed in Section IV), we modified the images of the first dataset [12] by excluding certain areas. There are two main strategies employed:

1) Overlaying a grid in various sizes over the face and blacking out one cell at a time. To be able to clearly see what modification has taken place, the resulting modifications for one randomly selected person are visualized in Fig. 1.
2) Removing landmarks of the face:
   - eye(s) (Fig. 2a, 2b, and 2c),
   - nose (Fig. 2e), or
   - mouth (Fig. 2d).

   In order to be able to objectively measure the impact of these landmarks on the face detection accuracy, for each of these settings we create another modification where the same amount of area is blacked out, on a random other part of the face (modifications ending with *-not*). Since the sizes of the images and therefore the faces vary significantly, the size of the blacked out area is proportional to the size of the face. The specific proportions have been empirically chosen such that the landmark is sufficiently removed. These values are easily retrievable in the published source code of this paper.

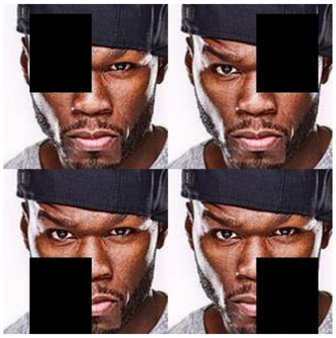
Manually modifying 5.000 images for all these settings is clearly not feasible. There are two requirements for creating these modifications:

1) *Background vs. face:* Even though all people are displayed in a portrait style where only one person is visible and takes up the majority of the space, the exact location of the face is not constant. For creating the modified versions *grid* and *grid-mask*, we do not want to distort the results by blackening out background pixels instead of pixels belonging to the face.
2) *Face landmarks:* In order to be able to remove face landmarks, we need to know their location.

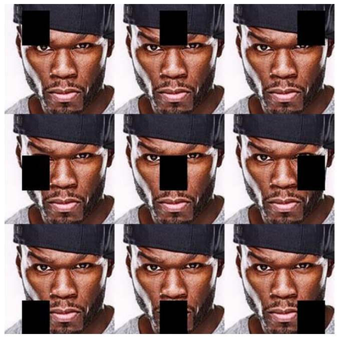
MTCNN returns the keypoints for the landmarks, and is therefore utilized in this work for automatically modifying the datasets. The ground truth for all experiments in this subsection are 4.978 faces, excluding 22 faces where MTCNN could not detect a face on the unmodified dataset. We defined the size of the rectangle through empirical experiments to properly cover the respective landmarks. For example, for removing the eyes, we chose the rectangles width to be 25 % and the height to be 15 % of the face width, as these values seem to properly cover the eyes in most instances. Even though we do not expect the specific values used in this paper to significantly impact the results, they are easily retrievable for every setting through the provided Github repository link.

IV. Experimental results

In order to verify which face regions are most important for face detection, we will check the accuracy of three state-of-

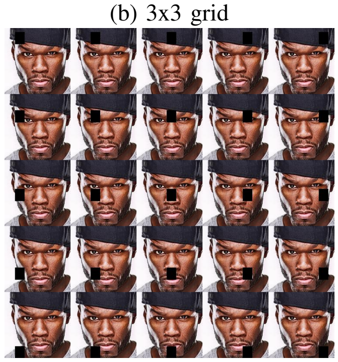


(a) 2x2 grid



(b) 3x3 grid

(c) 4x4 grid



(d) 5x5 grid

Fig. 1: Proposed modifications of the CFP dataset with respect to blacking out grid cells in various sizes.

(a) eyes

(b) eyeleft

(c) eyeright

(d) mouth

(e) nose

Fig. 2: Proposed modifications (landmarks-*) of the CFP dataset with respect to blacking out landmarks of the face.

Fig. 3: Proposed modifications (grid-mask-{00-15}) of the CFP dataset with respect to simulate a face mask.

TABLE I: Misclassification rates for grid-2 modification.

