Int. J. Nonlinear Anal. Appl. 14 (2023) 1, 1079–1091
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.6891

A new deep learning model to reduce Covid19-based face mask detection

Shwan Hadi Saeed[a], Bishar Rasheed Ibrahim[b,*], Nashwan Adnan Othman[a]

[a]*Department of Computer Engineering, College of Engineering, Knowledge University, Erbil 44001, Iraq*

[b]*Department of Computer Networks and Information Security, Technical College of Informatics – Akre, Duhok Polytechnic University, Duhok 42006, Iraq*

(Communicated by Ehsan Kozegar)

Abstract

The COVID-19 pandemic of the coronavirus is a serious health threat. Governments are taking specific protections, including lockdowns and the need that face masks to be used. Wearing a protective cover is one of the most efficient ways to fight the disease. Due to this reason, offerings are the detection of face masks that can be utilized by specialists to create moderation, prevention, and assessment. According to the recommendations of the World Health Organization (WHO), the best important prevention strategy is using a facial mask. Hence, the need of wearing a mask is so important to save our lives and also protect others appropriately in open places including shopping malls and general stores. Reports demonstrate that face mask wearing whereas at work clearly diminishes the chance of spread. This paper presents a rearranged approach to obtain this reason by utilizing machine learning for detecting face masks. A dataset is utilized to construct this detector of the face mask. Via computer vision techniques and algorithms using deep learning the goal can be achieved. It includes the architecture of the MobileNet model that is trained with Tensorflow and Keras libraries. In this paper, Jetson tx2 was utilized to implement a real-time face masks automatic detection that is embedded and powerful, running on an embedded system at a higher frames-per-second rate (FPS) based on IoT. The proposed system aids in monitoring, taking images, and identifying persons who were not wearing masks. Additionally, we employed IoT strategies to transmit the images and alerts to the closest police station so that forfeit could be applied when it discovered unmasked persons. We used an actual dataset to train our model, and this improvement makes the recommended approach possible to seek a high level of accuracy rate of unmasking people our model trained on the real datasets and this improvement makes the proposed model urge a high accuracy in the detection of unmasking persons.

Keywords: COVID-19 Virus, Masked Face Recognition and Detection, Masked Face Dataset, Safety, Deep Learning Algorithm, TensorFlow, Keras, Python Programming, Computer Vision, Neural Networks, Facial Landmarks, IoT Health Care, Raspberry Pi, MobileNetV2, Machine Learning
2020 MSC: 68T07

*Corresponding author
Email addresses: shwan.hadi@knu.edu.iq (Shwan Hadi Saeed), bashar.rashid@dpu.edu.krd (Bishar Rasheed Ibrahim), nashwan.adnan@knu.edu.iq (Nashwan Adnan Othman)

Received: April 2022 *Accepted:* August 2022

1 Introduction

On January 12th, 2019, the COVID-19 virus broke out in the United States, with China serving as the main point of transmission. Later, the virus spread to almost every country across the globe [14]. According to WHO, body interaction between different persons and fluids of an infected are the two primary causes that affect the transmission of this sickness [1]. Droplets of breath from other persons nearby (within 1 meter) may move via the air and reach other closer surfaces in the event that an infected person coughs or sneezes. Almost any surface can have this disease, which increases the risk of transmission.

Everybody is advised to utilize a face mask when out in public places due to the outbreak of COVID-19. According to the recommendations of WHO to protect the individual's life during the COVID-19 pandemic wearing masks can control the virus's spread (by infected people to prevent encouraging spread) or verify healthy persons [15].

Face detection and recognition for persons is an amazing computer vision-based Internet of Things application. In order to develop a more effective IoT security system for intelligent homes, computer vision and IoT may be combined. This is because these systems are able to detect humans in inappropriate areas who may be causing harm to the environment [3]. For surveillance applications and security systems, real-time face detection frameworks are essential [11]. The widespread use of face masks presents an issue to access face detection-based security frameworks, which are not capable of recognizing facial masks, like those used at airports. hence accurate performance at that point may be achieved by insulating uncovered faces.

Without a doubt, a contemporary measurement of the living being has been introduced into the Internet of Things through the connectivity of smart objects. It was conceivable to connect any device with any media at any time and location in this way. There would be an increase in the number of networked devices dramatically by 2021, according to expectations reaching more than 34 billion units in the IoT world. The Internet of Things (IoT), an underutilized concept, spans comprehensive appropriateness in numerous areas, including health care. [10].

Organic neural networks are used in deep learning, a creative part of machine learning approaches to address issues in bioinformatics, computer vision, natural network processing, and other areas [9]. To execute object detection and acknowledgment, researchers used the deep-learning library and OpenCV's deep neural network (DNN) module [16]. Caffe, TensorFlow, Burn, and Theano are just a few of the deep learning packages available.

