“Face Mask Detection”

Major project report submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

By

Varun Rangra (181451)

UNDER THE SUPERVISION OF Dr. Hari Singh

Department of Computer Science & Engineering and Information Technology,Jaypee University of Information Technology, Waknaghat, 173234, Himachal Pradesh, INDIA

TABLE OF CONTENT

I
DECLARATION

I hereby declare that this project has been done by me under the supervision of Dr. Hari Singh, Assistant Professor (SG) Jaypee University of Information Technology. I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:
Dr. Hari Singh
Assistant Professor(SG),
Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology

Submitted by:
Varun Rangra (181451)
Computer Science & Engineering Department
Jaypee University of Information Technology

II
CERTIFICATE

This is to certify that the work which is being presented in the project report titled " Face Mask Detection" in the partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science And Engineering and submitted to the Department of Computer Science And Engineering, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by "Varun Rangra (181451)" during the period from January 2022 to May 2022 under the supervision of Dr. Hari Singh in the Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

Varun Rangra (181451)

The above statement made is correct to the best of my knowledge.

Dr. Hari Singh
Assistant Professor(SG),
Computer Science & Engineering and Information Technology
Jaypee University of Information Technology, Waknaghat,

III

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the project work successfully.

I really grateful and wish my profound my indebtedness to Supervisor Dr. Hari Singh, Assistant Professor(SG), Department of CSE Jaypee University of Information Technology,Wakhnaghat. Deep Knowledge & keen interest of my supervisor in the field of "Computer Science" to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

I would like to express my heartiest gratitude to Dr. Hari Singh, Department of CSE, for his kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straight forwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

Varun Rangra (181451)

IV
Abstract

This put forward a novel mechanism in which good rules to restain Covid-19 epidemic needs a good attention to not to impact people health and global economy.In the absence of good antiviral and with the no great medical facilities,many measure are recommended by WHO to able to control the virus rate andrid of the limited medical resources.All countries are mandating to wear the mask in social event. To wards public health, our aim to be high efficiency and a good accuracy technique that can find non-mask faces in social gathering events. We will combine dataset to build the Covid-19 face mask detector with computer vision using Python,OpenCV,and Tensor Flow and Keras. Our goal is to identify whether the person on image/video stream that the person wearing a mask or notwith the help of computer vision and deep learning .

1.1 Introduction

Face Mask Detection is the approach for the reduce of corona virus.All over the countries faced this pandemic.The world had mad a great lose due to this pandemic.On August 2020,WHO report that the covid-19 disease is spread by the respiratory ailment.It has affected all worldwide people and theworld suffers.

1>According to reports and the health experts if we want to control Covid-19 virus we need tomaintained the social distance and boosts our health system.

2> In recent times, many University clears that wears a mask reduce the spread of covid-19.This is the only way that we can use this to reduce the spread of coronavirus.

We need made it compulsory to wear a mask in social gathering events,so now its our duty to providegood facility to force people to apply a mask.

The corona virus can spreads in between the people for example, speaking,coughing, or sneezing — even if those people are wearing masks.The recent information also gives the information the traces traces of a new strains of a corona virus, the mutant corona virus which, the virus has to change
its structure and its become strong mutant. The new strain is not even able to detect using theRT-PCR test we use now.
Our aim of the face mask detection is to determines that if there are any faces in the image or video.without masks.

There are multiple faces are present at the equal time, each face has encloses with boundry bins and hencewe understand the locations of the faces.

Human faces aren't an easy paintings for an version because there are numerous variable that can be exchange. For example :face expressions, orientations, lighting fixtures situations and in some conclusions inclusive of sunglasse,scarfs, mask and so on

The end result of the detection is to offers the face locations parameter that it could be required in lots of various shape of instances, a rectangle overlaying the crucial part of the face and eye centre.
It includes eye, nose and mouth corner, eyebrow, nose, ears,and many others.

We can use Face mask detection to see the people iswear a mask or not.With this system, we can help our government to checks who wears the mask and who not. We can use this system in Malls,social gathering and also for the security of people.It plays a very good rule in the computer visionand the recongnition.

