

Article

How to Correctly Detect Face-Masks for COVID-19 from Visual Information?

Borut Batagelj ^{1,*} , Peter Peer ¹ , Vitomir Štruc ²  and Simon Dobrišek ² 

¹ Faculty of Computer and Information Science, University of Ljubljana, Večna Pot 113, SI-1000 Ljubljana, Slovenia; peter.peer@fri.uni-lj.si

² Faculty of Electrical Engineering, University of Ljubljana, Tržaška Cesta 25, SI-1000 Ljubljana, Slovenia; vitomir.struc@fe.uni-lj.si (V.Š.); simon.dobrisek@fe.uni-lj.si (S.D.)

* Correspondence: borut.batagelj@fri.uni-lj.si

Abstract: The new Coronavirus disease (COVID-19) has seriously affected the world. By the end of November 2020, the global number of new coronavirus cases had already exceeded 60 million and the number of deaths 1,410,378 according to information from the World Health Organization (WHO). To limit the spread of the disease, mandatory face-mask rules are now becoming common in public settings around the world. Additionally, many public service providers require customers to wear face-masks in accordance with predefined rules (e.g., covering both mouth and nose) when using public services. These developments inspired research into automatic (computer-vision-based) techniques for face-mask detection that can help monitor public behavior and contribute towards constraining the COVID-19 pandemic. Although existing research in this area resulted in efficient techniques for face-mask detection, these usually operate under the assumption that modern face detectors provide perfect detection performance (even for masked faces) and that the main goal of the techniques is to detect the presence of face-masks only. In this study, we revisit these common assumptions and explore the following research questions: (i) How well do existing face detectors perform with masked-face images? (ii) Is it possible to detect a proper (regulation-compliant) placement of facial masks? and (iii) How useful are existing face-mask detection techniques for monitoring applications during the COVID-19 pandemic? To answer these and related questions we conduct a comprehensive experimental evaluation of several recent face detectors for their performance with masked-face images. Furthermore, we investigate the usefulness of multiple off-the-shelf deep-learning models for recognizing correct face-mask placement. Finally, we design a complete pipeline for recognizing whether face-masks are worn correctly or not and compare the performance of the pipeline with standard face-mask detection models from the literature. To facilitate the study, we compile a large dataset of facial images from the publicly available MAFA and Wider Face datasets and annotate it with *compliant* and *non-compliant* labels. The *annotation dataset*, called Face-Mask-Label Dataset (FMLD), is made publicly available to the research community.

Keywords: COVID-19; masked-face detection; face-mask classification; face-mask recognition; COVID-19 compliant mask detection



Citation: Batagelj, B.; Peer, P.; Štruc, V.; Dobrišek, S. How to Correctly Detect Face-Masks for COVID-19 from Visual Information?. *Appl. Sci.* **2021**, *11*, 2070. <https://doi.org/10.3390/app11052070>

Academic Editor: Rubén Usamentiaga

Received: 28 January 2021

Accepted: 22 February 2021

Published: 26 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In December 2019, a new and worryingly contagious primary atypical (viral) pneumonia broke out in Wuhan, China. The new disease, called COVID-19, was later found to be caused by a previously unknown zoonotic coronavirus, named SARS-CoV-2. To help limit the spread of this new coronavirus, the World Health Organization (WHO), medical experts as well as governments across the world now recommend that people wear face-masks if they have respiratory symptoms, if they are taking care of the people with symptoms or otherwise engage frequently with larger groups of people [1–4]. In response to these developments, research in face-mask detection has attracted the attention of the computer vision community recently and initiated efforts towards developing automatic detection

models that can help society (through monitoring, screening and compliance-assessment applications) containing the COVID-19 pandemic [5].

Face-mask detection represents both a detection as well as a classification problem because it requires first the location of faces of people in digital images and then the decision of whether they are wearing a mask or not. The first part of this problem has been studied extensively in the computer vision literature, due to the broad applicability of face-detection technology [6]. The second part, on the other hand (i.e., predicting whether a face is masked or not), has only gained interest recently, in the context of the COVID-19 pandemic. Although a considerable amount of work has been done over the last year on this part [7–17], it typically only tries to detect whether a mask is present in the image. No special attention is given to whether the masks are properly placed on the face and are, hence, worn in accordance with the recommendations of medical experts. This limits the application value of existing face-mask detection techniques and warrants research into computer vision models capable of not only detecting the presence of facial mask in images, but also of determining if the masks are worn correctly.

To further illustrate this issue, several sample images from the MAFA (MAsked FAces) dataset [18] are shown in Figure 1. MAFA represents one of the most popular datasets for training face-mask detectors, and in a similar manner to most other datasets publicly available for this purpose, contains only binary labels indicating whether face-masks are present in the images or not. As illustrated in Figure 1 masked facial images in such datasets typically belong to one of two groups: (i) faces with correctly worn masks (marked green), and (ii) faces with incorrectly worn masks (marked red). Because the *correctness* (or *compliance* with recommendations) of the mask placement is not annotated, existing mask detectors commonly learn from highly noisy data (considering the intended purpose of the detectors) and commonly do not flag faces where a mask is present, but does not cover the nose, mouth and chin. We note that this is a common issue seen across most of the existing work on face-mask detection [16,19–21] and has important implications for the usefulness of the designed detectors in practice.



Figure 1. Example images from the MAFA dataset [18]. Similar to other publicly available datasets of masked faces, the MAFA dataset contains images where facial masks are present, but are not necessarily placed/worn in accordance with recommendations from medical experts—which is critical for containing pandemics, such as COVID-19. All the presented examples are labeled as *masked faces*, while only the ones marked green have correctly placed masks. In this work, we compile a dataset of masked faces, annotate it manually with respect to the placement of the mask and then build a computer vision model for the detection of properly worn face-masks. The figure is best viewed in color.

In this paper, we try to address this shortcoming and focus on the problem of detecting properly placed face-masks. Towards this end, we construct a large experimental dataset of facial images from two publicly available donor datasets, i.e., MAFA [18] and Wider Face [22]. We select representative and challenging images from the donor datasets and use a semi-automatic procedure to partition them into groups of images with correctly and incorrectly placed face-masks. Labels for gender, ethnicity and pose are also added. Because face-mask detectors typically include two computational steps (i.e., detection and classification), we first use the constructed dataset to evaluate the performance of several recent face detectors with masked-face images. To the best of our knowledge, this problem has not been studied extensively in the literature before. Next, we train multiple classification models for the task of predicting correct face-mask placements and evaluate them on our experimental dataset. Finally, we construct a complete recognition pipeline

from the best performing (off-the-shelf) detection and classification models and compare the combined pipeline with existing face-mask detection models.

The main contributions of this paper are summarized below:

- We compile a dataset of facial images for the problem of predicting correct face-masks placements in the context of the COVID-19 pandemic and annotate it with respect to placement—correctness, gender, ethnicity and pose. We make the list of images and labels publicly available in the form of a dataset of annotations (*annotation dataset* hereafter) called Face-Mask Label Dataset (FMLD). FMLD can be downloaded from: <https://github.com/borutb-fri/FMLD> (accessed on 25 February 2021).
- We investigate the performance of various face-detection models with masked as well as mask-free face images and study the impact of face-masks on existing face detectors. Although the impact of face-masks has recently been studied for the problem of face recognition, e.g., in [23,24], this problem has not been investigated widely for face detectors.
- Using off-the-shelf deep-learning models, we implement and train a highly accurate and effective pipeline for detecting properly worn face-mask and show that existing solutions from the literature advocated for this task are in fact designed for a different problem and have only limited use for the task studied in this work.

2. Related Work

A considerable amount of research has been done over the last year with respect to the problem of face-mask detection. In this section, we review some of this research to provide the necessary background for our work. Specifically, we discuss two distinct topics, existing datasets and existing techniques for face-mask detection.

2.1. Face-Mask Datasets

Prior to the COVID-19 pandemic, computer vision problems related to masked faces received only limited attention from the vision community. Instead, masked faces were typically studied within a broader problem domain related to processing and classifying occluded facial images [25]. As a result, few datasets with annotated masked faces were available publicly for research purposes.

However, driven largely by the COVID-19 pandemic, such datasets started to appear recently. Wang et al. [5], for example, introduced three datasets to improve the performance of face-recognition methods with masked faces. These include the Masked-Face Detection Dataset (MFDD), the Real-world Masked-Face Recognition Dataset (RMFRD) and the Simulated Masked-Face Recognition Dataset (SMFRD). According to the authors, MFDD contains 24,771 masked faces and is an extended version of the dataset initially presented in [20]. The images in MFDD were crawled from the Internet and, therefore, contain faces captured in challenging conditions and in real-world settings. The dataset is labeled according to whether people wear masks and is primarily intended for training and evaluating face-mask detectors. The RMFRD includes 5000 images of 525 people with masks, and 90,000 images of the same 525 subjects without masks. Images labeled as wearing masks contain faces with correctly worn masks, but also faces where the masks are not placed correctly and, hence, do not conform to the recommendations issued during the COVID-19 pandemic. The SMFRD dataset is a simulated dataset of masked-face images. It contains 500,000 face images of 10,000 subjects, where facial masks were added artificially to simulate masked faces. It needs to be noted that only a subset of the images in these datasets is publicly available for download, as summarized in Table 1.

Ge et al. [18] presented the Masked Faces (MAFA) dataset (the term *masked* is used here to indicate general types of occlusions, not necessarily masks), a (masked/occluded) face-detection dataset, collected from the Internet. MAFA contains 30,811 images with real-world appearance variability. Images in the dataset are labeled according to the degree and type of occlusion, but not with respect to whether facial masks are placed properly or not—see Figure 2 for an illustration. The MAFA dataset is also used often for research on

face-mask detection together with datasets of unmasked faces, e.g., [20,21], but due to the annotation problem mentioned above, this typically results in detection techniques capable of detecting general occlusions rather than facial masks.



Figure 2. Examples of the MAFA images labeled as masked faces in [20]. Such incorrectly labeled data typically leads to detection models that respond to a broad range of facial occlusions and not only face-masks. Additionally, such detectors have limited use in the context of monitoring applications for COVID-19.