A framework that detects whether or not an individual is wearing a cover is referred to as mask detection. It is identical to the object identification and detection model, which recognizes a certain category of items. We hope that by constructing this system, we would be able to help ensure the security of persons in public areas. This technique may be applied in a number of places, such as stores and malls, educational institutions like colleges and universities, and public places like railway stations. This will be accomplished by optimizing the MobileNet V2 design, a highly effective structure that can be deployed on embedded systems with little computing power. In this case, the Internet of Things put up a Telegram application that was used to track activities, get notifications, and see images.

We used Keras/Tensorflow and OpenCV in this work, and we produced our show utilizing a huge dataset of faces both with and without masks. The haar cascade technique is then used to recognize faces in the acquired image. The photographs are then sent through the internet to a smartphone. The Telegram app in this instance, which was used to track the activity and get updates and pictures, was made possible by the Internet of Things. The Viola-Jones technique makes use of Adaboost learning to identify the face from Haar-like characteristics.

There are various ways in daily life for preparing to the Internet and to make our day by day life simpler and more pleasing like Smartphone, computers, tablets, smart cars and some Smart TVs [2].

Since TensorFlow is the strongest library for implementing a deep learning approach, we used it. The TensorFlow library has several appealing features, including excellent speed, ease of setup, support for CPU and GPU utilization, joins for Python, and easily readable source code.

The leftover portion of the paper is organized as takes after: The second section looks into similar work on facial cover detection. The nature of the dataset used is discussed in Section III.

2 Literature review

The author Das et al. [6] propose a disentangled approach using Keras, TensorFlow, Scikit-Learn, OpenCV, and other basic Machine Learning techniques to uncover this answer. The suggested approach works as intended in identifying the face in the photos and then ascertains whether a mask is worn or not. Additionally, when completing an observational job, it is possible to detect the face and the mask in motion. The model achieves an accuracy of up to 94.58% and 95.77%, independently, on two distinct datasets. In order to properly identify the nearness of face masks without producing over-fitting, the Sequential Convolutional Neural Network is employed to investigate the best parameter values.

Deena Nath [8] thinks to utilize a convolutional neural network model to decide whether or not an individual is utilizing a mask. The show makes use of the powerful TensorFlow library.

The exhibition is based on a collection of 3835 pictures, including photographs taken in 1919 of individuals without masks and photographs taken in 1916 of people wearing masks.

Kaggle Datasets, RMFD Datasets, and Bing Search API were used to compile the dataset's images. When produced with TensorFlow CPU 2.3.0, the proposed Deep Learning demonstration had a 99 percent accuracy. Supriya Kurlekar and colleagues [7] As a result of the Covid-19 episode, this framework can be employed in real-time applications that demand face-mask detection due to security concerns. To ensure that public security requirements are followed, this project can be coordinated with implanted systems for use in railroad stations, airports, schools, businesses, and public areas.

Aniruddha Srinivas Joshi and Shreyas Srinivas Joshi [13] provide a deep learning-based approach to detecting face masks in videos. The suggested approach shows that it can recognize the distinctive facial features(landmarks) that makeup faces inside a video frame using MTCNN face detection.

In the processing of these face photos and signals, a mystery classifier that uses the MobileNetV2 architecture as a protest detector for identifying masked regions does the classification. A dataset was used to test the suggested approach which may have been a collection of movies documenting people's movements in public locations while observing COVID-19 security guidelines. By achieving high accuracy the proposed technique demonstrated its efficacy in facials masks detection. Face cover detection, Deep Learning, and Computer Visio are all terms used in this record.

Adnane Cabania et al. [4] present a technique for image modification with three datasets for masked face detection: the Inaccurately Masked Face Dataset (IMFD), the Correctly Masked Face Dataset (CMFD), and their combination for universal masked face identification (MaskedFace-Net).

There are suggested useful masked face datasets with two purposes in mind: i) identifying persons whether or not their faces are covered with the mask, and ii) recognizing faces to determine whether or not masks are worn appropriately (e.g.; at air terminal portals or in swarms). No other large dataset of masked faces, to our knowledge, offers such granular classification for investigations of mask-wearing. Additionally, this paper proposed a universally applied mask-to-face changeable display that enables the era of extra masked face images, notably with distinctive masks.

3 System Structure

Technology affects practically everything in the twenty-first century. TensorFlow, OpenCV, and Computer Vision technologies are used in the Face Mask Detection system. Deep learning technology for protest recognition and detection, is similar to metal detectors used at malls, airports, and other public places. Recent research has shown that deep convolutional neural networks are superior to humans in object recognition and detection. In this work, algorithms of deep learning are employed to make distinctions between facial recognition and the presence or absence of a facemask.

