Hindawi
Scientific Programming
Volume 2021, Article ID 8340779, 21 pages
https://doi.org/10.1155/2021/8340779

Research Article

Real-Time Implementation of AI-Based Face Mask Detection and Social Distancing Measuring System for COVID-19 Prevention

Safa Teboulbi,[1] **Seifeddine Messaoud**,[1] **Mohamed Ali Hajjaji**,[1,2] **and Abdellatif Mtibaa**[1]

[1]*Université de Monastir, Laboratoire d'Electronique et de Microélectronique, LR99ES30, 5000, Monastir, Tunisia*
[2]*Higher Institute of Applied Sciences and Technology, Sousse University, Sousse, Tunisia*

Correspondence should be addressed to Mohamed Ali Hajjaji; daly_fsm@yahoo.fr

Received 25 May 2021; Revised 3 August 2021; Accepted 19 August 2021; Published 27 September 2021

Academic Editor: Shah Nazir

Since the infectious coronavirus disease (COVID-19) was first reported in Wuhan, it has become a public health problem in China and even around the world. This pandemic is having devastating effects on societies and economies around the world. The increase in the number of COVID-19 tests gives more information about the epidemic spread, which may lead to the possibility of surrounding it to prevent further infections. However, wearing a face mask that prevents the transmission of droplets in the air and maintaining an appropriate physical distance between people, and reducing close contact with each other can still be beneficial in combating this pandemic. Therefore, this research paper focuses on implementing a Face Mask and Social Distancing Detection model as an embedded vision system. The pretrained models such as the MobileNet, ResNet Classifier, and VGG are used in our context. People violating social distancing or not wearing masks were detected. After implementing and deploying the models, the selected one achieved a confidence score of 100%. This paper also provides a comparative study of different face detection and face mask classification models. The system performance is evaluated in terms of precision, recall, F1-score, support, sensitivity, specificity, and accuracy that demonstrate the practical applicability. The system performs with F1-score of 99%, sensitivity of 99%, specificity of 99%, and an accuracy of 100%. Hence, this solution tracks the people with or without masks in a real-time scenario and ensures social distancing by generating an alarm if there is a violation in the scene or in public places. This can be used with the existing embedded camera infrastructure to enable these analytics which can be applied to various verticals, as well as in an office building or at airport terminals/gates.

1. Introduction

Since the end of 2019, infectious coronavirus disease (COVID-19) has been reported for the first time in Wuhan, and it has become a public damage fitness issue in China and even worldwide. This pandemic has devastating effects on societies and economies around the world causing a global health crisis [1]. It is an emerging respiratory infectious disease caused by Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) [2]. All over the world, especially in the third wave, COVID-19 has been a significant healthcare challenge [3]. Many shutdowns in different industries have been caused by this pandemic. In addition, many sectors such as maintenance projects and infrastructure construction have not been suspended owing to their significant effect on people's routine life [4, 5].

By now, the virus has rapidly spread to the majority of the countries worldwide [2]. The last statistics (04/05/2021) provided by the World Health Organization (WHO) show 152,543452 confirmed cases and 3198528 deaths. According to the centers for Disease Control and Prevention (CDC), coronavirus infection is transmitted predominantly by respiratory droplets produced when people breathe, talk, cough, or sneeze [3] with common droplet size 5–10'm but aerosol emission increases when humans speak and shout loudly [6].

Therefore, to prevent rapid COVID-19 infection, many solutions, such as confinement and lockdowns, are suggested by the majority of the world's governments. However, this COVID-19 management inefficacy can be additionally explored with game-theoretic scenarios beyond the public goods game. In particular, some researchers have focused on the hesitancy of governments in enacting difficult but necessary virus containment measures (e.g., stay-at-home orders and lockdowns), as well as noncooperation for reasons other than free riding. For instance, authors in [7] argued that because strict stay-at-home measures can greatly impact people's livelihoods, the cost of staying home (coupled with lockdown fatigue) can end up outweighing the risk of infection from going out. As individual-level decisions have a direct impact on the society-level effectiveness of stay-at-home orders, governments may refrain from implementing them because of anticipated low rates of compliance, especially from socioeconomically disadvantaged individuals who do not have the luxury of staying home [8]. Some governments may have also been hopeful that herd immunity from recoveries and vaccinations would allow them to avoid imposing such unpopular measures altogether [9].

With rising numbers of cases and stretched health facilities, as well as the lack of a vaccine throughout 2020 and difficulties associated with achieving herd immunity for COVID-19 [10], government inaction became increasingly unviable. Hence, to increase people's adherence to strict regulations, authors in [7] suggested using social programs such as emergency relief funds and unemployment insurance to lower the costs of compliance, particularly for lower-paid workers [11]. As vaccines became available at the end of 2020, authors in [12] argued that programs driving vaccination uptake will surpass other aspects such as vaccine efficacy and isolation procedures in importance. Using EGT, social network analysis, and agent-based modeling, the authors proposed that individuals' vaccination decision-making will be influenced by "demographics, physical location, the level of interaction, the health of the vaccine, epidemic parameters, and perceptions about the vaccine being introduced, and similarly, the decision-making of the government will be influenced by epidemic parameters, the nature of the vaccine being introduced, logistics, the management of human resources needed for the vaccination effort, and the number of vaccine doses available" [12]. In summary, holistic COVID-19 management would involve an appreciation of the many factors that calibrate payoffs so that both individual and governmental decisions shift toward safety.

It is true that COVID-19 is a global pandemic and affects several domains. Nevertheless, it created a path for researchers in computer science. We have seen multiple research topics, such as creating new automatic detection methods of COVID-19 and detecting people with or without masks. Considering that there are some errors in the results of the early laboratory tests and their delays, researchers focused on different options [13]. Therefore, the application of advanced artificial intelligence (AI) techniques [14–17] coupled with chest radiological imaging (computed tomography (CT) and X-ray) can lead to a more accurate detection of the COVID-19 and can help to control the problem of loss of specialized physicians in isolated villages [18]. In this context, authors in [13] suggested a novel convolutional neural network (CNN)-based method for detecting COVID-19, with analyzing chest X-ray (CXR) images. This method allows to detect patients with COVID-19 at an accuracy of 91.34%. In [18], the authors introduced a new automatic COVID-19 detection model using CXR images. The model is called "DarkCovidNet." For binary classes (COVID-19 VS no findings), the classification accuracy produced by this model is 98.08%, but, for multiclass cases (COVID-19 VS pneumonia VS no findings), the accuracy is 87.02%. The main objective is to use such models to diagnose supplementary chest-related diseases such as tuberculosis and pneumonia. For the COVID-19 case detection from X-ray images, a deep CNN model is proposed in [19]. This model, denoted COVID-Net, is open source and accessible to the general public. The detection accuracy achieved by this model (93.3%) proves that the model makes good predictions in improved screening. In [20], distinct deep learning techniques are introduced to differentiate CT scan images of both COVID-19 and non-COVID-19. From the different techniques, we list a self-developed model (CTnet-10), VGG-16, ResNet-50, InceptionV3, VGG-19, and DenseNet-169, which have, respectively, accuracy values of about 82.1%, 89%, 60%, 53.4%, 94.52%, and 93.15%. The accuracy of VGG-19 is the highest one as compared to other models. The CTnet-10 method is a well-organized model, which is useful for doctors, especially in mass screening.

In [21], two deep learning models are suggested: a CNN and a convolutional long short-term memory (ConvLSTM). To simulate them, two datasets are assumed. A dataset includes CT images, and the other includes X-ray images. The models are tested four times. When they are examined on CT images, the dataset is split into 70% for the training set and 30% testing set. The accuracy value for the CNN model and for the ConvLSTM is the same, equal to 99%. When tested on the augmented dataset A, the testing accuracy of the CNN is 99%, but it is 100% for the ConvLSTM. When tested on the augmented dataset B, the testing accuracy of the CNN is 100%, but it is 99% for the ConvLSTM. When both models are tested on the combined dataset, containing both X-ray and CT images, the testing accuracy is 99% for the CNN and 98% for the ConvLSTM. Finally, when they are tested on the radiography dataset, the testing accuracy is 95% and 88% for the CNN and the ConvLSTM, respectively. We can consider this scenario as a challenging one, because it is called to distinguish between two diseases (COVID-19 and pneumonia) with a high closeness in features.

Before coronavirus, some people put masks to protect themselves from air pollution, while other people put face masks to hide their faces and their emotions from others. Protection against coronavirus is a mandatory counter measure, according to the WHO [1]. Indeed, wearing a mask is an effective method of blocking 80 of all respiratory infections [3]. Also, the WHO recommends practising physical distancing to mitigate the spread of the virus. All over the world, governments are struggling against this type

of virus. Many organizations enforce face mask rules for the personal protection. Checking manually if individuals entering an organisation are wearing masks is cumbersome and possibly conflicting [1]. In this context, authors in [6] proposed a deep learning-based model, named MobileNet Mask, to prevent human-to-human transmissions of the SARS-CoV-2 and detect faces with or without mask. Two different datasets (IDS1 and IDS2) with over 5200 images are used to train and test the model. All the experimental cases are controlled on Google Colab that runs in the cloud. In IDS1, the proposed model achieved a testing accuracy of 93%. However, in IDS2, the accuracy achieved is almost 100%.

In [22], the authors aim to detect and delimit medical face masks in real images. The proposed model is composed of YOLO-V2 and ResNet-50. In the training phase, the authors used two optimizers which are Adam and SGDM. During this process, SGDM is better than Adam on validation root mean square error (RMSE), time, and validation loss. However, Adam is better than SGDM in loss and minibatch RMSE. The average precision (AP) of the Adam optimizer is 0.81, which is better than the AP of SGDM that is equal to 0.61. Moreover, the log-average miss rates of the Adam optimizer are 0.4, which are better than the log-average miss rates of the SGDM optimizer (0.6), in all the recall levels. From these studies, we found also, in [23], a system which triggers an alarm in the operating room when the healthcare personnel do not wear face masks. The system combines two detectors, for faces and for masks. In the testing phase, by utilizing images from the BAO dataset and from an own image dataset, above 95% of true positive and below 5% of false positive rates are achieved.

Authors in [1] proposed a hybrid model utilizing deep learning with classical machine learning to detect masked faces. The proposed model consists of two components: the feature extraction applying ResNet-50 and the classification process. The three classifiers used are the decision trees (DTs), support vector machine (SVM), and ensemble algorithm. The Real-World Masked Face Dataset (RMFD), the Simulated Masked Face Dataset (SMFD), and the Labeled Faces in the Wild (LFW) are the three face masked datasets, selected for examination. The SVM classifier is greater than the other classifiers. It reached 99.64%, 99.46%, and 100% of testing accuracy, respectively, in RMFD, SMFD, and LFW.

In summary and without forgetting the legal side of AI, the deep learning technique inherently touches upon a full spectrum of legal fields, from legal philosophy, human rights, contract law, tort law, labor law, criminal law, tax law, procedural law, etc. While in practice, AI is just beginning to come into its own in terms of its use by lawyers, and within the legal industry, legal scholars have been occupied with AI for a long time. Furthermore, as collecting and analyzing data is progressively spreading from software companies to manufacturing companies, which have started to exploit the possibilities arising from collection and exploitation of potential data, so that added value can be created, this information deluge unlocks various legal concerns that could stimulate a regulatory backlash. Considering AI legal concerns and benefits in combating COVID-19 pandemic, AI technique-based solutions are still an open window for development and legal interpretation [24].

The reminder of this paper is organized as follows: Section 2 summarizes the recent related work in the proposed context. Section 3 presents the proposed framework. After that, a preliminary study is given in Section 4. Section 5 denotes the dataset collection. Thereafter, we introduce the evaluation metrics in Section 6, while the numerical result is discussed in Section 7. Finally, we conclude this paper in Section 8.

2. Related Works

Deep learning is an important breakthrough in the AI field. It has recently shown enormous potential for extracting tiny features in image analysis. Due to the COVID-19 epidemic, some deep learning approaches have been proposed to detect patients infected with coronavirus. In this context, and unlike bacterial pneumonia, many other types of lung infections caused by viruses are called viral pneumonia. These viruses, such as the COVID-19, infect the lungs by blocking the oxygen flow, which can be life-threatening. This motivated researchers to develop many frameworks and schemes based on AI tools in the fight against this dangerous virus. Hence, we divide this section into two sections to provide an in-depth overview of the proposed techniques.

