International Journal of Advanced Science and Computer Applications (2024); 3(2):
DOI: 10.47679/ijasca.v3i2.49
https://ijasca.org/index.php/ijasca

RESEARCH ARTICLE

A Comparative Study of Embedded Learning Models IoT-Based For Real time Mask Detection

Mohamed Amine Meddaoui, Mohammed Erritali, Françoise Sailhan

Published online: 25 September 2024

Abstract

Following the outbreak of the coronavirus, many preventive measures are implemented to slow down the transmission of the virus. Amongst others, face mask detection is a key innovative technology that allows the identification of the number of individuals wearing face masks. In this regard, this paper provides a comparative study of several machine learning and deep learning algorithms (e.g., SVM, RNN, Mask-RCNN, LSTM, CNN, Auto-Encoder, GAN, U-Net GAN) that support mask detection.

Keyword: Facemask detection, IOT, Classification, Segmentation

Introduction

Whilst wearing a face mask remains a cost-effective measure to prevent the spread of the virus, the adoption of face masks remains controversial. In this regards, the ability to detect mask adoption and violation in public/work spaces is of utmost importance for organisations that cater to a large population and wish to monitor exposure to take precautionary measures accordingly. So far, few research works (e.g., (16; 15; 21; 9; 25; 11; 5; 24)) attempt to determine whether a person is (appropriately) wearing a mask, using cameras. Unfortunately, these existing solutions suffer from several shortcomings. Solutions are cloud-centric: raw pictures flow directly from each camera to a remote server/service that further processes and classifies the data. This implemented centralisation/cloudification introduces user privacy leaks that limit the adoption of facemask detection.

Fig 1. Detection Process with GAN

We address this concern that is central to the effective deployment and execution of facemask detection through the introduction of a pervasive solution, which builds upon the following trend to support facemask detection: the solution leverages some microcontrollers embedding cameras that may be easily dis-seminated in a building to provide some videos (i.e., image sequences) and to support face mask recognition. Running necessary computations (a.k.a detection) on-device also eliminates the need for server-side processing and communication overheads. Our key contribution are the following: • a detection system (Figure 1) that determines whether a person is appropriately wearing a mask, using the information provided by cameras. • An end-to-end prototype that annotates the collected videos provided by cameras that is further used to support a comparative study of various classification models.

Faculty of Science and Technology - Béni Mellal

*) corresponding author

Email: meddaoui.med@gmail.com

Received 01 May 2024 | Received in revised form 20 June 2024 | Accepted 23 August 2024
Available online 25 September 2024

• An empirical evaluation with a discussion on the practical applicability of the various models.

There exists a literature in the image processing community on detecting facemask relying on machine learning models. Existing approaches generally make use of some purposely-built datasets (§ 2.1) that contain some pictures of people with or without facemask or with facemask put incorrectly. These datasets are typically used by a server that trains a model to latter detect facemasks on pictures (§ 2.2) or videos (§ 2.3).

FaceMask Dataset

Several detection systems (16; 15; 21; 9; 25; 11; 5; 24) 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 picture of masked people either (i) corresponds to realworld person that are pictured (real-world dataset) or (ii) results from the addition of a mask picture on an existing facial image (simulated dataset).

Table 1. Datasets containing real world images or simulated images

Dataset	Size	Content	Ref
Real world dataset			
Real World Masked Face Reco. set:RMFRD	14K	0.5K public figures with/without mask	(4)
Masked Face Detection Dataset: MFDD	24K	persons with mask	(4)
Face Mask Dataset: FMD	0.8K	with/without mask, mask put incorrectly	(1)
MaskedFace-Net: MFN	137K	with/without mask, mask put incorrectly	(7)
Medical Masks Dataset: MMD	6K	3k medical masked faces	(2)

Table 2. Simulated dataset

Simulated Masked Face Recog. Dataset: SMFRD	**500K**	**simulated facial images, 10K participants**	**(4)**
Custom Mask Community Dataset (CMCD)	1.2K	with/without simulated facemask	(3)