In this research, face mask detection is accomplished by combining an algorithm for machine learning with the MobileNetv2 photo categorization technique. MobileNetV2 is a Google-developed approach based on Convolutional Neural Networks (CNNs) that has enhanced performance and efficiency [12].

The research in this paper was done on two different datasets. The Real-World Masked Face dataset (RMFD) and Kaggle dataset were combined to create the first dataset, which was used for the model's testing, training, and validation stages.

The model may be made in a few stages. First, there's face recognition. Second, The deep learning system approach of face mask detectors is used after the face mask detection method to identify the face masks. Finally, based on the findings of stages 2 and 3, the face mask is estimated. Depending on whether someone is mask-wearing, the warning message is sent via the IoT system. Fig. 1 shows the proposal's flow chart in its entirety.

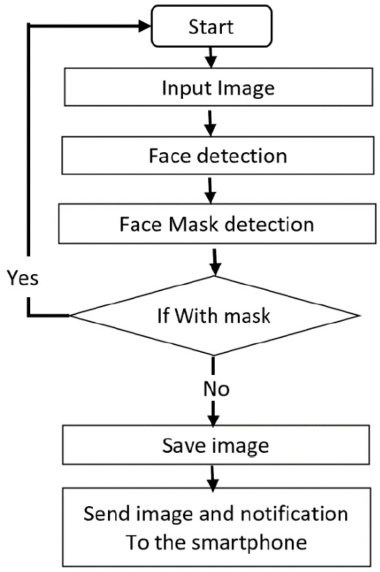

Figure 1: The suggested system's flowchart

3.1 Face Detection Stage

The main step in the system proposal is face detection. Regular face identification systems were built on conventional machine learning algorithms in the beginning. Viola and Jones introduced the most effective systems for detecting faces.

The final stage of the Viola-Jones face location calculation is the Cascade classifier organization. The cascade arrange is used to quickly dispose of face candidates. The channels of a cascade classifier are divided into several tiers that determine whether a given sub-window is certainly not a face or whether it is a face.

Viola Jones, the original [1] Face detection is used in a wide range of applications. Others are ignored when facial highlights are noticed. The Haar cascade technique is the most widely used face detection algorithm. It's used to distinguish between one or several faces. Due to its popularity as the most widely used open-source library for image processing, the OpenCV library is used to find faces [2].

They're easy to process and aren't overly complicated. This offered a quick response in recognizing the face in the image, however, it failed to detect faces from different perspectives. The mass of research done in the field of face identification is anticipated to employ a deep learning model due to the rapid growth of deep learning. In this work, a deep learning-based face detector is used. The ResNet-10 Architecture, which is built on the Single Shot Multibox detector, serves as the model's foundation [1].

In order to obtain fixed-size groups of bounding boxes and scores for the existence of object class occurrences in those boxes, the SSD technique first employs a feed-forward convolutional network. A non-maximum concealment step is then used to produce the final findings. The architecture of the SSD network is shown in Figure 2.

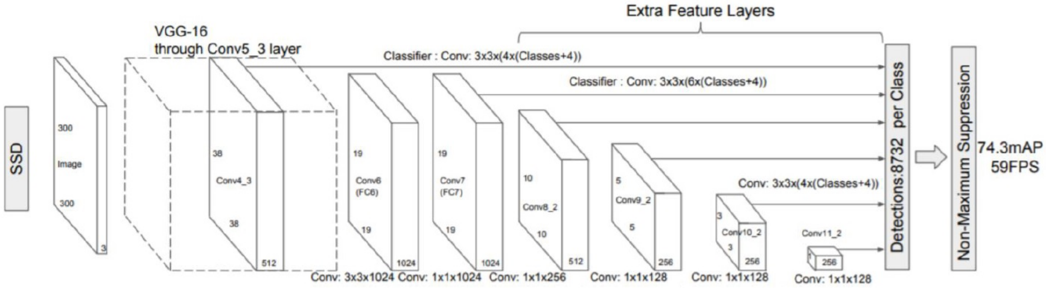

Figure 2: Single Shot Detector for object detection using MultiBox

3.2 Face mask detection algorithm Stage

3.2.1 Dataset

Our deep learning/computer vision pipeline is demonstrated through the two-phase COVID-19 face mask detection model. The dataset that we need to make our original face mask detector will then be covered. Then, We demonstrate how to create a Python script that uses TensorFlow and Keras to train a face mask detector on our dataset. This Python script will be used to train the model, and the output will be evaluated.

Two further Python scripts will be executed using the learned COVID-19 face mask detector model to:

- Recognize COVID-19 masks in photos
- Masks in real-time video feeds can be found.