This is highly effective and find the good accuracy

We need rules to restain Covid-19 epidemic needs a good attention to not to impact people health and globaleconomy.In the absence of good antiviral and withthe no great medical facilities, many such as measure are reco-mmended by WHO to able to control the virus rate andrid of the limited.

We need to implements the rules that everyone can wears masks in social events.

We create adataset to construct the Covid-19 face masks detector withcomputer imaginative and prescient the use of Python,OpenCV,and Tensor Flowand Keras. Our purpose is to become aware of whether the person onimage video

movement that the character wearing a mask or no longer/with the assist of laptop imaginative and prescient and deep getting to know .

1:This system is a great for the future to use this system in the public platforms, to get control in thespread of covid-19.

2:This system integrates with the highly resolutions,survellances devices.

1.2 Problem Statement

All over the countries faced this pandemic.The world had mad a great lose due to this pandemic. On August 2020,WHO report that the covid-19 disease is spread by the respiratory ailment.It has affected all worldwide people and the world suffers millions of deaths and tragedies.

The world has face the economy issues and people had lost their family members. Covid-19 made a huge impact in our daily to daily lifes and also change our lifestyle of living.We need this system

to apply on public platforms for the good of the people and it reduces the risk of spread the covid19.

According to reports and the health experts if we want to control Covid-19 virus we need to maintainedthe social distance and boosts our health system.

We need rules to restain Covid-19 epidemic needs a good attention to not to impact people health and global economy.In the absence of good antiviral and with the no great medical facilities, as many as measure are recommended by WHO to able to control the virus rate andrid of the limited.

We need to implements the rules that everyone can wears masks in social events.

The world of covid, multidisciplinary acts has been organizes to slows the spread of this pandemic. particular, they development the method for monitor social distance or identify face mask has made-the headlines in these days

But all they hypes to show off result as soon as possible, to add up the usuals AI overpromise factors,may be signals the wrong ideas.

That solves some of the use cases is almost trivials due to small powers of AI.

In this effort to paint a good picture, we decided to show the creative process behind a solutionfor a seems simple use case in computer visions.

1.3 Objectives:

The major goal of face mask detection system is to decrease the spread of coronavirus disease is caused by acute respiratory syndrome.I build a Face Mask Detector using Keras,Tenserflow, MobileNet and Open CV. With further need improvement these type of model could be integrated with CCTV cameras to detectand identify people without masks.

The symptoms of a person with coronavirus while it is spreading fast may experience include: illness, fever,pains in joints and respiratory problem.
The Face Mask Detector did not use any morphed masked photos into datasets. The model is accurate,and for the reason that MobileNetV2 architecture is used,its also computationally efficient and making it easierto set up the model to embedded systems(Raspberry Pi,Google).

This device can therefore be utilized in actual time packages whichrequire face mask detection for protection forpurpose due to outbreak of covid regulations.
We create adataset to construct the Covid-19 face mask detector with pc.Imaginative and prescient using Python,OpenCV,and Tensor Flowand Keras. Our intention is to
become aware of whether the person onimage videostream that the individual carrying a masks or now not/.
Vision using Python,OpenCV,and Tensor Flow and Keras. Our purpose is toidentify whether or not the character onimage video streamthat the man or woman carrying a masks or now not/with the help of laptop vision and deep studying .

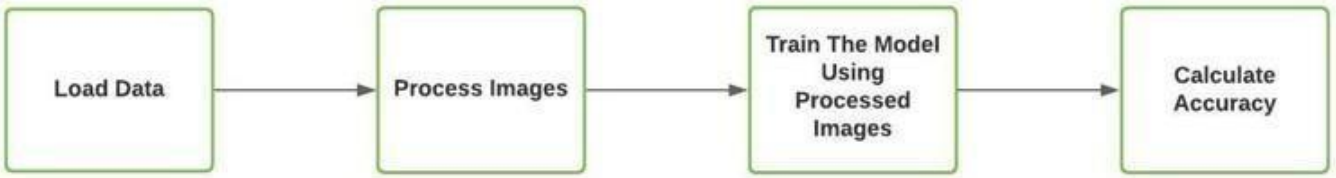
The world of covid, multidisciplinary acts has been organizes to slows the spread of this pandemic. All theAI communities has part of these endeavor.

