

Covid-19 Face Mask Detection Using Machine Learning

Sable Ujwala P.¹, Kale Nikita S.², Maskar Sayali B.³ and Prof. Khalate Y. R.⁴

¹Department of Computer Engineering, SVPMCOE Baramati, India

²Department of Computer Engineering, SVPMCOE Baramati, India

³Department of Computer Engineering, SVPMCOE Baramati, India

⁴Department of Computer Engineering, SVPMCOE Baramati, India

ABSTRACT

The rise of COVID-19 pandemic had a lasting impact in many countries worldwide since 2019. Face-mask detection had been significant progress in the Image processing and deep learning fields studies. Many face detection models have been designed using different algorithms and techniques. The proposed approach in this system developed to avoid mask-less people from entering to a desired places by detecting face mask using deep learning, TensorFlow, Keras, and OpenCV and sending a signal to Arduino device that connected to the gate to be open. it detect a face in a real-time and identifies whether the person wear mask or not. The method attains accuracy up to 97.80%. The dataset provided in this paper, was collected from various sources.

Keywords: : Face Mask Detection, Deep Learning, TensorFlow, Keras, OpenCV, Arduino.

1. INTRODUCTION

According to World Health Organization (WHO), since December 2019 more than 114 countries suffered from COVID-19 pandemic which has declared as a deadly diseases that has globally infected over 110 million people causing more than 2.43 million deaths in the worldwide as on Feb 18, 2021, In Iraq since the rise of COVID-19 more than 653000 effected case and more than 13000 deaths has been registered, this is Deu to the insufficient vaccinees to overcome this deadly disease till date [1]. Wearing a face-mask during this pandemic is a critical defensive in times when social distancing is hard to maintain.

Therefore, many face mask detection and monitoring systems have been developed to provide effective supervision for hospitals, public transportations, airports, retail locations, and sports venues. Over years, in the field of image processing, computer vision and pattern recognition, face detection is the very first step for various applications that depends facial analysis algorithms for identifies, recognize human faces and also to capturing facial motions in digital images, including the face recognition, face alignment, face verification, age recognition, face modelling, face authentication, access control, forensics, and human-computer interactions. face relighting, facial expression tracking, head pose tracking, facial expression recognition, gender recognition, and other face-detection based applications [3]

After the arise of Covid-19, the Face-Mask detection has widely considered problem in the image processing field. This technology is currently more appropriate because it is applied to detect faces and to identify people wearing masks in images, videos and also in real-time vision.[1] By using deep learning and convolution neural network (CNN) techniques, it become possible to achieve high accuracy results in image classification and object detection applications.[3] Creating a system for detecting the face-mask will provide a way for controlling the

people who enters any places. The proposed system in this paper uses deep learning, TensorFlow, Keras, and OpenCV which are used as an image classifier to detect face-mask and sends a signal to Arduino device that control the open and close of door.[2]

LITERATURE REVIEW

COVID-19 pandemic has rapidly affected our day- to-day life disrupting the world trade and movements. Wearing a protective face mask has become a new normal. In the near future ,many public service providers will ask the customers to wear masks correctly to avail of their services. This paper presents a simplified approach to achieve this purpose using some basic Machine Learning packages like TensorFlow, Keras, OpenCV and Scikit-Learn.[Arjya Das, Mohammad Wasif Ansari, Rohini Basak.]

Many face detection models have been designed using different algorithms and techniques. The proposed approach in this paper developed to avoid mask-less people from entering to adesired places by detecting face mask using deep learning, TensorFlow, Keras, and OpenCV and sending a signal to Arduino device.[Saman M. Almufti Ridwan Marqas]

Monitoring manually if the individuals are wearing facemask correctly and to notify the victim in public and crowded areas is a difficult task. This paper approaches a simplified way to achieve facemask detection and notifying the individual if not wearing facemask. Using Kaggle datasets,the proposed system/model is trained and examined.[Suresh K1 , Palangappa MB2 ,Bhuvan S3]

The rise of COVID-19 pandemic has had a lasting impact in many countries worldwide since 2019. Face-mask detection had been significant progress in the Image processing and deep Learning fields studies. Many face detection models have been designed using different algorithms and techniques. The proposed approach in this paper developed to avoid mask-less people from entering to a desired places (i.e. Mall, University,Office, . . . etc.) by detecting face Mask using deep learning, TensorFlow, Keras, and OpenCV and sending .A signal to Arduino device that connected to the gate to be open. It detect a face in a real-time and identifies whether the person wear mask or not.The method attains accuracy up to 97.80[Saman M. Almufti Ridwan B. Marqas Zakiya A. Nayef Tamara S. Mohamed]

2. PROJECT DESIGN

Creating a system for detecting the face-mask is providing a way for controlling the people who enters any public places. The system used TensorFlow, Keras, and OpenCV which is used as an image classifier to detect face-mask and sending signal to Arduino device that is controlling bar.

