Face Mask Detection

Prateek Choudhary, Syed Osama and Lakshya Jaiswal

July 5, 2021

Face Mask Detection Application

Dr. M. Kumaresan

Assistant Professor

Department of School of Computing Science and Engineering

Galgotias university (GU), Greater Noida, Uttar Pradesh

Syed Osama Islam

(1713101218)
Department of School of Computing Science and Engineering

Galgotias university (GU), Greater Noida, Uttar Pradesh
syedosama0055@gmail.com

Lakshya Jaiswal
(1713101178)
Department of School of Computing Science and Engineering

Galgotias university (GU), Greater Noida, Uttar Pradesh

lakshyajaiswal99@gmail.com

Prateek Choudhary

(1713101416)
Department of School of Computing Science and Engineering

Galgotias university (GU), Greater Noida, Uttar Pradesh

prateek.choudhary01_2017@galgotiasuniversity.edu.in

Abstract - Face recognition is of the most interesting modal of biometric. Due to its low interfering nature and to the constant decrease in image acquisition cost, it's particularly suitable for a wide number of real time applications. At this paper we are proposing a very fast image pre-processing by the introduction of a linearly shaded elliptical mask at the canter over our faces.

COVID-19 pandemic caused by the virus name coronavirus is continuously spreading until now to the various parts of the world. The impact of COVID19 have seen almost all sectors of development. The healthcare system is going through a crisis. Several precautions are measured had been taken to reduce the spread of this disease where wearing a mask is one of them. We propose a system that restrict the growth of COVID-19 by its feature of finding out people who are not wearing any facial mask in a smart city network where all the public places are monitored with CCTV cameras. When a person without a mask is detected, Automatic the corresponding authority is informed through the city network.

I. INTRODUCTION

A new strain which has not previously been identified in humans is novel coronavirus (covid). Coronaviruses (Covid) are a wide group of viruses which cause illness that range from colds to deadly infections like Middle East Respiratory Syndrome (MERS) and severe acute respiratory syndrome (SARS). The first infected tolerant coronavirus has been found in December 2019. After that period, COVID-19 has become a pandemic all over the world. All over the world people are facing challenging situations due to this pandemic. Every day a large scale of people is being infected by this virus and are being dead. At the time of writing this paper, almost 2.94Cr. infected cases have been confirmed where 3.7L are death. This number is increasing day by day. Cough, Fever, tiredness, diarrhea, loss of taste and smell are the major symptoms of coronavirus which is declared by the World Health Organization. Among them cleaning hands, social distancing, wearing a mask, refraining from touching eyes, nose, and mouth are the main, where wearing a mask is the simplest one. COVID-19 is a disease that spread from contacting with each other which can be controlled by ensuring proper use of a facial mask. The spread of COVID-19 can be controlled if all people strictly only this maintains social distancing and use a facial mask. Very sadly, people are not ready to obey these rules properly which is speeding the spread of this virus. Detecting the people not obeying these rules and informing the corresponding authorities can be a solution in reducing the spread of coronavirus.

A mask detection may be a technique to seek out whether someone is wearing a mask or not. It is similar to detect any particular view from a scene. Many systems have been introduced for object detection. Deep learning techniques are mostly used at medical applications. Recently, deep learning architectures have shown a remarkable role in object detection. These architectures can be inappropriate on detecting the mask on a face. Moreover, a smart city means an urban area that consists of many IoT sensors to collect data. Those collected

data is then used to perform various operations across the city. Currently, the growth of COVID-19 can be only reduced by detecting the facial mask in a smart city area.

At this paper we aim at designing a system to find out whether a person is using a mask or not and informing the corresponding authority in a smart city network. CCTV cameras are used to capture real-time video footage of different spots of that area. From that video footage, facial images are extracted and hence these images are used to identify the mask on the face. The algorithm of Convolutional Neural Network is used for feature extraction from the images then these features are learned by multiple hidden layers. Whenever the architecture identifies people without face mask this information is transferred through the network to the corresponding authority to take necessary actions. This system has promising output on data collected from different sources. It can also represent a system that can ensure proper law enforcement on people who are not following basic health guidelines in this pandemic situation.

II. LITERATURE REVIEW

Face Mask Detection can quickly identify and track everyone in a crowd as they move about and simultaneously help recognize guests and personnel who are not wearing masks and restrict access. In order to train a custom face mask detector, we need to break our project into two distinct phases, each with its own respective sub-steps: -

1. Training: Here we will focus on loading our face mask detection dataset from disk, training a model (using Keras/TensorFlow) on this dataset, and then serializing the face mask detector to disk
2. Deployment: Once the face mask detector is trained, we can then move on to loading the mask detector, performing face detection, and then classifying each face as with__mask or without__mask.