Finally, will examine the results of employing the model before we conclude.

a two-phase process COVID-19 is a detector for face masks.

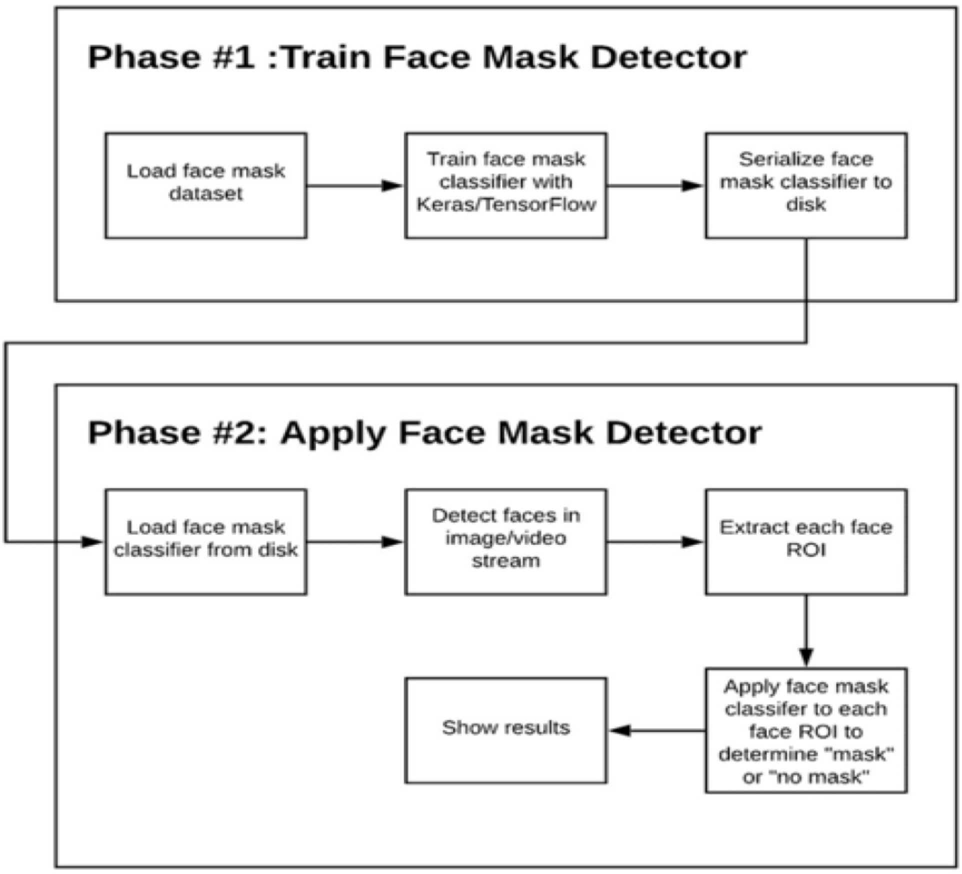

Figure 3: Stages and processes for developing a computer vision and deep learning model for detection of COVID-19 facial mask utilizing Open CV, Tensor Flow/Keras, and Python

As shown in Fig. 3, we need to split our work into 2 stages, each with its own set of sub-steps, to train customized face mask detectors.

1. **Training:** In this section, We'll concentrate on importing the dataset of face mask detection from the disk, train the model on it (via Keras/TensorFlow), then serialize the face mask detectors to the disk.
2. **Deployment:** When the model is trained it can be loaded to face detection, and classify each face as having a mask or not.

We'll go over each of these stages and the corresponding subsets in more detail later on in this section, but first, we will explain the trained dataset.

This dataset was created using common face photos. Also, a new computer vision Python script was developed to add masks to these pictures, producing an artificial but practical dataset. We use face landmarks to the problem to make this approach easier. Face landmarks help us to deduce the placement of facial elements including the eyes, brows, mouth, nose, and jawline automatically.