2.1 Problem Statement

The person who not wearing a mask are not allow to enter in the particular places and to Develop such a Machine Learning Model we are using CNN approach that detect face mask in real-time video stream. Face mask detection refers to detect whether a person is wearing a mask or not.

2.1.2 Algorithms Used

- **CNN Algorithm**

Input: Dataset including faces with and without masks

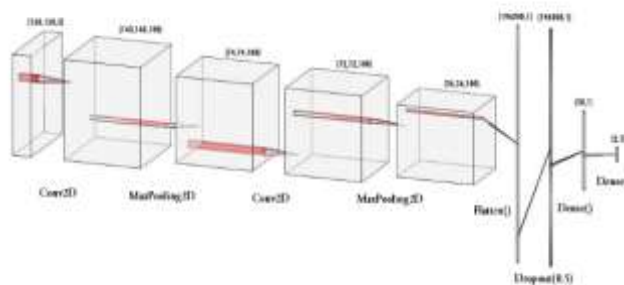
Output: Categorized image depicting the presence of face mask

1. for each image in the dataset do

2. Visualize the image in two categories and label them
3. Convert the RGB image to Gray-scale image
4. Resize the gray-scale image into 100 x 100
5. Normalize the image and convert it into 4 dimensional array
6. end
7. for building the CNN model do
8. Add a Convolution layer of 200 filters
9. Add the second Convolution layer of 100 filters
10. Insert a Flatten layer to the network classifier
11. Add a Dense layer of 64 neurons
12. Add the final Dense layer with 2 outputs for 2 categories
13. end
14. Split the data and train the model Project

3. What is the CNN Algorithm?

One of the main parts of Neural Networks is Convolutional neural networks (CNN). CNNs use image recognition and classification in order to detect objects, recognize faces, etc. They are made up of neurons with learnable weights and biases. Each specific neuron receives numerous inputs and then takes a weighted sum over them, where it passes it through an activation function and responds back with an output. CNNs are primarily used to classify images, cluster them by similarities, and then perform object recognition. Many algorithms using CNNs can identify faces, street signs, animals, etc.



How does CNN algorithm work?

A convolutional network ingests such images as three separate strata of color stacked one on top of the other. A normal color image is seen as a rectangular box whose width and height are measured by the number of pixels from those dimensions. The depth layers in the three layers of colours (RGB) interpreted by CNNs are referred to as channels.

Steps :-

- Step 1: Data Visualization In the first step, let us visualize the total number of images in our dataset in both categories.
- Step 2: Data Augmentation In the next step, we augment our dataset to include more number of images for our training. In this step of data augmentation, we rotate and flip each of the images in our dataset.
- Step 3: Splitting the data In this step, we split our data into the training set which will contain the images on which the CNN model will be trained and the test set with the images on which our model will be tested.
- Step 4: Building the Model In the next step, we build our Sequential CNN model with various layers such as Conv2D, MaxPooling2D, Flatten, Dropout and Dense. In the last Dense layer, we use the 'softmax' function to output a vector that gives the probability of each of the two classes.
- Step 5: Pre-Training the CNN model After building our model, let us create the

‘train generator’ and ‘validation generator’ to fit them to our model in the next step.

Step 6: Training the CNN model This step is the main step where we fit our images in the training set and the test set to our Sequential model we built using keras library.

Step 7: Labeling the Information After building the model, we label two probabilities for our results. [‘0’ as ‘without mask’ and ‘1’ as ‘with mask’].

Step 8: Importing the Face detection Program After this, we intend to use it to detect if we are wearing a face mask using our PC’s webcam. For this, first, we need to implement face detection. In this, we are going to use the Haar Feature-based Cascade Classifiers for detecting the features of the face.

