EasyChair Preprint

№ 7797

Face Mask Detection Using Machine Learning

Aasim Ismail and Anurag Kumar

April 18, 2022

FACE MASK DETECTION USING MACHINE LEARNING

Aasim Ismail

Bachelor of computer application

GALGOTIAS UNIVERSITY,

UTTAR PRADESH,

INDIA

19SCSE1040168

aasim_ismail.scsebca@galgotiasuniversity.edu.in

Anurag Kumar

Bachelor of computer application

GALGOTIAS UNIVERSITY,

UTTAR PRADESH,

INDIA

19SCSE1040118

anurag_kumar.scsebca@galgotiasuniversity.edu.in

Abstract

The purpose of the project "Face Mask Detection Using Machine Learning" is to create a tool that identifies the image of a human that can calculate the probability that he/she wearing a mask or not. Due to COVID, wearing a face mask is must in order to stay safe. As the country starts going through various stages of reopening, face masks have become an important element of our daily lives to stay safe. Wearing face masks will be required in order to socialize or conduct business. So, this application utilizes a camera to detect if a person is wearing a mask or not.

Introduction

The year 2020 has shown mankind some mindboggling series of events amongst which the COVID19 pandemic is the most life changing event which has startled the world since the year began. Affecting the health and lives of masses, COVID19 has called for strict measures to be followed in order to prevent the spread of disease. From the very basic hygiene standards to the treatments in the hospitals, people are doing all they can for their own and the society's safety; face masks are one of the personal protective equipment. People wear face masks once they step out of their homes and authorities strictly ensure that people are wearing face masks while they are in groups and public places

Face mask detection is a system that detects whether a person is wearing a mask or not. It is same as an object detection system in which a system detects a particular class of objects. Through building this system we are trying to help ensure people's safety at public places. This system can be implemented in many areas such as supermarkets and shopping malls, schools, colleges, stations and so on. To accomplish this task, we'll be fine-tuning the Mobile Net V2 architecture, a highly efficient architecture that can be applied to embedded devices with limited computational capacity

OBJECTIVE

The main objective of this project is to take care of human health and control the spread of deadly viruses. The sudden attack of COVID-19 has taught us many things in which it includes wearing a mask and sanitizing our hands each now and then. In order to avoid its spread mask is the most essential object. So, to prevent people from skipping wearing masks this project has come to the light. In order to avoid manual detection and for better accuracy this project has been introduced. This enables the people to remind them of wearing their masks and also allows only when they wear their masks. To summarise,

• The objective of the project is to protect people from contagious diseases.

• Reduce the time and cost taken for manual detection. FACE MASK DETECTION

• To avoid the spread of disease in public places like airports, bus stops, etc

SYSTEM ANALYSIS

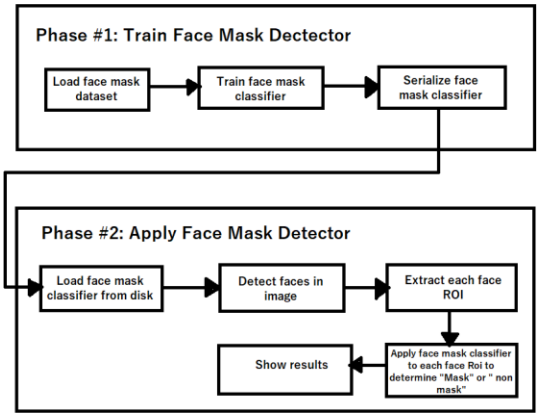
Introduction

The trend of wearing face masks in public is rising due to the COVID- 19 corona virus epidemic all over the world. Before Covid-19, People used to wear masks to protect their health from air pollution. While other people are self-conscious about their looks, they hide their emotions in the public to hide their faces. More than five million cases were infected by COVID- 19 in less than 6 months across 188 countries. The virus spreads through close contact and in crowded and overcrowded areas. We can tackle and predict new diseases by the help of new Technologies such as artificial intelligence, Iot, Big data, and Machine learning. People are forced by laws to wear face masks in public in many countries. These rules and laws were developed as an action to the exponential growth in cases and deaths in many areas. However, the process of monitoring large groups of people is becoming more difficult in public areas. So we will create a automation process for detecting the faces. Here we introduce a facemask detection model that is based on computer vision and deep learning. The proposed model can be integrated with Surveillance Cameras to impede the COVID-19 transmission by allowing the detection of people who are wearing masks not wearing face masks. The model is integration between deep learning and classical machine learning techniques with Open cv, Tensor flow and Keras.

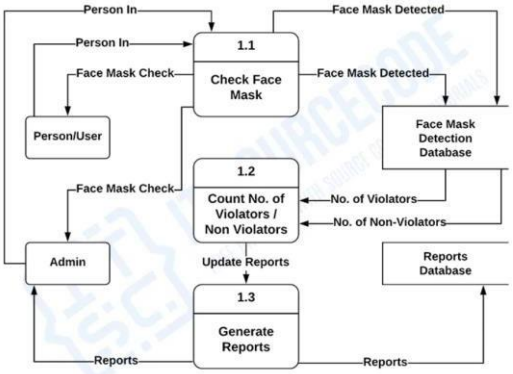
Project overview

To access the application authentication is provided. The user have username and password. User is two types, one admin and student. After login, it enters into different type of module, at the time of login it is decide. If the user is admin, it logon into admin module, otherwise it logon into the student module. Admin can have the control over the student module and after that admin can check whether the person is wearing the mask or not.