.

III. METHODOLOGY

Here, we had proposed an automatic smart framework for screening persons who are not wearing a face mask in this paper. In the smart city, all public places are monitored by CCTV cameras. The cameras are wont to capture images from public places; then these images are feed into a system that identifies if a person without mask appears within the image. If a person without a mask is detected then this information is shipped to the right authority to require necessary actions. The block diagram of the developed framework is depicted in the figure below.

Image Preprocessing: -

The images captured by the CCTV cameras required preprocessing before getting to subsequent step. In the preprocessing step, the image is transformed into a grayscale image because the RGB color image contains such a lot redundant information that's not necessary for mask detection. RGB color image stored 24 bits for each pixel of the image. On the other hand, the grayscale image stored 8 bits for each pixel and it contained sufficient information for classification. Then, we reshaped the images into (64×64) shape to maintain uniformity of the input images to the architecture. Then, the pictures are normalized and after normalization, the worth of a pixel resides within the range from 0 to 1. Normalization helped the training algorithm to find out faster and captured necessary features from the pictures.

Deep Learning Architecture: -

The deep learning architecture learns various important nonlinear features from the given samples. Then, this learned architecture is employed to predict previously unseen samples. For the use of our deep learning architecture, we must collect images from different sources. The architecture of the training technique highly depends on CNN.

The proposed model functions on generation and prediction of bounding boxes, which are categorized in two class, ground-truth box and detection box. The ground-truth is formed of annotations, which needs to be defined for each image, and specifies the ROI which is needed to be detect. The detection boxes are basically the outcome or predictions made by the algorithm. The confidence score for each detection depends on the extent of intersection of the bounding boxe. More the confidence score, better is the prediction accuracy. With each stage of detection, mobile net generates several boxes, which are further filtered through Non-Max Suppression algorithm, which filters out best detection.

Approach: -

1. Train Deep learning model (MobileNetV2)

2. Use mask detector over the images and live video streams.

Data at Source: -

The majority of the pictures were augmented by OpenCV. All of the set of images are already labeled as “mask” and “no mask”. All the images that were presented were all of unique sizes and resolutions, probably extracted from different sources or from cameras of different resolutions. optical Character Recognition (Often referred as OCR) is another major step of the algorithm which is responsible for character recognition after localization. The model invokes the usage of a python library named “EasyOCR”, which is responsible for recognition of characters from localized if the culprit is in moving vehicle. EasyOCR is highly efficient as it works efficiently on noisy images and also can be integrated with GPU for enhancing its performance.

Data preprocessing: -

Following processing steps as mentioned below was applied to all the raw input images to convert them into clean version, which could be fed to a neural network machine learning model:

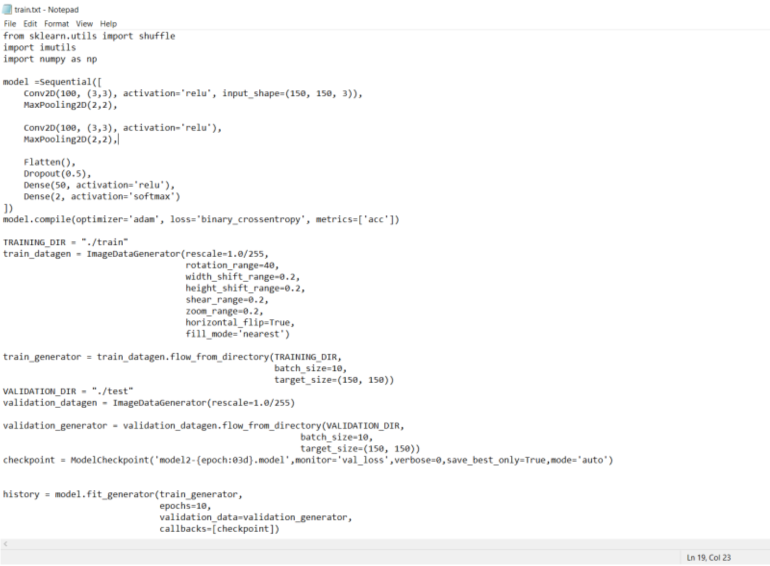
1. Resizing the input image (256 x 256)

