**PROJECT REPORT ON**

**Face Mask detection using Python**

***Prepared by***

***Subarna Chatterjee(Roll No:-18700120039)***

***Shreya Swar(Roll No:-18700120044)***

***Upayan Ghosh(Roll No:-18700320058)***

***Souvik Banerjee(Roll No:-18700220074)***

***Academic Year: 2021-2022***

SUBMITTED IN PARTIAL FULLFILLMENT OF THE REQUIREMENT FOR THE DEGREE BACHELOR OF TECHOLOGY (COMPUTER SCIENCE & ENGINEERING**)**

Logo, company name

Description automatically generated

**TECHNO INTERNATIONAL NEW TOWN**

**Department of Computer Science & Engineering**

**Block-DG, Action Area 1, New Town, Kolkata -700156, West Bengal, India**

**CERTIFICATE**

This is to certify that the project report entitled “ **Face Mask detection using Python** ” has been prepared by the following students of the Department of Computer Science & Engineering in partial fulfillment for the degree of Bachelor of Technology(B.Tech.) in Computer Science & Engineering which is affiliated to Maulana Abul Kalam Azad University of Technology, West Bengal (Formerly known as West Bengal University of Technology) in the academic year 2021-2022

It is to be understood that by this approval, the undersigned does not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn thereof, but approves the report only for the purpose for which it has been submitted.

|  |  |  |
| --- | --- | --- |
| **Sl No.** | **Name** | **Roll Number** |
| 1. | Subarna Chatterjee | 18700120039 |
| 2. | Shreya Swar | 18700120044 |
| 3. | Upayan Ghosh | 18700320058 |
| 4. | Souvik Banerjee | 18700220074 |

………………………………………….

Prof. Bikash Sadhukhan,

HOD, Computer Science & Engineering,

Techno International New Town

**ABSTRACT**

COVID-19 pandemic has rapidly increased health crises globally and is affecting our day-to-day lifestyle. A motive for survival recommendations is to wear a safe facemask, stay protected against the transmission of coronavirus. By wearing a facemask, the most effective preventive care must be taken against COVID-19. 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. The system runs in real-time and detects if an individual face has facemask if not then notify the individual personally through text message. The mask is extracted from real-time faces in public and is fed as an input into convolutional neural network (CNN).

**CONTENTS**

**Page No.**

1. **INTRODUCTION 1**
2. **PROBLEM DEFINITION 1**
3. **ARCHITECTURE & DETAILED WORKING 2 - 11**
4. **CONCLUSION 12**
5. **FUTURE SCOPE OF PROJECT 12**
6. **BIBLIOGRAPHY 12 - 13**

**LIST OF FIGURES**

**Figure Caption**……………………………………………………………**Page No**.

1. Steps / Process………………………………………………………………...2
2. Example of dataset……………………………………………………………3
3. Transfer Learning……………………………………………………………..4
4. With & Without mask………………………………………………………...5
5. Graph of training loss & Accuracy…………………………………………...9
6. **Introduction**

Since the declaration of the COVID-19 virus as a pandemic by WHO in the study by the reknown scientists from all over the world , the day today life of us is same as before . WHO stated that efforts have been made by various parties to reduce the spread of the virus. For now, there is various vaccine available. But due to spreading like a wild fire through out the glob , the covid 19 virus started mutation and get more stronger and dangerous than before . The countries like India , China , Indonesia and other countries are so densely populated that because of mutation covid 19 evolves in more fetal virus . The recent news coming from china millions of people is already effected by the new variant and could also spread more . So for instance, physical distancing and wearing a face mask in the public place to impede COVID-19 transmission . Furthermore, since the New Normal has been implemented, the people are forced by law to wear a face mask in the public place and wherever they interact with other people .

However, some difficulties are faced by the authorities in the process of monitoring a large population that has a different habit . The authorities need a solution to be able to validly control the implementation of the law, which begins with the availability of the data quickly and accurately. One of the solutions is to use a live face mask detection software to be implemented in the public place to monitor the people.