Area	Misclassification rate		
	MTCNN	Retinaface	DLIB
Top left corner	14.1%	9.96%	20.81%
Top right corner	18%	9.94%	28.65%
Bottom left corner	3.6%	4.3%	39.94%
Bottom right corner	9.78%	6.29%	63.1%

the-art face detection algorithms on computer-modified images and real-world images of people wearing face masks.

A. *Computer modified images from the CFP dataset*

We feed our dataset of modified images from the CFP dataset into MTCNN, Retinaface, and DLIB and analyze their accuracy.

1) Baseline: In order to be able to compare the performance of the face detection algorithms on differently modified CFP datasets, we first calculate the accuracy of the three analyzed algorithms on the dataset without any modification. In this work we are interested in correctly detecting the face. We are not differentiating between false positives (i.e. wrongly classifying part of the image as person) and false negatives (i.e. not detecting a person). They both count as *misclassification* and thus reduce the accuracy equally. From the 5.000 images, between 99.2% (DLIB, 4960 / 5000 images) and 99.74% (Retinaface, 4987 / 5000 images) of all visible humans are successfully detected.

B. *Grid*

The modifications are named after the amount of both horizontal and vertical cells.

a) Grid-2: In this setting, we blacked-out a quarter of the face. There is an interesting difference in accuracy between these quarters, as shown in Table I. These results suggest that the top half of the face is more important for face detection, as they have a higher misclassification rate. Interestingly, in all three face detection algorithms the bottom left corner has a significantly lower misclassification rate than the other 3 corners. This could be due to two facts:

1) The modified CFP dataset is biased and the bottom left quarter is not as informative as the remaining ones. Therefore the face detection algorithms (correctly) do not emphasis this part of the image.
2) The pre-trained face detection models are biased, e.g. by using a biased dataset for training.

In order to exclude the first possible explanation, we created another modification of the dataset by flipping the image vertically (*grid-flip-2*). If the first statement is true, we expect the misclassification rate to flip as well. Table II shows that this is not the case. The misclassification rate is still lowest if the bottom left quarter is blacked-out.

b) Grid-3: For all variations except for blacking-out the middle cell, all three algorithms perform pretty well:

1) MTCNN: 0.3% - 5.26% misclassification rate
2) Retinaface: 0.54% - 2.53% misclassification rate

TABLE II: Accuracy for the flipped image in the *grid-2* setting.

	MTCNN			Retinaface			DLIB		
	Correct	0 people	2 people	Correct	0 people	2 people	Correct	0 people	2 people
grid-flip-2/00	4303	674	2	4479	495	5	3951	1024	4
grid-flip-2/01	4139	836	4	4465	508	6	3616	1360	3
grid-flip-2/02	4758	218	3	4736	230	13	2899	2076	4
grid-flip-2/03	4571	406	2	4659	314	6	1727	3245	7

3) DLIB: 1.19% - 16.65% misclassification rate

Interestingly, the last case with a black middle cell achieves a significantly larger misclassification rate: 41.94% (MTCNN), 12.03% (Retinaface), and 55% (DLIB). This might be an indication that the nose might play an important role for face detection, which we will test in Section IV-B1c in more detail.

c) Grid-4 and Grid-5: These settings further indicate the importance of the nose, as every cell which "touches" the nose has a significantly higher misclassification rate.

1) Area around landmarks:

a) Eye region: The eye region is critical for face recognition [14]. Thus, it might also be of special importance for face detection. Therefore, as introduced in Section III we modified the CFP dataset, such that features around the eye region are removed.

MTCNN, Retinaface and DLIB achieve approximately the same accuracy (97.2%, 99.4%, and 98.4%, respectively) if the area around both eyes are removed.