2.1. COVID-19 Detection Methods

2.1.1. Deep Learning Tools and CXR Image-Based COVID-19 Detection. Radiography is a technique used to quantify the functional and structural consequences of chest diseases, to provide high-resolution images on disease progression. Several works have been carried out in this context. Echtioui et al., in [13], proposed a new CNN-based method for COVID-19 recognition, through analyzing radiographic images of a patient's lungs. The aim of this scheme is to provide clinical decision support for healthcare workers and also for researchers. Hence, performance results, as well as the accuracy value of about 91.34%, and the other metrics in terms of recall, precision, and F1-score, prove the efficiency of the method. In the same context, Ozturk et al., in [18], introduced a new automatic COVID-19 detection model using CXR images denoted by the "DarkCovidNet." It is used to provide correct diagnosis for both a binary classification (COVID-19 VS no findings) and a multiclass classification (COVID-19 VS pneumonia VS no findings). For binary classes, the classification accuracy produced by this model is about 98.08%, but, for multiclass cases, the accuracy is 87.02%. To validate their initial screening, radiologists can use the model to assist them. This model can be employed also via cloud to screen patients immediately. As a solution to the shortage of radiologists, this method can be used in remote places especially in countries affected by COVID-19. The most important advantage of this method is that such models can be used to diagnose supplementary chest-related diseases such as tuberculosis and pneumonia. However, the proposed work fits well into the COVID-19

detection phase, but to ensure its efficiency and model reliability, the authors may augment the dataset and retrain the proposed model. With the same idea, Wang et al. proposed a deep CNN model which is presented in [19]. Their model, called COVID-Net, is open source and accessible to the general public. The test accuracy achieved by this model is 93.3%. Therefore, this model makes predictions which can assist clinicians in improving screening, transparency, and trust.

2.1.2. Deep Learning Tools and CT Image-Based COVID-19 Detection. Computed tomography scan or CT scan is a medical imaging technique utilized in radiology in order to get detailed images of the body for diagnosis purposes. Accurate and fast COVID-19 screening is achievable using CT scan images. Various works have been carried out in this context. In [20], Shah et al. proposed distinct deep learning techniques to differentiate CT scan images of both COVID-19 and non-COVID-19, which helps in diagnosis. In the dataset, we find 349 images corresponding to patients with COVID-19 and 463 images corresponding to patients without COVID-19. These images were divided into three sets: 80% of them for training set, 10% for validation, and 10% for testing. From the different techniques presented in this work, we cite CTnet-10, which is a self-developed model having an accuracy of 82.1%. We can also cite VGG-16, ResNet-50, InceptionV3, VGG-19, and DenseNet-169, having an accuracy of 89%, 60%, 53.4%, 94.52%, and 93.15%, respectively. The accuracy of VGG-19 is the best as compared to other models. To predict the results, CTnet-10 takes only 12.33 ms. This method is well-organized. It is useful for doctors, especially in mass screening. All the automatic diagnosis methods presented previously can be used by doctors for COVID-19 screening.

2.1.3. Methods Using CXR and CT Images. Combining two types of images in one dataset is an effective method to detect a disease. In this context, in [21], Sedik et al. presented two deep learning models: CNN and ConvLSTM. To simulate the models, two datasets are assumed. The first dataset includes CT images while the second set includes X-ray images. Each dataset contains COVID-19 and non-COVID-19 image categories. The image categories, COVID-19 and pneumonia, were classified to certify the proposed models.

The first model based on CNN includes five convolutional layers (CNVLs) accompanied by five pooling layers (PLs). Two layers (fully connected layer (FC) and classification layer) make up the classification network. The second model is a hybrid one. It combines ConvLSTM and CNN at the same time.

The classification network, too, is in the first model. To reduce the complexity of the planed deep learning structure, training, validation, and testing are the three phases that make up the two modalities. An optimization methodology is necessary in the training. To minimize the errors between the real and the estimated targets, Sedik et al. used the Adam optimizer. This type of model needs to be held carefully. The proposed models are evaluated by measuring accuracy, Matthews correlation coefficient (MCC), and F1-score. The specificity, negative predictive value (NPV), sensitivity, and positive predictive value (PPV) are considered also in the evaluation process.

The models were tested four times: firstly, on the dataset containing CT images with 288 COVID-19 and 288 normal images, this dataset is augmented by diverse rotations and operations of scaling, and the number of COVID-19 and normal images becomes 2880 and 2880, respectively; secondly, on the dataset containing X-ray images, this dataset includes two distinct augmented subsets, and each subset, named "augmented dataset A" and "augmented dataset B," contains 304 COVID-19 and 304 normal images; thirdly, on the dataset, named COVID-19 radiography dataset, containing COVID-19 and viral pneumonia X-ray images; and fourthly, on a combined dataset, which combines both X-ray and CT images in the two cases, normal and COVID-19.

When the models were examined on CT images, the dataset is split into a training set (70%) and a testing set (30%). They were trained on 40 epochs. The testing accuracy for the CNN model and for the ConvLSTM was the same, equal to 99%. This is due to their methodical design and the nature of images. And when they are tested on the augmented dataset A, the testing accuracy was 99% for the first model and 100% for the second. However, when they were tested on the augmented dataset B, the testing accuracy was 100% for the first model and 99% for the second model. As for testing on the combined dataset, containing both X-ray and CT images, the testing accuracy was 99% for the first model and 98% for the second model. Finally, when they were tested on the radiography dataset, the testing accuracy of the first model was 95%, but 88% for the second model.

We can consider this scenario as a challenging one, because it is called for differentiating between two diseases (COVID-19 and pneumonia) with a high closeness in features. The proposed models achieved the same accuracy of 99% when they were tested on X-ray and CT images, while, in previous works, they achieved a range of 95% to 98% and a range of 83% to 90.1%, for X-ray and CT images, respectively. Therefore, the two proposed models can be considered as efficient COVID-19 detection systems. Table 1 presents a comparison of different COVID-19 methods.

2.2. Face Mask Detection. Another task in research is detecting people with or without mask, to prevent the transmission of SARS-CoV-2 between humans.

2.2.1. MobileNet Mask Model. All governments around the world are struggling against COVID-19, which causes serious health crises. Therefore, the use of face masks regulatory can slow down the high spread of this virus. In this context, Dey et al. proposed in [6] a deep learning-based model for detecting face mask. This model named "MobileNet Mask" is multiphase. A pretrained model of the ResNet-10 architecture is utilized to find faces in video stream. Also, numerous steps are used such as charging the classifier (MobileNet), building the FC layer, and testing phase. All the experimental cases are supervised on Google

Table 1: Comparison of different coronavirus detection methods.

Ref	Model used	Image type	Accuracy (%)	Advantages	Disadvantages
[13]	CNN-based model	Chest X-ray	91.34	A simple but effective CNN model for the detection of COVID-19	A limited number of images available
[18]	DarkCovidNet	Chest X-ray	87.02	Able to perform binary and multiclass tasks can be used to diagnose other chest-related diseases including tuberculosis and pneumonia	
[19]	COVIDX-Net	Chest X-ray	93.3	An open source and accessible to general public	
[20]	Self-developed model (CTnet-10)	CT	82.1	All the automatic diagnosis presented previously can be used by doctors as an efficient and quite method for coronavirus disease screening	
	VGG-16		89		
	ResNet-50		60		
	InceptionV3		53.4		
	VGG-19		94.52		
	DenseNet-169		93.15		
[21]	CNN modality	CT	99	The proposed modalities can be considered in an efficient diagnosis system for the detection of COVID-19 and other relevant infections	The proposed modalities achieve poor performance in some cases especially for the ConvLSTM modality due to the dependency on the previous states in the structure of the ConvLSTM modality
		Augmented dataset A	99		
		Augmented dataset B	100		
		Radiography X-ray	95		
		Combined dataset (CT scan and X-ray)	99		
	ConvLSTM	CT	99		
		Augmented dataset A	100		
		Augmented dataset B	99		
		Radiography X-ray	88		
		Combined dataset (CT scan and X-ray)	99		

Colab that runs in the cloud and is provided with over 12 GB of RAM. Different performance metrics (accuracy, F1-score, precision, and recall) are used to judge the performance of the proposed model.

Two distinct face mask datasets are used to train and test the model. The first dataset, named IDS1, consists of 3835 images, divided into two classes: 1916 images of faces with masks and 1919 images without masks. Kaggle dataset, RMFD, and Bing Search API are the source of the typical images of this dataset. The second dataset, named IDS2, consists of 1376 images, divided into two classes: 690 images of faces with masks and 686 images without masks. The sample images of this dataset are gathered from the SMFD.

For all experiments, 80% of the datasets are dedicated for training and 20% for testing. When testing the model, it achieved an accuracy of 93% in IDS1, but almost 100% in IDS2. Comparing the results of their model with those of state-of-art models available in the literature, Dey et al. found that the accuracy of theirs is higher. The main advantage is that their model can be implemented on lightweight embedded computing devices.

2.2.2. ResNet-50 with YOLO-V2 Model. Annotating and localizing medical face masks in real-life images is among the most important object detection applications. In this context, the main objective of Loey et al. in [22] is to explain and delimit the objectives of the medical face masks, especially in real images. They proposed a model consisting of two steps: medical face masks and feature extraction.

The two public datasets of medical face masks are merged in one dataset to be explored in their research. The first one is Medical Masks Dataset (MMD). It contains 682 images with more than 3000 faces wearing masks. The second one is Face Mask Dataset (FMD), which contains 853 images. Combining both datasets resulted in a dataset of just 1415 pictures after deleting bad quality pictures.

Three main components compose the introduced model: the number of anchor boxes, data augmentation, and the detector. To evaluate the performance of the YOLO-V2 with the ResNet-50 in noticing and isolating masked faces, Loey at al. conducted different experiments. The model was executed on the system which has specifications such as CuDNN that is a library of the deep neural network for GPU learning. 70% of the dataset is dedicated for the training phase, 10% for the validation phase, and 20% for the testing phase. Learning rate is initialized with $\sigma = 0.001$, the number of epochs with 60, and the minibatch size with 64. To ameliorate detector performance, they used Adam and Stochastic Gradient Descent with Momentum (SGDM) optimizers.

In the training, SGDM took less time than Adam. When comparing the performance of Adam and SGDM, in the process of training and validation, Loey et al. found that SGDM is better in time, validation loss, and validation

RMSE. However, Adam is better in loss and minibatch RMSE. AP and log-average miss rate scores are performance metrics studied for both Adam and SGDM optimizer experiments. In all the recall levels, the AP of the Adam optimizer (0.81) is better than the AP of SGDM (0.61) and the log-average miss rates of the Adam optimizer (0.4) are better than the log-average miss rates of the SGDM optimizer (0.6). Therefore, the proposed model is powerful in detecting masked faces.

2.2.3. Detecting Masks in the Operating Room. Wearing medical masks in an operating room is mandatory, to prevent the transmission of viruses from human to human. In this context, in [23], Nieto-Rodríguez et al. introduced a system which detects the existence or the absence of the medical masks in hospitals especially in the operating rooms. The system triggers an alarm for only the healthcare personnels who do not put on surgical masks. It consists of four components: two detectors, one for unmasked faces and one for masked faces, and also two color filters, one filter for each detector.

The well-known face detector, named "Viola and Jones," is the base of the face detector. Generally, face detectors utilize a succession of classifiers. In this paper, a form of AdaBoost, named LogitBoost, is used. The same strategy is followed by mask detectors. However, instead of LogitBoost, Gentle AdaBoost is used.

The two types of detectors work only on gray images. Reducing the number of false positives and trying to keep away from the false negative number is the objective of the stage of color filters. All the detections produced by the detectors which proceeded the color filters are then classified in the identical category. The face detectors include two stages: one stage on gray scale and the other stage with color.

For the gray-case sample, the training phase is accomplished in the LFW image dataset. 10000 positive images with 5000 negative images are used during the training process. The test process is run on the CMU Frontal Face Set. A succession of 20 classifiers is present in the face detector. Each one is a decision tree accompanied with two depth levels escorted by 0.999 threshold and 0.5 as false positive threshold. The color filter used for the face detection is built in the color space of HSV, for over 13000 pictures from LFW. More than 4000 images are used in the training phase. Images from the BAO dataset are used for testing.

In the training phase, on gray scale images, 4000 positive with 15000 negative images are used. The color filter for masks finds the position, definitions, threshold, and size of the features. The detectors, for face and for mask, are combined by the face classifier. Images with unmasked faces from the BAO dataset and images with masked faces from an own dataset are used in the test phase. As a result, the true positive ratio is throughout 95% and the false positives are under 40. The main objective of this system is working in real-time especially on a conventional PC.

