

Real Time Face Mask Detection

**Submitted in partial Fulfillment of the Requirements for the
award of Bachelor Of Computer Science**

By:-

Logitha M (39290058)

Hemamalini M (39290033)



Department of Computer Engineering and school of Computing

SATHYABAMA
INSITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
JAPPIAR NAGAR, RAJIVI GANDHI
CHENNAI - 600 019

SATHYABAMA
INSITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
JAPPIAR NAGAR,RAJIVI GANDHI
CHENNAI - 600 019



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of Logitha M (39290058) Hemamalini(39290033) who carried out the project entitled “Real time face mask detection” under my supervision from November 2020 to April 2021.

Internal Guide

Dr.S.Maheshwari

Head of the Department
Dr.Laskimanam

Submitted for Viva voice Examination held on -----

INTERNAL
EXAMINAIR

EXTERNAL
EXAMINAIR

DECLARATION

1.Logitha (39290058) 2.Hemamalini (39290033) hereby declare that the Project Report entitled “**Real time face mask detection**” done by us under the guidance of **Dr.MAHESHWARI** is submitted in partial fulfillment of the requirements for the award of Bachelor of Science degree in Computer Science.

DATE:

PLACE: CHENNAI

**SIGNATURE OF THE
CANDIDATES**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA, M.E., Ph.D.**, Dean, School of Computing and **Dr. S. VIGNESHWARI, M.E., Ph.D.**, and **Dr. L. LAKSHMANAN, M.E., Ph.D.**, Head of the Department, Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.Maheswari.Phd.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Computer Science and Engineering who were helpful in many ways for the Completion of the project.

HEMAMALINI M

LOGITHA M

ABSTRACT

Infection sickness 2019 has impacted the world genuinely. One significant security strategy for individuals is to wear covers in open regions. Besides, numerous public specialist co-ops expect clients to utilize the help provided that they wear veils accurately. Be that as it may, there are a couple of examination learns about facial covering recognition in view of picture analysis. this framework is an ongoing application to recognize individuals assuming they wearing mask or without wearing mask.

Wearing a mask is among the non-drug mediation estimates that can be utilized to cut the essential wellspring of sars-cov2 drops removed by a tainted individual. the informational index that contains around 1500 pictures or more of individuals with cover and without mask will be prepared and contrast individuals and cover and without mask. It tends to be right into it in open region like rail route station, bus stop, schools, medical clinics ..and so forth.,

Healthcare officials , WHO & local authorities are propelling people with face mask is the one of the comprehensive strategies to overcome the corona virus. Wearing a mask is among the non-pharmaceutical intervention measures that can be used to cut the primary source of SARS-CoV2 droplets expelled by an infected individual.

CHAPTER- 1

INTRODUCTION

The Covid-19 pandemic isn't finished. in the non-appearances of immunizations or drugs to stop COVID-19,one of the best technique to slow the spread of the infection is to wear facial covering , close by hand washing and rehearsing physical separating. "Concealing up" is the one of modest and viable approach to restricting the transmission of illness.

Now that everybody know the significance of wearing cover in their life Corona virus is a transferable infection that is communicated from an individual while talking, sniffing as respiratory drops But presently a days like wise a few people groups don't have a clue about the significant of wearing mask.

OVER VIEW OF PROJECT

we will figure out how to distinguish faces continuously utilizing Open CV. Subsequent to recognizing the face from the webcam stream, we will save the video containing the face. Later we will pass these video (pictures) to our mask indicator classifier to see whether the individual is wearing a mask or not .we will figure out how to distinguish faces continuously utilizing Open CV. Subsequent to recognizing the face from the webcam stream, we will save the edges containing the face. Later we will pass these video (pictures) to our mask indicator classifier to see whether the individual is wearing a mask or not.

LITERATURE REVIEW

1) M. S. Ejaz and M. R. Islam.(2019)

Masked face recognition using conventional neural network:

The essential worry to this work is about facial veils, and particularly to upgrade the acknowledgment precision of various concealed faces. A possible methodology has been recommended that comprises of first distinguishing the facial locales. The blocked face identification issue has been moved toward utilizing Multi-Task Cascaded Convolutional Neural Network (MTCNN). Then, at that point, facial highlights extraction is performed utilizing the Google Face Net implanting model. Lastly, the order task has been performed by Support Vector Machine (SVM).

Disadvantage (or) Disadvantage:

- High detection accuracy
- Low false positive rate.
- Longer training time.
- Less accurate on black faces.

