Mask Detection Using IoT - A Comparative Study of Various Learning Models

Mohamed Amine Meddaoui, Mohammed Erritali, Youness Madani, Francoise Sailhan

Mask Detection Using IoT - A Comparative Study of Various Learning Models

Mohamed Amine Meddaoui[1], Mohammed Erritali[1(✉)], Youness Madani[1], and Françoise Sailhan[2]

[1] Data4Earth Laboratory, Sultan Moulay Slimane University, Beni Mellal, Morocco
m.erritali@usms.ma
[2] Cedric Laboratory, CNAM Paris., Paris, France
francoise.sailhan@cnam.fr

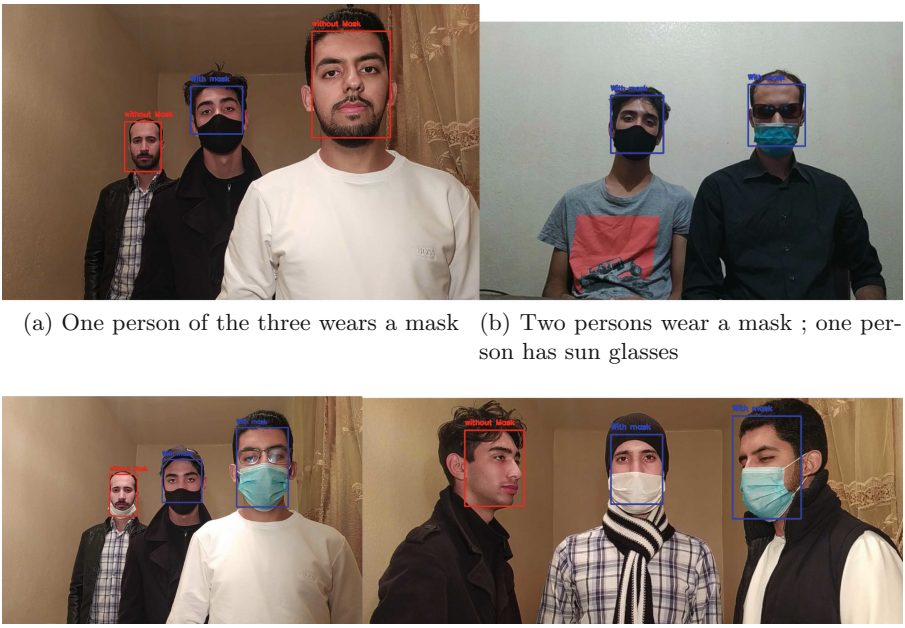
Abstract. Wearing a mask is an effective measure that prevents the spread of respiratory droplets into the air and thereby curtails the dissemination of coronavirus. Unfortunately, despite the proven effectiveness, the idea of wearing a face mask has difficulty being accepted by part of the population. To address this significant health concern, we present a monitoring system that automatically detects whether a mask is put appropriately over a face. The system annotates the videos that are provided by cameras. In this article, we present a comparative study of machine learning models (i.e., SVM, RNN, LSTM, CNN, auto-encoder, MobileNetV2, Net-B3, VGG-16, VGG-19, Resnet-152).

Keywords: COVID-19 · Face mask detection · IoT

1 Introduction

While there are multiple ways to fight the pandemic, a face mask remains a cost-effective measure that is widely practiced to prevent the spread of the virus. Nonetheless, the adoption of face-mask measures remains controversial. The ability to detect violations in public and work spaces is of utmost importance for organizations that host a large population and wish to monitor exposure to adopt precautionary measures. In this article, we propose a detection system that determines whether a person is appropriately wearing a mask, using the information provided by some cameras. Cameras may be disseminated in a building to provide some videos (i.e., image sequences) that are further used to support face mask recognition and ultimately issue proper directives. Still, detecting face masks in any situation is challenging because there are many variations in the appearance of the image that may alter the detection (Fig. 1): variations can be due to (i) a disruptive context (e.g. occlusion of the person, low light intensity), (ii) a rotation or an poor orientation of the person that hinders the detection, (iii) camera parameters (low resolution, excessive focus, noise) or poor positioning of the camera (misorientation, large distance to the scene). We