In particular, they development the method for monitorsocial distance or identify face mask has made-the-headlines in these days.

Our major approach is we biuild a two databases. In one database we use images

without masks of peoplesand the another one we uses images with masks.

METHODOLOGY



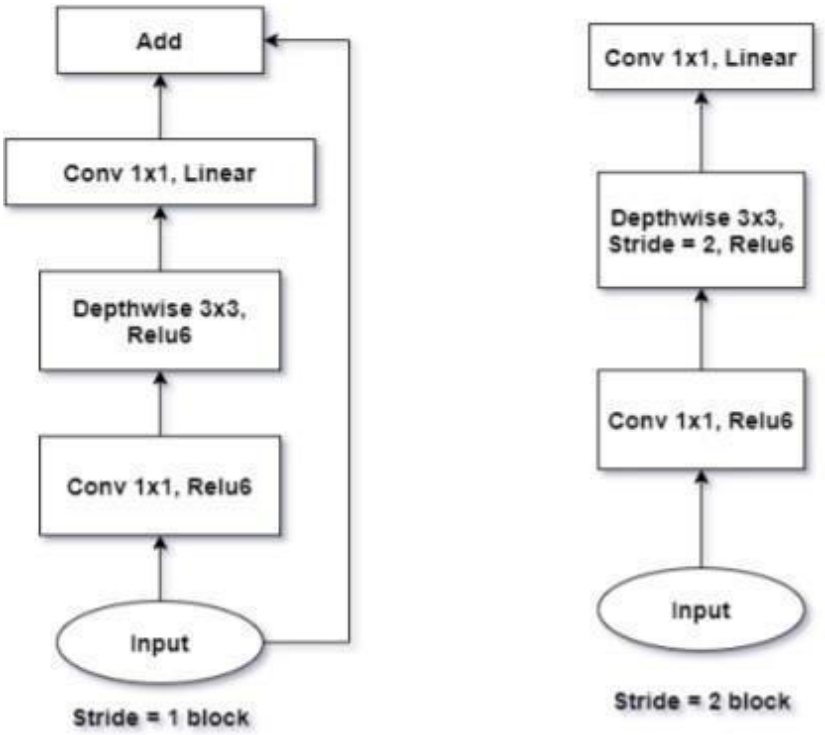


Figure1. MobileNetV2

(It shows firstly we load data and process and train model and then find accuracy)

There are two main approaches for Face Detection System:

1> Feature Based Approach

2> Image Based Approach

We use Image based approach.

Image Based approach:

It depend upon the strategies of statistical strategies and evaluation and device learning to find the applicable characters of face and non-face pics.The found out characterstics in the form of distribution models or discrimant characteristic which could beconsequently used for face detection.

Feature Base Approach

Object are bascially recognizes to their unique feature. There are many feature in a human face, which we can be recognizes between a face and many other object.

It located face by the extractes structural feature like eye, nose, mouth etc. Thus we uses them to detect the faces.

There are sort classifiers qualified to separate b/w facial and non-facial region.

1:Human face has particular texture which can be used for the differention between the face and otherobject.

2:Moreover,another feature can help us to detects the object from faces. 3:We

implements a feature-based approach by uses OpenCV.

CHAPTER 2

1. Face Mask Detector

Single shot detector structure is used for item detection motive. In this way to machine,face maskdetectorcan be developed in lots of areas like buying shops , airports and heavy traffic placesto monitorthe public and to keep away from the unfold of the sickness by way of the use of checking who's following fundamental rule and the who is not.

It takes excessive tgime in load Google Colab Notebook.We are processed a machine computionally green using MobileNetV2 which makes easy to Extract the Data set.

