FACE MASK DETECTION
USING
OPENCV, SKLEARN AND NUMPY

A PROJECT REPORT

SUBMITTED TO

DR. ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW

By:

ARISH AHMAD
MOHAMMAD ALI
ABU TALEY SIDDIQUEE

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VIVEKANANDA COLLEGE OF ENGINEERING AND TECHNOLOGY, ALIGARH

ABSTRACT

After the breakout of the worldwide pandemic COVID-19, there arises a severe need of protection mechanisms, face mask being the primary one. To monitor that people are following this basic safety principle, a strategy should be developed. A face mask detector system can be implemented to check this. Face mask detection means to identify whether a person is wearing a mask or not. The first step to recognize the presence of a mask on the face is to detect the face, which makes the strategy divided into two parts: to detect faces and to detect masks on those faces. Face detection is one of the applications of object detection and can be used in many areas like security, biometrics, law enforcement and more.

The basic aim of the project is to detect the presence of a face mask on human faces on live streaming video as well as on images. we develop a new facemask-wearing condition identification method which quantifies a three-category classification problem based on unconstrained 2D facial images. The proposed algorithm contains four main steps: Image pre-processing, facial detection and cropping, image super-resolution, and facemask-wearing condition prediction.

Keywords: Computer vision, Image processing, Open source computer vision library (OpenCV), scikit-learn (Sklearn), Support vector classification (SVC).

INTRODUCTION

Corona-virus disease 2019 (COVID-19) is an emerging respiratory infectious disease caused by Severe Acute Respiratory Syndrome corona-virus 2 (SARS-CoV2). At present, COVID-19 has quickly spread to the majority of countries worldwide, affecting more than 14.9 million individuals, and has caused 618,017 deaths, according to the report from the World Health Organization (WHO) on 23 July 2020 (https://covid19.who.int/). To avoid global tragedy, a practical and straightforward approach to preventing the spread of the virus is urgently desired worldwide. Previous studies have found that facemask-wearing is valuable in preventing the spread of respiratory viruses. For instance, the efficiencies of N95 and surgical masks in blocking the transmission of SARS are 91% and 68%, respectively. Facemask-wearing can interrupt airborne viruses and particles effectively, such that these pathogens cannot enter the respiratory system of another person. As a non-pharmaceutical intervention, facemask-wearing is a non-invasive and cheap method to reduce mortality and morbidity from respiratory infections. Since the outbreak of COVID-19, facemasks have been routinely used by the general public to reduce exposure to airborne pathogens in many countries. Facemasks, when fitted properly, effectively disrupt the forward momentum of particles expelled from a cough or sneeze, preventing disease transmission. However, the effectiveness of facemasks in containing the spread of airborne diseases in the general public has been diminished, mostly due to improper wearing. Therefore, it is necessary to develop an automatic detection approach for facemask-wearing condition, which can contribute to personal protection and public epidemic prevention.

In this project, we will be developing a face mask detector that is able to distinguish between faces with masks and faces with no masks. In this report, we have proposed a detector which employs support vector classification (SVC) to predict the presence or absence of a face mask. The implementation of the algorithm is on images, videos and live video streams.

The rest of the report is organized as follows. In Section 2, we will go through the literature review. In Section 3, the methodology of our proposed solution is discussed in detail. In Section 4, the model is evaluated, and results discussed. Section 5 and Section 6 discuss limitations and uses and finally with Section 7, the report is concluded.

Face Mask Detection Using OpenCV, Sklearn And Numpy

Introduction to Image Processing:

Object detection is one of the trending topics in the field of **image processing** and **computer vision**. Ranging from small scale personal applications to large scale industrial applications, object detection and recognition is employed in a wide range of industries. Some examples include image retrieval, security and intelligence, OCR, medical imaging and agricultural monitoring. In object detection, an image is read and one or more objects in that image are categorized. The location of those objects is also specified by a boundary called the bounding box. Traditionally, researchers used pattern recognition to predict faces based on prior face models. A breakthrough face detection technology then was developed named as Viola Jones detector that was an optimized technique of using Haar, digital image features used in object recognition. However, it failed because it did not perform well on faces in dark areas and non-frontal face.

Images are simply a collection of colors in red, green and blue format. As a human we see an image with some object or shape in it, but for computer it is just an array with color values range from 0 to 255.