H. Aloulou et al. (Eds.): ICOST 2022, LNCS 13287, pp. 1–12, 2022.
https://doi.org/10.1007/978-3-031-09593-1_23



(a) One person of the three wears a mask (b) Two persons wear a mask ; one person has sun glasses

(c) one mask is incorrectly put, a person has glasses (d) 2 persons in profile ; 2 persons wear a mask and one in front wears a cap

Fig. 1. Situations in which real-time videos are automatically annotated

propose a comparative study of different ML models and evaluate their ability to handle the above situations and their suitability for deployment on IoT devices or cloud servers. Overall, our key contribution includes:

- A face mask detection service that exploits videos.
- A comparative study of various ML models (i.e., CNN, LSTM, RNN, auto-encoder, SVM, MobileNetV2, Net-B3, VGG-16, VGG-19, Resnet-152) with a discussion on their practical applicability.
- A prototype that annotates the collected videos relying on a IoT device or on a remote (cloud) server ; the choice of the approach is based on requirements of the organization.

The paper is organized as follows. We present the related work (Sect. 2) and we introduce our comparative study (Sect. 3), which is further evaluated (Sect. 4).

2 Related Work

Several detection systems [5–7,10,11,14,16,17] have been proposed and evaluated using some purposely-built datasets (Table 1) that contain some pictures of people with facemask, without facemask or with facemask put incorrectly. The

pictures of masked people either (i) correspond to real-world person that are pictured (*real-world dataset*) or (ii) result from the addition of a mask picture on an existing facial image (*simulated dataset*). Leveraging existing datasets, detection system detects facemasks on pictures (Sect. 2.1) or videos (Sect. 2.2).

Table 1. Datasets containing real world images or simulated images

Dataset	Size	Content	Ref
Real world dataset			
Real World Masked Face Reco. Dataset:RMFRD	14 K	0.5 K public figures with/without mask	[4]
Masked Face Detection Dataset: MFDD	24 K	persons with mask	[4]
Face Mask Dataset: FMD	0.8 K	with/without mask, mask put incorrectly	[2]
Medical Masks Dataset: MMD	6 K	3 k medical masked faces	[3]
Simulated dataset			
Simulated Masked Face Recog. Dataset: SMFRD	500 K	simulated facial images, 10 K participants	[4]
Custom Mask Community Dataset (CMCD)	1.2 K	with/without simulated facemask	[1]

2.1 Facemask Detection in Pictures

The facemask detection system introduced in [11], consists of a feature extractor that uses convolutional neural network (Resnet50[1]) and a classifier that implements Support Vector Machine (SVM) and ensemble algorithm. Evaluations based on the RMFD, SMFD, CMCD datasets, show that the SVM classifier involves the fastest training and achieves the highest accuracy: 99.64% testing accuracy with RMFRD, 99.49% with SMFD, and, 100% with CMCD. In [10], the solution localizes medical face masks and annotates accordingly those images. Feature extraction process relies on the ResNet50 deep transfer learning model while the mask detection process uses YOLO v2 [12]. Following, authors rely on the Adam optimization algorithm [8] to improve the performance of the detector. Empirical evaluation shows that the Adam optimizer achieves the highest average precision percentage of 81%. In [14], face mask detection uses the faceNet image classifier [15] that implements a Convolutional Neural Network (CNN). This image classifier is trained using a purposely-built dataset including 4K images with half of the dataset containing some pictures of people wearing mask in public places (e.g. shops) while the rest concerns people without mask. Empirical results show that people wearing (or not) a face mask are detected with an accuracy of 96,85%. Arjya et al.[6] detect the facemask on image using a pre-trained CNN containing two 2D convolution layers connected to layers of dense neurons. The proposed method attains an accuracy up to 95.77% with SFC dataset and 94.58% respectively FMD dataset. In [7], a facial categorization system determines whether a person is wearing a mask or not. Face recognition is performed by a deep C2D CNN (Colour 2-Dimensional principal component analysis - Convolutional Neural Network) ; mask detection relies on special convolutional architecture that is best suited for the classification of RGB images.

[1] https://keras.io/api/applications/resnet.