If the eye region plays a more important role than other parts of the face, the amount of errors (false positives and false negatives) of face detection algorithms will be higher if compared to a dataset where rectangles with the same size are inserted on random positions (*eyes-both-not*). Our experiments clearly contradict this argument, as all three algorithms detect between 1.02 (Retinaface) and 7.33 percent points (DLIB) *more* faces if the rectangles are randomly located. This suggests that other parts of the face are more important for face detection accuracy. One possible explanation is that people in the *eyes-both* dataset look like they are wearing sunglasses, which face detection algorithms have already seen in the training phase.

Similar results are obtained if we occlude a single eye (datasets *eyes-{left—right}[-not]*).

b) Mouth: If we remove the mouth, we see similar results as with removing the eye region. Compared to the version where the *mouth* is covered, the face detection algorithms detect between 4.36 (Retinaface) and 14.81 (DLIB) percent points *more* faces if the rectangles are randomly distributed (*mouth-not*). Therefore, the mouth seems not to have a higher importance for face detection algorithms.

c) Nose: If we evaluate the face detection algorithms on images where the nose has been blacked-out, the algorithms achieve an accuracy of only 72.94% (MTCNN), 94.23% (Retinaface), and 58.62% (DLIB). If you remove a rectangle of similar size, accuracy increases to 98.03%, 98.65%, and 94.76%, respectively. One (partial) reason for this large difference (especially considering MTCNN and DLIB) might be that the nose is in the very center of the face.

Fig. 4: Misclassification results in percentage for simulated face mask modification.

In general, with an average accuracy of 97.4% Retinaface seems to handle occluded faces significantly better than MTCNN (91.6%) and DLIB (87.7%).

C. Mask

In this modification we simulated a face mask in various sizes. As expected, there is a positive correlation between the size of the face mask and the misclassification. The results are shown in Fig. 4.

D. Real world mask images

So far, we have only considered face occlusions which have been generated by a computer. In this subsection we will evaluate the performance on real world mask images. MTCNN and Retinaface both detected around 45% of the faces, DLIB only detected 3.4% of all faces (Table III). One possible reason for these low accuracy rates is the challenging dataset. Some people wear both a face mask and sunglasses, resulting in most of the face being occluded.

V. MTCNN FACE-IN-FACE MALFUNCTION

Since many state-of-the-art face recognition tools, such as ArcFace and SphereFace, recommend the use of MTCNN, we evaluated its performance on the real-world masked dataset [9]. As shown in Table III, face detection worked for 46% (1007/2203) of the images from the real-world mask dataset RMFD [9]. 15 randomly selected images where face detection did not work are shown in Fig. 5. In contrast, Fig. 6

TABLE III: Results of three face detection algorithms (MTCNN, Retinaface, and DLIB) on real-world mask dataset [9].

	MTCNN			**Retinaface**		**DLIB**	
	0 faces	2 faces	3 faces	0 faces	2 faces	0 faces	2 faces
RMFD [2203 images]	1196	1	1	1250	1	2129	1

Fig. 5: 15 exemplary images where MTCNN could not detect the person.

Fig. 6: 15 exemplary images where MTCNN could detect the person.

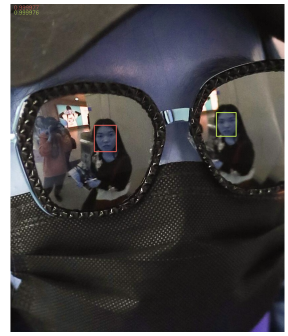


Fig. 7: MTCNN detects the reflected person in both lenses while missing the person wearing the eyeglasses.

(a) (b)

Fig. 8: MTCNN detects the original person (left-hand side in a) and b)). If there is another person inserted inside the head (right-hand side in a) and b)), the original person is not detected anymore.

shows 15 randomly selected images, where the face detection was successful.

After manually inspecting the cases where face detection did not work, we found an interesting behaviour of MTCNN. Fig. 7 shows a masked person wearing eyeglasses. MTCNN detects the reflected person in both lenses, while missing the person wearing the eyeglasses. This behaviour raises the question if MTCNN never detects a person if it has already detected a person in its subarea. Therefore, the following experiment has been conducted: Two images are manually constructed, each one featuring a person. Without any modification (left-hand side of both Fig. 8a and Fig. 8b) the person is detected. After inserting another image inside the (fore-)head (Fig. 8a) and inside the cheek (Fig. 8b)), MTCNN is not able to detect the original person anymore.