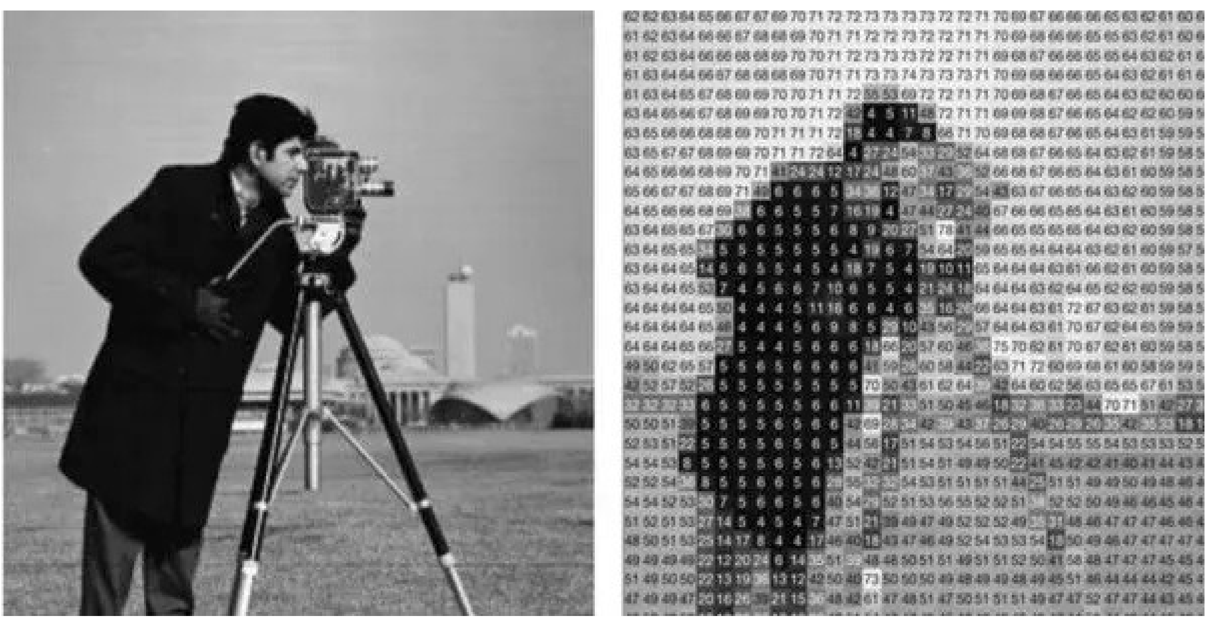

Figure 1: Grayscale photographer image

The way computer sees anything is different from the way human see an image. But that’s the good news for us because if we got an array of the image than it becomes simple for us to implement any algorithm on that array.

3. METHODOLOGY

3.1 Dataset

The dataset which we have used consists of 2000 total images array out of which 1000 are of masked faces and 1000 are of unmasked faces. All the images are actual images extracted from video capturing during training. The proportion of masked to unmasked faces determine that the dataset is balanced.

We need to split our dataset into two parts: training dataset, test dataset.The purpose of splitting data is to avoid overfitting which is paying attention to minor details/noise which is not necessary and only optimizes the training dataset accuracy. We need a model that performs well on a dataset that it has never seen (test data), which is called generalization. The training set is the actual subset of the dataset that we use to train the model. The model observes and learns from this data and then optimizes its parameters. The validation dataset is used to select hyperparameters (learning rate, regularization parameters). When the model is performing well enough on our validation dataset, we can stop learning using a training dataset. The test set is the remaining subset of data used to provide an unbiased evaluation of a final model fit on the training dataset. Data is split as per a split ratio which is highly dependent on the type of model we are building and the dataset itself. If our dataset and model are such that a lot of training is required, then we use a larger chunk of the data just for training which is our case. If the model has a lot of hyperparameters that can be tuned, then we need to take a higher amount of validation dataset. Models with a smaller number of hyperparameters are easy to tune and update, and so we can take a smaller validation dataset. In our approach, we have dedicated 75% of the dataset as the training data and the remaining 25% as the testing data, which makes the split ratio as 0.75:0.25 of train to test set.

3.2 Architecture

An SVM model is basically a representation of different classes in a hyperplane in multidimensional space. The hyperplane will be generated in an iterative manner by SVM so that the error can be minimized. The goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH).