The training relies on the RMFRD and Celeb Faces Attributes[2] dataset. In [5], the face mask detection system captures image, extracts features from image based on Principal Component Analysis (PCA), detects the human face using viola zones method and further uses the K-Nearest Neighbor (KNN) classifier. Experiments are based on the ORL[3] database in which the lower portion of detected face images is covered with black or white. Preliminary performance evaluation shows that the accuracy is around 98% with a principal component of two. Additionally, face recognition accuracy with face masks has been extensively investigated: interest reader may refer to the masked face recognition workshop and challenge[4].

2.2 Face Mask Detection in Video

Another line of research aims at providing a surveillance system [17] that identifies whether a person is wearing a mask using real-time videos. Mask detection is done by MobileNetV2 [13] that achieves high accuracy of 99.98% on training data, 99.56% on validation data, and 99.75% on testing data. In [16], a mobile robot automatically detects unmasked personnel in public spaces and provides a surgical mask to them to promptly remedy the situation. The mobile robot integrates deep residual learning (ResNet-50) with Feature Pyramid Network (FPN) to detect the existence of human subjects in video (feeds). Then, Multi-Task Convolutional Neural Networks (MT-CNN) detect and extract human faces from these videos. Ultimately, a convolutional neural network classifier detects (un)masked human subjects. Training leverages four publicly available datasets: Microsoft Common Objects in Context (COCO)[9], the CelebFaces Attributes Dataset[5] (CelebA), WIDER FACE dataset[6], CMCD. The proposed surveillance system is further evaluated using a dataset of videos collected by the robot in an educational institute. Results show a mask detection accuracy of 81.3% with a very high recall of 99.2%. While many detectors rely on pictures, only two approaches support real-time facemask detection leveraging videos. In this paper, we introduce a video-based system that incorporates several ML models and we provide a comprehensive comparison with the state of the art.

3 Mask Detection

Leveraging the videos delivered by the camera, our application detects the presence of any nearby person and determines whether the person has a mask and if the mask is correctly put. Then, the application labels the corresponding image. This detection requires converting the videos into an appropriate format, locating people face(s) (Sect. 3.1) and determining whether people wears mask (Sect. 3.2).

[2] https://www.kaggle.com/jessicali9530/celeba-dataset.
[3] https://cam-orl.co.uk/facedatabase.html.
[4] https://tinyurl.com/4xzjzvat.
[5] https://www.kaggle.com/jessicali9530/celeba-dataset.
[6] https://paperswithcode.com/dataset/wider-face-1.

3.1 Face Detection in Pre-processed Video

Face mask detection starts with the capture of a video partitioned in successive series of color pictures. Color pictures are further converted into RGB pictures, which render the process of discovering the face less complex comparing to color picture. For each picture, mean subtraction is applied to prevent illumination: the average intensity is computed accross all the images for each of the Red, Green, and Blue channel ; then, the mean is substracted per channel for each image. Following, each image is divided into $n \times n$ squares where the value of n depends of the object of interest (e.g. main feature of the face). Within each square, the face detection algorithm passes through each pixel of the image in addition to the adjacent pixels (i.e. the pixels located at the top - bottom - left - right - top right - top left - bottom right - and bottom left). This process is intended to store key facial features (e.g. eyes) that help in detecting face while irrelevant data that are located in the background (e.g. a car, tree, traffic light). Finally, our system determines the location of each face. Our approach consists in classifying into two classes, whether it is wearing appropriately a facemask or not. Note that the facemask is appropriately wear if the mouth, chin and part of the nose are well covered with the mask.

3.2 Classification

For facemask detection, we built some models using SVM, RNN, LSTM, Auto-Encoder, which are briefly presented, starting with SVM.

Support Vector Machines SVM separates the input data within the space by a hyperplane that linearly separates the data into classes (with and without mask). Input data typically refers to small training dataset made of support vectors. Herein we use a linear kernel. The hyperplane best separates the support vectors, by means of maximization of the distance between these vectors and the hyperplane. As shown in Sect. 4, the SVM remains efficient with little training data.