We can use CNN structure for better performance.We can restoration it any kind of cameras.

fig(2)

2 > Face Detection Techniques: a review, Artificial

Human beings have not the capability to identification diferent face than structures.So automatoc face detection device plays an critical function in face detectinghead pose estimation and so on. It have some mistakes like face occulusion,and non uniform illuminaton. We use Neural Network to hit upon face in Live video flow.

Tensor flow is likewise used inside the device. In modern they use Adaboostalgorithm, we're able to overcomedall issues in paper.

Three> Real Time Face Mask recongnition with alarm tool the use of deep studying:

This processed offers quicker consequences for face mask detection. Raspberry pi based real time face mask popularity that captured face photo. This device use the characteristic of VGG-sixteen as the muse networkfor face detector.

Deep mastering techniques are done to gather a clssifier the collect image of the individual wearinga mask and no masks.

It shows accuracy in detecting individual weared a face masks and no longer sporting it.

This research presences the useful tool within the preventing the unfold of covid 19 virus

The world of covid, multidisciplinary acts has been organizes to slows the spread of this pandemic. They development the method for monitor social distance or identify face mask has made-the-headlinein these days

But all they hypes to show off result as soon as possible, to add up the usuals.

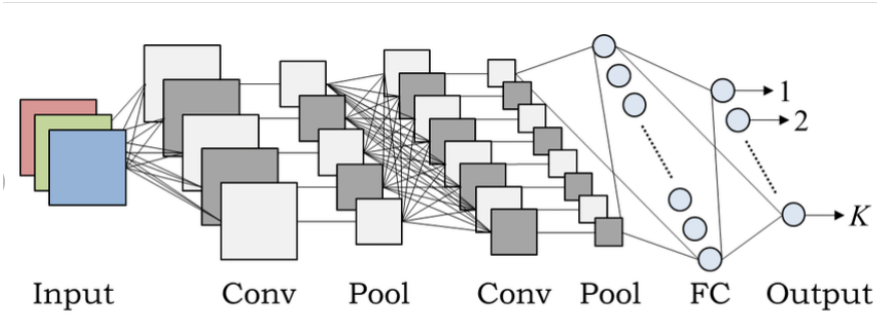
AI overpromise factors,may be signals the wrong ideas.

That solves some of the use cases is almost trivials due to small powers of AI.

4> Multiple Stage CNN Architecture for Face Mask Detection

This includes a dual degree CNN structure able to detecting masked and unmaksed faces and can be included with pre- installeed CCTV cameras. This may be helped tracked for protection violations, and promote the use of masks and ensured protection environment. Datasets wer collected from public domain at the side of some records scraped from the internet.

We can use any digital digicam to dtecting faces. It can be beneficial for public accumulating locations and society to save you them from virus transmission. Here we use live video detecrtion the use of open cv



An example of CNN architecture.

Fig-3(e.g input>conv>pool>conv>pool>fc>output)

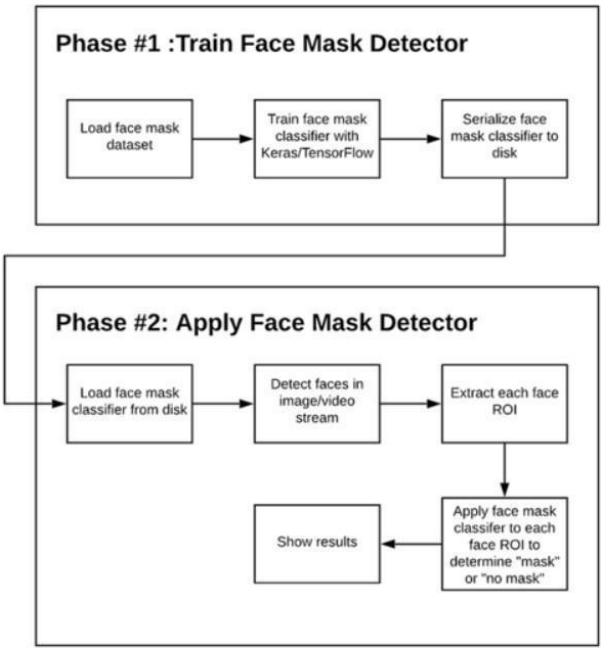


fig-4(Training and Data Loading)

First, we load the datasets with face masks and non face masks and then we train theface mask classifier with kears/tenser flow and serialize to disk.