- **Haar cascade classifier algorithm**

Input: Live video stream

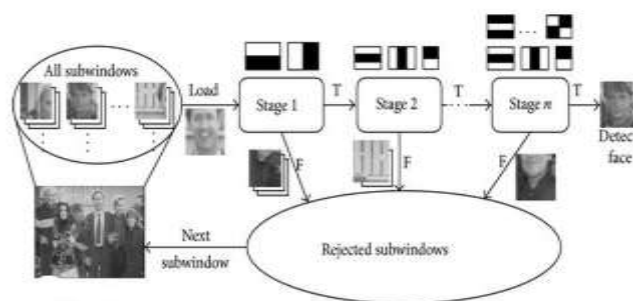
Output: Detected face

1. Loading the image
2. Converting the image to grayscale
3. Loading the required haar-cascade XML classifier file
4. Applying the face detection method on the grayscale image
5. Iterating through rectangles of detected faces
6. Displaying the detected image.

What is Haar cascade classifier?

Haar cascade classifier, is a machine learning object detection program that identifies objects in an image and video.

The term supervised refers to the fact that the attentional operator is trained to detect examples of a particular class. In the domain of face detection it is possible to achieve fewer than 1 from two Harr-like features. The effect of this filter is to reduce by over one half the number of locations where the final detector must be evaluated. Those sub-windows which are not rejected by the initial classifier are processed by a sequence of classifiers, each slightly more complex than the last. If any classifier rejects the sub-window, no further processing is performed. The structure of the cascaded detection process is essentially that of a degenerate decision tree, and as such is related to the work of Geman and colleagues [1, 4].



How does Haar cascade classifier algorithm work?

1. Importing OpenCV

In the command prompt type (pip install OpenCV-python) using this command, we can install and set up OpenCV to python. OpenCV provides a real-time optimized Computer Vision library, tools, and hardware and the same will be used in our project.

- 2.Importing XML file
- 3.Importing the test image.

imread() is a method of OpenCV to read the input. Similarly, imshow() is a method to display the processed input in the form of output.

4. Converting to grey Scale.

This project works on images that are in greyscale and hence we convert the image to greyscale for ease of face detection.

4. 5. Detecting Multi-scale faces. This function allows to detect objects of different sizes in the input image and hence an image with multi people with different sizes of the face can also be detected. Parameters of the functions are – detectMultiScale (InputArray image, double scaleFactor=1.1, int minNeighbors=6)

6. Mentioning sides of the rectangle for face detection.

This function helps us to mention the dimensions thickness and color of the rectangle that will be visible during the face detection.

7. Displaying the detected image.

4.1 System Architecture

In this system we have detected the faces with mask and without mask with the help of Arduino Uno. Firstly, User will come in front of camera. It detects faces by using Haar Cascade classifier. The algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it.

Then it will detect the face region in particular size.

Then the trained CNN model will detect the face mask which trained by Dataset. After that the output will show on screen as well as it will pass to Arduino Uno. Arduino Uno will Open/Close Bar as per the output. If the person is with Mask then it will open Bar and if person is without mask it will close Bar.

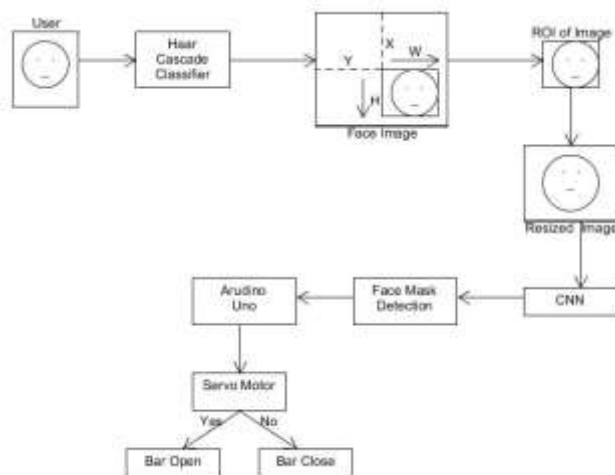
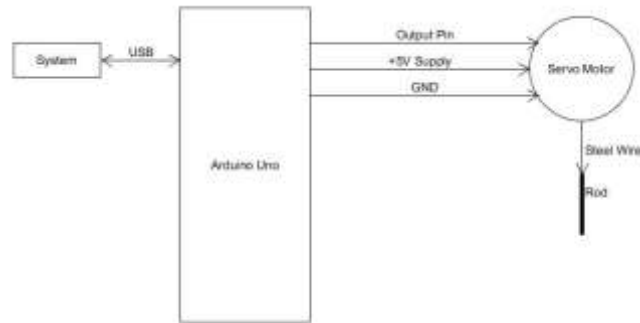