In addition, we consider three types of Convolutional Neural Networks (CNN) referring to Recurrent Neural Networks (RNN) and Long Short Term Memory Networks (LSTM) - that are multi-layered neural networks made of several hidden layers of neurons wherein the output of a neuron in a layer becomes the input of a neuron of the next layer. These networks have adopted diverse structures that meet different expectations. Convolutional Neural Network (CNN) is a neural made of two distinct parts: (i) the convolution layers extracts valuable features from the input (image); in practice, kernels automatically extract the relevant features based on the convolution operation, (ii) the fully connected layers leverage the data from convolution layer to generate the result.

Recurrent Neural Network (RNN) are a class of neural networks that differs from others in that they maintain internal hidden states and have cyclic/recurrent connections, which allow them (i) to capture the sequential information (i.e. dependencies) in the input data and (ii) information to persist. Still, RNN traditionally suffers from what is known as the problem of vanishing and exploding

gradient in which the network either stops learning (vanishing gradient) or never converges to the point of minimum cost (exploding gradient). LSTM are designed to remedy both problems and thereby have become popular in modelling complex sequential data.

Long Short Term Memory Networks consists of a set of recurrently connected subnetworks (also coined as memory blocks). Any block contains one or more self-connected memory cells storing historical states, as well as gates that control the flow of information through the cells. Thus, LSTM may store and access information over long period of time, which prevents the vanishing gradient problem. LSTM contains four layers of neural networks.

Auto Encoder is a specific type of neural network in which the encoder represents the input into a compressed and meaningful representation so that the decoder has the most relevant information to reconstruct the image. In particular, the encoder learns the most important components of an input and thereby gets the best possible compression. The error made by the encoder is established based on the differences between the reconstructed data and the initial data. The training consists in modifying the parameters of the auto-encoder so as to reduce the reconstruction error measured on the different samples of the dataset. While various neural network topologies exist (e.g., vanilla, convolutional, regularized, multi-layer), we used a multi-layer auto-encoder and we encoded in an unsupervised way. The encoder contains three hidden layers: the first one is four times larger than the input, and the second one is two times larger than the input, and the size of the third one is equal to the input size. Following, we optimize the model using adam optimiser [8].

4 Performance Evaluation

We assess the effectiveness of our detector relying on the following two training datasets: the Real World Masked Face Recognision Dataset [4] and the Face Mask Dataset [2] (FMD) that include some color pictures of different sizes. Together these two datasets include some pictures of people of different nationalities/ages, with/without facemask, with mask put incorrectly, with e.g., glasses, hat. As detailed in Sect. 3.1, pictures are normalized and expressed into a common format. We performed cross-validation, using scikit-learn platform[7], which splits the dataset into a training (80% of the original dataset) and test dataset. Unless explicitly mentioned, 100 epochs are for building models. In the following, we evaluate the performances associated with the training and the detection. In both cases, the experiments are run either on a IoT device (Raspberry Pi 4 with 1,6 GHZ and 2 GB of RAM) with 3,1 GHZ of CPU, and Machine ASUS with 8 GB in RAM, used to perform the training and the mask detection. In order to evaluate the detection of facemask, we also conducted series of experiments using dataset and a camera of 48 Mega pixel.

[7] https://scikit-learn.org.

Table 2. Size of dataset used to train and test the model

Size of dataset	Small	Medium	Large
Training	1000	2000	6000
Testing	200	400	1200

Delay associated with the training and detection. In Table 3, we evaluate the time associated with the learning process, using various sizes of dataset (Table 2). As expected, the larger the dataset, the longer the learning process. The device capacity also has a significant impact on its ability to learn (quickly). Contrary to the server, the IoT device can only handle small or medium datasets, regardless of the learning model. The learning process is on average 2.631 times longer with the IoT device whose capabilities are limited, compared to the server. On the other hand, detection is much faster than learning (Table 4). In particular, the time associated with detection is completely decoupled from that associated with learning. The detection with the Raspberry takes a little longer than with a server. For both (IoT device an server), CNN is the fastest, followed by autoencoder, LSTM, RNN whose results are close while SVM is the slowest.