Facemask Detection

Face mask detection is a form of object detection that is based on image or video and on classification techniques. As pointed in (17), the majority of the face mask detection algorithms that are detailed in the following are based on deep learning (DL). The facemask detection system introduced in (16), consists of a feature extractor that uses convolutional neural network (Resnet50) and a classifier that implements Support Vector Machine (SVM) and ensemble algorithms. 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 (15), 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 (18). Following, authors rely on the Adam optimization algorithm (13) to improve the performance of the detector. Empirical evaluation shows that the Adam optimizer achieves the highest average precision percentage of 81%. In (21), face mask detection uses the faceNet image classifier (23) 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.(9) 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 (11), 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. The training relies on the RMFRD and Celeb Faces Attributes 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 database in which a small portion of detected face images is covered with black or white boxes. Preliminary performance evaluation shows that the accuracy is around 98% with a principal component of two. In (6), authors introduce a cascaded CNN architecture to detect facemask, which comprises three binary CNN classifiers with a different number of layers that improve the detection accuracy. In (8), Amit et al. introduce a two-stages detector making use of two CNN models. In particular, they rely on existing pre-trained CNN that detect faces, and then, the next stage aims at classifying into a mask and no-mask class. Additionally, face recognition accuracy with face masks has been extensively investigated: in this regards, interest reader may refer to the masked face recognition workshop and challenge .

Face Mask Detection in Video

Another line of research aims at providing a surveillance system (25) that identifies whether a person is wearing a mask using real-time videos. Mask detection is done by MobileNetV2 (20) that achieves high accuracy of 99.98% on training data, 99.56% on validation data, and 99.75% on testing data. In (24), 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 (ResNet50) with Feature Pyramid Network (FPN) to detect the existence of human subjects in video (feeds). Then, Multi-Task Convolutional Neural Network (MT-CNN) detects and extracts 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)(14), the CelebFaces Attributes Dataset (CelebA), WIDER FACE dataset , 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

Method

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 locating people face (§ 3.1) and determining whether people wears mask (§ 3.2).

Face Detection using 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

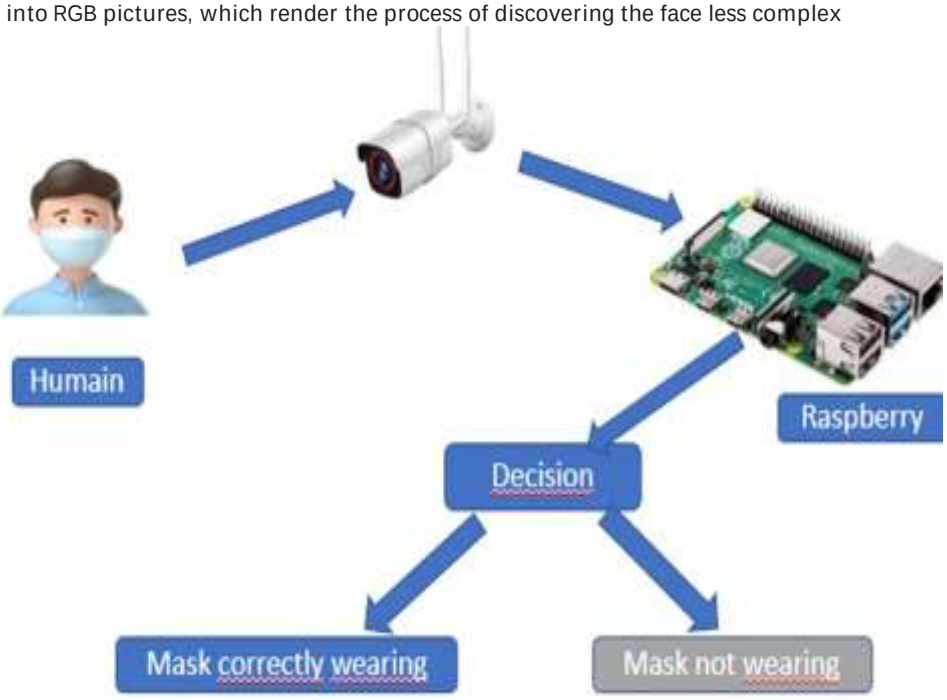

Fig 2. Detection Process

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.

Classification

Our approach consists in classifying images 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. For facemask detection, we built some models using SVM, RNN, CNN, Mask-RCNN LSTM, Auto-Encoder, DBN and GAN, U-Net

GAN which are briefly presented hereafter, 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 § 4, the SVM remains efficient with little training data.