CHAPTER 3- SYSTEM DEVELOPMENT

In Face Mask Detection,in recent trends , the approach of deep learning and detecting faces and with and without a masks was a good trend, as the Face Mask is obligatory for everyone while travelling without masks. I have created a model here that detects face masks with 3 colours channels trained on 7553 images(RGB)

The accuracy of training for the custom CNN architecture models amounted to 94% and validation exactitude to 98%.

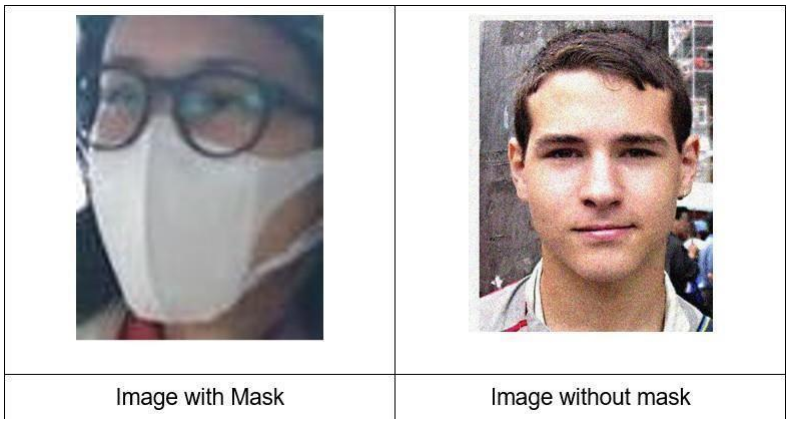


Fig-5(We make a two datasets images with masks and non -masks)

CONTENT

Data set comprises 7553 RGB images in mask and without mask in the two folders. Images are also reffered as an maskand mask-free images.

Mask pix are 3725 face photo and 3828 faceless maskimages.N order to educate a custom face mask detector , we want to break our challenge into distinct levels, every with its very own respective sub- steps:

1> Training:

Here we all recognition on loading our face masks detection dataset from the disk, schooling a model(the usage of keras/Tener waft) on thisdataset, and serialise the masks detector to disk.

2> Deployment:

When the face mask detector is skilled, we are able to then flow on to the loading the masks detector,acting face detection, after which it classifying each face as with mask or with out masks.