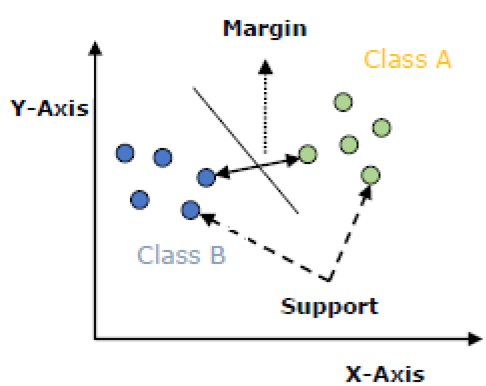

Figure 2: Working of support vector machine

The followings are important concepts in SVM −

- **Support Vectors**− Datapoints that are closest to the hyperplane is called support vectors. Separating line will be defined with the help of these data points.
- **Hyperplane**− As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes.
- **Margin−** It may be defined as the gap between two lines on the closet data points of different classes. It can be calculated as the perpendicular distance from the line to the support vectors. Large margin is considered as a good margin and small margin is considered as a bad margin.

The main goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH) and it can be done in the following two steps −

- First, SVM will generate hyperplanes iteratively that segregates the classes in best way.
- Then, it will choose the hyperplane that separates the classes correctly.

The problem can be solved in two parts: first detecting the presence of several faces in a given image or stream of video and then in the second part, detect the presence or absence of face mask on face. In order to detect the face, we have used the **OpenCV** library. Whenever a new test image is given, it is first converted into Binary array using N**umpy** module and then sent into the face detection algorithm.This algorithm is divided in four stages :

1. Haar Features Selection
2. Integral Images
3. AdaBoost
4. Cascading Classifier

which outputs the number of detected faces. Every face detected comes out with a level of confidence which is then compared with a threshold value to filter out the irrelevant detections. After we have the faces, we need to evaluate the bounding box around it and send it to the second part of the model to check if the face has a mask or not.

The second part of the model is trained by us using a dataset consisting of images with mask and without mask. We have used support vector machine (SVM) along with **Sklearn** to train our model. First part of the training includes storing all labels of the images in a **Numpy** array and the corresponding images are also reshaped (224, 244, 3) for the base model. Image augmentation is a very useful technique because it increases our dataset with images with a whole new perspective. For the image classification, it is now a common practice to use transfer learning which means using a model which has been pre-trained on millions of labels before and it has been tested that this method results in significant increase in accuracy.

4. EVALUATION

4.1 Testing

We have dedicated 25% as the testing data, We tried model for detecting ‘mask’ or ‘no mask’. The exercise was done to find the best fit in our scenario. The evaluation process consists of first looking at the classification report which gives us insight towards precision, recall. The equations of acuracy score is follow:

Accuracy Score = (True positives + True negatives) / Positives + Negatives

4.2 Result

Experimental results show that this model performs well on the test data with 100% and 99% precision and recall, respectively.

Following graph is generated by using matplotlib which showing the mask value and without mask value in frames per session.