Architecture Diagram



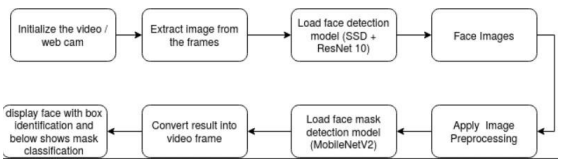
Data Flow Diagram



DFD level 2 lets you know the ideas on where does the data inputs goes and inputs comes within the Face Mask Detection System. Considering the dataflow levels mentioned above, you can determine well the importance of breaking the processes into more specific manner.

The presented level not only shows you the detailed processes of system, but also gives you precise destination of the data that flows in the system.

Flow Chart



Methodology

Data Pre-Processing

Data Pre-processing is the crucial steps to be carried for building the model. It is important to

clean the dataset and make it is suitable for model building. In the case of images, we are converting the images into an array. First, the dataset has to be collected from the directory with categories (with mask and without mask). Keras is a neural network library. It provides a high level API for building and training the models at ease. The conversion of the image into an array is done using img_to_array. Img_to_array comes from the keras. Pre-processing .image. After converting the image data into arrays, it is in the form of numerical data. Data Pre-processing is a method where all the data is generalized such that no redundancy will be present. So now, the rest of the data should be in the same format to avoid redundancy. The rest of the categorical values has to be converted into 0's and 1's. This conversion can be done by using Label Binaries. These labels will be now converted into arrays using NumPy. Train-Test split will be carried out with 80%ofthe images to the training set and 20% of them to the testing set. 2.2 Training of the Dataset Training the dataset is the important step for making the model. Here generally we use convolutional networks, but our idea is to neglect CNN we use for image processing and introduce Mobile Net essentially what we do is usually after the image is processed as an array, we will send that into Mobile Net and then do the maximum pooling. After the maxpooling, we will connect it and flatten it, which later results in the required output. The added advantage of using Mobile Net is thatitis faster compared to CNN and uses fewer parameters. Initially learning rate (INIT_LR) is assigned less because the lesser the learning rate higher the accuracy (INIT_LR (1e-4)) 20 Epochs Batch size (BS) =32 By using Mobile Net we are generating two types of models i) Mobile Net model (which is going to be passed into the normal model that is first developed). To explain clearly, the first generated model is the output that we are passing to the normal model we are going to develop. They can commonly be called as the head model and base model. Image Data generator is also used which is used to generate many images to single images by adding properties like flipping, rotating the image, etc. These properties are beneficial for generating more images in the dataset.

Modelling

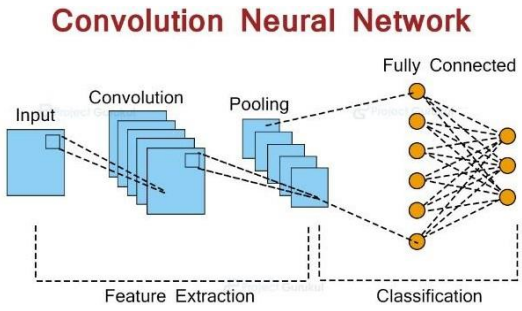
As Mobile Net V2 generated a Base model, for further development we are using weights= ImageNet which has pre-trained models for images, so those weights will be initialized for a better model. Input Tensor is the shape of the image that is going through which includes the size of the image and the number of channels. Input Tensor =Input (Shape (224, 224, 3)) The number of channels in the images we are using is RED, GREEN, and BLUE (RGB). Once the base model is built, we have to build the fully connected layer which is the head model. Already as mentioned, the Base model is passed through for Pooling, flattening, etc. to build the head model. We also used the go to activation function "Relu" for nonlinear use cases and generally used for images. We use dropout to avoid overfitting of models. And our output is two layers which are with mask and without the mask and our activation function used here is SoftMax. Our head model will be the output and the base model is the input which are the two parameters for the main model. The important measure taken here is to freeze the layers of the base model because they are just a replacement for CNN. For the loss calculation, we are using binary_crossentropy and the optimizer we used here is Adam optimizer which is similar to Relu and always a go-to optimizer for image Processing. For further training, we are using Image Data generator which is used to generate more images. And we also used Test X and Test Y for validation and also E Pochs. And finally, for plotting out accuracy and metrics we use matplotlib and we are also saving the image from matplotlib. So essentially, we need 2 files to be saved one is the model file and the other is the plot.png

Project Design

Convolution neural network

a convolutional neural network or CNN is a type of neural network that is commonly used for image processing tasks. It is specifically designed to process images. CNN takes some input arrays, then applies some filters, and gives

an output array. The filters help to extract features from the images.



We'll do this project in three parts.