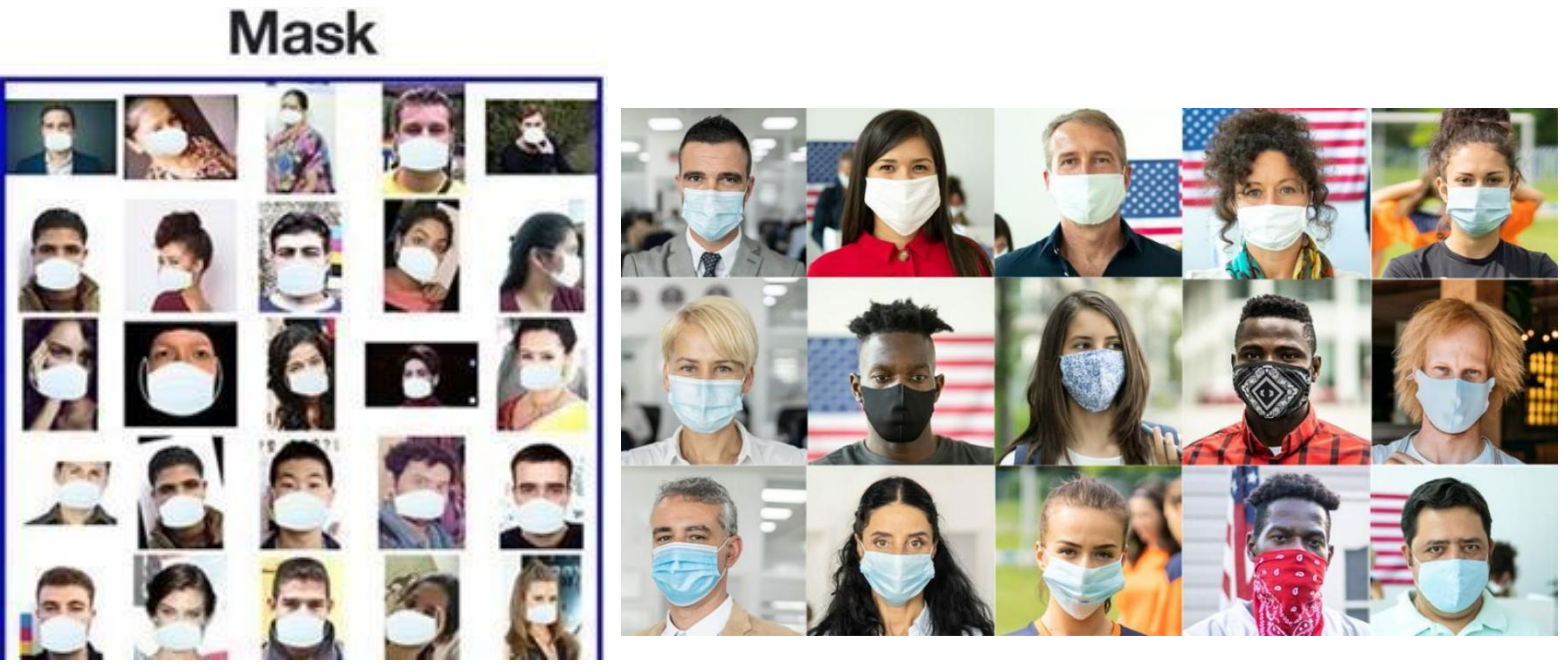


fig-6(It contains the dataset of people with masks.)

When the face mask detector is trained, we are able to then pass directly to loading the mask detector,performing face detection, after which it classifying every face as with mask or with out masks.

To create this dataset, we need to infer the vicinity of our facial systems, consisting of:

1 Eyes

2 Eyebrows

3 Nose

4 Mouth

5 Jawline

To use facial landmarks to construct a dataset of face carrying masks, we first need to go together with individual with out a masks.

We can use a hard and fast of photographs used to generate face mask samples as non-face mask samples,our version emerge as closely biased and fail to generaise properly.

Avoid this with the aid of taking the time to gather new examples of our faces
with out mask..

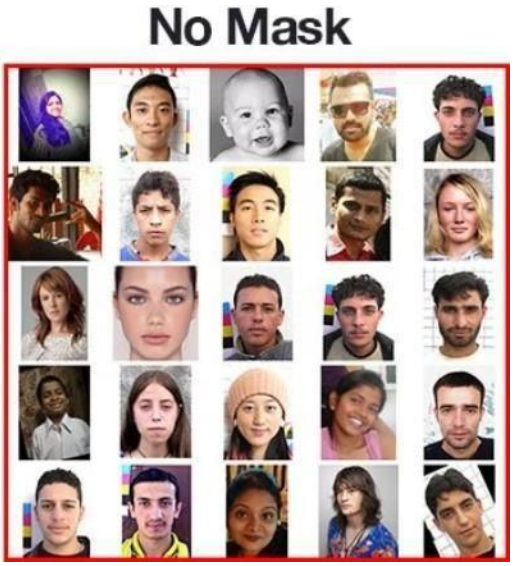


fig-7(Dataset of people without masks.)

Object are bascially recognizes to their unique feature. There are many feature in a human face, which we can be recognizes between a face and many other object.It located face by the extract structural feature like eye, nose, mouth etc.

Thus we uses them to detect the faces.

There are sort classifiers qualified to separate b/w facial and non-facial region.