2.2.4. Hybrid Model. All over the world, the trend of wearing masks is rising because of COVID-19 pandemic. In this context, a hybrid model utilizing deep learning with classical machine learning (ML) to detect masked faces is presented by Loey et al. in [1]. The model proposed by the authors includes two phases: the feature extraction process applying ResNet-50, and the classification process. The ResNet-50 model used like a feature extraction is composed of 50 deep layers. A convolutional layer (CNVL) is the start of the model, a fully connected layer (FCL) is the end of the model, and 16 residual bottleneck blocks are in between them.

To improve the performance of the model, three traditional classifiers replaced the last layer of the ResNet-50 during the classification process. The classification of face masks is released by three algorithms: the DT, the SVM, and the ensemble algorithm. The SVM is a machine learning algorithm designed for classification. It is one of the most popular supervised learning techniques. The DT is a model of classification based on information gain and entropy function. Ensemble methods are a combination of algorithms of machine learning which generate a collection of classifiers. The most adopted ensemble methods are linear regression, K-nearest neighbors (K-NNs) algorithm, and logistic regression (LR).

Three datasets are selected for examination: RMFD, SMFD, and LFW. The RMFD dataset contains 5000 masked faces with 90000 unmasked faces. However, Loey et al. used just 5000 masked face images and 5000 unmasked face images to stabilize the dataset. The SMFD dataset contains 785 simulated masked face images and 785 unmasked face images. As for the LFW dataset, it consists of 13000 masked face images for celebrities all over the world. It is used only in the testing phase. During the training and the testing phase, the RMFD is referred to as DS1, SMFD is referred to as DS2, a combination of DS1 and DS2 is referred to as DS3, and LFW is referred to as DS4 which is used only for the testing process. The datasets used for the training and the testing are divided into 70%, 10%, and 20%, for training, validation, and testing, respectively. The most frequent performance measures used to judge the performance of the three classifiers are accuracy, recall, precision, and F1-score.

Table 2 shows the performance of validation accuracy of 98%, achieved by the DT classifier. The highest testing accuracy is achieved by the DT classifier when the training is completed over DS3. On DS4, which is used for testing only, a competitive accuracy of 99.89% is achieved. The validation accuracy achieved by the SVM classifier for the different datasets exceeds the accuracy of the DT classifier.

Additionally, while the training is above DS2, the top validation accuracy possible reached by the SVM is 100%, while the highest validation accuracy achieved by the DT classifier in DS3 was 98%.

Another essential factor to evaluate the performance of a classifier is the time it takes to perform a task. For all the datasets, the time taken by the SVM classifier is shorter than that taken by the DT classifier. In terms of validation accuracy, testing accuracy, consumed time, and performance metrics, the SVM classifier is better than the DT classifier.

The same experimental cases conducted on the previous classifiers (DT and SVM) are performed on the ensemble algorithms classifier. During the validation on DS3, the ensemble achieved an accuracy of 100%. According to the

Table 2: Validation accuracy and testing accuracy of the three classifiers for different datasets.

Classifier	Datasets	Validation accuracy (%)	Testing accuracy (%)
DT	DS1	92 to 94	96.78
	DS2	96	95.64
	DS3	98	96.5
	DS4	—	99.89
SVM	DS1	98	99.64
	DS2	100	99.49
	DS3	99	99.19
	DS4	—	100
Ensemble algorithms	DS1	97	99.28
	DS2	94	99.49
	DS3	100	99.35
	DS4	—	100

obtained results, the ensemble classifier is better than the DT and the SVM classifier with regard to the validation accuracy, testing accuracy, and performance metrics, only when the training is on DS1 and DS3. Contrarily, when the training process is on DS2, the SVM classifier outperforms the other classifiers. Moreover, the time consumed by the SVM while the training process is the short one.

3. Proposed Face Mask Detection Framework Based on Deep Learning Models

Figure 1 depicts the whole proposed framework, in this paper, which consists of two principal blocks. The first block includes the training and the testing models, whereas the second block consists of the whole framework testing (the best model with social distancing step). For the first block, our labeled dataset was divided into three classes. The first class is focused on the training and represents 70% of the dataset images. However, the validation step required only 10% to validate the performance for the trained models. 20% of the dataset was devoted to the testing phase. For each epoch, each model is trained on the training dataset. The training results, as well as the training accuracy and the training loss, are presented in the form of curves in figures of "accuracy in terms of epoch" and "loss in terms of epoch," respectively. After training, each model is validated on the validation dataset. Like the training results, the obtained validation results are the validation accuracy and the validation loss. Then, the two results are compared with the loss function. An error function value tending toward zero means a well-trained model. Otherwise, the hyperparameters are tuned to train the model in another epoch. The process of calculating errors and updating the network's parameters is called backward propagation, which is the second important process elaborated in the training phase of any neural network, after the forward propagation process.

The hyperparameters, as well as learning rate, batch size, number of epochs, optimizer, anchor boxes, and loss function, are tuned to build an optimal model. However, the learning rate is denoted as the learning step where the model updates its learned weights. It contains inputs which are fed into the algorithm and also an output to calculate the errors. The batch size defines the number of trials to work along before updating the parameters of the internal model. In other words, it is the number of trials that will be proceeded across the network at the same time.

A training dataset could be dissected into just one or supplemental batches. The number of epochs is a hyperparameter defining the number of times the learning algorithm will labor through the full training dataset. Optimizers are assisted to minimize the loss function. They update the model in regard to the loss function output. The loss function is also called error function. We can say that the heart of the different algorithms of the ML is the loss functions. The loss function could be used to estimate the model's loss. Thus, the weights can be renovated to minimize the loss of the following evaluation. In the testing phase, our seven various models will be scanned to choose the best one to be exploited in the next step.

The second block, which is the testing framework phase, was developed to operate with the best model. The best loaded model is used to confirm the face mask detection technique. In addition, the pairwise distance algorithm was evolved to calculate social distance between peoples. However, the distance between the centers of the bounding box of detected people will be calculated. The center point $C(x, y)$ of bounding boxes is measured using the equation as seen in

$$C(x, y) = \frac{X_{\min} + X_{\max}}{2}, \cdot \frac{Y_{\min} + Y_{\max}}{2}, \tag{1}$$

where C is the center point of the bounding box. $X_m$in and $X_m ax$ are the minimum and maximum values for the corresponding width of the bounding box. $Y_{\min}$ and $Y_{\max}$ are the minimum and maximum values for the corresponding height of the bounding box. To measure the distance $C1$ $(X_{\max} - X_{\min})$ and $C2$ $(Y_{\max} - Y_{\min})$, between the center of each bounding box, we used the Euclidean formula, see equation (2), where the distance between pixels is translated in a metric distance (knowing the range and field of view covered by the camera) and then compared to a threshold value:

$$D(C_1, C_2) = \sqrt{(X_{\max} - X_{\min})^2 + (Y_{\max} - Y_{\min})^2}. \tag{2}$$

In case of finding color function detects two bounding boxes and the distance is less than the threshold value, these boxes will have a red color. If this function detects two bounding boxes and the distance is more than the threshold value, the color will be green for these boxes. Figure 2 provides the measured distance (D) between the center of each bounding box for a detected person, where D is the distance between the centers of bounding boxes [25].

After that, the proposed framework with the best trained deep learning model will be implemented on an embedded vision system that consists of Raspberry Pi 4 board and webcam.

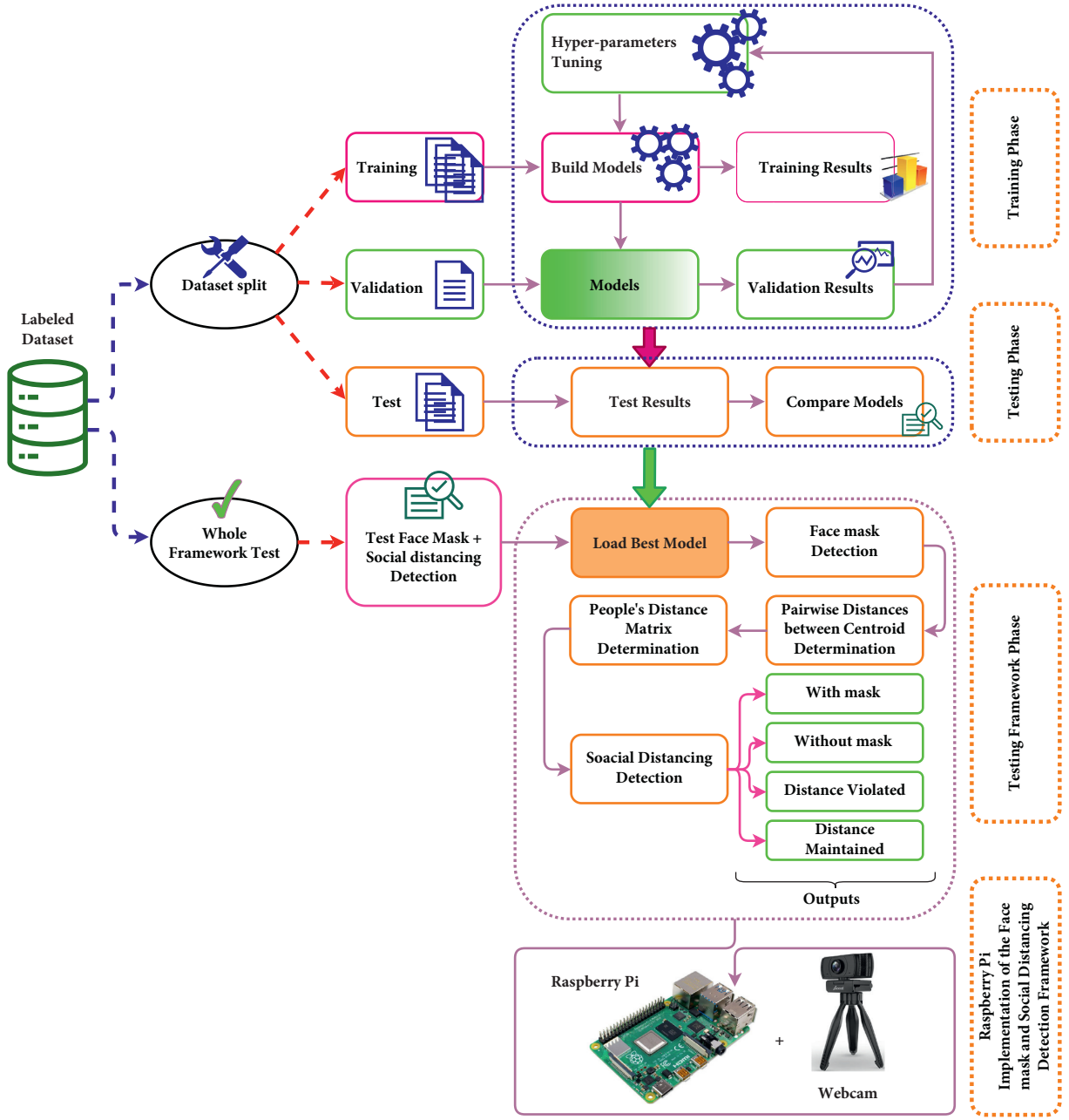


Figure 1: Proposed framework for the face mask and social distancing.

4. Preliminary Study

There are many categories of neural networks such as CNNs, which have proven very powerful in areas such as classification and face recognition. CNNs are a sort of feedforward neural networks which consists of many layers. The structure of CNNs mostly accommodates CNVLs, PLs, rectified linear unit (ReLU) layers, and FC layers. Other structures contain batch normalization layers and softmax and classification layer [26].

4.1. Convolutional Layer. Figure 3 represents the CNVL which is the key construction block of any convolutional networks. The main goal of CNVL is to take out features from the image's data (the input) [26]. In a considerable image, a small section is taken and passed throughout all points in the big image (the input). At the time of passing at every point, they are convoluted within a single position (the output). Each small section which passes over the big image is called kernel or filter [27]. This creates an activation map or a feature map in the output image. After that, the activation maps are sustained like input data to the following CNVL [26].