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) is a class of neural network 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.

Mask-RCNN (12) extends Faster R-CNN (19) by predicting segmentation masks on each Region of Interest (RoI), in parallel with the classification and bounding box regression. In particular, a convolutional network called backbone extracts primitives from the image. Based on these primitives, a Region Proposal Network (RPN) provides and refines a certain number of regions of interest (in the form of rectangular bounding boxes) likely to contain a parcel. Finally, the last part will retrieve the best proposals, refine them again, and produce a segmentation mask specific to each of them.

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 (13).

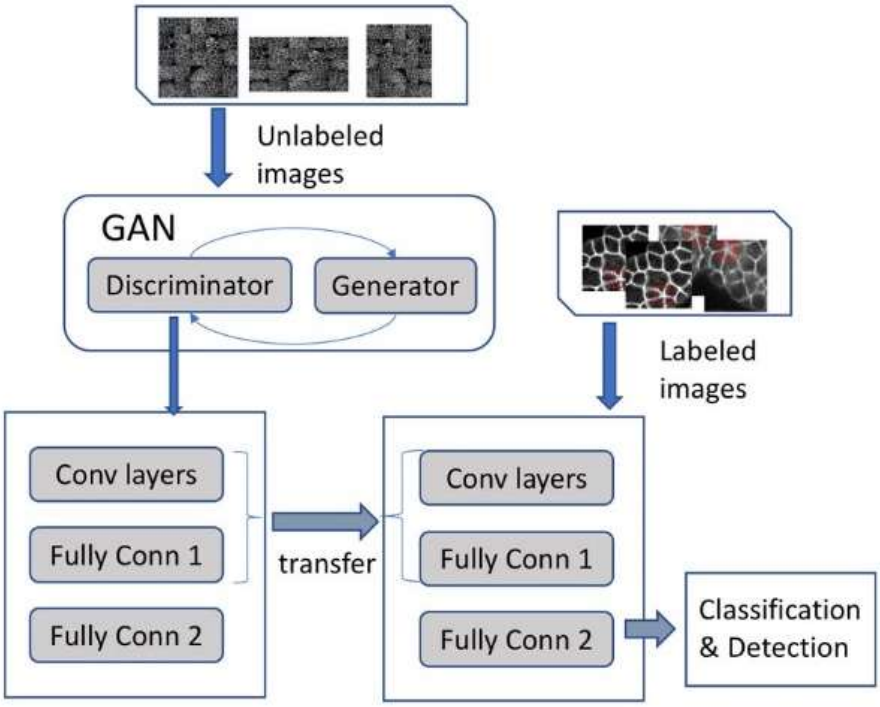

Fig 3. Detection Process with GAN

Generative Adversarial Network (GAN) (10) can be viewed as a zerosum two-players game in which two models are trained (Figure 3): a *generator* learns how to generate samples resembling real data so as to provide a generative model capturing the real data distribution, and with a *discriminator* that learns how to discriminate real and generated data to estimate the probability that a sample comes from the real data (or not). These two entities compete during the learning phase – this mechanism being called back propagation – so that both players improve their respective methods until the counterfeits are indistiguishable from the genuine samples. Nonetheless, one problem is related to the fact that the discriminator tends to learn a representation - often focusing on the global structure or local details - so as to efficiently penalize the generator.

U-Net Generative Adversarial Network (U-Net GAN) (22) U-Net GAN includes an alternative and stronger discriminator that acts at a classifier and segmenter and outputs simultaneously both global (over the whole image) and per-pixel decision of the image. Leveraging an U-Net [39], the decoder outputs per-pixel class decision, providing spatially coherent feedback to the generator and rendering the generator task of fooling the discriminator more difficult.