While this previously unknown behavior seems rather logical, it has a severe potential for abuse: face recognition relying on MTCNN for face detection can easily be evaded by smart placement of the image in a face, leading to the real face staying undiscovered and unrecognized. Furthermore, as shown in Fig. 7, MTCNN can be fooled if sunglasses reflect another face. This behaviour is particularly problematic since popular tools like ArcFace and SphereFace explicitly recommend the use of MTCNN.

VI. Conclusion

This paper analyzed the performance of three state-of-the-art face detection algorithms on occluded faces. Two different types of occlusions have been studied:

1) automatically modified versions of the CFP dataset, removing various parts of the face, and
2) real world images of people wearing masks.

The region around the nose plays an important role for face detection. Even though all three analyzed face detection algorithms achieve roughly the same accuracy on a dataset without occlusions, Retinaface outperforms both MTCNN and DLIB on most datasets where large parts of the face are missing.

Furthermore, this work found an interesting behaviour of the popular face detection algorithm MTCNN: If there is a face visible inside another face, the larger face will not be detected by MTCNN. This can significantly impact face recognition that relies on MTCNN for face detectio, such as state-of-the-art algorithms ArcFace and SphereFace.

Acknowledgment

This work has been carried out within the scope of Digidow, the Christian Doppler Laboratory for Private Digital Authentication in the Physical World, funded by the Christian Doppler Forschungsgesellschaft, 3 Banken IT GmbH, Kepler Universitätsklinikum GmbH, NXP Semiconductors Austria GmbH, and Österreichische Staatsdruckerei GmbH and has partially been supported by the LIT Secure and Correct Systems Lab funded by the State of Upper Austria.

References

[1] Zhang, Z., Shen, W., Qiao, S., Wang, Y., Wang, B., Yuille, A. (2020). Robust face detection via learning small faces on hard images. In The IEEE Winter Conference on Applications of Computer Vision (pp. 1361-1370). https://doi.org/10.1109/WACV45572.2020.9093445

[2] Fang, Z., Ren, J., Marshall, S., Zhao, H., Wang, Z., Huang, K., Xiao, B. (2020). Triple loss for hard face detection. Neurocomputing. https://doi.org/10.1016/j.neucom.2020.02.060

[3] Deng, J., Guo, J., Xue, N., Zafeiriou, S.: ArcFace: Additive Angular Margin Loss for Deep Face Recognition. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019). https://doi.org/10.1109/CVPR.2019.00482

[4] Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: A Unified Embedding for Face Recognition and Clustering. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015). https://doi.org/10.1109/CVPR.2015.7298682

[5] Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. IEEE Signal Processing Letters. 23, 1499–1503 (2016). https://doi.org/10.1109/LSP.2016.2603342.

[6] Deng, J., Guo, J., Ververas, E., Kotsia, I., Zafeiriou, S.: RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020). https://doi.org/10.1109/CVPR42600.2020.00525

[7] King, D.E.: Dlib-ml: A machine learning toolkit. The Journal of Machine Learning Research. 10, 1755–1758 (2009). https://dl.acm.org/doi/10.5555/1577069.1755843

[8] Damer, N., Grebe, J.H., Chen, C., Boutros, F., Kirchbuchner, F., Kuijper, A.: The Effect of Wearing a Mask on Face Recognition Performance: an Exploratory Study. arXiv:2007.13521 [cs]. (2020).

[9] Wang, Z., Wang, G., Huang, B., Xiong, Z., Hong, Q., Wu, H., Yi, P., Jiang, K., Wang, N., Pei, Y., Chen, H., Miao, Y., Huang, Z., Liang, J.: Masked Face Recognition Dataset and Application. arXiv:2003.09093 [cs]. (2020).