A typical convolution operation, shown in Figure 3, denotes the input image by X (n_H, n_W, and n_C), where n_H, n_W, and n_C are the height, the width size of the feature map, and the number of channels, respectively, while $K(f, f, n_C)$ is the filter kernel, where $f \times f$ is the size of the convolution

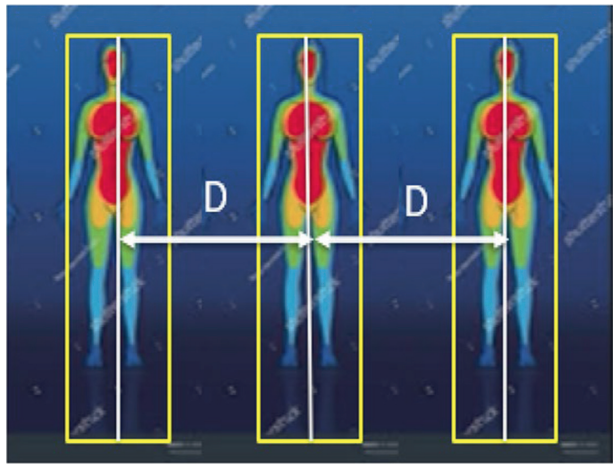


Figure 2: The measured distance (D) between the center of each bounding box.

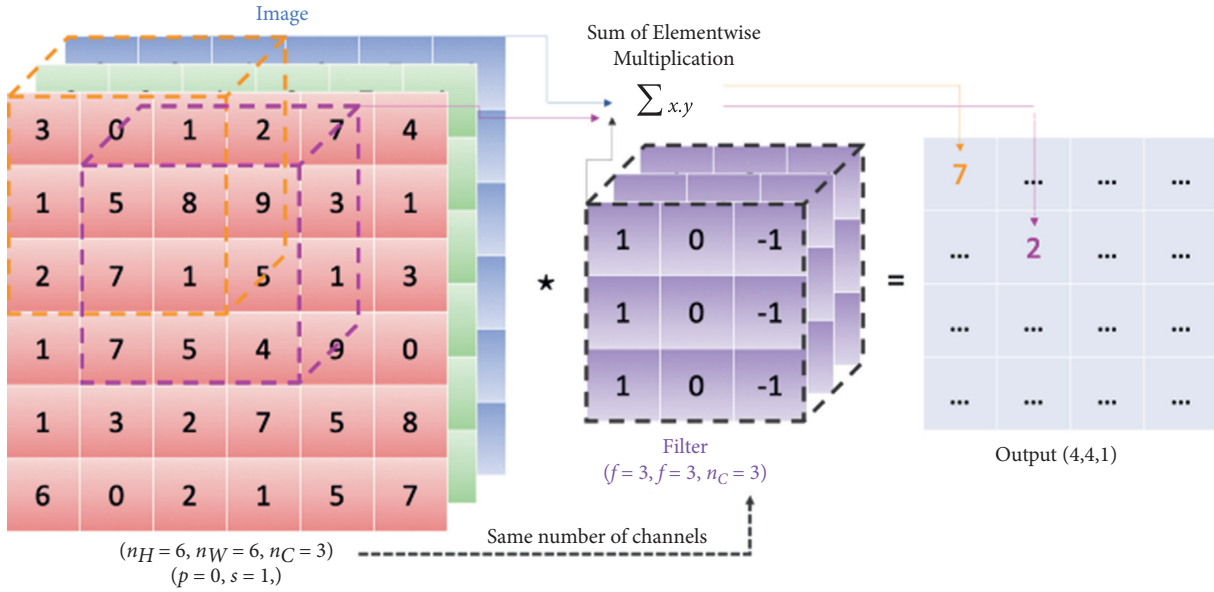


Figure 3: Convolutional layer.

kernel. Thus, the CONV formula is denoted in equation (3), and the output dimension is given by equation (4), where s designates the stride parameter [27, 28]:

$$\mathrm{CONV}(X,K)_{x,y} = \sum_{i}^{n_H}\sum_{j}^{n_W}\sum_{k}^{n_C} K_{i,j,k} X_{x+i-1,y+j-1,k}, \tag{3}$$

$$\begin{aligned}\mathrm{Dim}\Big(\mathrm{CONV}(X,K)_{x,y} &= \left(\left[\frac{n_H + 2_p - f}{s} + 1\right], \left[\frac{n_W + 2_p - f}{s} + 1\right]\right); s > 0,\\ &= \big(n_H + 2_p - f, n_W + 2_p - f\big); s = 0.\end{aligned} \tag{4}$$

4.2. Pooling Layer. Figure 4 depicts an example of max-pooling operation [27]. The pooling layer or subsampling means simply downsampling each image. It reduces the dimension of each activation map but keeps the most necessary information [26]. Therefore, a single output is produced by subsampling a small region of convolutional

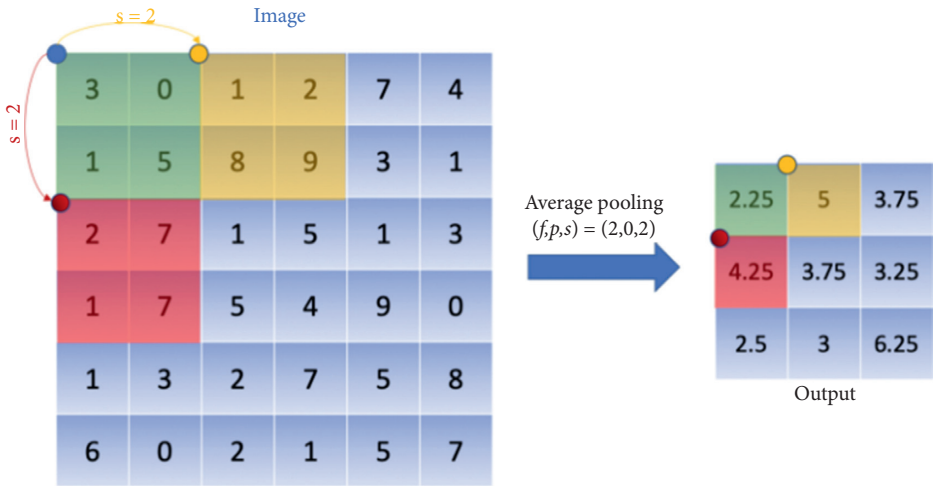


Figure 4: Pooling layer.

output. The max pooling, the average pooling, and the mean pooling are pooling techniques. Max pooling takes the biggest pixel value of the region [27]. Equations (5) and (6) present how to calculate the max pooling and the average pooling, respectively [29]. The main advantage of this layer is achieving faster convergence, better generalization, robust to distortion, and translation and is habitually placed in the middle of convolution layers [26]:

$$\text{Max}_i = \max_{1 \longrightarrow f \times s} (X), \tag{5}$$

$$\text{Avg}_i = \frac{1}{f \times s} \sum_{1}^{f \times s} X. \tag{6}$$

There are alternatives to CNNs which allow to further decrease the parameters. Among these options, one can cite stride [30].

4.3. Rectified Linear Unit Layer. The rectified linear unit layer, known as the ReLU layer, is a nonlinearity activation operation, applied in feature maps produced by the convolutional layers. Equation (7) presents how to calculate the ReLU [31]:

$$\text{Relu}\,(g(x) = \max(0, x)). \tag{7}$$

It is an operation which replaces all the negative values in each feature map by zero [26].

4.4. Fully Connected Layer. Figure 5 depicts the FCL [26, 27]. However, it is a finite number of neurons that takes one vector as input and return another. Let us consider a j^{th} node of an i^{th} layer, and the output Z_j^i is defined as equation (9):

$$Z_j^{[i]} = \sum_{l=1}^{n_{i-1}} W_{j,l}^{[i]} a_l^{[i-1]} + b_j^{[i]}. \tag{8}$$

The $a^{[i-1]}$ is denoted as a convolution or a pooling result with a dimension of $(n_H^{[i-1]}, n_W^{[i-1]}, n_C^{[i-1]})$. Therefore, to plug the fully connected layer, we flatten the tensor to a 1-dimension vector having the dimension $(n_H^{[i-1]} \times n_W^{[i-1]} \times n_C^{[i-1]}, 1)$; thus,

$$n_{i-1} = n_H^{[i-1]} \times n_W^{[i-1]} \times n_C^{[i-1]}. \tag{9}$$

However, the learned parameters at the l^{th} layer are the weights $w_{j,l} \times n_l$ parameters and the bias with n_l parameters.

4.5. The Batch Normalization Layers. To reduce the training time of any CNN and the sensitivity to initialize the network, we used the batch normalization layers.

The input (x_i), the minibatch mean (m_b), and also minibatch variance (v_b) are the three variables to compute the normalized activations. The formula is presented in the following equation:

$$\widehat{x}_i = \frac{x_i - m_b}{\sqrt{v_b^2 + \theta}}, \tag{10}$$

where θ is a constant which develops the numerical state if the v_b is small. Equations (11) and (12) present the calculation of m_b and v_b, respectively:

$$m_b = \frac{1}{n} \sum_{i=1}^{n} x_i, \tag{11}$$

$$v_b = \frac{1}{n} \sum_{i=1}^{n} (x_i - m_b)^2. \tag{12}$$

In the batch normalization layers, the activations are calculated as shown in

$$y_i = a * \widehat{x}_i + b, \tag{13}$$

where a is a balance factor and b is a scale factor. During the training process, these factors are two learnable factors renovated to the most suitable values [31].

4.6. Softmax and Classification Layer. The classification layer is habitually the last layer in a CNN. Softmax function is utilized generally in CNNs, in order to match

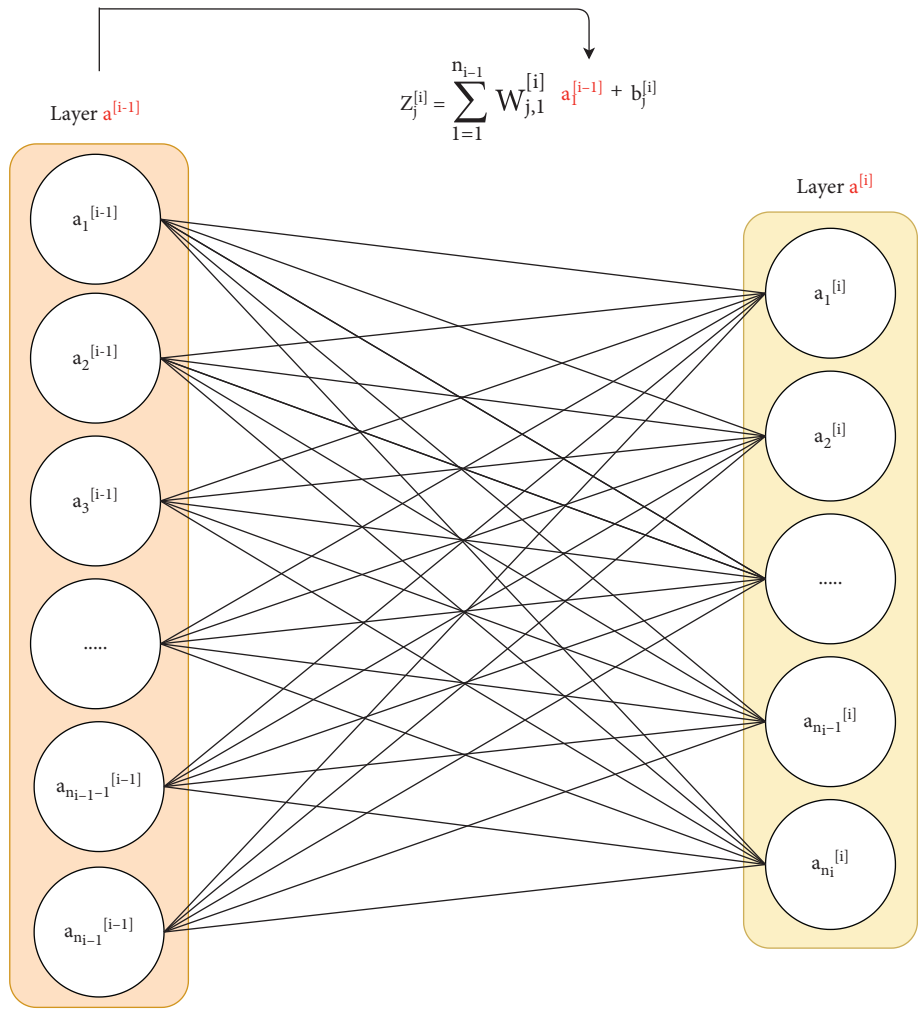


Figure 5: Fully connected layer.

nonnormalized values of the previous layer to allow distribution of above-predicted class scores. Equation (14) designs the softmax function:

$$\sigma(x_i) = \frac{e^{x_i}}{\sum_{K}^{j=1} e_j^x}, \quad j = 1, \ldots, K, \tag{14}$$

where σ refers to the softmax output corresponding to each x_i and x_j denotes the input vector values [31].