Although the datasets presented above do not distinguish between masked faces with correctly and incorrectly worn masks, there are two smaller datasets on the Kaggle portal that do make the distinction [26,27]. The first, the Face-Mask Detection (FMD) dataset [26], contains 853 images that are divided into three classes: (i) with mask, (ii) without mask and (ii) mask worn incorrectly. A total of 4072 faces are annotated in these images (with mask: 3232, without mask: 717 and mask worn incorrectly: 123). However, the main problem with this dataset is that most of the faces are very small, which limits the use of the dataset for the development and evaluation of face-mask detectors. The second dataset from Kaggle [27] consists of 6024 images acquired in unconstrained settings. The dataset was constructed with particular attention to diversity, featuring people of different ethnicities, ages, and regions. Images are labeled with respect to 20 classes, including faces with mask, faces with masks incorrectly worn, faces without masks, covered faces and 16 other classes for masks, face shields, hats, scarves, and other accessories. Out of the 6024 images, 4326 are annotated and available on Kaggle for training. The remaining (sequestered) images were later annotated in the form of a standalone dataset called Medical Mask Dataset (MMD) [28]. The MMD dataset has a total of 9067 annotated faces with labels that are relevant for studying face-mask detection problems, i.e., 6758 faces with masks, 2085 faces without masks and 224 faces with incorrectly worn masks.

In contrast to the work reviewed above, we do not collect any images for training and evaluating face-mask detection models in this paper. Instead, we construct a challenging dataset for experimentation from the existing MAFA [18] and Wider Face [22] datasets (akin to [20]) and make the generated labels together with a list of images publicly available in the form of an annotation dataset, called Face-Mask Label Dataset (FMLD). A high-level comparison of the main characteristics of FMLD and the reviewed datasets is presented in Table 1.

Table 1. High-level comparison of the main characteristics of existing datasets with masked faces and the annotation dataset FMLD presented in this work. Please note that the number of reported images and annotations does not always correspond to the actual numbers publicly available online.

Dataset	Type [†]	Mask Type	#Images	#Faces	Available Labels (Reported Only)			
					#Mask	#No Mask	#Inc. Worn	Other [‡]
MFDD [5]	IM	Real-world	4343	<i>n/a</i>	24,771	0	0	<i>n/a</i>
RMFRD [5]	IM	Real-world	92,671	92,671	2203	90,000	0	<i>n/a</i>
SMFRD [5]	IM	Simulated	500,000	500,000	500,000	0	0	<i>n/a</i>
MAFA [18]	IM	Real-world	30,811	36,797	35,806	991	0	G, E, P
FMD [26]	IM	Real-world	853	4072	3232	717	123	<i>n/a</i>
MMD [27,28]	IM	Real-world	6024	9067	6758	2085	224	<i>n/a</i>
FMLD (ours)	AN	Real-world	41,934	63,072	29,532	32,012	1528	G, E, P

[†] IM—Image dataset, AN—Annotation dataset; [‡] G—Gender, E—Ethnicity, P—Pose; *n/a*—Information not available.

2.2. Mask-Detection Methods

Multiple solutions for detecting masked-face images have been presented in the literature or made available online over the last year. Motivated by the success of the RetinaFace face detector [29], for example, the so-called RetinaFace AntiCov model was introduced in [19]. The model uses a MobileNet backbone and can detect whether or not faces are masked in the detected results. Similar work using another backbone model was also presented by Khandelwal et al. in [30].

Jiang and Fan [21] proposed a one-stage face-detection model capable of classifying detected faces with respect to whether they are wearing masks or not. The proposed approach was again inspired by the RetinaNet model and represents a one-stage object detector that consists of a Feature Pyramid Network (FPN) and a novel context attention module. The model comprises a backbone, a neck, and a head. The main (high accuracy) model uses a ResNet backbone, but a simpler model with a MobileNet backbone is also explored. For the neck of the model (the intermediate connection between the backbone and the heads of the model), the authors use an FPN. For the heads, the proposed approach relies on a structure similar to that used in single-stage detectors (SSD). The model is tested on selected subsets from the MAFA and Wider Face datasets that consist of a total of 7959 images with masked and unmasked faces. Despite the impressive detection performance the proposed models do not distinguish between faces that wear masks properly (in accordance with recommendations) and faces that do not.

The closest to our work is the recent paper by Qin and Li [31]. Here, the authors describe an approach (SRCNet) for classifying face-mask wearing. The approach incorporates an image super resolution model that makes it possible to process low-resolution faces and a classification network that predicts whether faces are masked, without masks or if the masks are worn incorrectly. The model is trained and evaluated on a dataset that contained a total of 3835 images, which unfortunately is no longer available. Out of the 3835 images, 671 contain faces without masks, 134 images contain faces with incorrectly worn masks and 3030 images contain faces with correctly worn face-masks. An accuracy of 98.70% is reported for the proposed model. Although this work shares the basic problem statement, we do not focus solely on low-resolution faces, but explore the general task of detecting whether face-masks are worn correctly or not regardless of the data characteristics.

There are also some commercial solutions available that offer face-mask detection [32,33]. The face-mask detection feature offered by these solutions allow the simple plugging in of a video stream from a selected (surveillance) camera and then the use of vision techniques to monitor crowds and generate alerts when detecting people without masks. Baidu, on the other hand, offers an open-source tool to detect faces without masks [34]. The Baidu PaddleHub mask-detection model is based on PyramidBox, a context-assisted single-shot face detector published in 2018 [35]. The model is open-sourced to help advance the field and help to prevent and control the pandemic.

3. The Face-Mask Label Dataset (FMLD)

There is an imminent need for well defined, annotated and structured datasets that can be used for research with masked-face images. Although there have been some datasets made available for this task recently, many have limitations, come only with a small number of images or lack suitable annotations, as illustrated by the literature review presented above. In this section, we present an annotation dataset, called Face-Mask Label Dataset (FMLD) that tries to address these issues.

FMLD represents a dataset constructed from images of two donor datasets, i.e., MAFA [18] and Wider Face [22]. The MAFA dataset mainly serves as the source of images with correctly and incorrectly worn face-masks, whereas Wider Face serves as the main source of images of faces without masks. All images are annotated with labels indicating the presence of face-masks, the placement of face-masks (i.e., correct or incorrect), the gender of the subjects, their ethnicity and head pose. The list of images included in the dataset and the corresponding annotations allow for experimentation with face-mask detection models and a fine-grained analysis of the generated results using a standard set of images. The overall aim of the FMLD data (i.e., list of images and annotations) is, hence, to provide a predefined data with accompanying labels to train, test and compare face-mask detection models. Details on the FMLD data are given below.

3.1. Dataset Construction

The first step in the construction of the FMLD dataset is the selection of suitable images for annotation that are reasonably balanced across different categories (e.g., masked vs. not masked, male vs. female), and representative of real-world imaging conditions, where appearance variability across lighting, pose, image quality and other similar factors is expected. In a similar manner to [20], we identify the MAFA and Wider Face datasets as suitable donors and select images from these datasets for labeling.

- *Selection of images with masked faces:* Masked faces are taken from the MAFA dataset only. MAFA has a total of 30,811 images that contain 35,806 occluded faces and 991 unoccluded faces. Faces in the dataset come in various poses. At least one part of each face is typically occluded, mostly by some sort of mask. A few example images from the MAFA are shown in Table 2, including faces with different types and degrees of occlusion—these are labeled in the dataset. As can be seen, some faces are masked by hands or other objects rather than physical masks, and are, hence, not useful for studying face-mask detection problems. For FMLD, we first identify candidate images for the group with *correctly placed masks*. For this group, we consider faces that have an occlusion of type 1 or 2 and an occlusion degree of 3 (illustrated with the first two images in the last row of Table 2 with a green border). Such faces typically have masks that cover the mouth, nose and chin. For the group of *incorrectly worn masks*, we consider faces with occlusions of type 1 and an occlusion degree of 1 or 2. The rest of the MAFA faces are added to the group of faces without masks. The presented prefiltering generates a fairly clean list of candidate images for FMLD, but still results in several incorrectly classified faces. We, therefore, inspect all images manually and modify incorrect annotations. The main criterium for annotating faces as compliant was that the nose, mouth, and chin were covered, even if the occluding object was not necessarily a mask, but something akin to a scarf or buff. The complete statistics of the MAFA part of FMLD are given in Table 3.
- *Selection of face images without masks:* Because MAFA contains many faces with masks, additional images of faces without masks are needed to balance the classes. Thus, we include images from the Wider Face-detection benchmark [22] in FMLD. Specifically, 11,123 images are selected that contain 26,278 faces with a high degree of variability in scale, pose and occlusion. No bounding box ground-truth is available for the test images, so we only consider images from the training and validation sets of Wider Face. We select images with faces that are at least 40 pixels in size ($\min(\text{width}, \text{height}) > 40$) and exclude faces that are marked as invalid, are

heavily blurred, have heavy occlusions and atypical poses. We also omit the entire “30—Surgeons” group because most of the faces contain masks. Through this prefiltering we generate an initial candidate list of faces without masks that are then manually inspected for the presence of facial masks. The presented procedure results in the overall statistics presented in Table 3 for the Wider Face part of FMLD.

We partition the selected images into a training and testing set that can be used in standardized experiments to evaluate and compare face-mask detection techniques. A summary of the selection is again presented in Table 3.

Table 2. Examples of masked faces from the MAFA [18] dataset with different occlusion types and with different number of occluded face parts (i.e., degrees of occlusion).



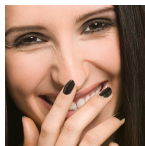






Degree	Occlusion Type (MAFA Labels)		
	1-Simple	2-Complex	3-Human Body
1			
			
			

Table 3. Overall FMLD statistics. Images annotated for FMLD were taken from the MAFA and Wider Face datasets and partitioned into three classes (correctly worn masks, incorrectly worn masks and without masks) and later equipped with additional labels.

Donor Dataset	Purpose	#Images	#Faces	Labels [†]		
				with Mask		without Mask
				Correctly Worn	Incorrectly Worn	
MAFA	Training	25,876	29,452	24,603	1204	3645
	Testing	4935	7342	4929	324	2089
Wider Face	Training	8906	20,932	0	0	20,932
	Testing	2217	5346	0	0	5346
FMLD	Training	34,782	50,384	24,603	1204	24,577
	Testing	7152	12,688	4929	324	7435
	Totals	41,934	63,072	29,532	1528	32,012

[†] Labels correspond to faces, not images.

3.2. Available Annotations

To make FMLD useful for experimentation with face-mask detection models and facilitate detailed analyses, we annotate all faces (that come with bounding boxes) with additional labels. Specifically, we add the following labels to all images considered in FMLD—in addition to the annotations related to face-mask presence and placement:

- *Gender information:* Each face bounding box is annotated with information on whether a male (6433 faces) or a female subject (6255 faces) is shown. The complete dataset is approximately gender balanced as shown in the corresponding pie chart in Figure 3.
- *Pose information:* Pose is annotated at a coarse level with respect whether faces are frontal or in profile view. As illustrated by the pie chart in the middle of Figure 3, most faces (9245) are frontal, and 3443 faces are shown in profile.
- *Ethnicity:* Three categories are considered to describe ethnic origin, i.e., Caucasian, Asian, and African. FMLD is dominated by faces of Asian origin (6962), followed by Caucasian (5131) and African (595) faces. The label distribution across ethnicity is shown in the last pie chart of Figure 3.