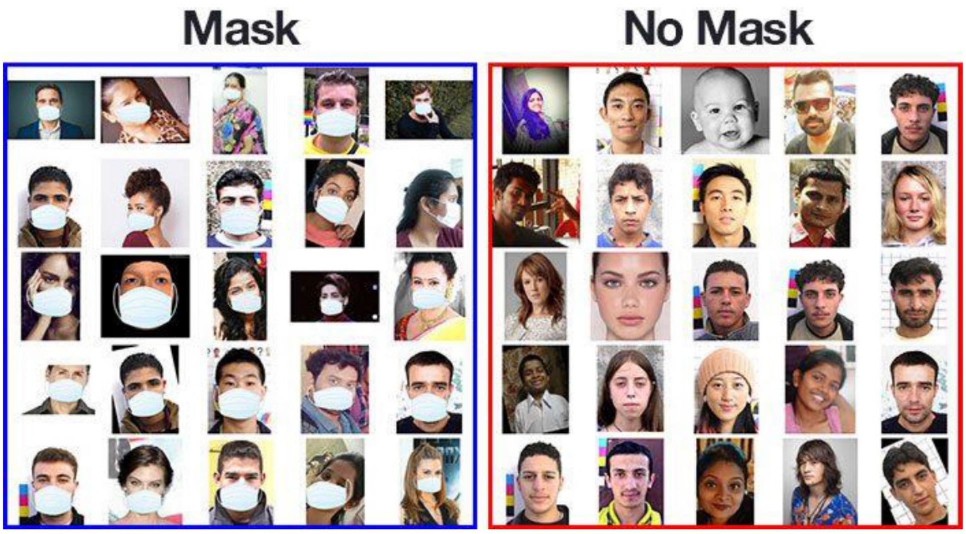

Figure 4: Two image categories, "with mask" and "without the mask", make up a face mask detection dataset

A COVID-19 face mask detector was created utilizing computer vision and deep learning methods with Python, OpenCV, and TensorFlow/Keras.

We begin with images of people who aren't wearing the mask and develop from there to get our artificial mask dataset. To locate the bounding box of a face in an image, face detection is utilized. When the face is detected in the picture, the Region of Interest (ROI) image is retrieved. To identify the mouth, nose, eyes, and other facial features, we employ facial landmark analysis. This technique resists facial deformations brought on by organic movements, positions, and expressions, both rigid and nonrigid.

Photographs are taken of a mask with a transparent backdrop. Employing facial landmarks that are retrieved along the nose and chin this type of mask will be automatically adjusted to fit the face in an accurate way. After that, the mask is rotated and enlarged before being put on the face. The technique is repeated for each of the input images to create the artificial face mask dataset.

There are 1,376 images in this collection, categorized into two groups:

- 686 images without the mask
- 690 images with mask

3.2.2 Training stage by using MobileNet Architecture

The MobileNetV2 architecture is based on an inverted residual architecture with narrow bottleneck layers at the output and input of the convolution layer. The MobileNetV2's middle expansion layer filters feature using lightweight depth-wise convolutions, in contrast to conventional residual models, which employ expanded representations as input. Also, we found that maintaining predictive accuracy requires removing non-linearities in the thin layers. We show how this enhances performance and go into the ideas that led to the design. Finally, our approach allows for a more simple conceptual approach by separating the input/output fields from the expressiveness of the transformation. We employ picture categorization, VOC image segmentation, and COCO object detection to evaluate our results. The accuracy, the quantity of multiply-add (MAdd) operations, and the quantity of parameters are compared.

MobileNetV2, which improves mobile model performance across a range of model sizes, activities, and benchmarks. We describe effective methods for exploiting these mobile models for object identification in a new framework we call

SSDLite. We also demonstrate how to develop mobile semantic segmentation models using Mobile DeepLabv3, a reduced version of DeepLabv3.

Prajna had an ingenious solution for creating datasets:

1. Capturing typical pictures of the faces
2. It is then feasible to apply face masks to them by building a special computer vision using a Python script, creating a fake (but still relevant) dataset.

For the purpose of identifying face landmarks in a picture, we'll utilize OpenCV and dlib.

To identify and depict important areas of the face, facial landmarks are employed, like:

- Jawline
- Mouth
- Nose
- Eyebrows
- Eyes

Face alignments, head posture estimate, blink detection, face swapping, and many more processes have all been made effective with the use of facial landmarks.

Additionally, we'll concentrate on the fundamental face landmarks, such as:

1. The purpose of and characteristics of facial landmarks.
2. Identify and extract face landmarks from a picture combining Python, OpenCV, and dlib.
3. Additionally, we focus deeply on landmarks of the face and learn how to extract certain facial parts based on these via Python. Face landmarks and why they're significant in computer vision will be dlib, OpenCV, discussed in the first section of this blog article.

Figure 5: Key face characteristics in an image are labeled and identified using facial landmarks

Face landmark detection is a subset of the shape prediction issue. Image input is used by a shape predictor to try to pinpoint important places along a form (and typically an ROI image describes the item of interest). In the context of facial landmarks, our objective is to employ shape prediction techniques to identify crucial facial components. Therefore, identifying face landmarks requires two steps:

- Step 1: Locate the face in the photo.
- Step 2: On the face ROI image, identify the important facial structures.

Applying Step 2: identifying important facial features in the face region

- Jaw
- Mouth
- Nose
- Right eye
- Left eye
- Right eyebrow
- Left eyebrow

Face landmark detection by dlib

We determine the positions of 68 (x, y)-coordinates that represent face structures using the pre-trained facial landmark detector from the dlib package.