5. Dataset Collection

Figure 6 illustrates an example of faces wearing and not wearing masks. The experiments of this research are conducted on one original dataset. It consists totally of 3835 images. This is a balanced dataset containing two categories, faces with masks (1919 images) and without masks (1916 images) with a mean height of 283.68 and mean width of 278.77. It comprises two categories. This dataset is used not only for training and validation, but also for testing, and if an individual is wearing a mask or not, then the social distance between two individuals will be estimated (violated distance alert or not) [32].

6. Evaluation Metrics

Accuracy is the overall number of the correct predictions fractionated by the whole number of predictions created for a dataset. It can inform us immediately if a model is trained correctly and by which method it may perform in general. Nevertheless, it does not give detailed information concerning its application to the issue. Precision, called PPV, is a satisfactory measure to determination, whereas the false positives cost is high. Recall is the model metric used to select the best model when there is an elevated cost linked with false negative. Recall helps while the false negatives' cost is high. F1-score is required when you desire to seek symmetry between both precision and recall. It is a general measure of the accuracy of the model. It combines precision and recall. A good F1-score is explained by having low false positives and also low false negatives.

Figure 6: Dataset example.

Equations (15), (16), (17), and (18) present how to calculate the accuracy, precision, recall, and F1-score, respectively [1]:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{(\text{TP} + \text{FP}) + (\text{TN} + \text{FN})}, \tag{15}$$

$$\text{precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})}, \tag{16}$$

$$\text{recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})}, \tag{17}$$

$$\text{F1} - \text{score} = \frac{2 * \text{Precision} * \text{Recall}}{(\text{Precision} + \text{Recall})}, \tag{18}$$

with TP being the computation of the samples of true positives, TN is the calculation of the samples of true negatives, FP is the counting of the samples of false positives, and FN is the enumeration of the samples of false negatives, from a confusion matrix. The sensitivity and specificity are two statistical measures of the binary classification test performance which are largely used in medicine. Sensitivity, known as true positive rate, measures the proportion of positives which are correctly identified. The specificity, known as true negative rate, measures the proportion of negatives which are correctly identified.

The terms "TP," "FP," "TN," and "FN" refer to the test result and the classification correctness. As a sample, if a disease is the condition, "TP" signifies "correctly diagnosed as diseased," "FP" refers to "incorrectly diagnosed as diseased," "TN" means "correctly diagnosed as not diseased," and "FN" denotes "incorrectly diagnosed as not diseased." Therefore, if the sensitivity of the test is 98% and the specificity is 92%, then the rate of the false negatives is equal to 2% and the rate of the false positives is equal to 8%.

Macro-averaging is used for models with 2 targets and more. Some macro-averaged measures are described [33]. First, macro-averaged precision computes the average precision per each class. It is known as macro-precision. Macro-precision score can be determined arithmetically by the mean of all the precision scores of the different classes. It is defined in equation (19) by

$$\text{macro} - \text{precision} = \frac{\sum_{i=1}^{n} \text{Precision}_i}{\text{number of classes}}. \tag{19}$$

Macro-precision is low for models which not only accomplish well on common classes but also accomplish poorly on rare classes. Thus, it is a harmonious metric to the all-inclusive accuracy. Second, macro-average recall is the mean of recall scores of all different classes. It is known as macro-recall. We can compute the macro-recall as

$$\text{macro} - \text{recall} = \frac{\sum_{i=1}^{n} \text{Recall}_i}{\text{number of classes}}. \tag{20}$$

Third, the macro-averaged F1-score, also called the macro F1-score, represented the harmonic mean of the macro-precision and the macro-recall. Equation (21) shows how to calculate macro F1-score:

$$\text{macro F1} - \text{score} = 2 * \frac{MAP * MAR}{MAP^{-}1 + MAR^{-}1}, \tag{21}$$

where MAP denotes macro average precision and MAR refers to the macro-average recall.

Weighted average (weighted avg) is a computation that accounts for the varying degrees of the numbers' importance in a dataset. When a weighted avg is calculated, each number

in the dataset is multiplied by a prearranged weight before the final calculation. A weighted avg can be more accurate than a simple average in which all numbers in a dataset are assigned an identical weight.

To explain in depth the formula of the weighted avg, we follow these steps: determine the weight of all data point, multiply the weight by every value, and the results of step two are added together. Among weighted avg scores, we find weighted avg precision, weighted avg recall, and weighted avg F1-score.

7. Numerical Results

In this section, the numerical results will be introduced. For all simulated deep learning models, as well as DenseNet, InceptionV3, MobileNet, MobileNetV2, ResNet-50, VGG-16, and VGG-19, the TensorFlow-GPU is used as a deep learning framework to train the deep learning models. The hyperparameters used in our experiments are summarized as follows: the batch size is set to 32, the training epochs are from 20 to 40, and the learning rate is set to 0.0001, with the Adam optimizer used to update network weights. The training platform uses Windows 10 OS with Intel® Core TM i7-3770 @3.4 GHz CPU and 16 GB RAM and an NVIDIA GeForce RTX 2070 GPU.

7.1. DenseNet Results. DenseNet is a contemporary architecture of CNN. Among distinct DenseNet (DenseNet-201, DenseNet-160, DenseNet-121, etc.), in our study we employed DenseNet-201. Our DenseNet model was trained to classify images into masked faces and unmasked faces for 20 epochs. The training and validation loss and accuracy graphs of DenseNet-201 are shown in Figures 7(a) and 7(c), respectively. Figure 7(a) shows that after inputting the data of our algorithm, this model nearly tends to maintain a high accuracy greater than 80% without overfitting.

A confusion matrix is a particular table layout which allows to visualise the performance of the algorithm. For the trained model, we can compute a plenty of parameters, based on TP, FP, FN, and TN. In our case, TP means that a human is wearing a mask and the system shows a person wearing a mask, FN means that a human is not wearing a mask but the system shows a person not wearing a mask, FP means that a human is wearing a mask but the system shows a person not wearing a mask, and TN means that the human is not wearing a mask and the system shows that person not wearing a mask. Figure 7(b) illustrates the confusion matrix for the DenseNet model in the testing phase. Table 3 shows all the evaluation parameters: precision, recall, F1-score, support, accuracy, sensitivity, and specificity, for masked faces and unmasked faces cases, macro avg precision, macro avg recall, macro avg F1-score, macro avg support, weighted avg precision, weighted avg recall, weighted avg F1-score, and weighted avg support. For the cases of masked faces, the resulted surpassed 0.92 accuracy and F1-score, with 0.91 recall, accuracy, and sensitivity.

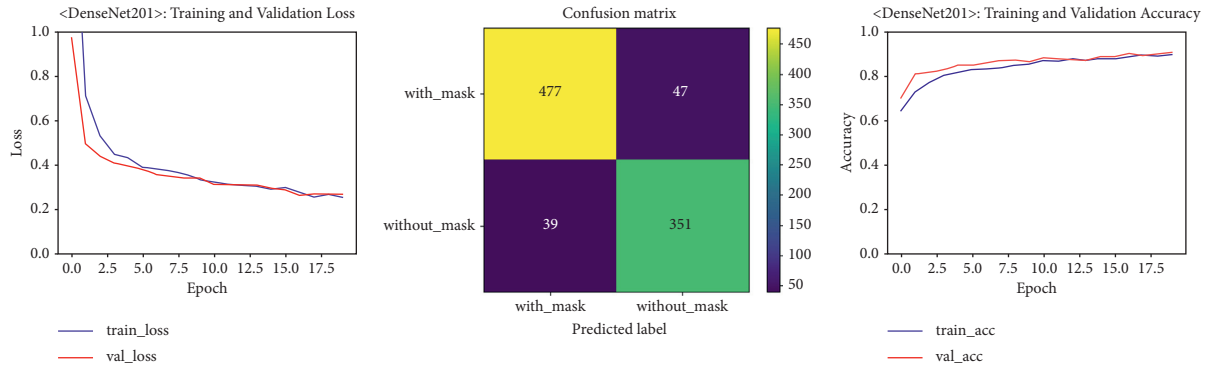
Different DenseNet architectures have been used in research. In this context, numerous works focused on detection of COVID-19 from CXR or CT images. In [34], a DenseNet-121 model used a promising technique to predict COVID-19 patients from 2482 CT images using CNN. It achieved a total accuracy of 92%, a precision of 84%, a recall of 95%, an F1-score of 89%, and a macro-precision, F1-score, recall, and weighted-precision of 92%. All parameters are over 84%. Therefore, DenseNet-121 is an efficient model to detect COVID-19. In [35], an approach used DenseNet-121 to detect COVID-19 patients from radiology images. The model was trained and tested on a COVIDx dataset of 13800 images corresponding to 13725 patients. It achieved 96.49% and 93.71% accuracies for two-class and three-class classifications. These results outperformed the robustness of the model. In [36], an automated method was used to classify CT and X-ray images into coronavirus, bacterial pneumonia, and normal classes using DenseNet-201 architecture. Regarding coronavirus class, specificity and precision were satisfactory, with rates of 96.10% and 90.33%, respectively. As the sensitivity, it was equitable and reached 75.14%. We can explain these values by the fact that the sum of the TNs was high, the sum of the FPs was low, and the sum of the FNs was low, respectively. For bacteria class, the sum of FNs was low, which justifies the acceptable sensitivity (92.95%). The sum of TNs was relatively high, and the sum of the FPs was relatively low, which justifies the reasonability of specificity and precision values. For normal class, the sum of FNs was low, the sum of FPs was low, and the sum of TNs was high, which justifies the good sensitivity (95.88%), precision (92.59%), and specificity (96.28%), respectively. Here, FNs, TNs, and FPs mean false negatives, true negatives, and false positives, respectively.

In the end, we can say that the family of DenseNet is a methodical architecture to detect COVID-19 patients and faces with or without mask.

7.2. InceptionV3 Results. InceptionV3 is one of the CNNs dedicated for classification. It contains 48 deep layers and employs inception modules, which requires a connected layer with 1×1, 3×3, and 5×5 convolutions. It is referred to as a GoogleNet architecture. Figure 8(a) shows the training and validation loss for our InceptionV3 model, as it decreases with the continuous epochs, to achieve 0.4. Figure 8(c) depict the training and validation accuracy of our model as it upgrades with the successive epochs achieving a high accuracy greater than 80% without overfitting. Figure 8(b) shows the confusion matrix of testing data of the InceptionV3 model.

When discussing the evaluation parameters in Table 3, we note that precision, recall, F1-score, and support, obtained in "masked faces" case, are higher than "unmasked faces" case. The accuracy and the sensitivity are over 80%. To conclude, InceptionV3 learned the information well, but DenseNet is better.

Among works which interested in detecting and analyzing COVID-19 on chest X-ray images, we cite the work in [37], which utilized a method based on the InceptionV3 model. 6432 images collected from the Kaggle repository were used for training, validation, and testing phases. Three



(a) (b) (c)

Figure 7: DenseNet evaluation metrics, (a) training and validation loss, (b) confusion matrix, and (c) training and validation accuracy.

Table 3: Performance evaluation of the proposed models.