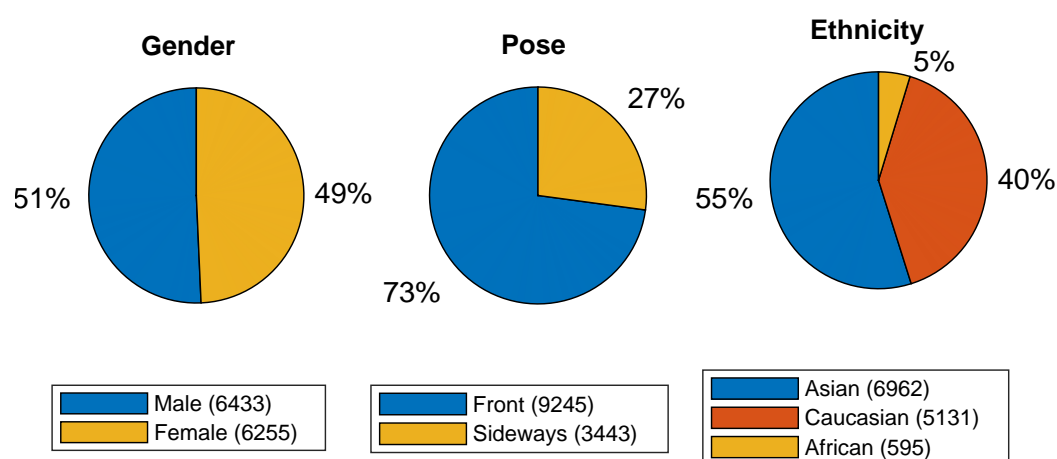


Figure 3. Available labels and the label distribution across the faces annotated for FMLD. As can be seen, the dataset contains labels for gender, pose and ethnicity in addition to the main labels indicating the presence of face-masks and their correct/incorrect placement. The figure is best viewed in color.

3.3. FMLD Availability

FMLD is publicly available from <https://github.com/borutb-fri/FMLD> (accessed on 25 February 2021). The dataset is distributed in the form of a single archive that contains: (i) a list of images from the MAFA dataset, (ii) a list of images from the Wider Face dataset, (iii) labels/annotations for all images, and (iv) an experimental protocol that partitioned the data into separate training (can be split further into training and validation) and testing splits. This information allows for reproducible research into face-mask detection techniques and comparisons between published models in the literature.

We used XML files in the PASCAL VOC file format for annotations. The annotation file for each image contains information about the name of the original image, its size and the source dataset. The annotated face in the image can be with mask (face_mask), without mask (face_nomask) and with mask worn incorrectly (face_incorrect). Each face object has bounding box information and labels for gender (male/female), ethnicity (Asian/Caucasian/African) and pose (frontal/profile).

4. Evaluation Methodology

For the experimental evaluation we construct a two-stage pipeline, as illustrated in Figure 4. The pipeline consists of a face detection and a face classification stage and allows us to explore the performance of each stage separately, but also the complete pipeline as a whole. A detailed description of each stage including the models considered and the performance metrics used is given in the following sections.

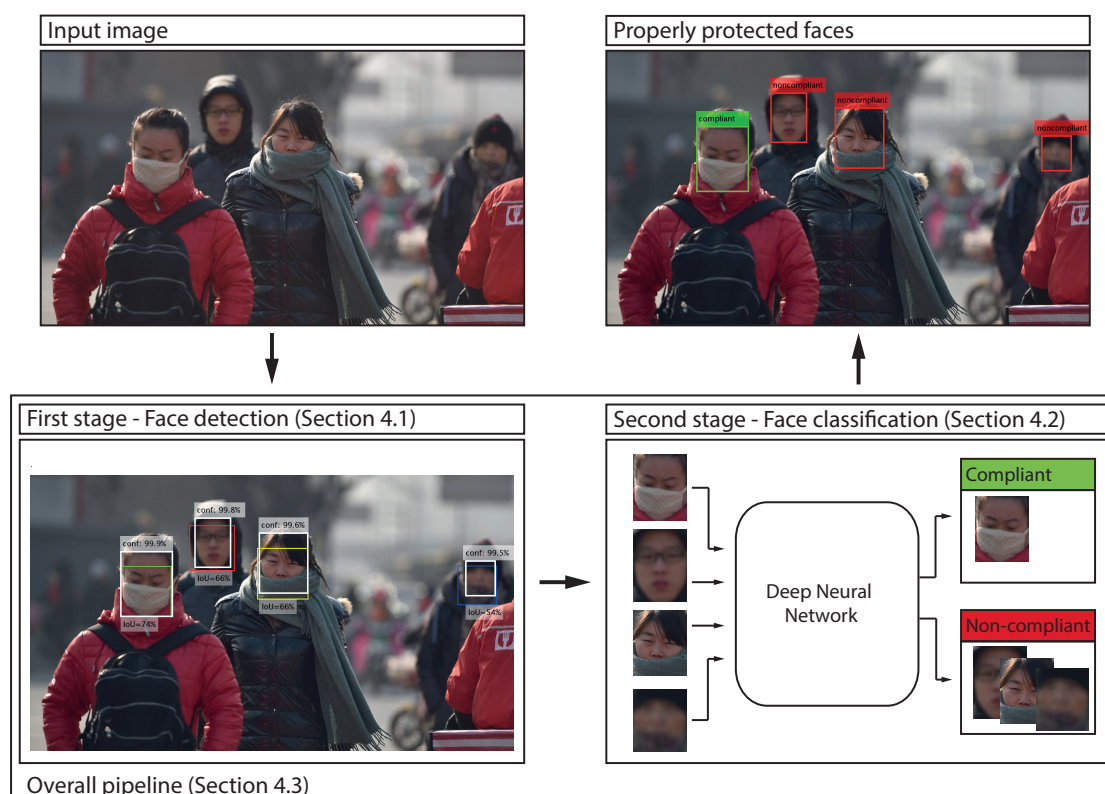


Figure 4. Overall pipeline designed for the evaluation procedure. A two-stage pipeline consisting of a face detection and a face classification stage is used. Unlike existing studies, we are not interested in whether face-masks are present in the images or not, but whether they are worn in a recommendation compliant manner.

4.1. First Stage—Face Detection

Detection models. Face detection has made significant progress over the last decade, mainly due to advances in deep-learning and convolutional neural networks (CNNs). As a result, most of the existing state-of-the-art (SOTA) face-detection techniques are CNN-based and are today able to efficiently detect faces with challenging characteristics and variability across pose, scale, illumination, from poor-quality data and in the presence of various other nuisance factors. Although face detection has been studied for faces with various occlusions, not much research has been done on investigating detection performance with masked faces specifically. In this paper, we explore this issue using the following (pretrained) models:

- Multi-Task Cascaded Convolutional Neural Network (MTCNN) [36]—MTCNN represents a carefully designed cascaded CNN-based framework for joint face detection and alignment. The detection process of MTCNN consists of three stages of convolutional networks that can predict face and landmark locations in a coarse-to-fine manner. A publicly available version of MTCNN is used for the experiments (<https://github.com/matlab-deep-learning/mtcnn-face-detection/releases/tag/v1.2.3>, accessed on 25 February 2021).
- The OpenCV single-shot detector (OCVSSD)—OCVSSD is a Single-Shot Detector (SSD) framework with a ResNet base network [37]. The detector is Caffe-based and has been part of OpenCV since OpenCV 3.3 (https://github.com/opencv/opencv/tree/master/samples/dnn/face_detector, accessed on 25 February 2021).
- The Dual Shot Face Detector (DSFD) [38]—The DSFD is an extension of the standard single-shot face detector (SSD) that improves upon SSD in terms of more descriptive feature maps, a more efficient learning objective (loss) and an improved strategy of matching predictions to the faces in the input images. We use improved Pytorch imple-

mentation of the original algorithm (<https://github.com/hukkelas/DSFD-Pytorch-Inference>, accessed on 25 February 2021).

- RetinaFace [29]—RetinaFace is a single-shot multi-level face localization method that performs pixel-wise face localization on various scales by taking advantage of multi-task learning. It jointly performs three different face localization tasks, i.e., face detection, 2D face alignment and 3D face reconstruction within a single framework (https://github.com/biubug6/Pytorch_Retinaface, accessed on 25 February 2021).
- BAIDU detector—The Baidu detector is based on PyramidBox: A context-assisted single-shot face detector [35] that is used as part of Baidu’s face-mask detector. PyramidBox implements multiple strategies to use context information to improve the face-detection results.
- AntiCov—The RetinaFace AntiCov is a customized one-stage face detector based on RetinaFace [29]. The model is publicly available as part of the *InsightFace: open-source 2D and 3D deep face analysis toolbox* (<https://github.com/deepinsight/insightface/tree/master/detection/RetinaFaceAntiCov>, accessed on 25 February 2021). It uses a RetinaFace MobileNet as the backbone network.
- The Mask detector by AIZooTech—The AIZoo face-mask detector is an open-source detector again based on the SSD framework [39] (<https://github.com/AIZOOTech/FaceMaskDetection>, accessed on 25 February 2021). It uses a lightweight backbone network to ensure low computational complexity.

The models were selected for our experiments because of their competitive detection performance and the fact that pretrained models are publicly available. Some of the models (Baidu and AIZooTech, for example) are also part of face-mask detection frameworks and, therefore, designed specifically for the studied detection problem.

Performance measures. To evaluate performance of the detection models we use *average precision* (AP), which is a standard performance measure also used in visible competitions, such as PASCAL VOC [40], ImageNet [41] and COCO [42]. When computing performance scores, we consider all detections with a confidence score greater than 0.02.

To match the detections with ground-truth annotations, we define a criterion that states: two bounding boxes represent the same face if their IoU (Intersection over Union) is equal or greater than a specific threshold (typically 0.5). This “match” is considered a true positive (TP) if that ground-truth annotation has not been already used (to avoid multiple detections of the same face). IoU is defined as:

$$IoU = \frac{A \cap B}{A \cup B} \quad (1)$$

where A is the area of the predicted bounding box and B is the area of the ground-truth bounding box.

A good way to characterize the performance of face detectors (also used in this paper) is to look at how precision and recall change with respect to changes of the confidence threshold. Using this procedure, we calculate precision/recall curves, which visualize the trade-off between the precision p and the recall r of the evaluated detection model. Here, precision tells us how many of the detected faces are relevant (i.e., present in the ground-truth) and is defined as:

$$p = \frac{TP}{TP + FP} \quad (2)$$

where TP stands for the number of true positives and FP stands for the number of false positives. Recall, on the other hand, tells us how many relevant faces are detected and is defined as:

$$r = \frac{TP}{TP + FN} \quad (3)$$

where FN stands for the number of false negatives.