Fig. 1

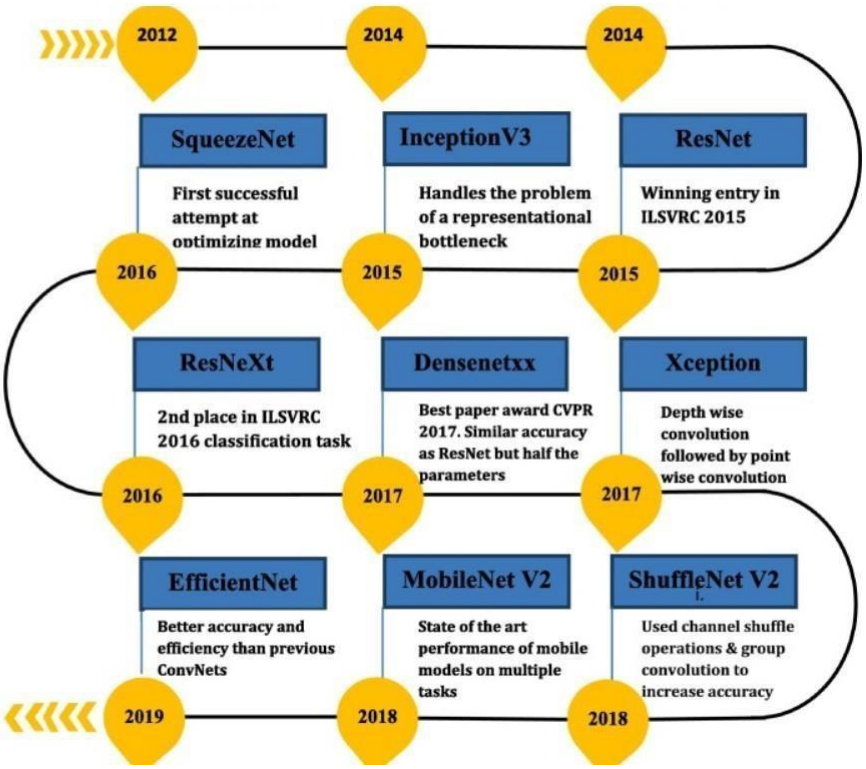


fig-8(It shows that sequence of every year for the better accuracy or multiple tasks.)

Face Mask Detector

Single shot detector structure can be used for object detection.In this device,face masks detector canbe evolved in lots of regions like shopping branch stores , airports and heavy website site visitors locations to screen the general public and to avoid the spread of the disease with the resource of checking who's following primary rule and who isn't always. It takes excessive time in load Google Colab Notebook.We re processed a device computionally efficient the usage of theMobileNetV2 which makes smooth to Extract the Data set.

1: We can use CNN structure for higher performance.

2: We can repair it any type of cameras

3: Each image is divided into three binss 0f 1*1,2*2,3*3 size.

4: We can use that for VGG-f for v-support regression.

5: We can use MobileNet model for predicte the class of images and the fast R- CNN model on ResNet50 for predicte hard image.

Training & Loading :