[10] Kaziakhmedov, E., Kireev, K., Melnikov, G., Pautov, M., Petiushko, A.: Real-world Attack on MTCNN Face Detection System. In: 2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON). pp. 0422–0427 (2019). https://doi.org/10.1109/SIBIRCON48586.2019.8958122.

[11] Komkov, S., Petiushko, A.: AdvHat: Real-world adversarial attack on ArcFace Face ID system. arXiv:1908.08705 [cs]. (2019).

[12] Sengupta, S., Chen, J.-C., Castillo, C., Patel, V.M., Chellappa, R., Jacobs, D.W.: Frontal to profile face verification in the wild. In: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1–9 (2016). https://doi.org/10.1109/WACV.2016.7477558.

[13] Ngan, M.L., Grother, P.J., Hanaoka, K.K.: Ongoing Face Recognition Vendor Test (FRVT) Part 6A: Face recognition accuracy with masks using pre- COVID-19 algorithms. (2020). https://doi.org/10.6028/NIST.IR.8311

[14] Saavedra, C., Smith, P., Peissig, J.: The Relative Role of Eyes, Eyebrows, and Eye Region in Face Recognition. Journal of Vision. 13, 410–410 (2013). https://doi.org/10.1167/13.9.410.

[15] Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: SphereFace: Deep Hypersphere Embedding for Face Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017). https://doi.org/10.1109/CVPR.2017.713

[16] Jin, Q., Mu, C., Tian, L., Ran, F.: A Region Generation based Model for Occluded Face Detection. Procedia Computer Science. 174, 454–462 (2020). https://doi.org/10.1016/j.procs.2020.06.114.

[17] Zeng, D., Veldhuis, R., Spreeuwers, L.: A survey of face recognition techniques under occlusion. arXiv:2006.11366 [cs]. (2020).

[18] Zheng, W., Gou, C., Wang, F.-Y.: A novel approach inspired by optic nerve characteristics for few-shot occluded face recognition. Neurocomputing. 376, 25–41 (2020). https://doi.org/10.1016/j.neucom.2019.09.045.

[19] Ge, S., Li, C., Zhao, S., Zeng, D.: Occluded Face Recognition in the Wild by Identity-Diversity Inpainting. IEEE Transactions on Circuits and Systems for Video Technology. 1–1 (2020). https://doi.org/10.1109/TCSVT.2020.2967754.

[20] Song, L., Gong, D., Li, Z., Liu, C., Liu, W.: Occlusion Robust Face Recognition Based on Mask Learning With Pairwise Differential Siamese Network. Proceedings of the IEEE/CVF International Conference on Computer Vision (2019). https://doi.org/10.1109/ICCV.2019.00086

[21] Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: VGGFace2: A Dataset for Recognising Faces across Pose and Age. In: 2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018). pp. 67–74 (2018). https://doi.org/10.1109/FG.2018.00020.

[22] Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J.: MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition. In: Leibe, B., Matas, J., Sebe, N., and Welling, M. (eds.) Computer Vision – ECCV 2016. pp. 87–102. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_6.

[23] AIZOOTech: https://github.com/AIZOOTech/FaceMaskDetection. (2020).

[24] Tolba, A.S., El-Baz, A.H., El-Harby, A.A.: Face recognition: A literature review. International Journal of Signal Processing. 2, 88–103 (2006). https://doi.org/10.5281/zenodo.1334652

[25] Ngan, M.L., Grother, P.J., Hanaoka, K.K.: Ongoing Face Recognition Vendor Test (FRVT) Part 6B: Face recognition accuracy with face masks using post-COVID-19 algorithms. (2020). https://doi.org/10.6028/NIST.IR.8331

[26] Zhao, Z.-Q., Zheng, P., Xu, S., Wu, X.: Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems. 30, 3212–3232 (2019). https://doi.org/10.1109/TNNLS.2018.2876865