AP is precision averaged across all recall values between 0 and 1 and represent an overall performance measure often used to characterize the performance of detection models. AP corresponds to the area under the precision—recall curve and is defined as:

$$AP = \int_0^1 p(r)dr, \quad (4)$$

where $p(r)$ is precision at recall r . To compute the above integral, we use in this paper the methodology from VOC2012 (Visual Object Classes Challenge 2012), which can be computed from the corresponding Matlab Development Kit or Cartucho's open-source Python implementation [43].

4.2. Second Stage—Face Classification

To determine whether face-masks are worn correctly or not, all faces detected in the first stage of our pipeline need to be classified into one of two classes: (i) compliant, and (ii) non-compliant, as illustrated in the third image of Figure 4. For the first class we consider faces, where the mask is placed properly and covers the nose, mouth and chin. For the second class, we consider faces that do not have masks at all or have incorrectly placed masks. Under this setup, we crop the facial regions from the input images, rescale them to a standard size (defined by the given model) and then subject them to a recognition procedure. Details on the considered recognition models are given next.

Classification models. For the second stage, we consider several recent CNN models that were shown to ensure competitive recognition performance in different problem domains [44–53]. Specifically, we use the following models in the experimental evaluation:

- AlexNet [54]—AlexNet is one the first large-scale convolutional neural networks that showed the potential of the deep-learning models ImageNet. The model consists of five conv layers of decreasing filter size followed by three fully connected layers upon which a learning objective can be defined [55]. One of the main characteristics of the model is the very rapid downsampling of the intermediate representations through strided convolutions and max-pooling layers.
- VGG-19 [56]—The VGG network is a 19-layer CNN model that uses a series of convolutional layers with small filters sizes 3×3 , to facilitate training. As with AlexNet, the last two layers of VGG-19 are again fully connected and used to define the learning objective for the model.
- ResNet [57]—The model is built around the residual CNN architecture and uses skip connections to allow for training of very deep CNN models. We use three ResNet variants with 18 (ResNet-18), 34 (ResNet-34), 50 (ResNet-50) and 101 (ResNet-101) layers. The skip connections allow for efficient training by ensuring appropriate gradient flow.
- SqueezeNet [58]—SqueezeNet is a convolutional neural network that employs design strategies to reduce the number of parameters, notably with the use of fire modules that “squeeze” parameters using 1×1 convolutions. In general, SqueezeNet is a special variant of ResNet, optimized for efficiency (in space and time). We used two version of the model for the experiments, the initial SqueezeNet v1.0 from [58], but also SqueezeNet v1.1 (https://github.com/forresti/SqueezeNet/tree/master/SqueezeNet_v1.1, accessed on 25 February 2021), which is an improved version that has $2.4 \times$ less computations and slightly fewer parameters than SqueezeNet v1.0.
- DenseNet [59]—DenseNet is a powerful recent CNN model that reuses features from preceding layers in all subsequent layers within a DenseNet block. This architecture makes it possible to learn highly descriptive features that are very different from competing models. Several dense blocks are then concatenated in a ResNet manner to allow for gradient flow and efficient training.
- GoogLeNet [60]—GoogLeNet is another type of CNN model based on the Inception architecture. It uses so-called Inception modules, which allow the network to analyze images at different scales within each block. GoogLeNet stacks these modules in a

feed forward architecture, with occasional max-pooling layers in between to reduce data dimensionality along the model.

- MobileNet [61]—MobileNet is a convolutional neural network architecture optimized for mobile devices. It is based on an inverted residual structure where the residual connections are between bottleneck layers. The intermediate expansion layer uses lightweight depthwise convolutions to filter features as a source of non-linearity. MobileNet version 2 is used in our experiments.

To ensure reproducibility, the models are taken from the torchvision module of PyTorch. Instead of learning the models from scratch, we use transfer learning and initialize the models with parameters learned on a subset of the ImageNet database [62], which is used in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC). We then fine-tune all models on the training part of the FMLD dataset using a standard cross-entropy loss. For the optimization algorithm, we use Stochastic Gradient Descent (SGD).

Performance measures. Classification accuracy is used to report performance for the evaluated classification models. It is defined as the ratio between the number of correct predictions and the total number of input samples. For binary problems, accuracy can be computed as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

where faces compliant to recommendations are considered the positive class. Classification accuracy is known to be sensitive to imbalanced classes. However, in our case this is not a problem as both classes (i.e., compliant vs. non-compliant) are well balanced.

4.3. Overall Pipeline

The experimental pipeline as a whole takes arbitrary images (or video frames) as input and classifies all detected faces as either compliant or non-compliant with respect to COVID-19 recommendations related to face-mask placement. Using the best performing models from the first and second stage, we implement a working recognition system for detecting compliant face-mask placement and then compare it to publicly available face-mask detection models.

To account for both detection and classification performance, we consider compliant and non-compliant faces as two separate classes (or objects) within a detection evaluation framework. Thus, for the complete pipeline we report performance in terms of the average precision (AP) for each of the two classes and to have a single performance measure for the pipeline in the form of the *mean average precision* (mAP).

5. Results and Discussion

In this section, we present the results of our experimental evaluation. We discuss results from three groups of experiments, i.e., (i) detection results with images of faces with masks, (ii) recognition results for classifying images with respect to face-mask placement (i.e., compliant or non-compliant), and (iii) result from the complete pipeline that includes both the detection as well as the classification task. The goal of these experiments is to provide insight into the performance of detection techniques with masked-face images and to evaluate performance of recognition models that in addition to the presence of face-masks also determine whether the masks are worn correctly or not.

5.1. Face-Detection Results

In the first series of experiments, we explore the performance of existing face detectors on images from the testing splits of the MAFA and Wider Face datasets—see Table 3. Modern face-detection models often return bounding boxes that tightly fit facial regions. The ground-truth annotations, on the other hand, are coarse and at times not consistently placed. As a result, correct detector results are also generated at lower value of IoU than the typical 50% cutoff. This is illustrated in Figure 5. Due to this characteristic, we report AP scores at two different IoU thresholds, i.e., at 40% (AP_{40}) and 50% (AP_{50}).

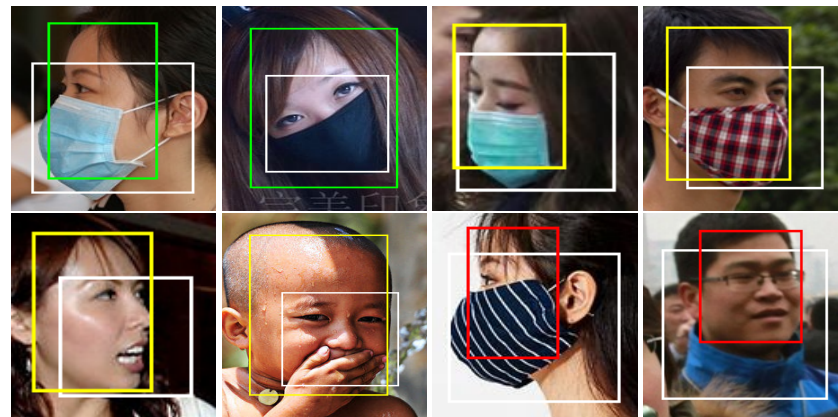


Figure 5. Results of the preliminary evaluation. Modern face detectors often return detection results (colored bounding boxes) that tightly fit the facial area, whereas the ground-truth is often only coarsely annotated (white bounding boxes). Results in this work are, therefore, reported for different IoU thresholds. The example images show detections with typical IoU threshold of 50% in green, between 50% and 40% in yellow, and below 40% in red. Please note that all results represent solid detections. The figure is best viewed in color.

We note that it is not possible to explore detection performance (for faces with and without masks) on the same set of images. We, therefore, compare performance across image populations with different characteristics. Specifically, we set up three test sets for the experiments: (i) the first contains 4935 images with masked faces from MAFA (images without masks are excluded), (ii) the second contains 2217 images from Wider Face and features only faces without masks, and (iii) the third contains the combined set of all the MAFA and Wider Face images. To have consistent sample sizes and to be able to estimate confidence measures for the reported scores, we use bootstrapping (i.e., random sampling with replacement) and randomly sample sets of 1000 images from all three test sets 100 times. From these we then compute average AP scores and the corresponding standard deviations. The results of this experiment are reported in Table 4 and visualized in Figures 6 and 7.

Table 4. Evaluation of detection models with masked-face images. Results are presented for three sets of images, i.e., MAFA—for faces with masks, Wider Face—for faces without masks, and the combined datasets for mixed faces. Reported are AP scores at IoU thresholds of 0.5 and 0.4.

Metric	$AP_{50}(\mu \pm \sigma) [\%]$			$AP_{40}(\mu \pm \sigma) [\%]$		
Dataset	MAFA	Wider	DB	MAFA	Wider	DB
MTCNN [63]	43.47 \pm 1.68	85.40 \pm 0.67	67.11 \pm 1.44	56.47 \pm 1.73	87.05 \pm 0.62	73.99 \pm 1.19
OCVSSD [64]	76.39 \pm 1.70	80.15 \pm 0.96	78.67 \pm 1.25	87.29 \pm 1.18	82.47 \pm 0.79	85.86 \pm 0.97
DSFD [65]	86.55 \pm 1.04	98.67 \pm 0.29	91.77 \pm 0.80	95.75 \pm 0.61	99.06 \pm 0.27	97.21 \pm 0.47
RetinaFace [66]	86.61 \pm 1.22	99.23 \pm 0.13	92.93 \pm 0.88	96.43 \pm 0.57	99.50 \pm 0.09	97.96 \pm 0.36
Baidu [34]	59.40 \pm 1.68	87.71 \pm 0.65	76.32 \pm 1.55	73.40 \pm 1.64	90.33 \pm 0.54	84.67 \pm 1.03
AntiCov [19]	80.55 \pm 1.50	95.98 \pm 0.40	88.91 \pm 0.90	92.49 \pm 0.83	96.73 \pm 0.32	95.29 \pm 0.52
AIZooTech [20]	85.44 \pm 1.30	87.77 \pm 0.66	84.72 \pm 1.10	95.06 \pm 0.73	91.45 \pm 0.56	93.25 \pm 0.75