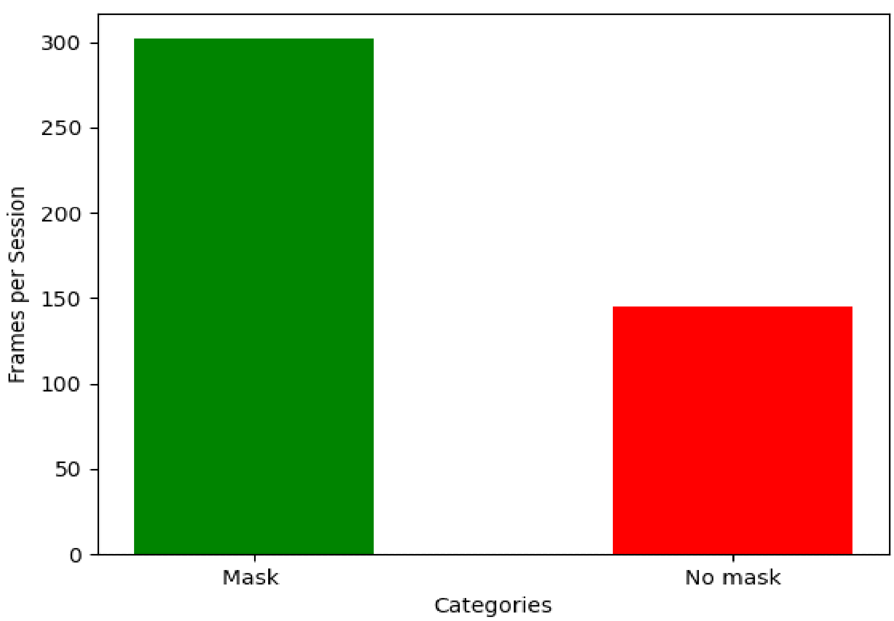

Figure 3: Classification graph

Accuracy Rate: 0.9820359281437125
mask time: 302
without mask time: 145

4.3 Inference

We implemented our model on images containing one and more faces. We also implemented it on videos and live video streams by removing and wearing masks one by one. Some screenshots of the results are shown below:

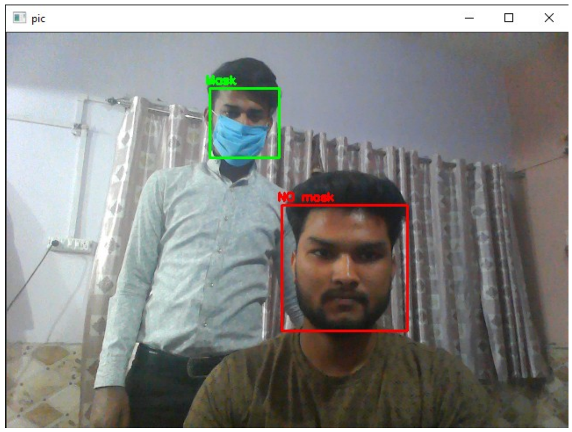

Figure 4: One person with mask and one person without mask.

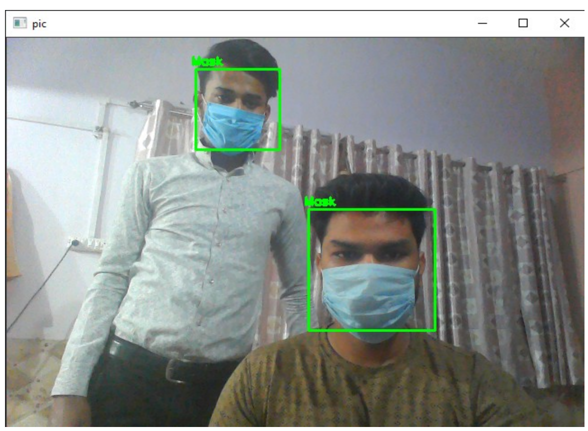

Figure 5: Both persons with masks.

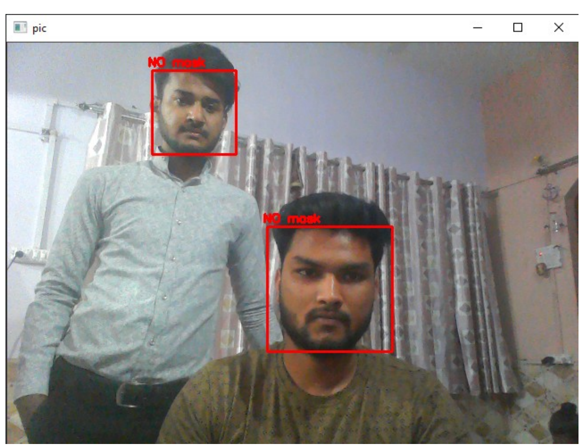

Figure 6: Both persons without masks

5. LIMITATIONS

We observed a few limitations in our current model. The model is not able to correctly classify face images where the face is partially hidden by a person in the front as shown in Figure 7. Also the model is not able to detect faces if the camera height is greater than 10 feet.

Figure 7: Shows faces which get incorrectly identified as the face gets occluded by the person in the front

CONCLUSION

Mask detector using SVM architecture to train, and test the model, we used the dataset that consisted of 1000 masked faces images and 1000 unmasked faces images. These images were taken from video capturing during training. The model was inferred on images and live video streams. To select a base model, we evaluated the metrics like accuracy, precision and recall and selected SVM architecture with the best performance having 100% precision and 99% recall. It is also computationally efficient using SVM which makes it easier to install the model to embedded systems. This face mask detector can be deployed in many areas like shopping malls, airports and other heavy traffic places to monitor the public and to avoid the spread of the disease by checking who is following basic rules and who is not.