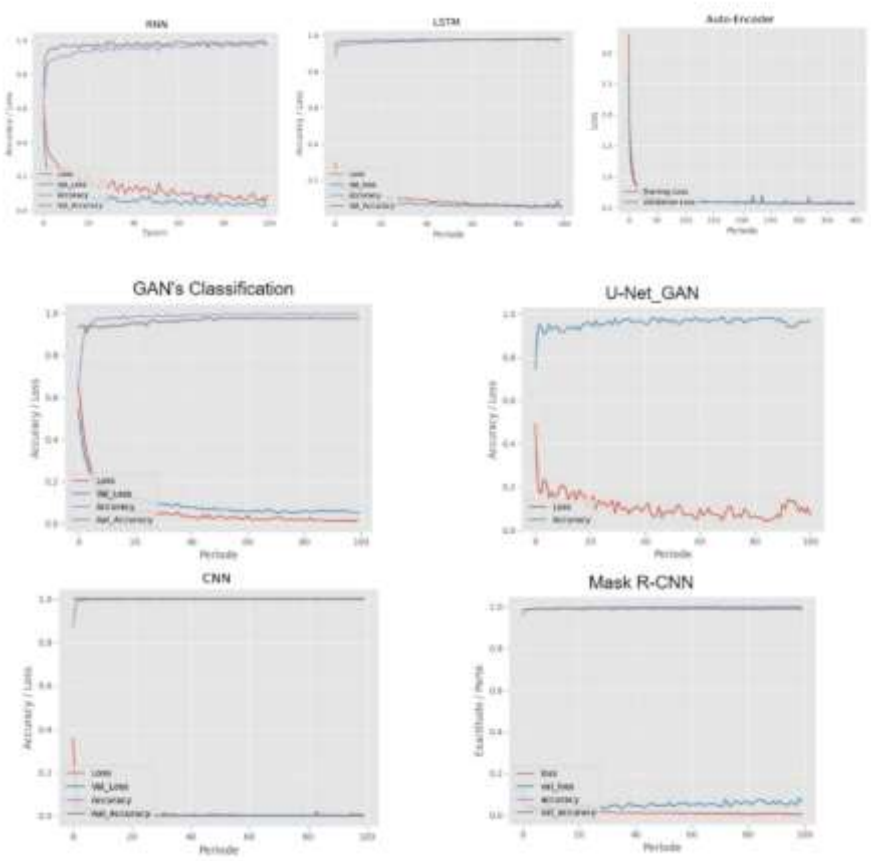

Fig 4. Accuracy and Loss associated with the training and validation with large dataset.

Results and Discussion

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 (1) (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 § 3.1, pictures are normalized and expressed into a common format. We performed cross-validation, using scikit-learn platform, 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 detection. 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, which is used to perform the training and the mask detection.

Figure 4 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 fine tuning of the parameters and of the model structure. Loss corresponds to binary-cross entropy. Overall, 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, R-CNN mask and then CNN converge the most quickly (the accuracy becomes high and the loss low after only 40 epochs) while RCNN shows a low fluctuation of losses which reflects a very small uncertainty in the classification. Other models exhibit poorer and more fluctuating levels of accuracy and loss. With LSTM (and resp. autoencoder), the accuracy and loss take a little bit of time to stabilize (around 20 epochs). RNN, GAN and U-Net GAN slowly converge and exibit comparatively some variations of accuracy and loss.

In addition, we evaluate the performances associated with the detection in terms of precision, recall, F1 score and accuracy. U-Net GAN, Mask-RCNN and CNN give the best results in terms of precision, recall, F1 score and accuracy. In particular, CNN performs better than the others but involves a significantly longer learning period than U-Net GAN.

Table 3. Models Efficiency

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	
GAN	Without Mask	0.92	0.93	0.94	0.95
	With Mask	0.94	0.95	0.95	
	Macro Avg	0.94	0.95	0.95	
	Weighted Avg	0.95	0.95	0.95	
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	Without Mask	0.91	0.92	0.92	0.94
Encoder	With Mask	0.92	0.93	0.92	
	Macro Avg	0.93	0.92	0.92	
	Weighted Avg	0.94	0.94	0.94	
U-Net GAN	Without Mask	0.96	0.97	0.97	0.96
	With Mask	0.96	0.96	0.96	
	Macro Avg	0.95	0.96	0.96	
	Weighted Avg	0.96	0.96	0.96	
Mask-RCNN	Without Mask	0.97	0.98	0.98	0.98
	With Mask	0.96	0.97	0.97	
	Macro Avg	0.97	0.98	0.97	
	Weighted Avg	0.98	0.98	0.98	

Table 4. Training Delay

Algorithm	Time
SVM	14h30mn
RNN	13h50mn
LSTM	14h10mn
CNN	15h25 mn
Mask-RCNN	16h 15 min
GAN	12 h 30 mn
U-Net GAN	11 h 10 min
Auto-Encoder	16h30 mn

.