		Precision	Recall	F1-score	Support	Accuracy	Sensitivity	Specificity
DensNet	With mask	0.92	0.91	0.92	524	0.91	0.91	0.99
	Without mask	0.88	0.90	0.89	390			
	Macro avg	0.90	0.91	0.90	914	—	—	—
	Weighted avg	0.91	0.91	0.91	914	—	—	—
InceptionV3	With mask	0.83	0.85	0.84	524	0.88	0.86	0.77
	Without mask	0.80	0.77	0.78	390			
	Macro avg	0.81	0.81	0.81	914	—	—	—
	Weighted avg	0.82	0.82	0.82	914	—	—	—
MobileNet	With mask	0.98	0.98	0.98	524	0.98	0.98	0.98
	Without mask	0.97	0.98	0.98	390			
	Macro avg	0.98	0.98	0.98	914	—	—	—
	Weighted avg	0.98	0.98	0.98	914	—	—	—
MobileNetV2	With mask	0.95	0.96	0.96	524	0.95	0.96	0.94
	Without mask	0.95	0.94	0.94	390			
	Macro avg	0.95	0.95	0.95	914	—	—	—
	Weighted avg	0.95	0.95	0.95	914	—	—	—
ResNet-50	With mask	0.99	0.98	0.99	524	0.99	0.98	0.99
	Without mask	0.97	0.99	0.98	390			
	Macro avg	0.98	0.99	0.99	914	—	—	—
	Weighted avg	0.99	0.99	0.99	914	—	—	—
VGG-16	With mask	0.99	0.98	0.99	524	0.99	0.98	0.99
	Without mask	0.98	0.99	0.99	390			
	Macro avg	0.99	0.99	0.99	914	—	—	—
	Weighted avg	0.99	0.99	0.99	914	—	—	—
VGG-19	With mask	0.99	0.99	0.99	524	0.99	0.99	0.99
	Without mask	0.98	0.99	0.99	390			
	Macro avg	0.99	0.99	0.99	914	—	—	—
	Weighted avg	0.99	0.99	0.99	914	—	—	—

classes of classification (normal, COVID-19, and pneumonia) are discussed. The highest precision (0.97) is obtained for detecting COVID-19. The highest recall (0.97) and F1-score (0.95) are obtained for detecting pneumonia. And the overall accuracy of this model is 0.93. We can say that InceptionV3 is a satisfactory model. Also, in [36], an automated method aimed to classify CT and X-ray images into coronavirus, bacterial pneumonia, and normal classes using InceptionV3 architecture. Regarding coronavirus class, it was identified quite well because of the reasonable sensitivity and precision values and the good specificity whose respective values were 75.88%, 88.06%, and 95.01%. These values are explained as follows: the sum of FNs is practically low, the sum of FPs is relatively low, and the sum of TNs is high, respectively. Bacteria class is distinguished well since specificity and sensitivity are equivalent to 90.00% and 92.42% and with the tolerable precision (83.06%). The high value of the sum of TNs and the low value of the sum of FNs explain the values of specificity and sensitivity, respectively. On the other hand, the low value of the sum of TNs justifies the value of precision (83.06%). The normal class is well identified since sensitivity, precision,

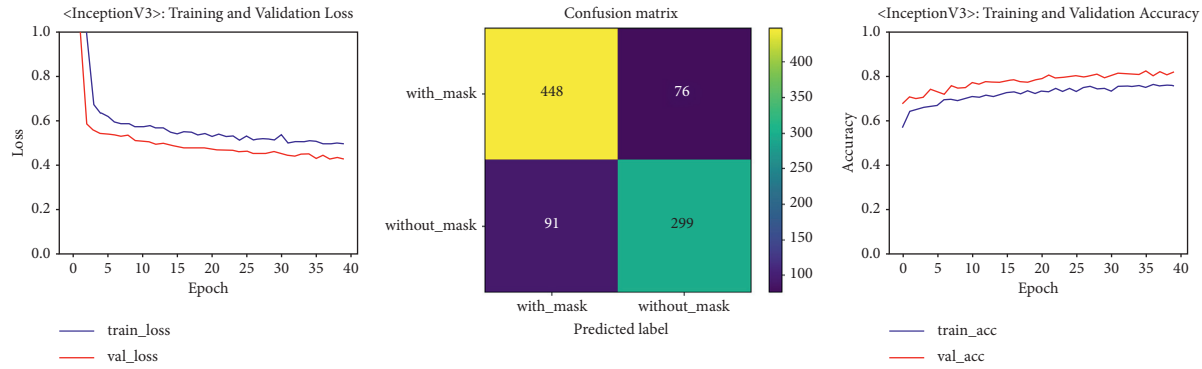


Figure 8: InceptionV3 evaluation metrics, (a) training and validation loss, (b) confusion matrix, and (c) training and validation accuracy.

and specificity attained 95.51%, 93.76%, and 96.91%, respectively. The low values of the sum of FNs and FPs and the high value of the sum of TNs explain the performance of the evaluation parameters, respectively.

To conclude, we can affirm that the InceptionV3 is a systematic architecture to detect COVID-19, pneumonia, or normal patients as well as masked and unmasked faces.

7.3. MobileNet Results. MobileNets are one of CNN-based networks, which are primarily built from depthwise separable convolutions. Figures 9(a) and 9(c) analyze the results of the training and validation loss and accuracy of the MobileNet model, respectively. After inputting the data of our algorithm, the graphs of loss nearly tended to zero and the graphs of accuracy showed that after five training epochs, the model maintained a high accuracy close to 100% without overfitting. The confusion matrix of the MobileNet model during the testing phase is shown in Figure 9(c). Table 3 shows that all the evaluation parameters values, especially support, are over 0.97. Therefore, the MobileNet is a methodical model to detect masked faces and unmasked faces. In fact, MobileNet outperforms previous models.

Among the several works interested in detecting COVID-19, a DNN named RAM-Net is presented in [38]. The RAM-Net is a combination of MobileNet with DDSC which is a dilated depthwise separable convolution, three residual blocks and two attention augmented convolution. One of the most popular public datasets containing CXR images is used to learn and validate the network. This dataset is called COVIDx. Thanks to this network, positive COVID-19 cases can be identified. While testing the network on the COVIDx dataset, the overall accuracy, precision, and sensitivity achieved are, respectively, 95.33%, 99%, and 92%.

In conclusion, we can say that when using MobileNet architecture solely or combined with other blocks for object detection, we always achieve high performances. Moreover, it is useful for a lot of object detection.

7.4. MobileNetV2 Results. MobileNet is one of the deep learning models intended to be utilized in low-hardware cost gadgets. Classification, segmentation, and object identification can be performed by operating the MobileNet model. The MobileNetV2 model is developed from the MobileNetV1. Figure 10(a) presents the MobileNetV2 training and validation loss. Also, Figure 10(c) shows the graphs of training and validation accuracy. Then, after seven training epochs, the graph shows that this model is prone to overfit but still gives a high accuracy close to 100%. Figure 10(b) illustrates the confusion matrix of the MobileNetV2 model in the testing phase. In Table 3, we note that all evaluation parameters except for the support are over 0.94. This means that the MobileNetV2 is well trained and is an efficient model in detecting faces with and without masks. Comparing this model with the previous ones, we see that it is a little bit less efficient than MobileNet, but more coherent than DenseNet and InceptionV3.

Many studies are focused on detecting COVID-19 using MobileNetV2 as a classifier. Among them, we cite the classifier presented in [39]. The principal aim of the authors is to distinguish between people who are normal, people who have pneumonia, and people having COVID-19 (with damaged lungs), from CXR images. The overall testing accuracy of this model is 96.32%. The testing F1-score, sensitivity, specificity, precision, and accuracy obtained when classifying COVID-19 data are, respectively, 99.43%, 100%, 97.72%, 98.87%, and 99.24%. The values of these metrics in the classification of normal and pneumonia individuals are all over 89%.

It is obvious that the MobileNetV2 model could contribute to detecting not only COVID-19 disease efficiently but also faces with or without masks.

7.5. ResNet-50 Results. ResNet is the abbreviation of Residual Networks. It is a network employed as a backbone for countless computer vision tasks and a winner of the Image Net challenge in 2015. It is a variant of the ResNet model family. It consists of 48 convolutional layers with 1 max pooling and also 1 average pooling layer. As Figures 11(a) and 11(c) of the training and validation accuracy, and loss, respectively, show, the loss nearly tends to zero and the accuracy is high close to 100%. The confusion matrix after testing is given in Figure 11(b). When evaluating the parameters in Table 3 of the ResNet-50 model, we note that all parameters' values except for "support" are over 0.97.

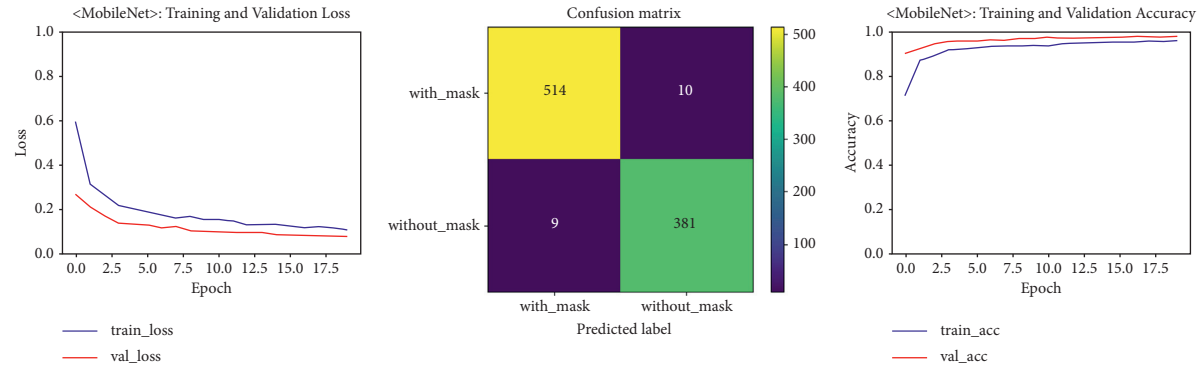


Figure 9: MobileNet evaluation metrics, (a) training and validation loss, (b) confusion matrix, and (c) training and validation accuracy.

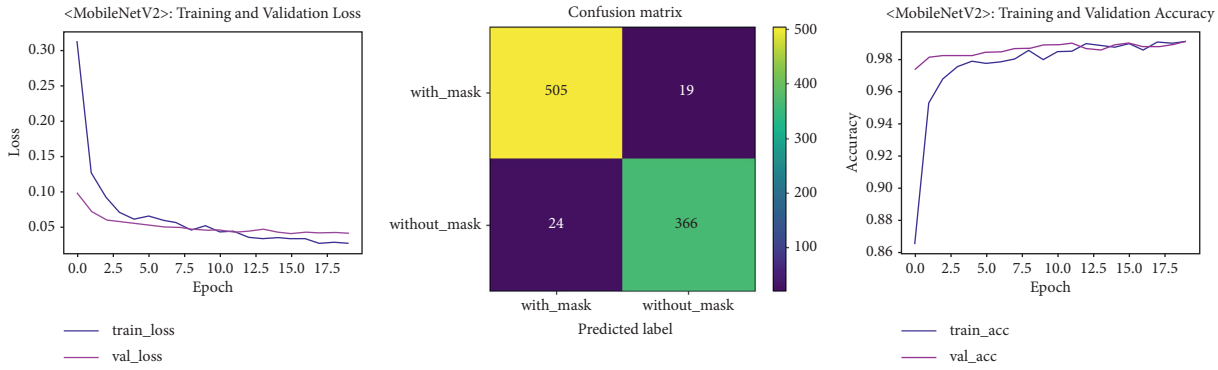


Figure 10: MobileNetV2 evaluation metrics, (a) training and validation loss, (b) confusion matrix, and (c) training and validation accuracy.

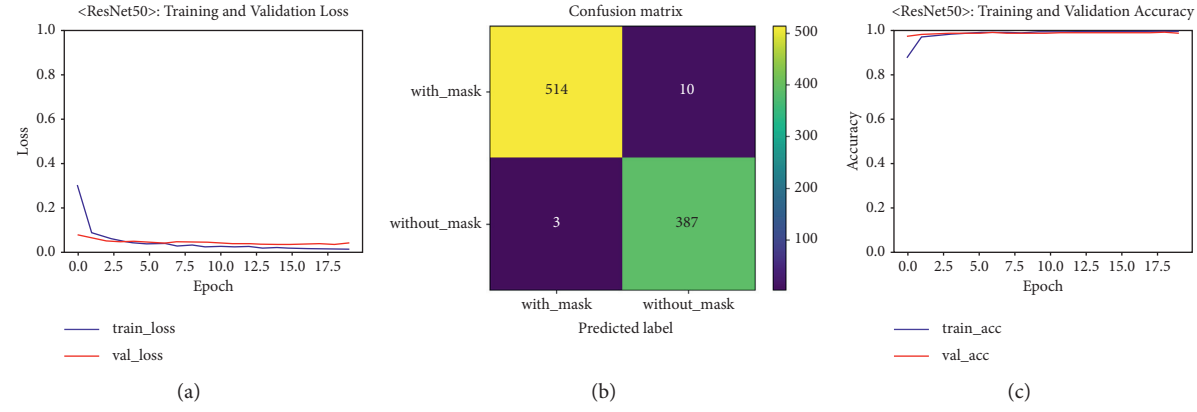


Figure 11: ResNet-50 evaluation metrics, (a) training and validation loss, (b) confusion matrix, and (c) training and validation accuracy.

Compare ResNet-50 with the other four preceding models, and we note that ResNet is the best.

In AI, many research works are focused on detecting objects using the ResNet-50 model as a classifier. In [40], the authors are concentrated on detecting and classifying COVID-19 individuals with the ResNet CNN architecture. The used CXR images are created with more than one dataset. The sources of the created dataset are as follows: SIRM, which is the Italian Society of Medical Radiology, dataset generated by assembling diagnosed images from different articles, coronavirus open-source shared dataset, and CXR image dataset. Augmentation techniques are elaborated due to the tiny dataset. The dataset consists of two classes: people having COVID-19 and normal people. By

appealing 5-fold-cross-validation, the results are evaluated. As a result, a classification accuracy of 99.5% is achieved. Therefore, the obtained results are encouraging regarding the exploitation of computer-aided models, especially in the pathology field. It can be also operated in situations when the possibilities are deficient, such as RT-PCR tests, radiologist, and doctor.