2) M. Inamdar and N. Mehendale.(2020)

Real-time face mask identification using face mask net deep learning network

The model called "Face mask net" was proposed by to recognize an individual wearing a face mask appropriately or not. The data set prepared to machine contained 35 pictures including the two pictures of veiled countenances and un-concealed faces.

Disadvantage (or) Disadvantage

Computationally simple and fast. Less accurate.

LITERATURE REVIEW

3).V. Vinitha and V(2021)

Covid-19 face mask detection with deep learning and computer vision.

This paper gives a straightforward method for accomplishing this goal using some major Machine Learning apparatuses as TensorFlow, Keras, Open CV and Scikit - Learn. The recommended strategy effectively perceives the face in the picture or video and afterward decides if it has a veil on it. As a reconnaissance work entertainer, it can likewise perceive a face along with a veil moving as well as in a video. The method achieves brilliant exactness.

Disadvantage (or) Disadvantage

Shorter training time.Robust to occlusion.

High false positive rate.Less accurate

4) A.S. Kinjal Goswami.(2020)

"Detection of face mask through machine learning"

The specific model so evolved utilizes profound learning. Profound learning is man-made consciousness (AI) work that impersonates the operations of the human cerebrum in handling information and making designs for use in choice making. Profound learning is a subset of AI in computerized reasoning that has networks prepared to do gaining unaided from information that is unstructured or unlabeled. Otherwise called profound brain learning or profound brain organization. As the title of this paper suggests that use of open CV, keras and TensorFlow have been made to do the real time detection of face mask which predicted mask or no mask; they obtained an accuracy of 95% by taking the input as a video from the camera.

CHAPTER-2

AIM AND SCOPE OF PROJECT

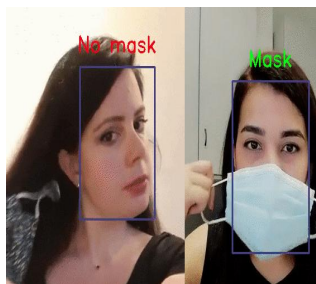
The profound learning is kind of AI based on artificial neural network in which different layers of handling are utilized to extricate advancing more significant level highlights from information. Profound learning is a subset of AI, which is basically a neural network with at least 3 layers. In this way neural networks endeavor to reenact the way of behaving of human mind albeit its performance matching capacity permitting it to "Learn" from large measure of information.

The aim is to develop an application to investigate mask detection to prevent COVID-19. Because, the use of masks can prevent the transmission of sparks that can be used to protect others, and help contaminate the environment due to this spark to minimize the spread of COVID-19 due to the use of masks. But some people are not wearing masks properly in public places.

Real-time face mask detection using AI (Artificial Intelligence) or deep learning is a variant of a deep neural network that detects objects in an image and generates a high face mask detection system by using a camera combined with a computer vision platform to detect people without masks.

OBJECTIVE

Face mask detection refers to detect whether a person is wearing a mask or not. In fact, the problem is reverse engineering of face detection where the face is detected using different machine learning algorithms for the purpose of security, authentication and surveillance. Face detection is a key area in the field of Computer Vision and Pattern Recognition.



CHAPTER-3

PROPOSED SYSTEM:

The proposed system consists of 4 steps ,

Includes,

- I. training of real time images
- II. multiple face detection using Haar-classifier
- III. comparison of trained real time images with images from the surveillance camera
- IV. result based on the comparison.



We implement this application in an official environment, when an employee enters into the zone, the application automatically scales the current location with the static company's zone. Then the employee needs to do face verification. When the face is matched then the attendance will be added into the database. Thus, the attendance count will be added to the server. This project was created to detect multi-faces at a time. The accuracy level of detection was increased from the existing systems.

METHODOLOGY

The system consists of a camera that captures the images of the employee and sends it to the image enhancement module. After enhancement the image comes within the Face mask Detection and Recognition modules then the it will show thus the person is wearing mask or not . At the time of enrollment, templates of face images of individual employees are stored in the Face database. Here all the faces are detected from the input image and therefore the algorithm compares them one by one with the face database. If any face is recognized without mask in the server it will show as 100% no mask. If it is a company the employee is not wearing the mask in front or around the surveillance camera it will show as 100% no mask by marking around the person or the employee face in the red square as shown in fig below.