This solution is basically created with the help of machine learning algorithm and to make the accuracy even more higher than the normal we use transfer learning provided by the MobileNetV2 where the model is already pretrained with mllions of image data and for the data set we use a large amount of data set of people with mask and without mask and train the local model accordingly.

The built model in this study can be implemented on the surveillance cameras to impede the transmission of COVID-19 transmission by detecting the people who are not wearing a face mask. Each camera point is supplied with location data, so the data can be used to determine which locations require more attention from the authorities.

1. **Problem Definition**

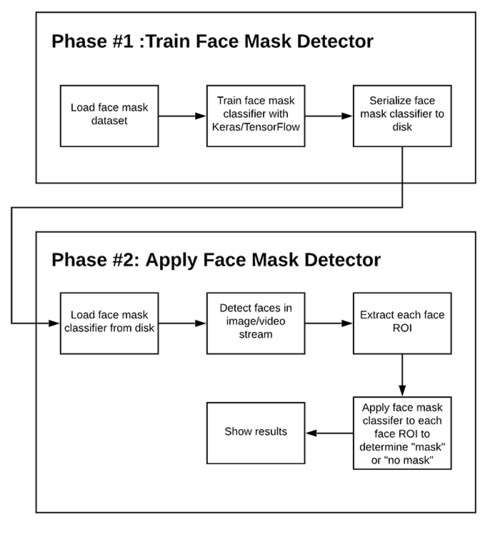
The problem statement is face mask detection using python. To accomplish the solution we use machine learning algorithm to build a model that can detect whether the person in front is wearing mask or not.

**1.**

1. **Architecture**

The face mask recognition in this study is developed with a machine learning algorithm through the image classification method: MobileNetv2. MobileNetV2 is a method based on Convolutional Neural Network (CNN) that developed by Google with improved performance and enhancement to be more efficient .

The dataset was taken from the Kaggle dataset and the Real-World Masked Face dataset (RMFD); used for the training, validation, and testing phase so the model can be implemented to the dataset. The model is made by following some steps which are 1) data collecting, 2) pre-processing, 3) split the data, 4) building the model, 5) testing the model, and finally 5) implement the model on real time . The complete steps as shown in Figure.



1) Steps

**2.**

The data set contain two sub directory With\_Mask and Without\_Mask containing almost

2000 images each . In the model building and training we use the transfer learning to

make the efficiency more better and the accuracy go high. Transfer learning is a machine-

learning method where the application of knowledge obtained from a model used in one

task can be reused as a foundation point for another task.



2) Example of dataset

Machine-learning algorithms use historical data as their input to make predictions and

produce new output values. They are typically designed to conduct isolated tasks. A

source task is a task from which knowledge is transferred to a target task. A target task is

where improved learning occurs because of the transfer of knowledge from a source task.

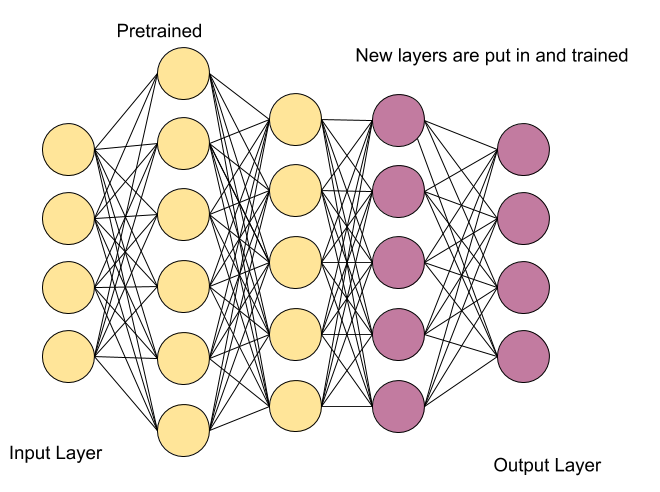
During transfer learning, the knowledge leveraged and rapid progress from a source task

is used to improve the learning and development to a new target task. The application of

knowledge is using the source task’s attributes and characteristics, which will be applied

and mapped onto the target task.