1. Data Collection
2. Model training
3. Model Testing

Feature-based face mask detection- Every object has its unique features and our face has it too when we wear mask. By extracting those unique features from an image, we can detect different objects and their location. Our face has two eyes, two eyebrows above each eye, a nose under the eyes, and person will be wearing a mask etc. using these features we can determine whether person is wearing it or not.

Machine learning-based face detection: - In this method, some Face images are required to train a machine learning model to detect faces in an image. This method requires lots of data and pre-processing to build a face detector but it gives better results than a feature-based face detector.

In this face mask detection project, we'll use a deep learning approach to detect faces. We'll use the media Pipe framework to detect a face.

Media Pipe is a machine learning solution framework that has several pre-trained models inbuilt. It is a very fast and lightweight multi-platform framework. Media Pipe is developed by Google's brain team

Steps:

- Import necessary packages.
- Create the Face detector function.
- Data collection.

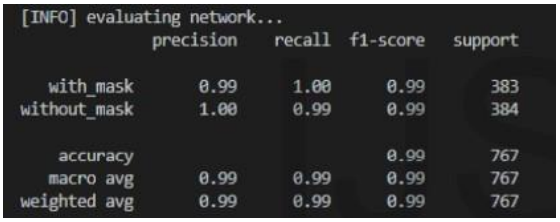
Model Usage in Real-Time

Camera

We have developed a model for mask detection and now we need to have a model for face detection. So, the deep learning model will detect the mask and we use Open Cv for performing the camera operation. The two types of parameters Face net and Mask net. Face net contains the files that detect the face and Mask net has the datasets with mask and without the mask. Deep Neural Networks (DNN) has been developed in is the module developed by Cv2 which contains many methods in them. In order to load the camera, we use src=0, which is the source i.e., the main camera which is inbuilt into the system. After the face is detected, a frame will be loaded in the video with a width of 400. Then we detect the face and mask together as a single function which later returns the location and predictions. Location is x and y coordinates of the rectangle surrounding the face and predictions are nothing but the accuracy of the model. Beep Alert: The additional feature is that the person detected without the mask is alerted with a beep sound.

Result:

Real time testing :



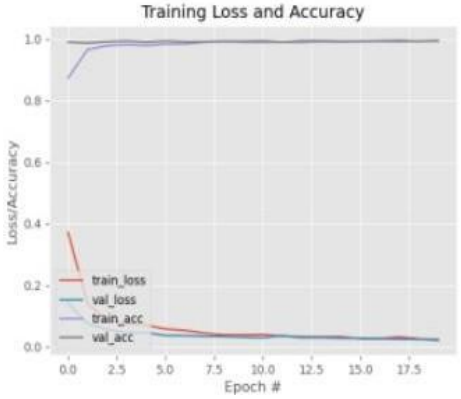
We have successfully trained our model and tested it on a real time face using the laptop's camera. Our face mask detector correctly labelled the person's face as either 'Mask' or 'No Mask'. As you can see in this image that the face is labelled as 'Mask' when the person is wearing a mask and labelled as 'No Mask' when the person is not wearing a mask.

In this as we can see the person is wearing mask so the output or result will be that its allow the person as person is wearing mask

Now in the second pic we can see that the person is not wearing mask so it will not allow the person so it will be the output of "no mask detection"

Visualization

After training the model, the accuracy and the training loss with the epoch are plotted and saved in the directory mentioned. The model is saved in two forms one as the model file other as the .png file. The below are the results of the training data



Conclusion

The goal of the project is to ensure that the working of a few industries and professions progress well without any economic loss. Since man-power is mandatory and safety should be ensured in these pandemic COVID-19 conditions, it is important and would be efficient to install such models at workplaces. Apart from this, public transportation like airports can also be made available with these systems for safety and sanity. The experimental analysis shows that the proposed method can be successfully exploited for face mask violation detection. It is a real time software application which can be deployed in smart cc tv surveillance, public areas like airports, malls, Etc where mask is necessary. Only the software it can be extensible to work along with other IOT devices

Reference

[1] Ahmed I., Ahmad M., Rodrigues J.J., Jeon G., Din S. A deep learning based social distance monitoring framework for COVID-19. Sustainable Cities and Society. 2020 [PMC free article] [PubMed] [Google Scholar].

[2] Kumar P., Hama S., Omidvarborna H., Sharma A., Sahani J., Abhijit K.V.…Tiwari A. Temporary reduction in fine particulate matter due to 'anthropogenic emissions switch-off' during COVID-19 lockdown in Indian cities. Sustainable Cities and Society. 2020;62 [PMC free article] [PubMed] [Google Scholar].

[3] Wang Z., Wang G., Huang B., Xiong Z., Hong Q., Wu H.…Chen H. 2020. Masked face recognition dataset and application. arXiv preprint arXiv:2003.09093. [Google Scholar].

[4] Huang C., Ai H., Li Y., Lao S. High-performance rotation invariant multi view face detection . IEEE Transactions on Pattern Analysis and Machine Intelligence. 2007; 29(4):671–686. [PubMed] [Google Scholar