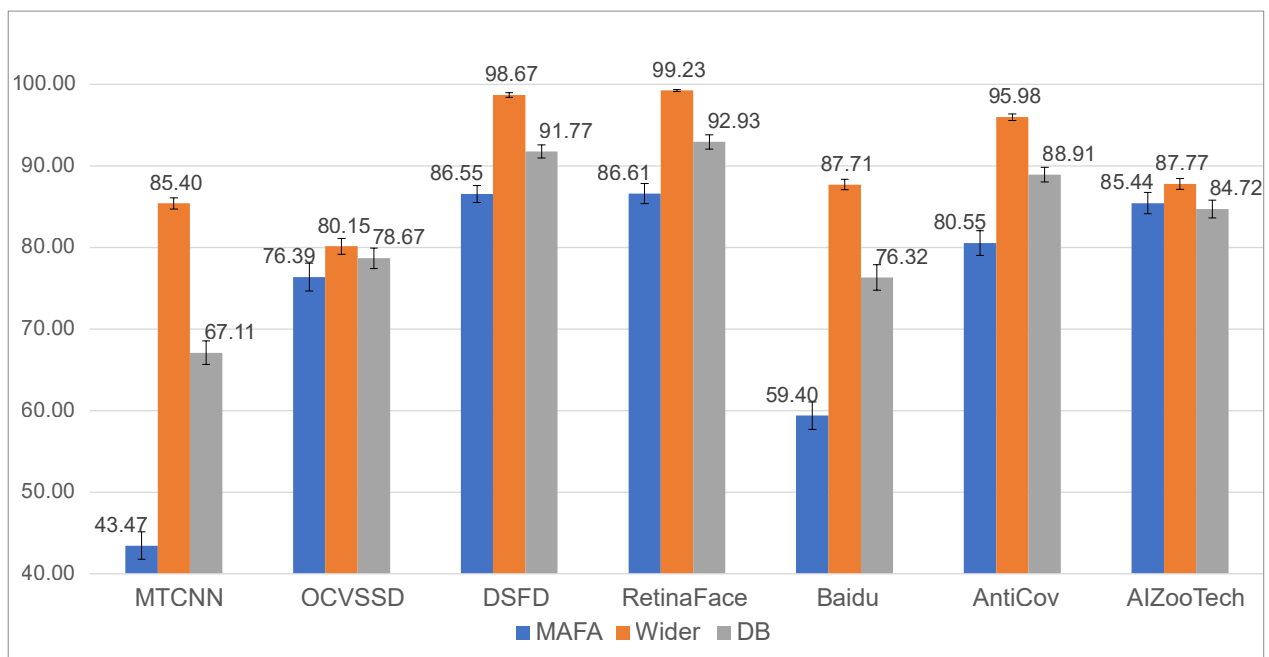


Figure 6. Average precision (AP_{50}) results for different face-detection models for three sets of test images, i.e., MAFA—only faces with masks, Wider Face—face without masks (Wider), and the combined dataset (DB)—faces with and without masks.

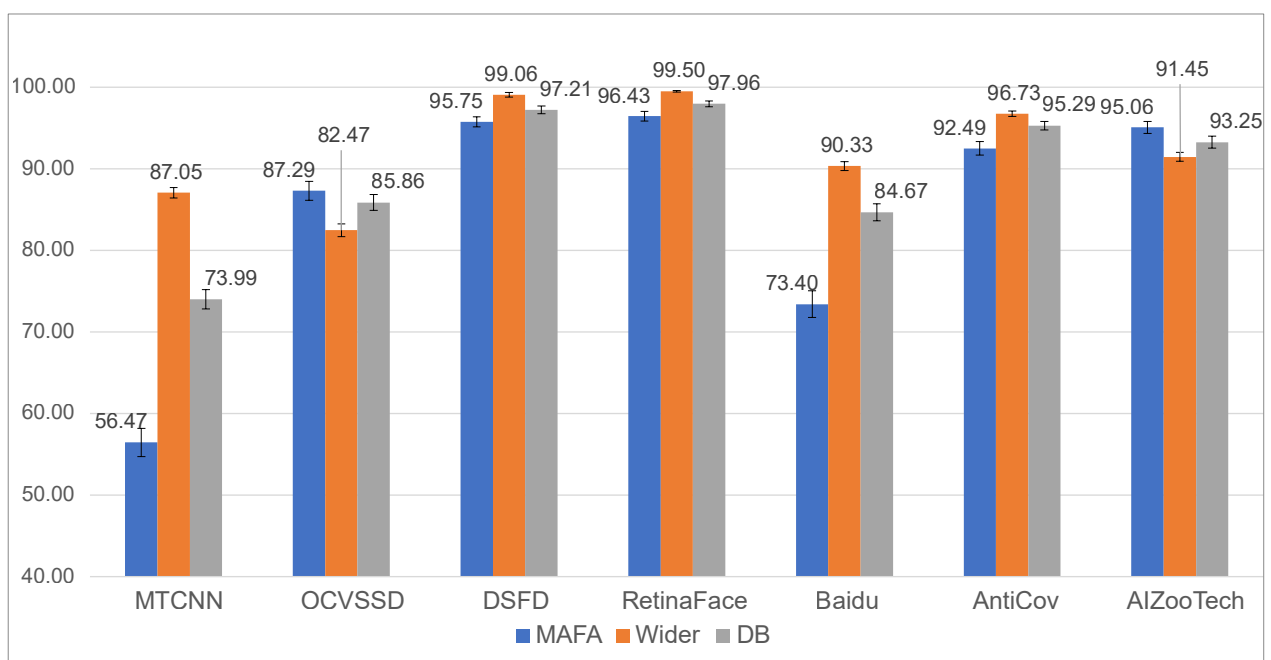


Figure 7. Average precision (AP_{40}) results for different face-detection models for three sets of test images, i.e., MAFA—only faces with masks, Wider Face—face without masks (Wider), and the combined dataset (DB)—faces with and without masks.

In general, masked faces are still a challenge for most detectors. On average, the performance for masked faces is 73%, while for unmasked faces the detectors achieve an average precision over 90%. Among the most successful are the DSFD and RetinaFace detectors, which have an efficiency of around 99% for mask-free and over 85% for masked faces at an IoU threshold of 50% (Figure 6). This further improves if an overlap of 40% is considered. In this setting the difference in performance between masked and mask-free is only around 3% for DSFD and RetinaFace. As expected, the performance on the combined dataset (marked DB) is between masked and mask-free face for most methods.

Among the face-detection techniques taken from the mask-detection models (Baidu, AntiCov and AIZooTech) the most successful is AntiCov with an average precision of around 88% at an overlap of 50% on DB and an AP score of around 95% at a 40% overlap. Interestingly, though, this model exhibits a significant gap between the performance on masked and mask-free faces, i.e., appx. 15% at an overlap of 50%. This suggests that considerable performance variations can be expected in real-world applications, where the characteristics of the input images cannot be controlled. An even bigger performance gap can be observed for the Baidu detector. The most stable detector is the AIZooTech detector, which has the smallest performance gap between masked and mask-free faces at 50% overlap and even performs better with masked faces at an overlap of 40%.

Overall, the best among the evaluated detectors was RetinaFace [29], which on the entire dataset (DB) achieved a mean accuracy of $AP = 92.93\%$ for a 50% overlap with the ground-truth. The model also achieved a competitive AP_{50} score of 86.61% with masked face. The performance and robustness of the model is likely a consequence of the multi-task nature of RetinaFace each of the face-detection candidates is also reconstructed in 3D, making the model highly robust to partial occlusions.

When manually reviewing errors of the best detector (RetinaFace), we observe that many detections are flagged as type I errors because of low IoU scores—caused by the ground-truth annotations. A few examples of such detection are shown with red bounding boxes in Figure 5. Even though a considerable number of additional face annotations have been generated for FMLD, the best detectors still (correctly) detect many faces that are not marked in the annotation files and are, therefore, flagged as false positives. Most type II errors (false negatives) are caused by excessive occlusions, as shown in the first row of Figure 8, and profile views of poor quality, as illustrated in the second row of Figure 8.

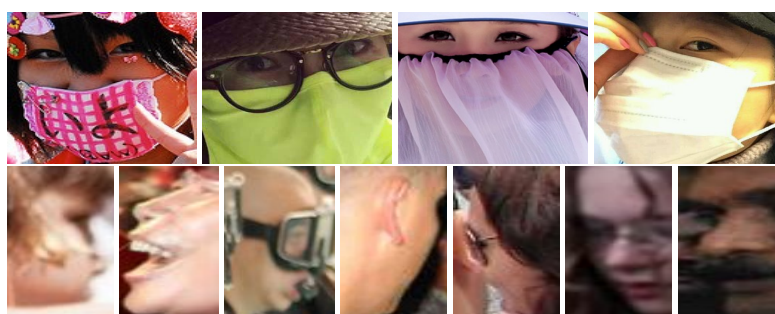


Figure 8. Example face images that were not detected by RetinaFace (type II errors), the best performing face detector from the experimental evaluation. The detector was adversely affected by extreme face occlusion (top row) and profile and poor-quality faces (bottom row).

5.2. Classification Results

In the second series of experiments, we investigate the performance of different recognition models for predicting compliant and non-compliant face-mask placements. We implement multiple versions of the models discussed in Section 4.2 with different numbers of layers—as indicated by the suffixes next to the model names below. We train all models on the training part of FMLD, as summarized in Table 3. All considered models expect (3-channel) RGB images of size $H \times W$ as input, where H and W are expected to be at least 224. The images are loaded in floating point format (in a range of $[0, 1]$) and normalized in terms of mean and standard deviation. Here, the vector $[0.485, 0.456, 0.406]$ is used as the mean and $[0.229, 0.224, 0.225]$ as the standard deviation.

As already emphasized in the previous section, we perform transfer learning to learn the recognition models. Thus, we initialize the models with weights learned in ImageNet and fine-tune all parameters for the new task. The training data is randomly divided in a ratio of 80:20 into a learning and a validation set. We use 40,307 face images (19,682 compliant, 20,625 non-compliant) for training and 10,077 face images (4921 compliant, 5156 non-compliant) for validation and monitoring model convergence. Cross-Entropy is used

as the learning objective. For the optimization algorithm, we rely on Stochastic Gradient Descent with a learning rate of 0.001, which is reduced by a factor of 0.1 every 7 epochs, and a momentum of 0.9. The training is repeated 10 times with 10 epochs for each model architecture and with a random division into learning and validation sets. Images are shuffled for each epoch. The final results from the different training runs vary by 0.1% on average on the validation data and best overall model on the validation set is selected for the final performance reporting. With this procedure we ensure that all models are optimized for the best possible performance.