In the figure below, the 68 coordinates' indexes can be seen:

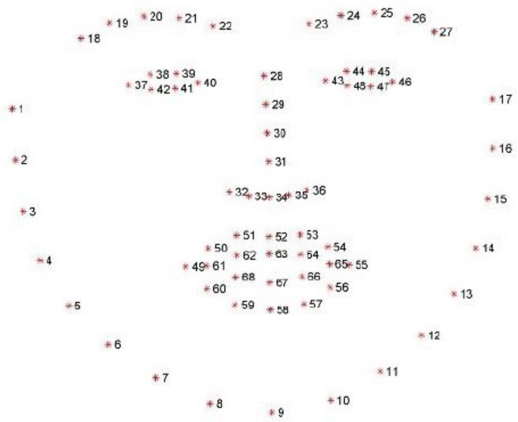

Figure 6: Displaying the 68 facial feature positions from the iBUG 300-W dataset

Additionally, there are other varieties of facial landmark detection, for instance, the 194 position model that may be trained with the dataset.

Facial landmark visualizations

Figure 7: Using Python, OpenCV, and dlib for face landmark identification

As illustrated in Fig. 7, we'll then require a mask picture with a transparent backdrop.

Figure 8: Using a COVID-19 mask. This mask is applied to the original face ROI image since we are aware of the locations of the facial landmarks

Based on facial landmarks, Automatic application of the mask on the face (specifically, the spots around the chin and nose).

The mask is then adjusted in size and orientation before being put on the face:

Figure 9: In the first frame of this image, the person's face is covered by a face mask. At first glance, it's impossible to tell that the COVID-19 mask was created using computer vision using dlib face landmarks and OpenCV

This technique may then be done for all of our input photos, yielding the following dataset:

Figure 10: COVID-19 face mask photos are exhibited in an artificial set

This dataset will be utilized in our COVID-19 face mask detection with deep learning and computer vision using TensorFlow/Keras, OpenCV, and Python "with mask" / "without mask" dataset.

The structureof project ids like this

$ Face-Mask-Detection

- **Dataset**
  - **With_mask[690 entries]**
  - **Without_Mask[686 entries]**
- **Face_detector**
  - **Deploy.prototxt**
  - **Rres10+300X300_ssd_iter_140000.caffemodel**
- **Detect_mask_video.py**
- **Mask_detector.model**
- **Plot.png**
- **Train_mask_detector.py**

  2 directories,8 files

3.2.3 Classification stage

After reviewing our face mask dataset, now we use TensorFlow and Keras to develop a classifier that can be trained to recognize mask wearers.

We used images of people who were not masking their faces to train the images. The application of facial detection is the next phase. Face detection with OpenCV was achieved using a deep learning algorithm. The face ROI images are then obtained using NumPy slicing and OpenCV. Afterward, we identify the nose, mouth, eyes, and other characteristics by using facial landmarks.

The objective of our method is to recognize those who aren't wearing face masks. Based on the input picture, the learning architecture generates a response that classifies the photo as having a mask or not. Anytime a person is observed without a face mask on, an alert beeps. Additionally, if everyone dons a mask, they will be immune to the illness.

Import the libraries of TensorFlow. Keras enables the following steps;

- Pre-processing
- MobilNetV2 classifier loading (it is pre-trained with ImageNet weights)
- Developing a novel fully-connected (FC) head
- Augmentation of data
- Picture data loading

To segment our dataset, binarize the class labels, and produce a classification report, we utilized scikit-learn (sklearn).

In order to fine-tune, MobileNetV2 was created. We almost always propose fine-tuning as an approach for establishing a baseline model while saving time. This procedure includes three main steps:

- Pre-trained ImageNet weights should be loaded into MobileNet, ignoring the network's head.
- Replace the older FC head by creating a new one and attaching it to the base.
- The base layers of the network should be frozen. Backpropagation will not change the weights of these base layers; however, it will change the weights of the head layers.

In order to increase generalization, on-the-fly mutations were applied to our photos during training. Finding the random rotations, zooming, shearing, shifting, and flipping parameters is the process of data augmentation.

We were able to discover and list images in our dataset thanks to the imutils paths implementation. Our training curves were plotted using matplotlib.

With our deep learning and computer vision expertise, we prepared a face mask recognition using TensorFlow/Keras, Python, and OpenCV.