**3.**

 3) Transfer Learning

Trainning :- After reshaping and resizing the batches of image in proper size , split the

data in 3 part training, testing and validation . During the training of the data we build a

model using mobilenetV2 .

Deployment :- When the model accuracy is proper and the validation accuracy also

higher , finalize the model and deploy it for real time detection.

1. **Detailed Working**
   1. **Data Collecting**

The development of the Face Mask Recognition model begins with collecting the data.

The dataset train data on people who use masks and who do not. The model will

differentiate between people wearing masks and not.

For building the model, this study uses 2564 data with mask and 1.930 data without a

mask.

**4.**

1. b)

4) With mask (a) & Without mask (b)

The next step is to label the data. The data which has been collected labeled into two

groups; with and without a mask. After the data has been labeled, it is grouped into

those two categories.

Datadirectory = "Dataset/"#tranning data

Classes = ["With\_Mask","Without\_Mask"]#list of classes

for category in Classes:

    path = os.path.join(Datadirectory,category)

    for img in os.listdir(path):

        img\_array = cv2.imread(os.path.join(path,img))

        plt.imshow(cv2.cvtColor(img\_array, cv2.COLOR\_BGR2RGB))

        plt.show()

        break

    break

* 1. **Pre-processing**

The pre-processing phase is a phase before the training and testing of the data. There are

four steps in the pre-processing which are resizing image size, converting the image to the

array, pre-processing input using MobileNetV2, and the last is performing hot encoding

on labels.

**5.**

The resizing image is a critical pre-processing step in computer vision due to the

effectiveness of training models. The smaller size of the image, the better the model will

run. In this study, the resizing an image is making the image into 224 x 224 pixels.

img\_size = 224

new\_array = cv2.resize(img\_array, (img\_size,img\_size))

plt.imshow(cv2.cvtColor(new\_array, cv2.COLOR\_BGR2RGB))

plt.show()

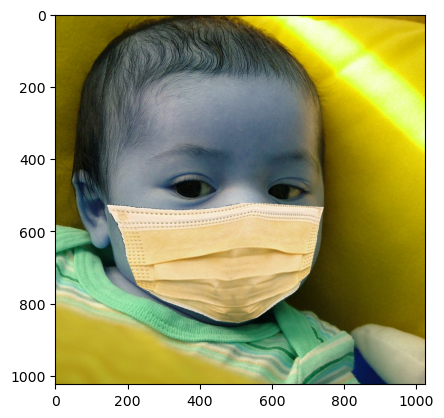
The next step is to process all the images in the dataset into an array. The image is

converted into the array for calling them by the loop function. After that, the image will be

used to pre-process input using MobileNetV2.

img\_array = cv2.imread("Dataset/With\_Mask/00000\_Mask.jpg")

plt.imshow(img\_array)



**6.**

And the last step in this phase is performing hot encoding on labels because many

machine learning algorithms cannot operate on data labeling directly. They require all

input variables and output variables to be numeric, including this algorithm. The labeled

data will be transformed into a numerical label, so the algorithm can understand and

process the data

import pickle

pickle\_out = open("x.pickle","wb")

pickle.dump(x, pickle\_out)

pickle\_out.close()

pickle\_out = open("y.pickle","wb")

pickle.dump(y, pickle\_out)

pickle\_out.close()

**4.3 Split the Data**

After the pre-processing phase, the data is split into two batches, which are training data namely 75 percent, and the rest is testing data. Each batch is containing both of with-mask and without-mask images.

training\_data = []

def create\_training\_data():

    for category in Classes:

        path = os.path.join(Datadirectory, category)

        class\_num = Classes.index(category)

        for img in os.listdir(path):

            try:

                img\_array = cv2.imread(os.path.join(path,img))

                new\_array = cv2.resize(img\_array,(img\_size,img\_size))

                training\_data.append([new\_array,class\_num])

            except Exception as e:

                pass

create\_training\_data()