Figure 1: System Architecture

4.2 Hardware Architecture

Face Mask Detection system is detecting the face mask and output of the system is stored in the form of 0 and 1. So, we are passing the output to Arduino uno. 0 means Mask and 1 means No Mask. With the help of servo motor we are opening and closing bar. There are three pins of Arduino uno that is Output pin, 5+ Supply and GND. That are connected to servo motor. Servo motor will open bar if person is with mask and closes bar if person is without mask.



5. RESULTS AND DISCUSSION

The figure shows the model trained by Convolutional Neural Network.

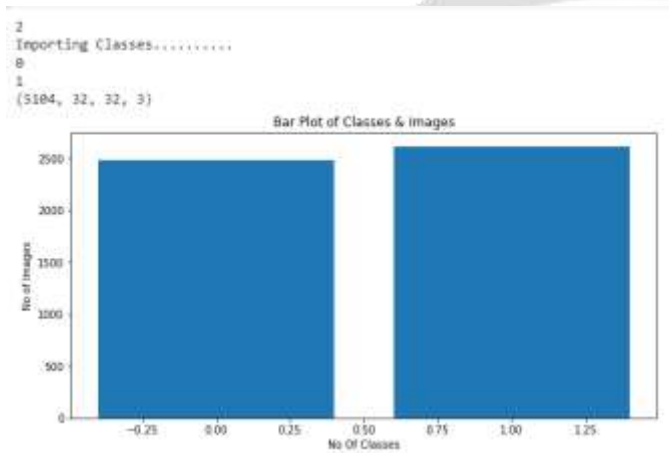


Figure 1: Training Data

6. CONCLUSIONS

Using basic ML tools and simplified techniques our system achieved reasonably high accuracy. Today it has become mandatory for all the citizens to wear a face mask to protect themselves. This application become helpful for all the shop owners, offices, banks or any public place because if anyone is not wearing a mask then he or she must not be allowed in that area. So, to take care of this problem we don't need any guard or person who keeps a watch on people. We integrated a camera which continuously clicks pictures of humans and detect from there faces whether they are wearing a face mask or not.

REFERENCES

- [1] Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9342585>
- [2.] Real Time Face-mask Detection with Arduino to Prevent COVID-19 Spreading <https://journal.qubahan.com/index.php/qaj/article/view/47> [Accessed 4 April 2021].
- [3.] COVID-19: Face Mask Detection using CNN model to detect if a person is wearing a face mask or not with your webcam. <https://towardsdatascience.com/covid-19-face-mask-detection-using-tensorflow-a>
- [4]. Face Mask Detection”, Kaggle.com, 2020. [Online]. Available: <https://www.kaggle.com/andrewmvd/face-mask-detection>
- [5]. “TensorFlow White Papers”, TensorFlow, 2020. [Online]. Available:

<https://www.tensorflow.org/about/bib>. 2020.

[6]. K. Team, "Keras documentation: About Keras", Keras.io, 2020. [On- line]. Available:

<https://keras.io/about>. 2020.

[7]. "OpenCV", Opencv.org, 2020. [Online]. Available:

<https://opencv.org/2020>.

[8]. "Coronavirus Disease 2019 (COVID-19) – Symptoms", Centers for Disease Control and Prevention, 2020. [Online]. Available:

<https://www.cdc.gov/coronavirus/2019-ncov/symptomstesting/symptoms.html>. 2020.

[9]. "Coronavirus — Human Coronavirus Types — CDC", Cdc.gov, 2020. [Online]. Available:

<https://www.cdc.gov/coronavirus/types.html>. 2020.

[10]. P. Kazarinoff, "Using Python to control an Arduino", Python for Undergraduate Engineers, 2021. [Online]. Available:

<https://pythonforundergradengineers.com/python-arduino-LED.html>. [Ac-cessed: 05- Apr- 2021]