Table 3. Training Delay associated with a dataset of varying size, using an IOT device and a server

Dataset	Device	SVM	RNN	LSTM	CNN	AUTOENCODER
Small	Raspberry	13 h 20 mn	14 h 35 mn	15 h 30 mn	13 h 07 mn	15 h 20 mn
Small	Server	5 h 10 mn	5 h 15 mn	4 h 50 mn	5 h 20 mn	6 h 05 mn
Medium	Raspberry	—	—	—	—	—
Medium	Server	6 h 50 mn	7 h 30 mn	8 h 04 mn	8 h 30 mn	10 h 15 mn
Large	Raspberry	—	—	—	—	—
Large	Server	14 h 30 mn	13 h 50 mn	14 h 10 mn	15 h 25 mn	16 h 30 mn

Table 4. Delay associated with detection

Model	SVM	RNN	LSTM	CNN	AUTOENCODER
Raspberry	4 s	2.8 s	2.6 s	1.5 s	2.2 s
Server	3 s	1.9 s	1.5 s	0.8 s	1.4 s

Facemask Detection Efficiency We compare the effectiveness of four strategies used to detect face masks (as defined in Sect. 3). Experiments were run on the IoT device and on the server; results reported bellow are the same for both. AQ3
Figure 2 provides the accuracy and the loss functions associated with the training (*accuracy* and *loss*) and the validation (*val_accuracy*, *val_loss*) sets; the training set is used to build the model while the validation set supports the

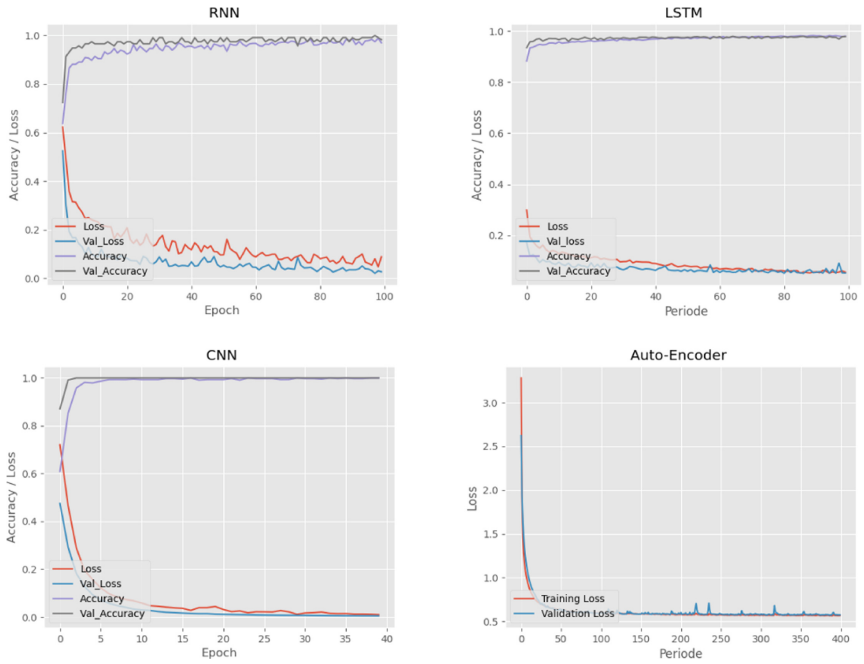


Fig. 2. Accuracy and Loss associated with the training and validation with large dataset

Table 5. Dataset size

Class	Small dataset	Medium dataset	Large Dataset
With mask	1000	2000	4718
With mask	1000	2000	1759

fine tuning of the parameters and of the model structure. Loss corresponds to binary-cross entropy. For the RNN, LSTM and CNN models, accuracy and loss appear to be inversely proportional: after few iterations of optimization, the loss reduces drastically while the accuracy greatly increases. As intended, the shape of accuracy and loss functions are quite similar with the training and validation sets. At first sight, CNN becomes quickly accurate (accuracy is high and loss is small after only 40 epochs). With LSTM (and resp. RNN), the accuracy and loss stabilize after 60 epoch (resp. 80 epochs). The auto-encoder takes much more time (400 epochs) comparing to the other methods: the loss curve still fluctuates for epoch ranges of [200,250] and [300,350]. In addition, we evaluate the performances associated with the detection in terms of precision, recall, F1 score and accuracy. Relying on these performance metrics, we evaluate in Tables 6, 7 and 8 the efficiency of the detection models considering a small, medium and large dataset (as defined in Table 5). As expected, the larger the dataset, the