REQUIRMENTS

Hardware Requirements:

Hard Disk : 500GB and Above

RAM : 4GB and Above

Processor : I3 and Above

Webcam – 1

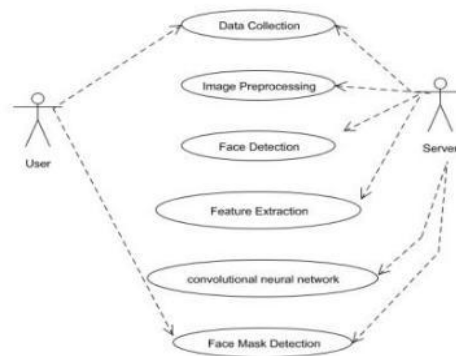
Software Requirements:

Operating System : Windows 7, 8, 10 (64 bit)

Software : Python

Tools : Python 3.7 IDLE, Anaconda

MODULES



- Data set
- Face detection
- Training process
- Face mask detection

MODULES

Data set:

Data set are often accomplished by digitally scanning an existing photograph or by using an electro - optical camera to accumulate a live picture of a topics.

Face Detection:

After collecting the data set we categorized in to 2 parts. Detect the faces using deep leaning algorithms. algorithms are used to detect the faces accurately.

Training Process:

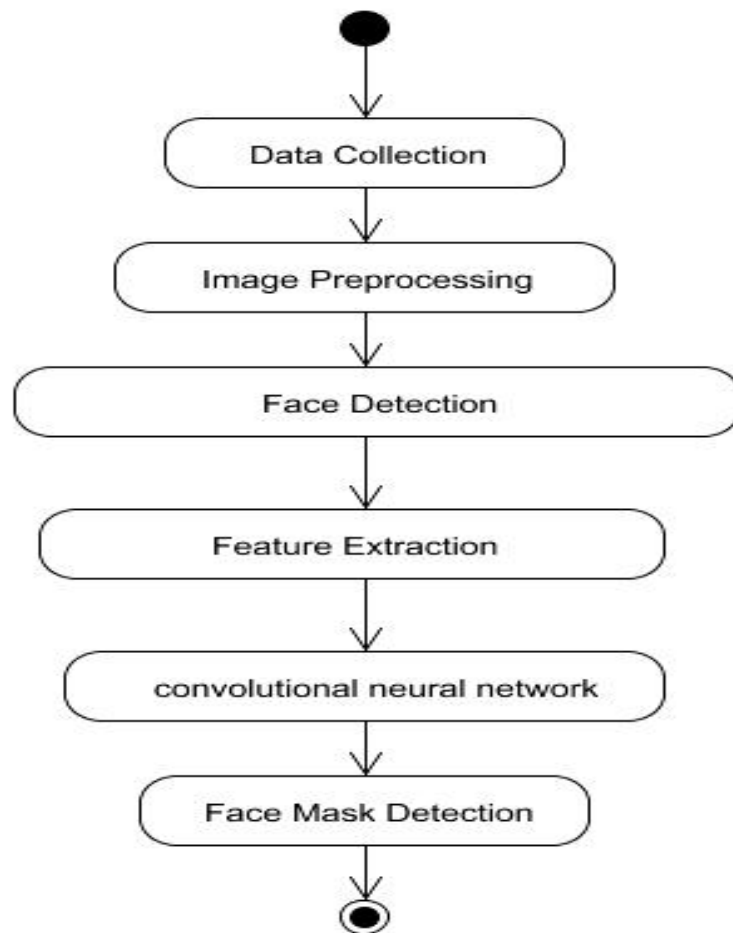
We are train the collected video frames using Deep leaning techniques. Because, it will train to multiple times and push a very fast results. Deep Learning is a one of the best algorithm for making a project and it is also give an accurate output.

Face mask detection:

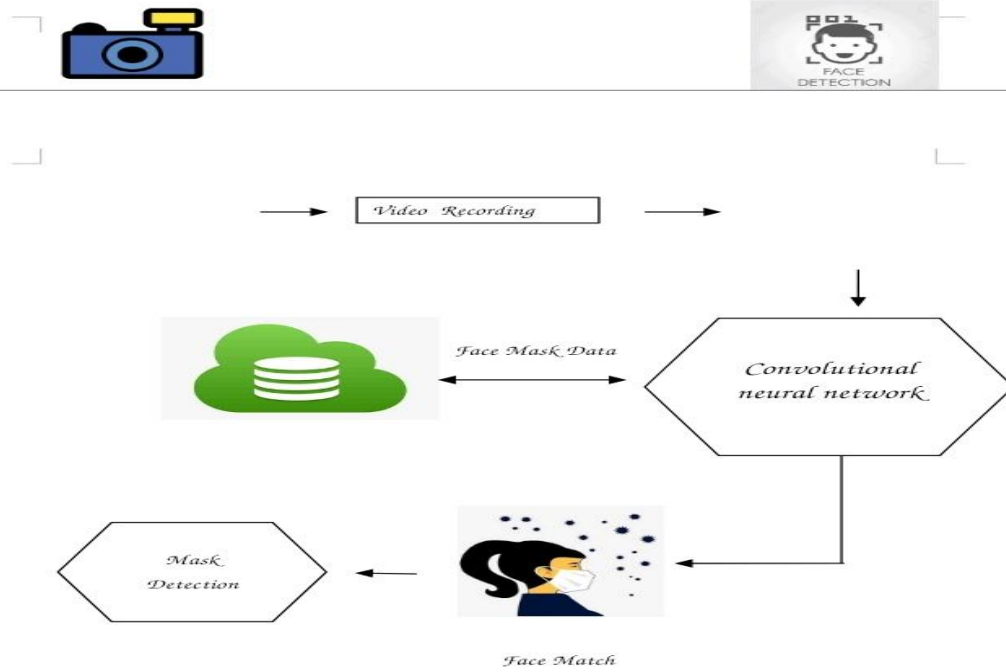
Finally, it will working in real time detection with mask or without mask.



DIAGRAM



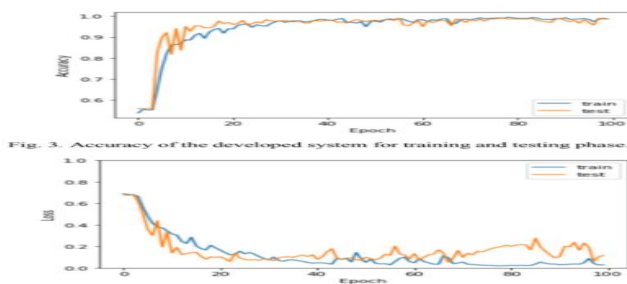
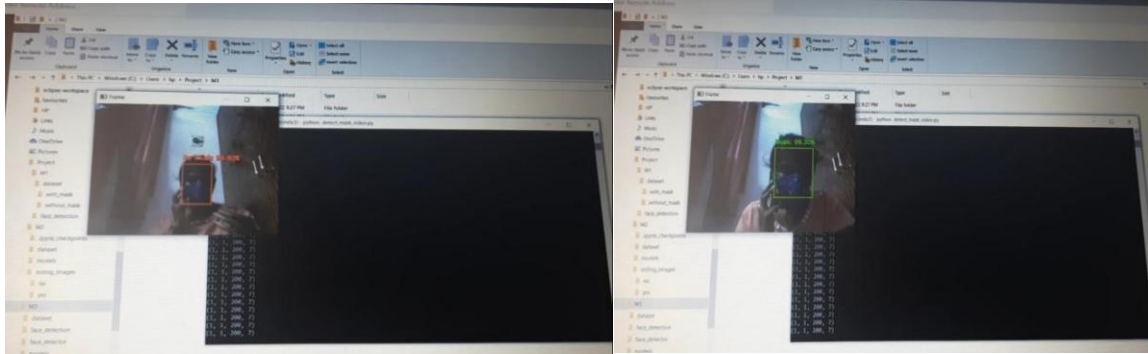
Architecture Diagram:



APPLICATION SNAPSHOT

person without mask or properly wearing

person with mask:



Results and Discussions

Discussions:

Given a still image or video of a scene, identify or verify one or more persons in this scene using a stored database of facial images.

It works as similar as face recognition system, which is used to recognize faces for various purposes, in which face mask is recognized.

Different persons may have very similar appearance

- Twins
- Relatives

Difference between two images of the same subject under different view angles is greater than differences between images of two different subjects under the same view. This device clearly identifies the images and recognition was perfectly done.

Males are easier to recognize than females (5-9%) Younger peoples are easiest to acknowledge than older people

Result:

By preserving a reasonable proportion of different classes, the data set is partitioned into training and testing set. The data set comprises of 1539 samples in total where 80% is use in training phase and 20% is used in testing phase. The training and testing data set contains 1231 and 308 images respectively. The developed architecture is trained for 100 epochs since further training results cause over fitting on the training data. Over fitting occurs when a model learns the unwanted patterns of the training samples

. Hence, training accuracy increases but test accuracy decreases. Fig. 3 and Fig. 4 show the graphical view of accuracy and loss respectively. The trained model showed 98.7% accuracy and AUC of 0.985 on the unseen test data.