In a similar manner as with the detection experiments, we use bootstrapping to generate results with confidence scores. Specifically, we sample test sets of 5000 images from the test part of FMLD 100 times and observe the mean recognition accuracy and the corresponding standard deviation. The results of this experiment are summarized in Table 5. For each tested model, the computational complexity and model size is also reported. Here, the computational complexity is measured through the prediction speed for 12,688 face images on the NVIDIA Titan Xp GPU graphics card with mini-batch size of 128.

Table 5. The average classification accuracy of each model on the FMLD test set together with the prediction speed and the size of the model on disk, arranged in descending order. The processing time is computed over all 12,688 face images and on a per—image basis on a NVIDIA Titan Xp GPU. The prediction accuracy is reported for a bootstrapping protocol with 100 sampled test sets of 5000 images.

Model	Model Size [MB]	Time		Accuracy ($\mu \pm \sigma$) [%]
		Overall [s]	Per Image [ms]	
ResNet-152	223	79	6.23	98.93 \pm 0.10
DenseNet-161	103	76	5.99	98.75 \pm 0.12
DenseNet-201	71	60	4.73	98.62 \pm 0.14
ResNet-101	163	64	5.04	98.55 \pm 0.13
DenseNet-121	28	51	4.02	98.50 \pm 0.12
VGG-19	513	62	4.89	98.49 \pm 0.12
DenseNet-169	49	54	4.26	98.48 \pm 0.13
ResNet-50	90	51	4.02	98.42 \pm 0.14
ResNet-34	82	41	3.23	98.28 \pm 0.13
ResNet-18	43	38	2.99	98.24 \pm 0.12
MobileNet	8.8	38	2.99	98.19 \pm 0.14
SqueezeNet v1.1	2.8	35	2.76	98.09 \pm 0.15
SqueezeNet v1.0	2.9	39	3.07	98.06 \pm 0.15
GoogLeNet	22	38	2.99	98.01 \pm 0.16
AlexNet	218	33	2.60	97.81 \pm 0.16

As can be seen, all tested models achieve an average recognition accuracy of over 97%. The difference between the worst performing model, AlexNet, and the best performing model, ResNet-152, is only 1.12% in absolute terms. These results show that all tested recognition models have sufficient capacity to efficiently discriminate between a compliant and a non-compliant placement of face-masks and can do so very accurately.

Although the performance difference between the top performer, ResNet-152, and several other tested models is statistically significant, this difference (around 1%) has limited impact in operational settings, where other aspects of the models may be more important. Figure 9, therefore, compares the tested models with respect to multiple criteria, i.e., processing time, recognition accuracy and model size. Here, the processing time is reported relative to the fastest network (AlexNet). In this comparison, an ideal model would be close to the left and to the top and be represented by a circle with a small diameter.

We can see that different models offer different trade-offs. ResNet-152, for example, performs best, but is among the slowest in terms of processing speed. DenseNet-121 performs slightly worse, but is considerably faster and has a smaller memory footprint.

At the other end of the spectrum are the SqueezeNet models, which show an even weaker recognition performance than DenseNet-121, but are, therefore, among the fastest and smallest of all tested models. If we disregard AlexNet, which is somewhat older than the rest of the tested models, the results point to a trend, in that the smaller models (represented by small radii in Figure 9) are typically slightly worse in terms of performance compared to the larger models, but have an edge with respect to processing speed.

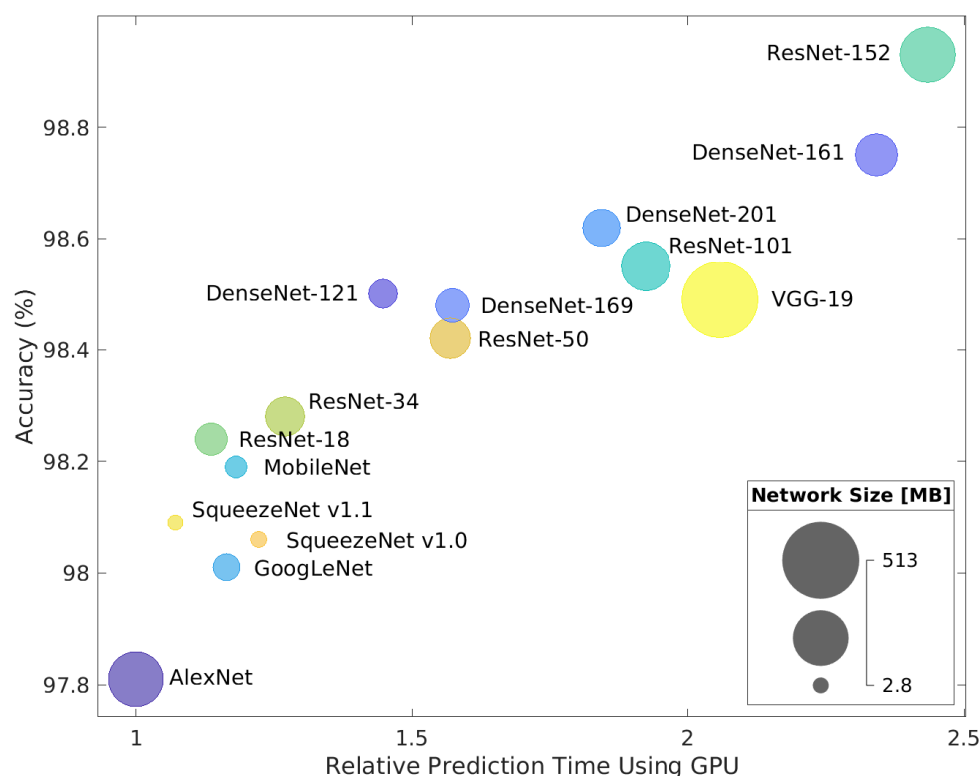


Figure 9. Comparison of different recognition models in terms of recognition accuracy, model size and processing speed. The processing time is measured using a GPU and a mini-batch size of 128. The prediction time is reported relative to the fastest network (AlexNet).

5.2.1. Analysis of Results

To get better insight into the behavior of recognition models trying to detect whether face-masks are worn in a compliant manner or not, we now investigate the results for the best performing model from the previous round of experiments, i.e., ResNet-152, in more detail.

We first look at the errors generated by the model. If we consider the entire test set of FMLD (without bootstrapping) the overall accuracy of the ResNet-152 model is 98.79% and the model makes a total of 154 errors. In 47 cases, it classifies faces with compliantly placed masks as non-compliant and for 107 faces from the non-compliant class it incorrectly predicts that they feature compliantly placed masks. Among these 107 faces, 37 have incorrectly placed masks and 70 faces in fact do not wear a mask at all. This error distribution is presented in the form of a confusion matrix on the left side of Figure 10. Below the confusion matrix we show class-wise precisions and on the right class-wise recalls for the tested model.

Since all faces in FMLD have labels for gender, pose and ethnicity, we next explore how successful the ResNet-152 model is with population subgroups defined by these labels. Table 6 and the right part of Figure 10 show the performance of the model with respect to the available annotations. As we can see, there is no significant difference in recognition accuracy with respect to gender and even pose. However, we do observe a bigger performance difference when ethnicity is considered. Here, the models perform comparable with Caucasian and African faces, but exhibits a slightly weaker performance

for Asian faces. Overall, this difference is in the range of 1.5%. A detailed analysis of these results showed that compared to African or Caucasian faces, the Asian subgroup contained significantly more masked faces. As a result, Asians faces dominate among the errors.

Table 6. Error distribution by labels for the evaluated classification model.

	Gender		Pose		Ethnicity		
	Male	Female	Front	Sideways	Asian	Caucasian	African
Faces	6433	6255	9245	3443	6962	5131	595
#Errors	69	85	102	52	126	27	1
Acc [%]	98.93	98.64	98.90	98.49	98.19	99.47	99.83

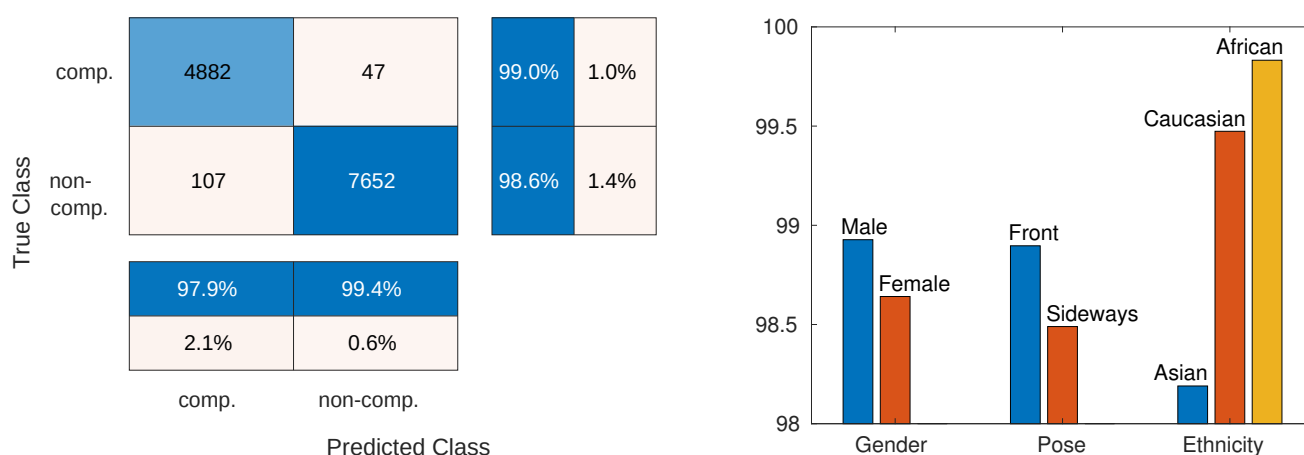


Figure 10. Results analysis for the best performing ResNet-152 model. **Left:** confusion matrix, **right:** accuracy results for subgroups of images.

Finally, we investigate what the ResNet-152 model has learned. To get insight into what contributed to the decisions, we use the gradient-weighted class activation mapping technique (Grad-CAM) [67]. Grad-CAM uses the gradient of the classification score with respect to the convolutional features determined by the network to understand which parts of the image are most important for classification. A few sample images that were correctly classified as not wearing masks in a compliant manner and corresponding Grad-CAM results are shown in Figure 11. As can be seen, uncovered facial features such as the nose and mouth appear to contribute to the decision the most.