**7.**

**4.4 Building the Model**

The next phase is building the model. There are six steps in building the model which are

constructing the training image generator for augmentation, the base model with

MobileNetV2, adding model parameters, compiling the model, training the model, and the

last is saving the model for the future prediction process.

model = tf.keras.applications.mobilenet.MobileNet()

model.summary()

base\_input = model.layers[0].input

base\_output = model.layers[-4].output

Flat\_layer = layers.Flatten()(base\_output)

final\_output = layers.Dense(1)(Flat\_layer)

final\_output = layers.Activation('sigmoid')(final\_output)

new\_model = keras.Model(inputs =base\_input ,outputs = final\_output)

new\_model.summary()

* 1. **Testing the Model**

To make sure the model can predict well, there are steps in testing the model. The first

step is making predictions on the testing set. The result for 10 iterations in checking the

loss and accuracy when training the model is shown in Table of 2.

new\_model.compile(loss="binary\_crossentropy",optimizer = "adam",metrics=["accuracy"])

new\_model.fit(x,y, epochs= 2,validation\_split = 0.1)

Epoch 1/2 81/81 [==============================] - 144s 2s/step - loss:

0.0337 - accuracy: 0.9915 - val\_loss: 0.0014 - val\_accuracy: 1.0000

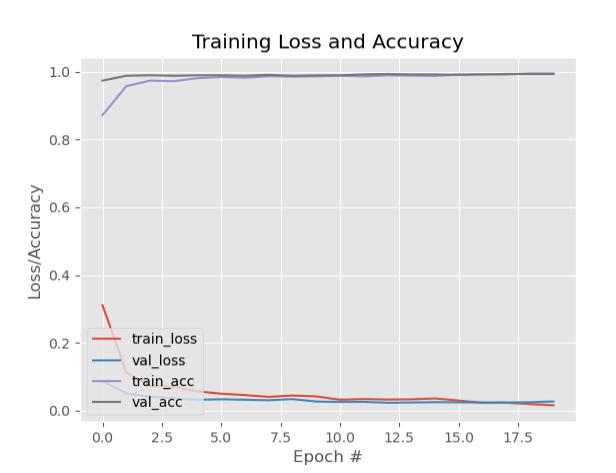
Epoch 2/2 81/81 [==============================] - 127s 2s/step - loss:

0.0110 - accuracy: 0.9973 - val\_loss: 0.0031 - val\_accuracy: 0.9965

For the model face mask detection and once the person is found with mark the person

with the square box.

8.

****

5) Graph of Training Loss and Accuracy

* 1. **Implementing the model**

The model implemented in the video. The video read from frame to frame, then the face

detection algorithm works. If a face is detected, it proceeds to the next process. From

detected frames containing faces, reprocessing will be carried out including resizing the

image size, converting to the array, pre-processing input using MobileNetV2.

The next step is predicting input data from the saved model. Predict the input image that

has been processed using a previously built model. Besides, the video frame will also be

labeled that the person is wearing a mask or not along with the predictive percentage.

Next step is real time mask detection , in this we use opencv and

haarcascade\_frontalface\_default.xml file to detect the only the frontal face of the person

to detect the face more preciously and the output to be more clear.

import cv2

path = "haarcascade\_frontalface\_default.xml"

font\_scale = 1.5

font = cv2.FONT\_HERSHEY\_PLAIN

rectangle\_bgr = (255, 255, 255)

img = np.zeros((500, 500))

text = "Some text in a box!"