Table 6. Models with small dataset

Algo		Precision	Recall	F1-score	Accuracy
SVM	Without mask	0.83	0.90	0.85	0.84
	With mask	0.84	0.83	0.92	
	Macro avg	0.82	0.83	082	
	Weighted avg	0.84	0.84	0.84	
RNN	Without mask	0.85	0.84	0.84	0.85
	With mask	0.86	0.85	0.84	
	Macro avg	0.86	0.85	0.86	
	Weighted avg	0.86	0.86	0.86	
LSTM	Without mask	0.89	0.84	0.85	0.88
	With mask	0.89	0.87	0.86	0.86
	Macro avg	0.86	0.86	0.85	
	Weighted avg	0.87	0.87	0.87	
CNN	Without mask	0.93	0.93	0.4	0.95
	With mask	0.96	0.97	0.95	
	Macro avg	0.94	0.95	0.95	
	Weighted avg	0.95	0.95	0.95	
Auto encoder	Without mask	0.88	0.87	0.85	0.91
	With mask	0.90	0.88	0.86	
	Macro avg	0.89	0.88	0.86	
	Weighted avg	0.90	0.89	0.87	

Table 7. Efficiency of learning models with medium dataset

Model		Precision	Recall	F1-score	Accuracy
SVM	Without Mask	0.85	0.86	0.86	0.87
	With Mask	0.86	0.85	0.84	
	Macro Avg	0.86	0.87	0.85	
	Weighted Avg	0.87	0.87	0.87	
RNN	Without Mask	0.87	0.85	0.88	0.89
	With Mask	0.89	0.90	0.90	
	Macro Avg	0.90	0.89	0.90	
	Weighted Avg	0.90	0.90	0.90	
LSTM	Without Mask	0.89	0.88	0.90	0.91
	With Mask	0.91	0.91	0.91	
	Macro Avg	0.90	0.89	0.91	
	Weighted Avg	0.91	0.91	0.91	
CNN	Without Mask	0.91	0.91	0.92	0.97
	With Mask	0.98	0.95	0.96	
	Macro Avg	0.95	0.96	0.96	
	Weighted Avg	0.97	0.96	0.97	
Auto Encoder	Without Mask	0.91	0.90	0.92	0.94
	With Mask	0.92	0.93	0.92	
	Macro Avg	0.91	0.92	0.92	
	Weighted Avg	0.92	0.93	0.92	

Table 8. Efficiency of models with large dataset

Model		Precision	Recall	F1-score	Accuracy
SVM	Without Mask	0.86	0.89	0.88	0.89
	With Mask	0.91	0.89	0.90	
	Macro Avg	0.88	0.88	0.89	
	Weighted Avg	0.89	0.88	0.89	
RNN	Without Mask	0.91	0.90	0.91	0.90
	With Mask	0.93	0.92	0.92	
	Macro Avg	0.93	0.91	0.93	
	Weighted Avg	0.94	0.94	0.94	
LSTM	Without Mask	0.94	0.91	0.92	0.96
	With Mask	0.97	0.95	0.96	
	Macro Avg	0.95	0.94	0.95	
	Weighted Avg	0.96	0.96	0.96	
CNN	Without Mask	0.98	0.99	0.99	0.99
	With Mask	0.99	0.99	0.99	
	Macro Avg	0.98	0.98	0.98	
	Weighted Avg	0.99	0.99	0.99	
Auto Encoder	Without Mask	0.91	0.92	0.92	0.94
	With Mask	0.92	0.93	0.92	
	Macro Avg	0.93	0.92	0.92	
	Weighted Avg	0.94	0.94	0.94	