7.6. VGG-16 Results. VGG-16 is a CNN proposed by A. Zisserman and K. Simoniyan. They utilized small convolutional filters and a stride of 1. This CNN has 16 layers. Figure 12(a) shows that both the training loss and validation loss are reduced ensuing each epoch for the VGG-16 model. The graph nearly tends to zero after 10 epochs. Moreover, Figure 12(c) shows the overall training and validation accuracy throughout each epoch for the VGG-16 model. Then, after six training epochs, the graph shows that this model maintains a high accuracy close to 100% at epochs 6 to 20. The confusion matrix for the VGG-16 model in the testing phase is presented in Figure 12(b). Table 3 reveals all the evaluation measures of VGG-16. It shows over 0.98 performance beyond all measures except for the "support." These values are indicative of strong performance of VGG-16. Compared with other models, we can say that VGG-16 and ResNet-50 have the same effectiveness and outperformed the DenseNet, InceptionV3, MobileNet, and MobileNetV2 models.

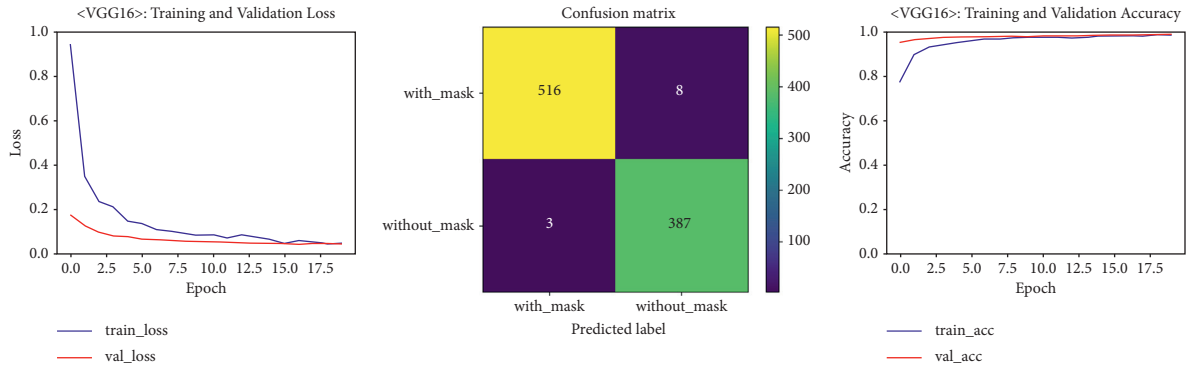
VGG-16 is exploited in many research works. In [41], a deep CNN method "VGG-16" is demonstrated not only to detect but also to diagnose COVID-19 cases using CXR images. Three separate studies in this article with three different datasets are used. According to study one, a miniature and balanced dataset is used. It contains CXR images of 50 patients acquired from an open-source repository given by Dr. Joseph Cohen. The performance of the VGG-16 is weighted on both the training and the test sets. It showed a 100% performance covering all measures (accuracy, precision, recall, and F1-score) in the two sets. Concerning study two, an imbalanced and a larger dataset is elaborated. It includes 1845 CXR images of patients, obtained from the Kaggle COVID-19 dataset. VGG-16 achieves 99% across all metrics in the training phase. However, the performance results of accuracy, precision, recall, and F1-score are, respectively, 99%, 100%, 97%, and 98%. During stage three, the VGG-16 is implemented on the multiclass dataset with 2637 images procured also from the Kaggle COVID-19 X-ray dataset. The accuracy remained over and above 90% on the training and test runs. Therefore, VGG-16 shows an extremely high performance with both binary and multiclass datasets.

7.7. VGG-19 Results. VGG-19 is a CNN also proposed by A. Zisserman and K. Simoniyan. It has 19 layers. As a result of the three more layers than VGG-16, the number of parameters in VGG-19 is greater than VGG-16. Therefore, it is more costly to train. Figure 13(a) provides evidence that both training and validation losses were minimized following each epoch for the VGG-19 model. It shows that the graph nearly tends to zero. Moreover, Figure 13(c) suggests that the overall training and validation maintain a high accuracy close to 100% without overfitting, after seven training epochs. Figure 13(b) illustrates the confusion matrix for the VGG-19 model in the testing phase. It shows that the VGG-19 performance is satisfactory on the test set. Table 3 expresses the evaluation metrics of the VGG-19 model. All the metric values, except for the support, are over 0.98. Compare the performance of this model with the previous ones, and we note that the VGG-19 is the best one.

Many researchers applied nonidentical deep learning algorithms for detecting COVID-19 automatically, and many of these algorithms reported a significant accuracy. In [42], a current deep learning framework is proposed to identify the presence or absence of COVID-19. The main purpose of the authors is to judge the effectiveness of the pretrained VGG-19 architecture in detecting COVID-19 cases using CXR images. The used dataset is gathered from different hospitals in Tehran, Iran. It includes 464 high quality images; 345 of them are for COVID-19 cases, and 119 are for normal cases. Posterior-anterior (PA) projection is used to train the model. 5-fold-cross-validation is applied to use all images in the training phase. In each fold, 20% of all images are used for testing and 80% for both training and validation.

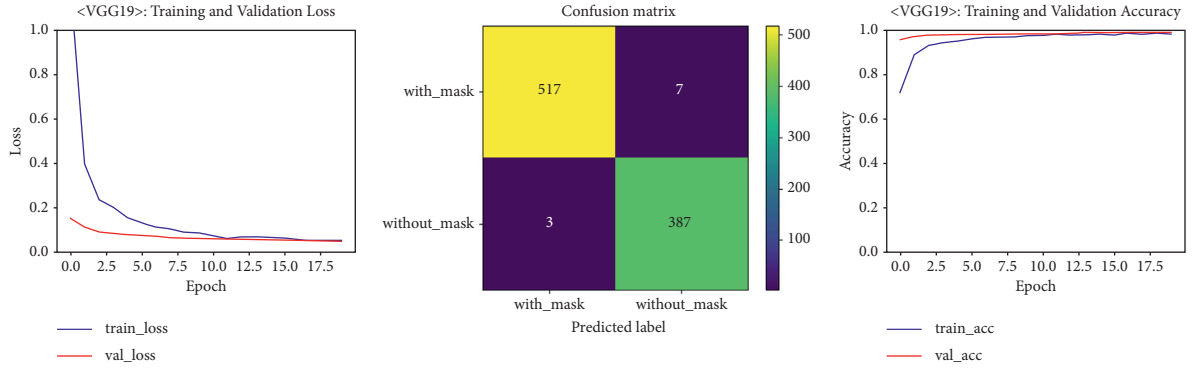
The network's weights are initialized with the pretrained model's weights on the ImageNet database. The Adam optimizer with two standard parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and the batch size equal to 16 are used in the training phase of the model. The learning rate is initialized at 1e-5. After that, it decreased by 0.2. 1e-7 is the minimum learning rate. If the accuracy of validation does not boost after 20 epochs, the training will stop. Furthermore, a heatmap is created to aid radiologists in refining decision-making. A PC, with Intel Core i9 CPU and NVIDIA GeForce GTX 2070 with 8 GB of RAM, is used to conduct the experiment. Many metrics, such as accuracy, specificity, sensitivity, F1-score, and AUROC [43], are calculated in order to estimate the model's effectiveness. The heatmaps are plotted to confirm that the VGG-19 is extracting the valid features in order to distinguish COVID-19 versus normal cases. A high AUROC equal to 0.91(0.03) is achieved by the neural network in 2-class classification. When classifying COVID-19 VS normal cases, VGG-19 achieved different sensitivities equal to 94.21%, 87.18%, and 71.91% when the specificities are equal to 60%, 75%, and 90%. Therefore, the VGG-19 is a model structured to detect COVID-19 cases, in addition to masked and nonmasked faces.

7.8. Comparative Study between the Proposed Models. Table 3 shows the performance of the different models network. These metrics are as follows: precision, recall, F1-score, support, accuracy, sensitivity, and specificity, for masked and unmasked cases, macro avg precision, macro avg recall, macro avg F1-score, macro avg support, weighted avg precision, weighted avg recall, weighted avg F1-score, and weighted avg support. We note that the highest precision in detecting masked faces case is for ResNet-50, VGG-16, and VGG-19 models. The highest precision in detecting unmasked faces case, highest



(a) (b) (c)

Figure 12: VGG-16 evaluation metrics, (a) training and validation loss, (b) confusion matrix, and (c) training and validation accuracy.



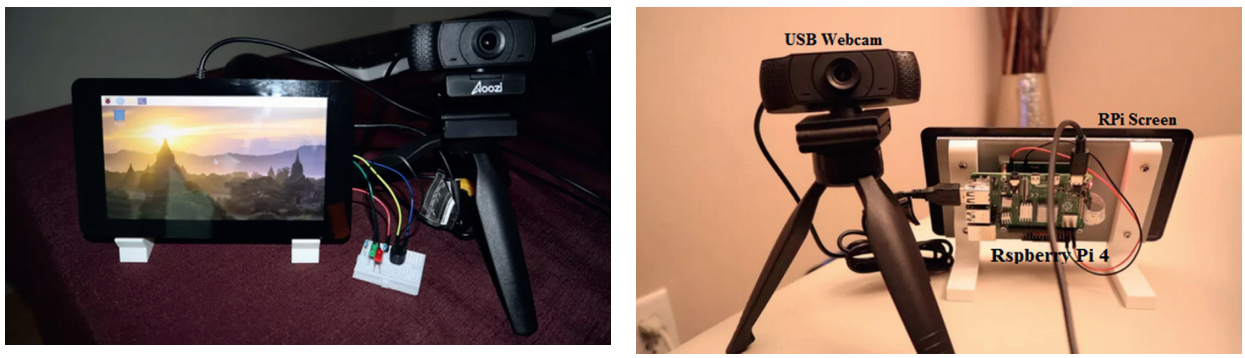
(a) (b) (c)

Figure 13: VGG-19 evaluation metrics, (a) training and validation loss, (b) confusion matrix, and (c) training and validation accuracy.

macro avg precision, and weighted avg precision are for VGG-16 and VGG-19 models. The highest recall in detecting people wearing mask is for the VGG-19 model. The highest recall in detecting people not wearing mask case, the highest macro avg recall, and the highest weighted avg recall are for ResNet-50, VGG-16, and VGG-19 models. The highest F1-score in wearing mask case, the highest macro avg F1-score, and the highest macro weighted F1-score are for ResNet-50, VGG-16, and VGG-19 models. However, the highest F1-score in not wearing mask case is only for VGG-16 and VGG-19 models. The support in wearing and not wearing mask cases, macro avg support, and weighted avg support have the same values for the different models. The highest accuracy and specificity are for ResNet-50, VGG-16, and VGG-19 models. Finally, the highest sensitivity is for the VGG-19. Therefore, we can say that the VGG-19 is the best trained model, when we compared it with the other models.

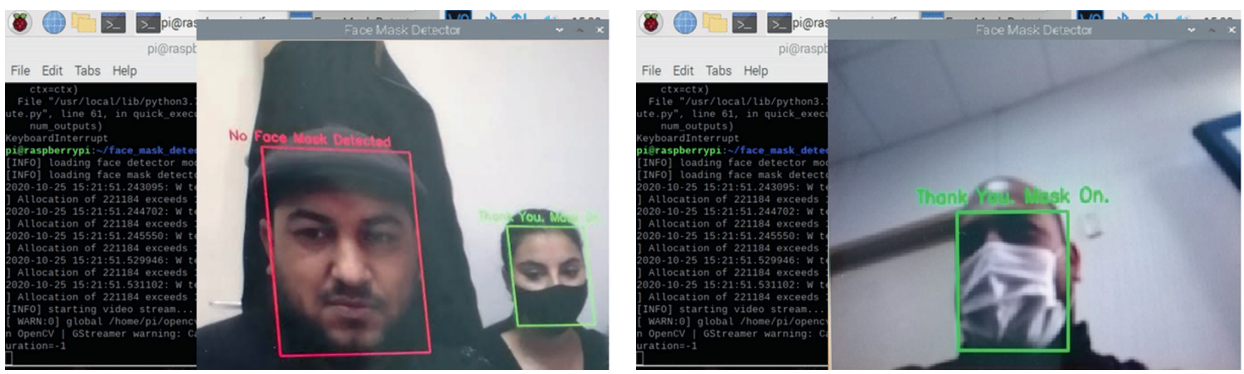
7.9. Embedded Face Mask and Social Distancing Detection: Raspberry Pi Implementation. After evaluating the proposed face mask detection models, in this step, the best model with high accuracy rate (ResNet-50, VGG-16, and VGG-19) will be applied to the embedded vision system. Figure 14 depicts the proposed embedded vision system that consists of a Raspberry Pi 4 platform coupled with a webcam and touchscreen and sounds a buzzer when someone is not wearing their face mask (green or red LED) or social distancing is violated.