Conclusions and Recommendations

This article proposes a new detection system that automatically determines whether a person wears a facemask, which is put appropriately. Our face-mask detection is supported by Raspberry pi 4 that is connected to a camera and that hosts a facemask detection system, which locates faces and determines if the facemask is properly wear. For that purpose, the detection system relies on a model (e.g. SVM, RNN, LSTM, CNN, Mask-RCNN, GAN, U-Net GAN, Auto-Encoder). Experimental results show that classification may be performed by an IoT device.

References

[1] Face mask dataset.https://www.kaggle.com/andrewmvd/face-maskdetection.

[2] Medical masked faces. https://www.kaggle.com/vtech6/medical-masksdataset.

[3] SLFW. https://github.com/prajnasb/observations/tree/master/experiements/data.

[4] Z. Wang abd G. Wang, B. Huang, and al. Masked face recognition dataset and application. arXiv preprint arXiv:2003.09093, 2020.

[5] Suparna Biswas, Senjuti Mazumdar, Sangeeta Rana, S.B. Amreen Saba, and al. Face detection based approach to combat with COVID-19. IOP Publishing, 1797(1), feb 2021.

[6] Wei Bu, Jiangjian Xiao, Chuanhong Zhou, Minmin Yang, and Chengbin Peng. A cascade framework for masked face detection. In IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), 2017.

[7] Adnane Cabani, Karim Hammoudi, Halim Benhabiles, and Mahmoud Melkemi. Maskedface-net - a dataset of correctly/incorrectly masked face images in the context of covid-19. Smart Health, 19, 2021.

A. Chavda, J. Dsouza, S. Badgujar, , and A. Damani. Multi-stage cnn architecture for face mask detection. In 6th International Conference for Convergence in Technology (I2CT), 2020.

[8] Arjya Das, Mohammad Wasif Ansari, and Rohini Basak. Covid-19 face mask detection using tensorflow, keras and opencv. In IEEE India Council International Conference (INDICON), pages 1–5, 2020.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, and al. Generative adversarial networks. Communications of the ACM, 63, 2020.

[10] Sandeep Gupta, S.V.N. Sreenivasu, Kuldeep Chouhan, and al. Novel face mask detection technique using machine learning to control covid'19 pandemic. Materials Today: Proceedings, 2021.

[11] Kaiming He, Georgia Gkioxari, Piotr Doll´ar, and Ross Girshick. Mask r-cnn. In IEEE International Conference on Computer Vision (ICCV), 2017.

[12] D.P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In International Conference on Learning Representations (ICLR), 2014.
[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, and al. Microsoft coco: Common objects in context. In European Conference on Computer Vision (ECCV), 2014.
[14] Mohamed Loey, Gunasekaran Manogaran, Mohamed Hamed N. Taha, and al. 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, 65, 2021.
[15] Mohamed Loey, Gunasekaran Manogaran, Mohamed Hamed N. Taha, and 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, 2021.
[16] Afsana Nowrin, Sharmin Afroz, MD. Sazzadur Rahman, and al. Comprehensive review on facemask detection techniques in the context of covid-19. volume 29, 2020.
[17] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger.CoRR, 2016.
[18] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems (NIPS), 2015.
[19] Mark Sandler, Andrew Howard, Menglong Zhu, and al. Mobilenetv2: Inverted residuals and linear bottlenecks. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[20] Samuel Ady Sanjaya and Suryo Adi Rakhmawan. 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.
[21] Edgar Schonfeld, Bernt Schiele, and Anna Khoreva. A u-net based discriminator for generative adversarial networks. In IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR), 2020.
[22] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
[23] Shay E. Snyder and Ghaith Husari. Thor: A deep learning approach for face mask detection to prevent the covid-19 pandemic. In SoutheastCon, 2021.
[24] Soham Taneja, Anand Nayyar, Vividha, and Preeti Nagrath. Face mask detection using deep learning during covid-19. In International Conference on Computing, Communications and Cyber-Security, 2021.,