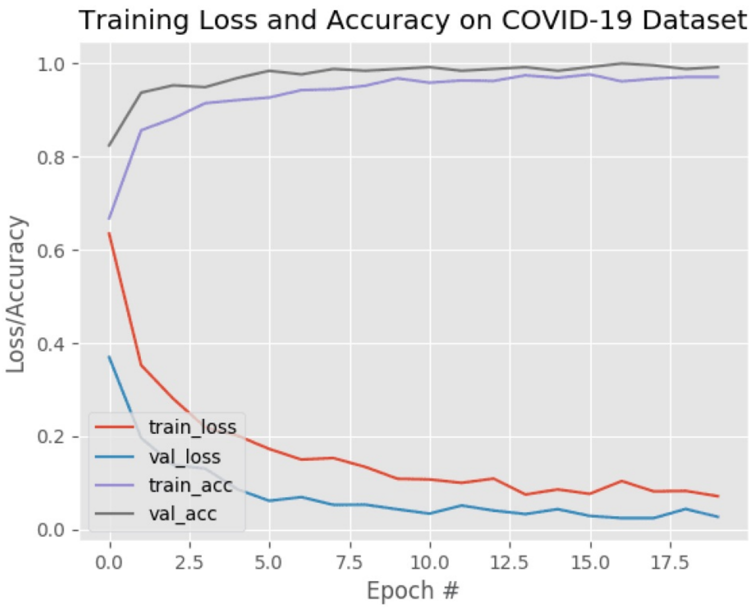

Figure 11: The curves of accuracy for training and loss for the COVID-19 face mask detector demonstrate high accuracy and minimal sign of overfitting the data

As you can see, our test set has a 99 percent accuracy rate. Figure 11 shows that the validation loss is less than the training loss, there are no clear signs of overfitting. We believe the proposed model will perform well when applied to photos that are not part of our test and training collection, based on these findings.

3.3 IoT API (Telebot API)

Smart security, smart transportation, smart cities, health care, smart grids, and online commerce are just a few of the industries where the technology is gaining traction.

Wireless Sensor Networks, which are at the heart of the Internet of Things is the most crucial. A Wireless Sensor Network (WSN) is a network that connects sensors to a server or a single device to collect data and can be used to automate routine chores in one area [5].

Figure 12: The IoT System

IoT has been used in a variety of applications. Some of these IoT systems offer features that others do not.

We describe how to create a Telegram bot and set up the rule engine of ThingsBoard so that it may send alerts to the Telegram application using the Rest API Call extension. You may build Telegram Bots, which are third-party apps, with the help of the platform.

The capability of converting existing protocols and payload formats to ThingsBoard format messages makes HTTP Integration beneficial in a variety of deployment scenarios.

- Stream devices and/or assets data from a back end of a connection provider, IoT platform, or external system.
- From any proprietary cloud-based app, stream asset or/and device data.
- Via a unique HTTP-based protocol, link the existing devices to ThingsBoard.
- the steps to follow in order to generate a token authorization for a new bot after it has been created. A token is a group of letters that resembles this —

'110201543:AAHdqTcvCH1vGWJxfSeofSAs0K5PALDsaw' it is necessary to give the bot authorization.

How to Get the Chat ID We'll need to get a Chat ID in the next stage. To send messages over the HTTP API, you'll need the Chat ID.

Adjust Send Email, Create/Clear Alarm

Adding the required nodes

- Transform Script
- With a relation type of Created, add the Transform Script node and link it to the Create Alarm node.
- The body of the notification message will be created using this node.
- REST API Call

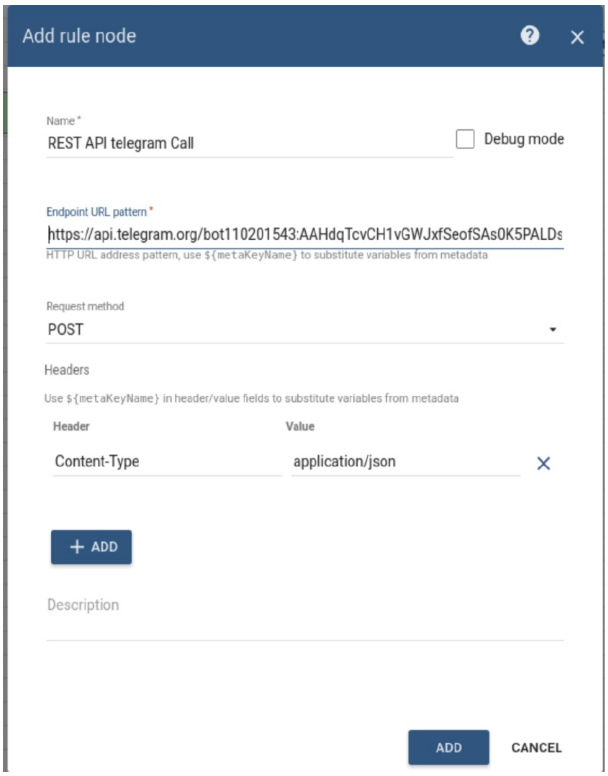

Figure 13: Modify Create/Clear Alarm & Send Email

We'll use the Rest APIs and Telemetry upload APIs to post device telemetry. The Thermostat Home device's device access token must be copied for this.