Conclusion

This paper presents a system for a smart city to reduce the spread of corona virus by informing the authority about the person who is not wearing a facial mask that is a precautionary measure of COVID-19. The motive of the work comes from

the people disobeying the rules that are mandatory to stop the spread of corona virus. The system contains a face mask detection architecture where a deep learning algorithm is used to detect the mask on the face. To train the model, labeled image data are used where the images were facial images with masks and without a mask. The proposed system detects a face mask with an accuracy of 98.7%. The decision of the classification network is transferred to the corresponding authority. The system proposed in this study will act as a valuable tool to strictly impose the use of a facial mask in public places for all people.

Conclusion

This paper presents a system for a smart city to reduce the spread of corona virus by informing the authority about the person who is not wearing a facial mask that is a precautionary measure of COVID-19. The motive of the work comes from the people disobeying the rules that are mandatory to stop the spread of corona virus. The system contains a face mask detection architecture where a deep learning algorithm is used to detect the mask on the face. To train the model, labeled image data are used where the images were facial images with masks and without a mask. The proposed system detects a face mask with an accuracy of 98.7%. The decision of the classification network is transferred to the corresponding authority. The system proposed in this study will act as a valuable tool to strictly impose the use of a facial mask in public places for all people.

References:

- M . S. Ejaz and M. R. Islam, "Masked face recognition using convolutional neural network", 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), pp. 1-6, 2019.
- M. Inamdar and N. Mehendale, Real-time face mask identification using facemasknet deep learning network, 2020.
- V. Vinitha and V, Covid-19 facemask detection with deep learning and computer vision, vol. 07, pp. 3127-3132, 2020.
- M. Jiang, X. Fan and H. Yan, Retinamask: A face mask detector, 2020.
- P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria and J. Hemanth, "Ssdmnv2: A real time dnn-based face mask detection system using single shot multibox detector and mobilenetv2", Sustainable Cities and Society, vol. 66, pp. 102692, 2021A. S. Kinjal Goswami, "Detection of face mask through machine learning", International Research Journal of Modernization in Engineering Technology and Science, vol. 02, pp. 450-455, 2020.
- M. R. Bhuiyan, S. A. Khushbu and M. S. Islam, "A deep learning based assistive system to classify covid-19 face mask for human safety with yolov3", 2020 11th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1-5, 2020.
- M. S. Islam, E. Haque Moon, M. A. Shaikat and M. Jahangir Alam, "A novel approach to detect face mask using cnn", 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), pp. 800-806, 2020.
- A. S. Joshi, S. S. Joshi, G. Kanahasabai, R. Kapil and S. Gupta, "Deep learning framework to detect face masks from video footage", 2020 12th International Conference on Computational Intelligence and Communication Networks (CICN), pp. 435-440, 2020.
- S. A. Sanjaya and S. A. Rakhmawan, "Face Mask Detection Using MobileNetV2 in the Era of COVID-19 Pandemic", 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy ICDABI 2020, pp. 1-5, 2020.

PROGRAM CODES

IMPORT LIBRARIES

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import tensorflow as tf
import cv2
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
import itertools
from keras.utils.np_utils import to_categorical
from keras.models import Sequential
from keras.callbacks import ModelCheckpoint
from keras.preprocessing.image import ImageDataGenerator
from sklearn.metrics import confusion_matrix
```

```
import itertools
```

STORE EVERY IMAGE IN AN ARRAY

```
import os
data_dir = (r'.\testing_images')
categories = ['no', 'yes']
for i in categories:
    path = os.path.join(data_dir, i)
    for img in os.listdir(path):
        img_array = cv2.imread(os.path.join(path,img))
```

Resize each image to same size for fast processing

```
img_size = 128
image_array = cv2.resize(img_array, (img_size,img_size))
```

```
dno = cv2.imread('./testing_images/no/1 (9).jpg')
dyes = cv2.imread('./testing_images/yes/1 (19).jpg')
```

```
plt.rcParams["figure.figsize"] = (5,5)
plt.imshow(dno)
plt.axis('off')
OUT[]:(-0.5, 149.5, 144.5, -0.5)
```

```
plt.rcParams["figure.figsize"] = (9,9)
plt.imshow(dyes)
plt.axis('off')
OUT[]:(-0.5, 421.5, 636.5, -0.5)
```

Convert each image to grayscale and append into an array