(text\_width, text\_height) = cv2.getTextSize(text, font, fontScale=font\_scale,thickness=1)[0]

text\_offset\_x = 10

text\_offset\_y = img.shape[0] - 25

box\_coords = ((text\_offset\_x, text\_offset\_y), (text\_offset\_x + text\_width + 2, text\_offset\_y - text\_height - 2))

**9.**

cv2.rectangle(img, box\_coords[0], box\_coords[1], rectangle\_bgr, cv2.FILLED)

cv2.putText(img, text, (text\_offset\_x,text\_offset\_y), font, fontScale=font\_scale, color=(0, 0, 0), thickness=1)

cap = cv2.VideoCapture(1)

if not cap.isOpened():

    cap = cv2.VideoCapture(0)

if not cap.isOpened():

    raise IOError("Cann't open webcam")

while True:

    ret,frame = cap.read()

    faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade\_frontalface\_default.xml')

    gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

    faces = faceCascade.detectMultiScale(gray, 1.1,4)

    for x,y,w,h in faces :

        roi\_gray = gray[y:y+h, x:x+w]

        roi\_color = frame[y:y+h, x:x+w]

        cv2.rectangle(frame, (x, y), (x+w, y+h), (255,0,0), 2)

        facess =faceCascade.detectMultiScale(roi\_gray)

        if len(facess) == 0:

            print("Face not detected")

        else:

            for (ex,ey,ew,eh) in facess:

                face\_roi = roi\_color[ey: ey+eh, ex:ex+ew]

    final\_image = cv2.resize(face\_roi, (244,244))

    final\_image = np.expand\_dims(final\_image,axis = 0)

    final\_image = final\_image/255.0

    font = cv2.FONT\_HERSHEY\_SIMPLEX

    predictions = new\_model.predict(final\_image)

    font\_scale = 1.5

    font = cv2.FONT\_HERSHEY\_PLAIN

    if(predictions < 0.1):

        status = "No Musk"

        x1,y1,w1,h1 = 0,0,175,75

**10.**

        cv2.rectangle(frame, (x1, x1), (x1 + w1, y1 + h1), (0,0,0), -1)

        cv2.putText(frame, status, (x1 + int(w1/10),y1 +int(h1/2)), cv2.FONT\_HERSHEY\_SIMPLEX, 0.7, (0,0,255), 2)

        cv2.putText(frame,status,(100, 150),font, 3,(0, 0, 255),2,cv2.LINE\_4)

        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255))

    else:

        status = "Face Mask"

        x1,y1,w1,h1 = 0,0,175,75

        cv2.rectangle(frame, (x1, x1), (x1 + w1, y1 + h1), (0,0,0), -1)

        cv2.putText(frame, status, (x1 + int(w1/10),y1 +int(h1/2)), cv2.FONT\_HERSHEY\_SIMPLEX, 0.7, (0,255,0), 2)

        cv2.putText(frame,status,(100, 150),font, 3,(0, 255, 0),2,cv2.LINE\_4)

        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0))

    cv2.imshow('Face Mask Detection ',frame)

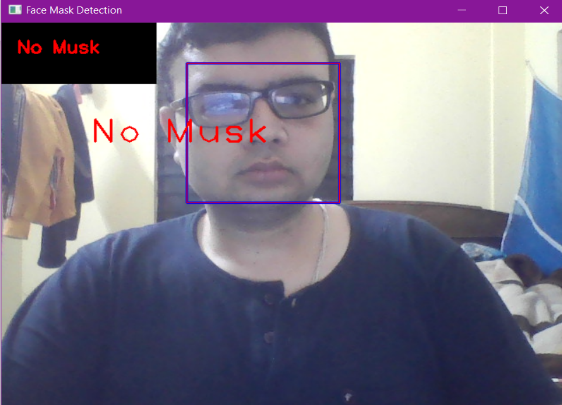
    if cv2.waitKey(2) & 0xFF == ord('q'):

        break

cap.release()

cv2.destroyAllWindows()

**Output :**

** **

In real time when the model predict and show the output in text at the corner and over the face and for with mask the text is in green color and for without mask the text is in red color .

**11.**

1. **Conclusion**

In our research we have proposed a system that automatically identifies whether or not a person is wearing a face mask and notify the higher authorities if not wearing a mask. This proposed system uses Computer Vision and MobileNet to help the public ensure that they are wearing face masks and to keep away from the spread of COVID-19 virus. Our research also helps police or higher authorities that makes it easier to identify whether a person is wearing a mask, if not then they will be also having the victim's photo by which they can take further actions. The proposed system can be implemented in places like railway stations, shopping malls, offices, schools, airports, etc.