2. ON applying all the color filtering over the channels

3. Normalizing and scanning of the images using the standard mean of Py Torch build in weights

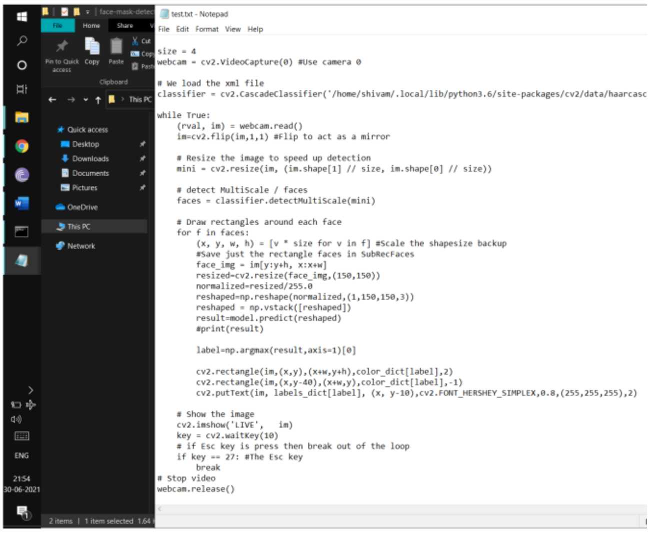
4. And cropping the image at the pixel value of 224x224x3

5.At Last than Converting them into tensors (Similar to NumPy array)



```
from sklearn.utils import shuffle
import imutils
import numpy as np

model =Sequential([
    Conv2D(100, (3,3), activation='relu', input_shape=(150, 150, 3)),
    MaxPooling2D(2,2),

    Conv2D(100, (3,3), activation='relu'),
    MaxPooling2D(2,2),

    Flatten(),
    Dropout(0.5),
    Dense(50, activation='relu'),
    Dense(2, activation='softmax')
])
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['acc'])

TRAINING_DIR = "./train"
train_datagen = ImageDataGenerator(rescale=1.0/255,
                                   rotation_range=40,
                                   width_shift_range=0.2,
                                   height_shift_range=0.2,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   horizontal_flip=True,
                                   fill_mode='nearest')

train_generator = train_datagen.flow_from_directory(TRAINING_DIR,
                                                    batch_size=10,
                                                    target_size=(150, 150))
VALIDATION_DIR = "./test"
validation_datagen = ImageDataGenerator(rescale=1.0/255)

validation_generator = validation_datagen.flow_from_directory(VALIDATION_DIR,
                                                              batch_size=10,
                                                              target_size=(150, 150))
checkpoint = ModelCheckpoint('model2-{epoch:03d}.model',monitor='val_loss',verbose=0,save_best_only=True,mode='auto')


history = model.fit_generator(train_generator,
                              epochs=10,
                              validation_data=validation_generator,
                              callbacks=[checkpoint])
```

Fig. The code is for ml training (train.py)



```
size = 4
webcam = cv2.VideoCapture(0) #Use camera 0

# We load the xml file
classifier = cv2.CascadeClassifier('/home/shivam/.local/lib/python3.6/site-packages/cv2/data/haarcasc

while True:
    (rval, im) = webcam.read()
    im=cv2.flip(im,1,1) #Flip to act as a mirror

    # Resize the image to speed up detection
    mini = cv2.resize(im, (im.shape[1] // size, im.shape[0] // size))

    # detect MultiScale / faces
    faces = classifier.detectMultiScale(mini)

    # Draw rectangles around each face
    for f in faces:
        (x, y, w, h) = [v * size for v in f] #Scale the shapesize backup
        #Save just the rectangle faces in SubRecFaces
        face_img = im[y:y+h, x:x+w]
        resized=cv2.resize(face_img,(150,150))
        normalized=resized/255.0
        reshaped=np.reshape(normalized,(1,150,150,3))
        reshaped = np.vstack([reshaped])
        result=model.predict(reshaped)
        #print(result)

        label=np.argmax(result,axis=1)[0]

        cv2.rectangle(im,(x,y),(x+w,y+h),color_dict[label],2)
        cv2.rectangle(im,(x,y-40),(x+w,y),color_dict[label],-1)
        cv2.putText(im, labels_dict[label], (x, y-10),cv2.FONT_HERSHEY_SIMPLEX,0.8,(255,255,255),2)

    # Show the image
    cv2.imshow('LIVE',   im)
    key = cv2.waitKey(10)
    # if Esc key is press then break out of the loop
    if key == 27: #The Esc key
        break
# Stop video
webcam.release()
```

Fig. The code for the ml testing (test.py)

IV. OUTCOME

As the technologies are blooming with new trends the use so we have our face mask detector which can contribute a vital role in today s crisis. The architecture consists of MobileNet because the backbone it often used for top and low computation scenarios. In order to extract more robust features, we utilize transfer learning to adopt weights from an identical task face detection, which is trained on a really large dataset.