4 Conclusion

COVID-19's spreading has presented a number of issues to the world, and the virus's transmission must be slowed because it has affected more than one crore people worldwide. Wearing a mask is the most significant safeguard, as it prevents the spreading of contaminated individuals' respiratory droplets through coughing or sneezing. Healthy people should also wear masks. We learned how to use Deep Learning approaches, OpenCV, and Keras/TensorFlow to propose a COVID-19 facial mask detection model. Also, the research puts on two different cases show with those who wear masks and those who weren't wearing masks to create our face mask locator. With our masked/no masked dataset, we fine-tuned MobileNetV2 and obtained a 99-percent-accurate classifier. We coupled this face mask classifier to both photos and real-time video streams. Face recognition in photos and videos Getting each person out of their situation Our face cover classifier in action due to the MobileNetV2 architecture, our face mask detection is accurate, Its computational effectiveness makes it simpler to transmit the show to embedded systems (the Raspberry Pi 3 was used to run the frameworks because it is a low-cost embedded hardware stage with great details).

References

[1] A.A.R. Alsaeedy and E. Chong, *Detecting regions at risk for spreading covid-19 using existing cellular wireless network functionalities*, IEEE Open J. Eng. Med. Biol. **1** (2020), 187–189.

[2] I. Aydin and N.A. Othman, *A new IoT combined face detection of people by using computer vision for security application*, Int. Artific. Intel. Data Process. Symp.(IDAP), IEEE, 2017, pp. 1–6.

[3] P.N. Belhumeur, J.P. Hespanha and D. Kriegman, *Eigenfaces vs. fisherfaces: Recognition using class specific linear projection*, IEEE Trans. Pattern Anal. Machine Intel. **19** (2007), no. 7, 711–720.

[4] A. Cabani, K. Hammoudi, H. Benhabiles and M. Melkemi, *MaskedFace-Net–A dataset of correctly/incorrectly masked face images in the context of COVID-19*, Smart Health **19** (2021), 100144.

[5] N.M. Cuong, D.N. Au, Q.N. Doan, P.H. Bui, D.C. Do and H. Pham–Viet, *A real-time drowsiness detection system based on raspberry*, https://www.researchgate.net/publication/352559269_A_Real-time_Drowsiness_Detection_System_Based_on_Raspberry, 2020.

[6] A. Das, M.W. Ansari and R. Basak, *Covid-19 face mask detection using TensorFlow, Keras and OpenCV*, IEEE 17th India Council Int. Conf.(INDICON), IEEE, 2020, pp. 1–5.

[7] M.S. Kurlekar, *Face mask detection system using deep learning*, Turkish J. Comput. Math. Educat. **12** (2021), no. 7, 1327–1332.

[8] D. Nath, *Face mask detection*, EasyChair, (2021), no. 5976.

[9] N.A. Othman and I. Aydin, *A new deep learning application based on movidius NCS for embedded object detection and recognition*, 2nd Int. Symp. Multidiscip. Stud. Innov. Technol.(ISMSIT), IEEE, 2018, pp. 1–5.

[10] Y. Perwej, K. Haq, F. Parwej, M. Mumdouh and M. Hassan, *The internet of things (IoT) and its application domains*, Int. J. Comput. Appl. **182** (2019), no. 49, 36–49.

[11] G. Sánchez-Arias, C.G. García and B.C.P. G-Bustelo, *Midgar: Study of communications security among smart objects using a platform of heterogeneous devices for the internet of things*, Future Generat. Comput. Syst. **74** (2017), 444–466.

[12] S.A. Sanjaya and S.A. Rakhmawan, *Face mask detection using MobileNetV2 in the era of COVID-19 pandemic*, Int. Conf. Data Anal. Bus. Ind.: Way Towards Sustain. Econ. (ICDABI), IEEE, 2020, pp. 1–5.

[13] A. Srinivas Joshi, S. Srinivas Joshi, G. Kanahasabai, R. Kapil and S. Gupta, *Deep learning framework to detect face masks from video footage*, arXiv e-prints, arXiv-2011, (2020).

[14] A. Waheed, M. Goyal, D. Gupta, A. Khanna, F. Al-Turjman and P.R. Pinheiro, *CovidGAN: Data augmentation using auxiliary classifier GAN for improved covid-19 detection*, IEEE **8** (2020), 91916–91923.

[15] WHO, *Coronavirus disease (COVID-19) advice for the public: When and how to use masks*, https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public/when-and-how-to-use-masks, (2021).

[16] http://opencv.org/.