1. **Future Scope of the Project**

There are many more different cases in which this model can be integrated for the safety of the public:

* Identify a person if he is doing any crime by wearing face mask.
* Identify what type of mask is the person wearing.
* Coughing and Sneezing Detection.
* Temperature Screening

1. **Bibliography**
2. Lin, Y.-H., Liu, C.-H., & Chiu, Y.-C, “Google searches for the keywords of “wash hands” predict the speed of national spread of COVID-19 outbreak among 21 countries. Brain, Behavior, and Immunity”, 2020.
3. Murray, O. M., Bisset, J. M., Gilligan, P. J., Hannan, M. M., & Murray, J. G., “Respirators and surgical facemasks for COVID-19: implications for MRI. Clinical Radiology, 75(6), 405–407. https://doi.org/10.1016/j.crad.2020.03.029, 2020.
4. Fadare, O. O., & Okoffo, E. D. “Covid-19 face masks: A potential source of microplastic fibers in the environment”. *Science of the Total Environment*, 737, 140279, 2020.
5. Feng, S., Shen, C., Xia, N., Song, W., Fan, M., & Cowling, B. J., “Rational use of face masks in the COVID-19 pandemic”. *The Lancet Respiratory Medicine*, 8(5), 434–436, 2020.
6. Fund, U. N. C., “COVID-19 and Children in Indonesia. COVID-19 and Children in Indonesia” *UNICEF Press*, 11 May, 2020.

**12.**

1. Wilder-Smith, A., & Freedman, D. O., “Isolation, quarantine, social distancing and community containment: pivotal role for old-style public health measures in the novel coronavirus (2019-nCoV) outbreak”. *Journal of Travel Medicine*, 27(2)-2020 https://doi.org/10.1093/jtm/taaa020
2. Setiati, S., & Azwar, M. K., “COVID-19 and Indonesia”. *Acta Medica Indonesiana*, 52(1), 84–89, 2020.
3. Loey, M., Manogaran, G., Taha, M. H. N., & Khalifa, N. E. M. “A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic”.
4. Ejaz, M. S., Islam, M. R., Sifatullah, M., & Sarker, A., “Implementation of Principal Component Analysis on Masked and Non-masked Face Recognition”. 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), 1–5.
5. Hussain, S. A., & Al Balushi, A. S. A. “A real time face emotion classification and recognition using deep learning model”. *Journal of Physics: Conference Series*, 1432(1), 12087, 2020.
6. Qin, B., & Li, D. “Identifying Facemask-wearing Condition Using Image Super-Resolution with Classification Network to Prevent COVID-19”, 2020.
7. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. “Mobilenetv2: The next generation of on-device computer vision networks. URL Https://Ai. Googleblog. Com/2018/04/Mobilenetv2-next-Generation-of-on. 2020.
8. OECD. “Handbook for Constructing Composite Indicators : Methodology and User Guide (Vol. 63, Issue 5 Year 2008)”. https://doi.org/10.1111/jgs.13392
9. Greenhalgh, Trisha, et al. “Face masks for the public during the covid-19 crisis”. *Bmj*, 2020, 369.
10. Cheng, Kar Keung; Lam, Tai Hing; Leung, Chi Chiu, “Wearing face masks in the community during the COVID-19 pandemic: altruism and solidarity”. *The Lancet*, 2020.
11. Pemprov DKI. “Peraturan Gubernur Nomor 20 Tahun 2020 Tentang Pengendalian Kegiatan Bepergian di Provinsi DKI Jakarta Dalam Upaya Pencegahan Penyebaran Corona Virus Desease (COVID-19)” Tanggal 26 Juni 2020
12. Fekete A, “Spatial disaster vulnerability and risk assessments: challenges in their quality and acceptance”. doi:10.1007/s11069-011-9973-7, 2020

**13.**