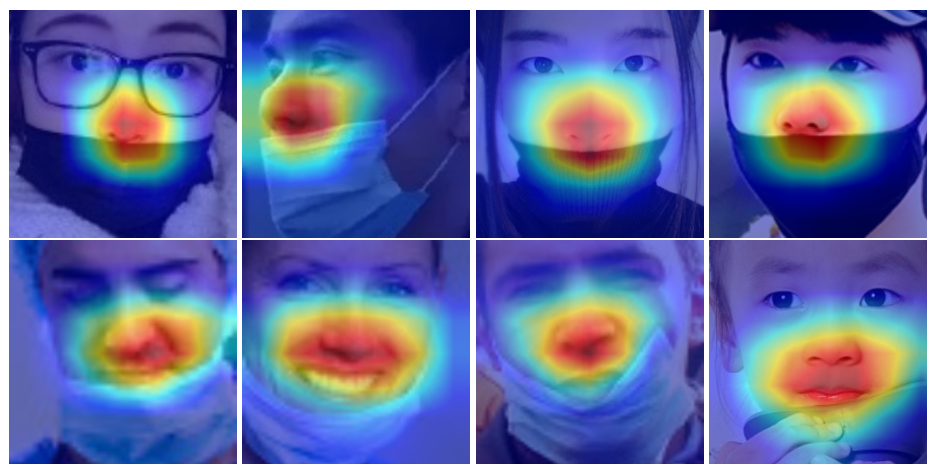


Figure 11. Properly classified faces that do not wear the mask correctly.

5.2.2. Error Analysis

The best models from our evaluation make an average of 150 mistakes, of which 2/3 are type I errors (false positives), i.e., the model predicts that the faces have compliantly placed face-masks, but in fact this is not the case. A few example images that generated this type of error with the ResNet-152 model are shown in Figure 12. As can be seen, type I errors belong to two groups of images, i.e., (1) images where no mask is actually present, but some sort of occluding object covers the nose, mouth and chin (top row in Figure 12), and (2) images that do wear a face-mask, but the placement is not compliant with recommendations (bottom row in Figure 12). The type I errors are distributed in a ratio of 2:1 in favor of the first group mentioned above.



Figure 12. Example face images incorrectly classified as being compliant (type I errors), first row: faces without masks, second row: faces with non-compliant mask placement.

The remaining 1/3 of errors are type II errors (false negatives), where the model predicts that faces are not masked in a compliant manner, but in fact they are wearing the masks properly. Some examples of such images are presented in Figure 13. As can be seen, these faces usually correspond to borderline cases (labeled as compliant in the dataset), where the nose, mouth and chin are covered correctly, but the face is also occluded by other objects, e.g., hands, or is covered by an object that is not actually a face-masks.

5.3. Evaluation of Overall Pipeline

In the last series of experiments, we evaluate the performance of the overall recognition pipeline. That is, we evaluate how successfully the complete processing pipeline detects faces and recognizes that they are masked in either a compliant or a non-compliant manner.

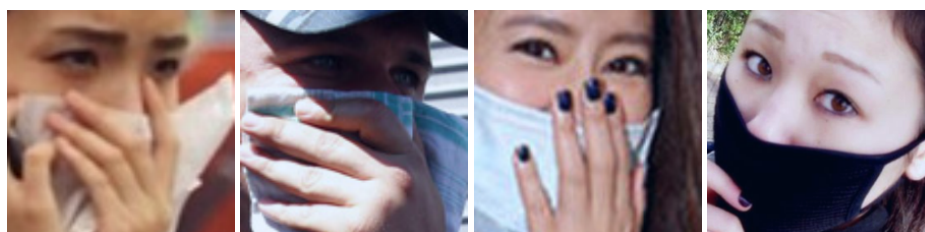


Figure 13. Examples of type II errors. The figure shows example faces labeled as compliant that are classified as being non-compliant. Please note that the errors are generated mostly for borderline cases, where the nose, mouth and chin are covered, but other objects occluding the face are also presented in the images, e.g., hands.

We design the pipeline with the best performing detection and recognition models from the previous section, i.e., RetinaFace and ResNet-152, and compare its performance to

existing one-stage models from the literature, i.e., Baidu, AntiCov and AIZooTech. We note that the considered baseline models have been designed to detect whether subjects are wearing masks and not if the masks are being worn correctly. Since only the latter is critical for containing the COVID-19 pandemic, this series of experiments also aims at evaluating the usefulness of existing models for this task.

To facilitate a detailed analysis, we report results separately for each class (compliant/non-compliant) in the form of average precision scores at two IoU thresholds. Additionally, we also report results for both classes in the form of mean average precisions (mAP). Results are again generated using a bootstrapping procedure with tests sets of 1000 images sampled 100 times from the test part of FMLD.

As can be seen from Tables 7 and 8 all evaluated baseline models are in general more effective at detecting non-compliant faces. The exception is AIZooTech's method, which is the only approach that is more successful with faces that wear masks in a compliant manner. This approach is also the most successful among all baseline models at detecting compliant faces with AP scores of 81.3% and 87.1% at IoU thresholds of 50% and 40%, respectively. In contrast to the compliant faces, where a considerable difference in performance is observed for the tested baseline models, the performance with non-compliant faces is very similar in terms of AP scores for both considered IoU thresholds of all three tested approaches, i.e., Baidu, AntiCov and AIZooTech.

Table 7. Average Precision results for each class, i.e., Compliant/Non-compliant and the Mean Average Precision for both classes. Results are reported for an IoU threshold of 50%.

Method	Compliant, $AP_{50}(\mu \pm \sigma)$ [%]	Non-Compliant, $AP_{50}(\mu \pm \sigma)$ [%]	$mAP_{50}(\mu \pm \sigma)$ [%]
Baidu	62.63 ± 2.16	78.34 ± 2.01	70.49 ± 1.36
AntiCov	69.77 ± 2.24	78.55 ± 1.56	74.16 ± 1.66
AIZooTech	81.30 ± 1.79	77.73 ± 1.79	79.52 ± 1.26
This paper	88.27 ± 1.75	93.23 ± 0.88	90.75 ± 0.99

Table 8. Average Precision results for each class, i.e., Compliant/Non-compliant and the Mean Average Precision for both classes. Results are reported for an IoU threshold of 40%.

Method	Compliant, $AP_{40}(\mu \pm \sigma)$ [%]	Non-Compliant, $AP_{40}(\mu \pm \sigma)$ [%]	$mAP_{40}(\mu \pm \sigma)$ [%]
Baidu	69.12 ± 1.98	84.92 ± 1.42	77.02 ± 1.32
AntiCov	76.55 ± 1.91	83.06 ± 1.28	79.80 ± 1.32
AIZooTech	87.07 ± 1.58	86.36 ± 1.23	86.72 ± 1.05
This paper	94.40 ± 0.81	97.04 ± 0.49	95.72 ± 0.55

AIZooTech is also the most successful baseline model in our experiments when both classes are considered jointly with an mAP_{50} score of close to 80% and an mAP_{40} score of close to 87%. Nonetheless, this is significantly behind our pipeline, which outperforms AIZooTech by around 10% in terms of both mAP_{50} as well as mAP_{40} score. The pipeline is also the clear top performer for each of the two classes at both IoU thresholds. Additionally, the pipeline also has the narrowest deviations around the computed mean scores.

Although the pipeline constructed in this paper achieves the best overall results, it needs to be noted again that it also represents the only tested approach specifically designed for detecting compliant and non-compliant faces. The baseline models, on the other hand, were designed only for detecting whether face-masks are present in the images or not. In terms of real-world use, where the goal is to identify people not properly protected by face-masks (i.e., people not wearing masks in accordance with recommendations), such models have limited value and fall behind techniques that are trained to directly identify whether the masks are worn in a compliant manner or not, as demonstrated by the presented results.

6. Conclusions

In this paper, we studied the problem of face-mask detection relevant in the scope of monitoring applications for the COVID-19 pandemic. We introduced a novel (annotation) dataset for studying face-mask detection problems and conducted an experimental study that looked at: (i) the performance of existing face detectors with masked-face images, (ii) the feasibility of recognition techniques aiming at the detection of properly worn face-masks and (iii) the usefulness of existing face-mask detection models for monitoring applications in the fight against COVID-19.

Our results showed that all tested detection models significantly deteriorate in performance when trying to detect masked faces compared to the performance observed with faces without masks. The most stable here was the RetinaFace model that also includes a generative component in the detection procedure. Furthermore, we observed that it is possible to design efficient techniques for recognizing faces with properly placed masked and that the selection of model architecture plays only a limited role in the final recognition performance. Finally, we demonstrated that existing models for face-masked detection have only limited value for real-life applications, as they only detect the presence of facial masks in the images, but not how these masks are placed.

Because the tested models work well and in real time, we plan to integrate the best performing approaches into a real-world monitoring system. We also plan to extend our analysis to other datasets that contain a wider range of mask types. Since measures to contain the spread of COVID-19 infections go in the direction that a certain group of people must use a certain type of mask, it would also make sense to design a classifier that can differentiate between different types of face-masks.

Author Contributions: Conceptualization: B.B., P.P., V.Š. and S.D.; Methodology: B.B., V.Š., P.P. and S.D.; Software: B.B.; Validation: V.Š. and S.D.; Formal analysis: B.B.; Investigation: B.B. and S.D.; Resources: B.B., P.P. and V.Š.; Data curation: B.B.; Writing—original draft: B.B.; Writing—review and editing: V.Š., P.P. and S.D.; Visualization: B.B.; Supervision: P.P., V.Š. and S.D.; Project administration: P.P. and V.Š.; Funding acquisition: P.P. and V.Š. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in parts by the Slovenian Research Agency (ARRS) through the ARRS Research Programmes P2-0214 (A) “Computer Vision” and P2-0250 (B) “Metrology and biometric systems”. The research was made possible by the increased funding for research programme P2-0250 (B) in the scope of the COVID-19 call. We also gratefully acknowledge the support of the NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The FMLD dataset presented in this study is openly available from GitHub: <https://github.com/borutb-fri/FMLD> (accessed on 25 February 2021).