```
train_data = []

for i in categories:
    train_path = os.path.join(data_dir,i)
    tag = categories.index(i)
    for img in os.listdir(train_path):
        try:
            image_arr = cv2.imread(os.path.join(train_path , img),
cv2.IMREAD_GRAYSCALE)
            new_image_array = cv2.resize(image_arr, (img_size,img_size))
            train_data.append([new_image_array , tag])
        except Exception as e:
            pass
```

SPLIT THE FEATURES AND TARGET IN X AND Y

```
X = []
y = []
for i,j in train_data:
    X.append(i)
    y.append(j)
X = np.array(X).reshape(-1,img_size,img_size)
print(X.shape)
X = X/255.0
X = X.reshape(-1,128,128,1)
(115, 128, 128)
```

ONE-HOT ENCODE THE TARGET COLUMN

```
from keras.utils.np_utils import to_categorical

y_enc = to_categorical(y, num_classes = 4)
```

```
X_train , X_test, y_train, y_test = train_test_split(X , y_enc , test_size = 0.1,  
random_state = 42)  
X_train , X_val, y_train, y_val = train_test_split(X_train , y_train , test_size = 0.1,  
random_state = 42)
```

```
from sklearn.metrics import confusion_matrix  
import itertools
```

```
from keras.utils.np_utils import to_categorical  
from keras.models import Sequential  
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D  
from keras.optimizers import RMSprop,Adam  
from keras.preprocessing.image import ImageDataGenerator  
from keras.callbacks import ReduceLROnPlateau
```

```
from keras.callbacks import EarlyStopping
```

```
model = Sequential()
```

```
model.add(Conv2D(filters = 64, kernel_size = (5,5),padding = 'Same', activation  
='relu', input_shape = (128,128,1)))  
model.add(MaxPool2D(pool_size=(2,2)))  
model.add(Dropout(0.2))
```

```
model.add(Conv2D(filters = 128, kernel_size = (3,3),padding = 'Same', activation  
='relu'))  
model.add(MaxPool2D(pool_size=(2,2), strides=(2,2)))  
model.add(Dropout(0.2))
```

```
model.add(Conv2D(filters = 128, kernel_size = (3,3),padding = 'Same', activation  
='relu'))  
model.add(MaxPool2D(pool_size=(2,2), strides=(2,2)))  
model.add(Dropout(0.2))
```

```
model.add(Conv2D(filters = 128, kernel_size = (2,2),padding = 'Same', activation  
='relu'))
```

```
model.add(MaxPool2D(pool_size=(2,2), strides=(2,2)))
```

```
model.add(Dropout(0.2))
```

```
model.add(Conv2D(filters = 256, kernel_size = (2,2),padding = 'Same', activation  
='relu'))
```

```
model.add(MaxPool2D(pool_size=(2,2), strides=(2,2)))
```

```
model.add(Dropout(0.2))
```

```
model.add(Flatten())
```

```
model.add(Dense(1024, activation = "relu"))
```

```
model.add(Dropout(0.5))
```

```
model.add(Dense(4, activation = "softmax"))
```

```
optimizer = Adam(lr=0.001, beta_1=0.9, beta_2=0.999)
```

```
model.compile(optimizer = optimizer , loss = "categorical_crossentropy",  
metrics=["accuracy"])
```

```
epochs = 10
```

```
es = EarlyStopping(  
    monitor='val_acc',  
    mode='max',  
    patience = 3  
)
```

```

batch_size = 16
imggen = ImageDataGenerator(
    featurewise_center=False,
    samplewise_center=False,
    featurewise_std_normalization=False,
    samplewise_std_normalization=False,
    zca_whitening=False,
    rotation_range=0,
    zoom_range = 0,
    width_shift_range=0,
    height_shift_range=0,
    horizontal_flip=True,
    vertical_flip=False)

from keras.callbacks import ModelCheckpoint

checkpoint = ModelCheckpoint("models/model_weights.h5", monitor='val_acc',
verbose=1, save_best_only=True, mode='max')
callbacks_list = [checkpoint]

imggen.fit(X_train)
history = model.fit_generator(imggen.flow(X_train,y_train,batch_size = batch_size),
                             epochs = epochs, validation_data = (X_val,y_val),
                             steps_per_epoch = X_train.shape[0] // batch_size,
                             callbacks=callbacks_list)

# serialize model structure to JSON
model_json = model.to_json()
with open(r".\models\mask_detector.model", "w") as json_file:
    json_file.write(model_json)

os.listdir(r"models")

```

```
OUT[:]['mask_detector.model', 'model_weights.h5']
```