Thus, after installing Raspberry Pi OS and all libraries, such as TensorFlow, OpenCV, and imutils, the embedded vision system will be able to detect if a user is wearing a face mask or not and if the distance between peoples is maintained or violated. Figure 15 shows the implementation results. Hence, when someone is not wearing a face mask, it will be designated with a red box around their face with the text, "No Face Mask Detected," and when wearing a face mask, it will be seen a green box around their face with the text, "Thank you. Mask On." The same thing is depicted by Figures 15(a) and 15(b). On the other hand, the proposed model with social distancing task detects peoples and provides the bounding box information. After that, the Euclidean distance between each detected centroid pair is computed using the detected bounding box and its centroid information based on (x, y) dimensions for each bounding box. Figure 15(c) illustrates the social distancing detection task where an alert message displayed with a red box for violated distance and a green box for the maintained distance.



(a) (b)

Figure 14: Embedded vision system for face mask and social distancing detection.



(a) (b)

(c)

Figure 15: Test of the proposed framework on the embedded vision system.

8. Conclusion

Due to the urgency of controlling COVID-19, the application value and importance of real-time mask and social distancing detection are increasing. This work reviewed, firstly, many research works that seek to surround COVID-19 outbreak. Then, it clarified the basic concepts of deep CNN models. After that, this paper reproduced the training and testing of the most used deep pretrained-based CNN models (DenseNet, InceptionV3, MobileNet, MobileNetV2, ResNet-50, VGG-16, and VGG-19) on the face mask dataset. Finally and after evaluated the numerical results, best models are tested on an embedded vision system consisted of Raspberry Pi board and webcam where efficient real-time deep learning-based techniques are implemented with a social distancing task to automate the process of detecting masked faces and violated or maintained distance between peoples.

This embedded vision-based application can be used in any working environment such as public place, station, corporate environment, streets, shopping malls, and examination centers, where accuracy and precision are highly desired to serve the purpose. It can be used in smart city innovation, and it would boost up the development process in many developing countries. Our framework presents a chance to be more ready for the next crisis or to evaluate the

effects of huge scope social change in respecting sanitary protection rules.

In future works, we will exploit this methodology on smart sensors or connected RP nodes that will be considered as an Edge Cloud to collect multimedia data, e.g., an autonomous drone system, which can provide capture (by the camera) of the detected objects from different angles and send them to the Edge Cloud system to be analyzed.

Data Availability

The data in the study are available from the corresponding authors on reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic," *Measurement*, vol. 167, Article ID 108288, 2021.

[2] B. Qin and D. Li, "Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19," *Sensors*, vol. 20, no. 18, p. 5236, 2020.

[3] Z. Wang, P. Wang, P. C. Louis, L. E. Wheless, and Y. Huo, "Wearmask: fast In-browser face mask detection with serverless edge computing for COVID-19," 2021, https://arxiv.org/abs/2101.00784.

[4] X. Zhang, H. Saleh, E. M. Younis, R. Sahal, and A. A. Ali, "Predicting coronavirus pandemic in real-time using machine learning and big data streaming system," *Complexity*, vol. 2020, Article ID 6688912, 10 pages, 2020.

[5] M. Razavi, H. Alikhani, V. Janfaza, B. Sadeghi, and E. Alikhani, "An automatic system to monitor the physical distance and face mask wearing of construction workers in COVID-19 pandemic," 2021, https://arxiv.org/abs/2101.01373.

[6] S. K. Dey, A. Howlader, and C. Deb, "MobileNet mask: a multi-phase face mask detection model to prevent person-to-person transmission of SARS-CoV-2," in *Proceedings of International Conference on Trends in Computational and Cognitive Engineering*, pp. 603–613, Springer, Dhaka, Bangladesh, December 2021.

[7] K. M. A. Kabir and J. Tanimoto, "Evolutionary game theory modelling to represent the behavioural dynamics of economic shutdowns and shield immunity in the COVID-19 pandemic," *Royal Society Open Science*, vol. 7, no. 9, Article ID 201095, 2020.

[8] R. Blundell, M. Costa Dias, R. Joyce, and X. Xu, "COVID-19 and inequalities*," *Fiscal Studies*, vol. 41, no. 2, pp. 291–319, 2020.

[9] J. S. Weitz, S. J. Beckett, A. R. Coenen et al., "Modeling shield immunity to reduce COVID-19 epidemic spread," *Nature Medicine*, vol. 26, no. 6, pp. 849–854, 2020.

[10] D. Sridhar and D. Gurdasani, "Herd immunity by infection is not an option," *Science*, vol. 371, no. 6526, pp. 230-231, 2021.

[11] J. Wong and N. Wong, "The economics and accounting for COVID-19 wage subsidy and other government grants," *Pacific Accounting Review*, vol. 33, no. 2, pp. 199–211, 2021.

[12] M. Piraveenan, S. Sawleshwarkar, M. Walsh et al., "Optimal governance and implementation of vaccination programmes to contain the COVID-19 pandemic," *Royal Society Open Science*, vol. 8, no. 6, Article ID 210429, 2021.

[13] A. Echtioui, W. Zouch, M. Ghorbel, C. Mhiri, and H. Hamam, "Detection methods of COVID-19," *SLAS TECHNOLOGY: Translating Life Sciences Innovation*, vol. 25, no. 6, pp. 566–572, 2020.

[14] N. Abbassi, R. Helaly, M. A. Hajjaji, and A. Mtibaa, "A deep learning facial emotion classification system: a VGGNet-19 based approach," in *Proceedings of the 2020 20th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, pp. 271–276, IEEE, Monastir, Tunisia, December 2020.

[15] R. Helaly, M. A. Hajjaji, F. M'Sahli, and A. Mtibaa, "Deep convolution neural network implementation for emotion recognition system," in *Proceedings of the 2020 20th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, pp. 261–265, IEEE, Monastir, Tunisia, December 2020.

[16] S. Bouaafia, S. Messaoud, R. Khemiri, and F. E. Sayadi, "COVID-19 recognition based on deep transfer learning," in *Proceedings of the 2021 IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems (DTS)*, pp. 1–4, IEEE, Sfax, Tunisia, July 2021.

[17] l. Khriji, A. Ammari, S. Messaoud, S. Bouaafia, A. Maraoui, and M. Machhout, "COVID-19 recognition based on patient's coughing and breathing patterns analysis: deep learning approach," in *Proceedings of the 2021 29th Conference of Open Innovations Association (FRUCT)*, pp. 185–191, IEEE, Tampere, Finland, May 2021.

[18] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. Rajendra Acharya, "Automated detection of COVID-19 cases using deep neural networks with X-ray images," *Computers in Biology and Medicine*, vol. 121, Article ID 103792, 2020.

[19] L. Wang, Z. Q. Lin, and A. Wong, "Covid-net: a tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images," *Scientific Reports*, vol. 10, no. 1, pp. 1–12, 2020.

[20] V. Shah, R. Keniya, A. Shridharani, M. Punjabi, J. Shah, and N. Mehendale, "Diagnosis of COVID-19 using CT scan images and deep learning techniques," *Emergency Radiology*, vol. 49, pp. 1–9, 2021.

[21] A. Sedik, M. Hammad, F. E. Abd El-Samie, B. B. Gupta, and A. A. Abd El-Latif, "Efficient deep learning approach for augmented detection of Coronavirus disease," *Neural Computing & Applications*, vol. 18, pp. 1–18, 2021.

[22] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, "Fighting against COVID-19: a novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection," *Sustainable Cities and Society*, vol. 65, Article ID 102600, 2021.

[23] A. Nieto-Rodríguez, M. Mucientes, and V. M. Brea, "System for medical mask detection in the operating room through facial attributes," in *Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis*, pp. 138–145, Springer, Santiago de Compostela, Spain, June 2015.

[24] M. Perc, M. Ozer, and J. Hojnik, "Social and juristic challenges of artificial intelligence," *Palgrave Communications*, vol. 5, no. 1, pp. 1–7, 2019.

[25] S. Saponara, A. Elhanashi, and A. Gagliardi, "Implementing a real-time, AI-based, people detection and social distancing

measuring system for Covid-19," *Journal of Real-Time Image Processing*, vol. 11, pp. 1–11, 2021.

[26] M. Coşkun, A. Uçar, O. Yildirim, and Y. Demir, "November). Face recognition based on convolutional neural network," in *Proceedings of the 2017 International Conference on Modern Electrical and Energy Systems (MEES)*, pp. 376–379, IEEE, Kremenchuk, Ukraine, November 2017.

[27] F. Sultana, A. Sufian, and P. Dutta, "Advancements in image classification using convolutional neural network," in *Proceedings of the 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, pp. 122–129, IEEE, Kolkata, India, November 2018.

[28] S. M. Jameel, M. A. Hashmani, M. Rehman, and A. Budiman, "Adaptive CNN ensemble for complex multispectral image analysis," *Complexity*, vol. 2020, Article ID 83561989, 21 pages, 2020.

[29] Q. Zhao, S. Lyu, B. Zhang, and W. Feng, "Multiactivation pooling method in convolutional neural networks for image recognition," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 8196906, 15 pages, 2018.

[30] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proceedings of the 2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, Ieee, Antalya, Turkey, August 2017.

[31] F. Demir, D. A. Abdullah, and A. Sengur, "A new deep CNN model for environmental sound classification," *IEEE Access*, vol. 8, pp. 66529–66537, 2020.

[32] https://drive.google.com/drive/folders/1IPwsC30wNAc74_GTXuEWX_F8m2n-ZBCH.

[33] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: an overview," 2020, https://arxiv.org/abs/2008.05756.

[34] N. Hasan, Y. Bao, A. Shawon, and Y. Huang, "Densenet convolutional neural networks application for predicting COVID-19 using CT image," *SN Computer Science*, vol. 2, no. 5, 2020.

[35] L. Sarker, M. M. Islam, T. Hannan, and Z. Ahmed, "Covid-densenet: a deep learning architecture to detect Covid-19 from chest radiology images," vol. 21, 2020.

[36] K. El Asnaoui and Y. Chawki, "Using X-ray images and deep learning for automated detection of coronavirus disease," *Journal of Biomolecular Structure and Dynamics*, pp. 1–12, 2020.

[37] R. Jain, M. Gupta, S. Taneja, and D. J. Hemanth, "Deep learning based detection and analysis of COVID-19 on chest X-ray images," *Applied Intelligence*, vol. 51, pp. 1–11, 2020.

[38] M. A. R. Ratul, M. T. Elahi, K. Yuan, and W. Lee, "RAM-Net: a residual attention MobileNet to detect COVID-19 cases from chest X-ray images," in *Proceedings of the 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 195–200, IEEE, Miami, FL, USA, December 2020.

[39] M. Toğaçar, B. Ergen, and Z. Cömert, "COVID-19 detection using deep learning models to exploit Social Mimic Optimization and structured chest X-ray images using fuzzy color and stacking approaches," *Computers in Biology and Medicine*, vol. 121, Article ID 103805, 2020.

[40] Z. Karhan and F. Akal, "Covid-19 classification using deep learning in chest X-ray images," in *Proceedings of the 2020 Medical Technologies Congress (TIPTEKNO)*, pp. 1–4, IEEE, Antalya, Turkey, November 2020.

[41] M. M. Ahsan, M. T. Ahad, F. A. Soma et al., "Detecting SARS-CoV-2 from chest X-ray using artificial intelligence," *IEEE Access*, vol. 9, pp. 35501–35513, 2021.

[42] S. S. Paima, N. Hasanzadeh, A. Jodeiri, and &H. Soltanian-Zadeh, "Detection of COVID-19 from chest radiographs: comparison of four end-to-end trained deep learning models," in *Proceedings of the 2020 27th National and 5th International Iranian Conference on Biomedical Engineering (ICBME)*, pp. 217–221, IEEE, Tehran, Iran, November 2020.

[43] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.