Acknowledgments: The authors would like to thank Miha Zadavec who contributed to the creation of the annotation dataset and Selim Yahia-Messaoud who contributed to the review and supplementation of facial labels.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. World Health Organization. Coronavirus Disease (COVID-19) Advice for the Public. Available online: <https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public> (accessed on 1 December 2020).
2. Centers for Disease Control and Prevention. Considerations for Wearing Masks. USA. 2020. Available online: <https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/cloth-face-cover-guidance.html> (accessed on 12 November 2020).
3. Leung, N.H.; Chu, D.K.; Shiu, E.Y.; Chan, K.H.; McDevitt, J.J.; Hau, B.J.; Yen, H.L.; Li, Y.; Ip, D.K.; Peiris, J.M.; et al. Respiratory virus shedding in exhaled breath and efficacy of face masks. *Nat. Med.* **2020**, *26*, 676–680. [CrossRef] [PubMed]
4. Feng, S.; Shen, C.; Xia, N.; Song, W.; Fan, M.; Cowling, B.J. Rational use of face masks in the COVID-19 pandemic. *Lancet Resp. Med.* **2020**, *8*, 434–436. [CrossRef]

5. Wang, Z.; Wang, G.; Huang, B.; Xiong, Z.; Hong, Q.; Wu, H.; Yi, P.; Jiang, K.; Wang, N.; Pei, Y.; et al. Masked face recognition dataset and application. *arXiv* **2020**, arXiv:2003.09093.
6. Zafeiriou, S.; Zhang, C.; Zhang, Z. A survey on face detection in the wild: past, present and future. *Comput. Vis. Image Underst.* **2015**, *138*, 1–24. [[CrossRef](#)]
7. Wang, Y. Which Mask Are You Wearing? Face Mask Type Detection with TensorFlow and Raspberry Pi. Available online: <https://towardsdatascience.com/which-mask-are-you-wearing-face-mask-type-detection-with-tensorflow-and-raspberry-pi-1c7004641f1> (accessed on 26 November 2020).
8. Pranilshinde. Face-Mask Detector Using TensorFlow-Object Detection (SSD_MobileNet). Available online: <https://medium.com/pranil-shinde/face-mask-detector-using-tensorflow-object-detection-ssd-mobilenet-37f233202c67> (accessed on 26 November 2020).
9. Bhandary, P. Mask Classifier. Available online: <https://github.com/prajnasb/observations> (accessed on 26 November 2020).
10. Rosebrock, A. COVID-19: Face Mask Detector with OpenCV, Keras/TensorFlow, and Deep Learning. Available online: <https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning> (accessed on 26 November 2020).
11. Khanna, R. COVID-19-Authorized-Entry-Using-Face-Mask-Detection. Available online: <https://github.com/Rahul24-06/COVID-19-Authorized-Entry-using-Face-Mask-Detection> (accessed on 26 November 2020).
12. Bornstein, A.A. Personal Face Mask Detection with Custom Vision and Tensorflow.js. Available online: <https://medium.com/microsoftazure/corona-face-mask-detection-with-custom-vision-and-tensorflow-js-86e5fff84373> (accessed on 26 November 2020).
13. Yicong, O. Python Face Masks Detection Project. Available online: <https://github.com/ohyicong/masksdetection> (accessed on 26 November 2020).
14. K, G.M. COVID-19: Face Mask Detection Using TensorFlow and OpenCV. Available online: <https://towardsdatascience.com/covid-19-face-mask-detection-using-tensorflow-and-opencv-702dd833515b> (accessed on 26 November 2020).
15. Song, W. COVID19 Face Mask Detection Using Deep Learning. Available online: <https://www.mathworks.com/matlabcentral/fileexchange/76758-covid19-face-mask-detection-using-deep-learning> (accessed on 26 November 2020).
16. Loey, M.; Manogaran, G.; Taha, M.H.N.; Khalifa, N.E.M. A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. *Measurement* **2021**, *167*, 108288. [[CrossRef](#)] [[PubMed](#)]
17. Chowdary, G.J.; Pun, N.S.; Sonbhadra, S.K.; Agarwal, S. Face mask detection using transfer learning of inceptionv3. In Proceedings of the International Conference on Big Data Analytics, Sonapat, India, 15–18 December 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 81–90.
18. Ge, S.; Li, J.; Ye, Q.; Luo, Z. Detecting masked faces in the wild with LLE-CNNs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2682–2690.
19. RetinaFace Anti Cov Face Detector. Available online: <https://github.com/deepinsight/insightface/tree/master/detection/RetinaFaceAntiCov> (accessed on 26 November 2020).
20. Daniell Chiang, Y.F. Detect Faces and Determine Whether they are Wearing Mask. Available online: <https://github.com/AIZOOTech/FaceMaskDetection> (accessed on 26 November 2020).
21. Jiang, M.; Fan, X. RetinaMask: A Face Mask detector. *arXiv* **2020**, arXiv:2005.03950.
22. Yang, S.; Luo, P.; Loy, C.C.; Tang, X. Wider face: A face detection benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 5525–5533.
23. 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. In *NIST Interagency/Internal Report (NISTIR)*; NIST: Gaithersburg, MD, USA, 24 July 2020.
24. 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. In Proceedings of the 2020 International Conference of the Biometrics Special Interest Group (BIOSIG), Darmstadt, Germany, 16–18 September 2020; pp. 1–6.
25. Wang, J.; Yuan, Y.; Yu, G. Face attention network: An effective face detector for the occluded faces. *arXiv* **2017**, arXiv:1711.07246.
26. Larxel. Face Mask Detection, 853 Images Belonging to 3 Classes. Available online: <https://www.kaggle.com/andrewmvd/face-mask-detection> (accessed on 26 November 2020).
27. Wobot Intelligence. Face Mask Detection Dataset. Available online: <https://www.kaggle.com/wobotintelligence/face-mask-detection-dataset> (accessed on 26 November 2020).
28. Humans in the Loop. Medical Mask Dataset. Available online: <https://humansintheloop.org/medical-mask-dataset> (accessed on 26 November 2020).
29. Deng, J.; Guo, J.; Ververas, E.; Kotsia, I.; Zafeiriou, S. RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Conference on Computer Vision and Pattern Recognition (CVPR2020), Seattle, WA, USA (Virtual), 14–19 June 2020; pp. 5203–5212.
30. Khandelwal, P.; Khandelwal, A.; Agarwal, S. Using Computer Vision to enhance Safety of Workforce in Manufacturing in a Post COVID World. *arXiv* **2020**, arXiv:2005.05287.
31. Qin, B.; Li, D. Identifying Facemask-wearing Condition Using Image Super-Resolution with Classification Network to Prevent COVID-19. *Sensors* **2020**, *20*, 5236. [[CrossRef](#)]
32. Trident. Face Mask Detection System Using Artificial Intelligence. Available online: <https://www.tridentinfo.com/face-mask-detection-systems> (accessed on 26 November 2020).

33. Leewayhertz. Face Mask Detection System Using Artificial Intelligence. Available online: <https://www.leewayhertz.com/face-mask-detection-system> (accessed on 26 November 2020).
34. Udemans, C. Baidu Releases Open-Source Tool to Detect Faces without Masks. Available online: <https://technode.com/2020/02/14/baidu-open-source-face-masks> (accessed on 26 November 2020).
35. Tang, X.; Du, D.K.; He, Z.; Liu, J. Pyramidbox: A context-assisted single shot face detector. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 797–813.
36. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Process. Lett.* **2016**, *23*, 1499–1503. [\[CrossRef\]](#)
37. Villán, A.F. *Mastering OpenCV 4 with Python: A Practical Guide Covering Topics from Image Processing, Augmented Reality to Deep Learning with OpenCV 4 and Python 3.7*; Packt Publishing Ltd.: Birmingham, UK, 2019.
38. Li, J.; Wang, Y.; Wang, C.; Tai, Y.; Qian, J.; Yang, J.; Wang, C.; Li, J.; Huang, F. DSFD: Dual shot face detector. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 5060–5069.
39. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
40. Everingham, M.; Eslami, S.A.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [\[CrossRef\]](#)
41. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 248–255.
42. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
43. Cartucho, J.; Ventura, R.; Veloso, M. Robust Object Recognition Through Symbiotic Deep Learning In Mobile Robots. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 2336–2341.
44. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [\[CrossRef\]](#)
45. Valueva, M.V.; Nagornov, N.; Lyakhov, P.A.; Valuev, G.V.; Chervyakov, N.I. Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Math. Comput. Simulat.* **2020**, *177*, 232–243. [\[CrossRef\]](#)
46. Van den Oord, A.; Dieleman, S.; Schrauwen, B. Deep content-based music recommendation. *Adv. Neural Inform. Process. Syst.* **2013**, *26*, 2643–2651.
47. Collobert, R.; Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 160–167.
48. Kwon, O.Y.; Lee, M.H.; Guan, C.; Lee, S.W. Subject-independent brain-computer interfaces based on deep convolutional neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 3839–3852. [\[CrossRef\]](#) [\[PubMed\]](#)
49. Tsantekidis, A.; Passalis, N.; Tefas, A.; Kannianen, J.; Gabbouj, M.; Iosifidis, A. Forecasting stock prices from the limit order book using convolutional neural networks. In Proceedings of the 2017 IEEE 19th Conference on Business Informatics (CBI), Thessaloniki, Greece, 24–27 July 2017; Volume 1; pp. 7–12.
50. Bortolato, B.; Ivanovska, M.; Rot, P.; Križaj, J.; Terhorst, P.; Damer, N.; Peer, P.; Štruc, V. Learning privacy-enhancing face representations through feature disentanglement. In Proceedings of the FG 2020, IEEE International Conference on Automatic Face & Gesture Recognition, Buenos Aires, Argentina (Virtual), 16–20 May 2020; pp. 495–502.
51. Štepec, D.; Emeršič, Ž.; Peer, P.; Štruc, V. Constellation-Based Deep Ear Recognition. In *Deep Biometrics: Unsupervised and Semi-Supervised Learning*; Jiang, R., Li, C., Crookes, D., Meng, W., Rosenberger, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; [\[CrossRef\]](#)
52. Vitek, M.; Rot, P.; Štruc, V.; Peer, P. A comprehensive investigation into sclera biometrics: A novel dataset and performance study. *Neural Comput. Appl.* **2020**, 1–15. [\[CrossRef\]](#)
53. Grm, K.; Štruc, V. Deep face recognition for surveillance applications. *IEEE Intell. Syst.* **2018**, *33*, 46–50.
54. Krizhevsky, A. One weird trick for parallelizing convolutional neural networks. *arXiv* **2014**, arXiv:1404.5997.
55. Grm, K.; Štruc, V.; Artiges, A.; Caron, M.; Ekenel, H.K. Strengths and weaknesses of deep learning models for face recognition against image degradations. *IET Biometrics* **2017**, *7*, 81–89. [\[CrossRef\]](#)
56. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
57. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
58. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
59. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
60. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.

61. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
62. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
63. Pinkney, J. MTCNN Face Detection. Available online: <https://github.com/matlab-deep-learning/mtcnn-face-detection/releases/tag/v1.2.3> (accessed on 26 November 2020).
64. Rosebrock, A. Face Detection with OpenCV and Deep Learning. Available online: <https://www.pyimagesearch.com/2018/02/26/face-detection-with-opencv-and-deep-learning/> (accessed on 26 November 2020).
65. Hukkelås, H. DSFD-Pytorch-Inference. Available online: <https://github.com/hukkelas/DSFD-Pytorch-Inference> (accessed on 26 November 2020).
66. biubug6. RetinaFace in PyTorch. Available online: https://github.com/biubug6/Pytorch_Retinaface (accessed on 26 November 2020).
67. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.