Table 9. Efficiency of state of the art models using dataset3

Algo		Precision	recall	F1-score	Accuracy
MobileNetV2	Without Mask	0.95	0.94	0.95	0.96
	With Mask	0.96	0.95	0.96	
	Macro Avg	0.96	0.95	0.95	
	Weighted Avg	0.96	0.96	0.96	
EfficientNet-B3	Without Mask	0.93	0.92	0.91	0.94
	With Mask	0.93	0.92	0.92	
	Macro Avg	0.93	0.92	0.91	
	Weighted Avg	0.93	0.93	0.92	
VGG-16	Without Mask	0.96	0.95	0.97	0.98
	With Mask	0.98	0.98	0.96	
	Macro Avg	0.97	0.97	0.96	
	Weighted Avg	0.98	0.97	0.97	
VGG-19	Without Mask	0.96	0.96	0.92	0.98
	With Mask	0.98	0.95	0.96	
	Macro Avg	0.97	0.95	0.94	
	Weighted Avg	0.98	0.96	0.94	
ResNet-152	Without Mask	0.98	0.99	0.99	0.98
	With Mask	0.99	0.99	0.98	
	Macro Avg	0.98	0.98	0.98	
	Weighted Avg	0.99	0.99	0.99	

more effective the training and subsequent detection, as shown by the increase in precision, recall, F1 score and accuracy for any model. Note that data augmentation may be relevant to increase the dataset size (and thereby the efficiency of the ML models) by creating modified versions of images. Regardless of the dataset size and of the model, the accuracy associated with the detection of people with mask is better than that of people without mask. Regardless of the dataset size, CNN always gives the best result in terms of precision, recall, F1 score, accuracy. Then, LSTM and auto-encoder provide lower but high efficiency, followed by RNN and SVM. Table 9 compares various models. CNN gives the best results in terms of accuracy while ResNet-152 is characterised by a slighly lower accuracy but a quite similar precision, recall and F1-score.

5 Conclusion

We introduce a new detection system that automatically determines whether a person wears facemask, which is put appropriately. In practice, the system detects and determines the position of the face(s) in the videos provided by cameras. For each detected face, the system determines whether a facemask is put appropriately, leveraging some machine learning models. Experimental results show that classification may be performed by a server or IoT device. Empirical results demonstrated that CNN gives is the most accurate (99.8% with a large training dataset). Future work involves improving the detection in presence of low quality picture (e.g., low light level, presence of obstacles) and evaluating the energy associated with the detection.

References

1. https://github.com/prajnasb/observations/tree/master/experiements/data
2. Face mask dataset. https://www.kaggle.com/andrewmvd/face-mask-detection
3. Medical masked faces. https://www.kaggle.com/vtech6/medical-masks-dataset
4. Wang, Z., Wang, G., Huang, B., et al.: Masked face recognition dataset and application. arXiv preprint arXiv:2003.09093 2020
5. Biswas, S., Mazumdar, S., Rana, S., Saba, S.A., et al.: Face detection based approach to combat with COVID-19, vol. 1797(1). IOP Publishing (2021)
6. Das, A., Ansari, M.W., Basak, R.: Covid-19 face mask detection using TensorFlow, Keras and OpenCV. In: IEEE India Council International Conference (INDICON), pp. 1–5 (2020)
7. Gupta, S., Sreenivasu, S.V.N., Chouhan, K., et al.: Novel face mask detection technique using machine learning to control COVID'19 pandemic. Mater. Today Proc. (2021)
8. Kingma, D.P., Ba, J.: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2014)
9. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48

10. Loey, M., Manogaran, G., Taha, M.H.N., et al.: Fighting against COVID-19: a novel deep learning model based on yolo-v2 with ResNet-50 for medical face mask detection. Sustain. Cities Soc. **65**, 102600 (2021)
11. Loey, M., Manogaran, G., Taha, M.H.N., et al.: A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. Measurement **167**, 108288 (2021)
12. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. CoRR (2016)
13. Sandler, M., Howard, A., Zhu, M., et al.: Mobilenetv 2: inverted residuals and linear bottlenecks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
14. Sanjaya, S.A., Rakhmawan, S.A.: Face mask detection using MobileNetV2 in the era of COVID-19 pandemic. In: International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI) (2020)
15. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: a unified embedding for face recognition and clustering. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
16. Snyder, S.E., Husari, G.: Thor: a deep learning approach for face mask detection to prevent the COVID-19 pandemic. In: SoutheastCon (2021)
17. Taneja, S., Nayyar, A., Nagrath, P.: Face mask detection using deep learning during COVID-19. In: International Conference on Computing, Communications and Cyber-Security (2021). https://doi.org/10.1007/978-981-16-0733-2_3