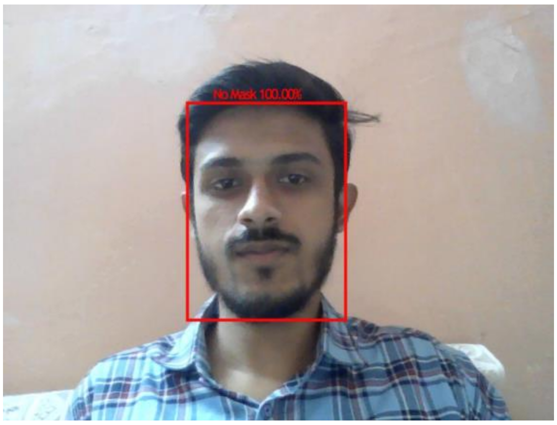


Fig. Face without mask

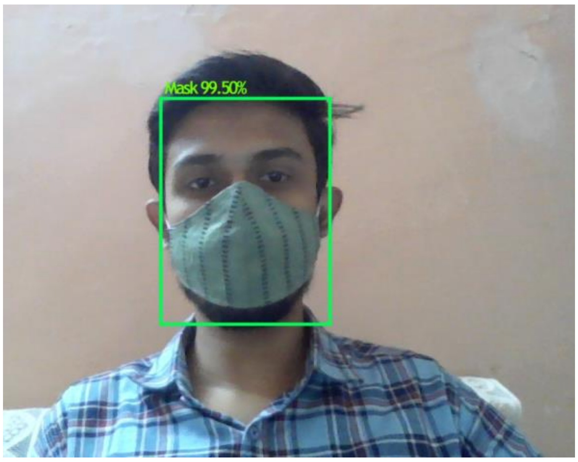


Fig. Face with mask

We used Machine Learning to detect whether people were wearing face masks or not. These models are always tested with images and real time video streams. The accuracy of the model is achieved and, the optimization of the model could also be endless process which we are building a highly accurate solution by tuning the hyper parameters. For this current specific model could be used as a use case for analytics. Afterword, the proposed methods achieve state of the art results on a public face mask dataset. AT The development of face mask detection, we can detect if the person is wearing a face mask and allow their entry at any of the public areas would be a great use for the government.

V. FUTURE SCOPE

The proposed system is works quite well but still there is some areas or part which can be improved and enhance the accuracy of system more. This system is used in many purposes like face detection with ROI detection of two faces with mask and without mask. In this the camera used by this system to detect the real time object is of very average quality and unable to detect the high-speed moving people in car or bike due to the long shutter time. So, to improve the robustness and speed of the system the high-speed cameras is used in place of the low-speed camera. The model does mistakes or problem in detecting the similar dimensions of character in the plate. To improve the number plate recognition system the model is trained with more set of accurate data. The future advancement in the technology with the time makes more powerful frameworks to make this system to provide more accuracy.

REFERENCE

[1] P. A. Rota, M. S. Oberste, S. S. Monroe, W. A. Nix, R. Campagnola, J. P. Icenogle, S. Penaranda, B. Bonampak. Maher, M.-h. Chenet "Of a novel coronavirus associated with severe acute respiratory syndrome, "science, vol. 300, no. 5624, pp. 1394–1399, 2003.

[2] Z. A. Memis, A. I. Zulma, R. F. Al-Hakeem, A. A. Arabieh, and G. M. Stephens, "Family cluster of middle east respiratory syndrome coronavirus infections, "New England Journal of Medicine, vol. 368, no. 26, pp.2487–2494, 2013.

[3] Kang, H.-J. (2019). Real-Time Object Detection on 640x480 Image With VGG16+SSD. 2019 International Conference on Field-Programmable Technology (ICFPT). doi:10.1109/icfpt47387.2019.00082

[4] A M. Al-bakry, S O. Al-mamory, H HMushatet. In Annual Conference on New Trends in Information & Communications Technology Applications-(NTICT'2017), Irak, 7-9 Mars,2017.