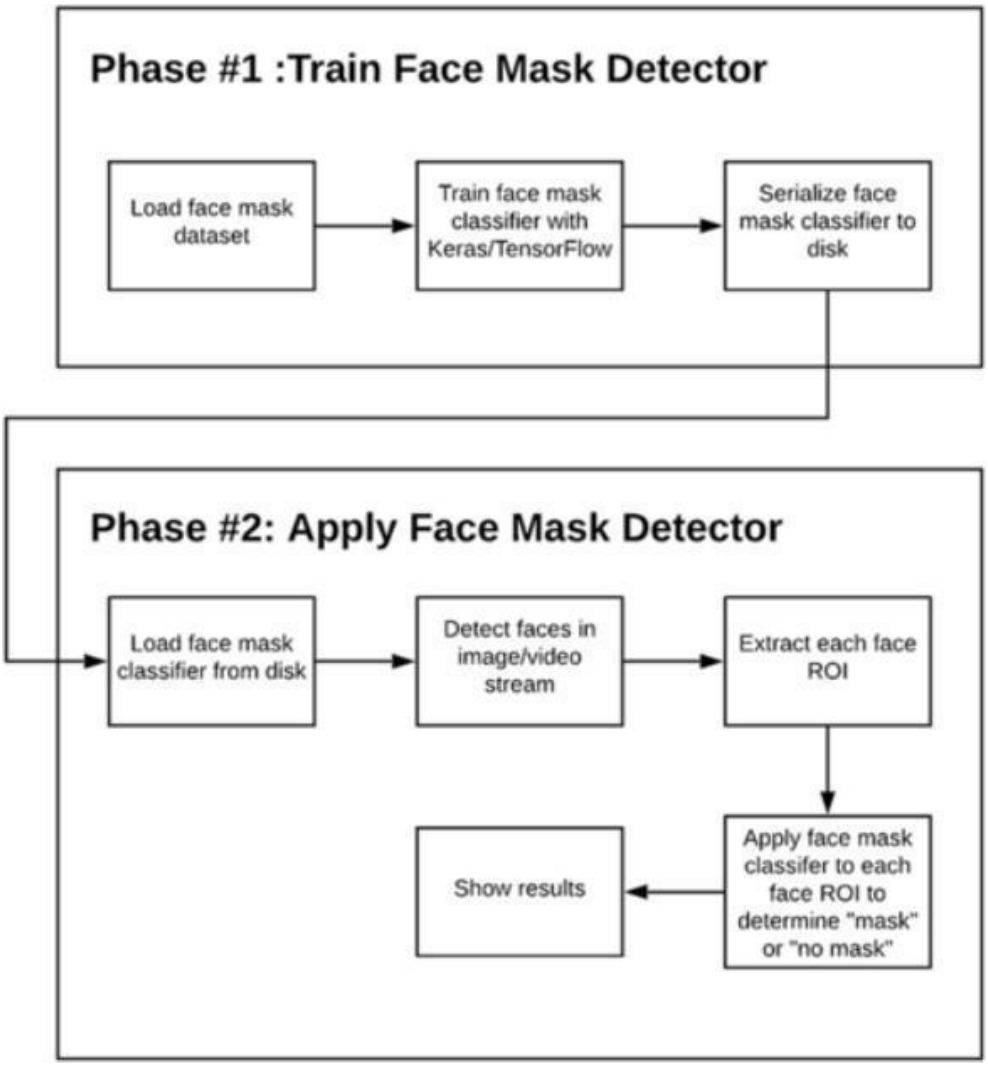


fig-9(It shows the training of data face masks and non-masks. We train the data with Keras and tenserflow)

Comparison between MobileNet-SSD, ResNet50 and Their Various Combinations based on Random vs. Hard/Soft Complexity of Test Data.

Comparison Parameters	MobileNet-SSD to ResNet50 (Left to Right)				
	100–0%	75–25%	50–50%	25–75%	0–100%
Random split (mAP)	0.8868	0.9095	0.9331	0.9650	0.9899
Soft/hard split (mAP)	0.8868	0.9224	0.9631	0.9892	0.9899
Image complexity prediction time (ms)	–	0.05	0.05	0.05	–
Mask detection time (ms)	0.05	1.92	3.08	5.07	6.02
Total Computation Time (ms)	0.05	1.97	3.13	5.12	6.02

fig-10(We compare image complexity prediction time b/w Mobile NetSSD to ResNet50)

We compare all the data with various test data parameters.

Design of Problem Statement:

After detect the face with wear masks and not wear masks in searches ,the unweared masked face are passed separate to a neural networks for the further explorate of the person safety and to violate facemask rules.These steps required a fixed sizes and inputs. Only one way to get fixed sizes inputs to reshape the face into boxes 96*96 pixel.

The system is designed up by the load different pretrained model uses the torch vsion packages. These model are fine tuned on to datset with images.

:Created a Face Mask Detector includes data and collection of the person images.

Our system can be used in real time software which require face- face detection for protection purposes.

: Working at the dataset and preparing it for records analysis.

This model used MobileNetV2 architecture and it is also efficient it's miles less difficult to install the version to embedded system like raspeb -erry Pi,Google Coral, and so forth.

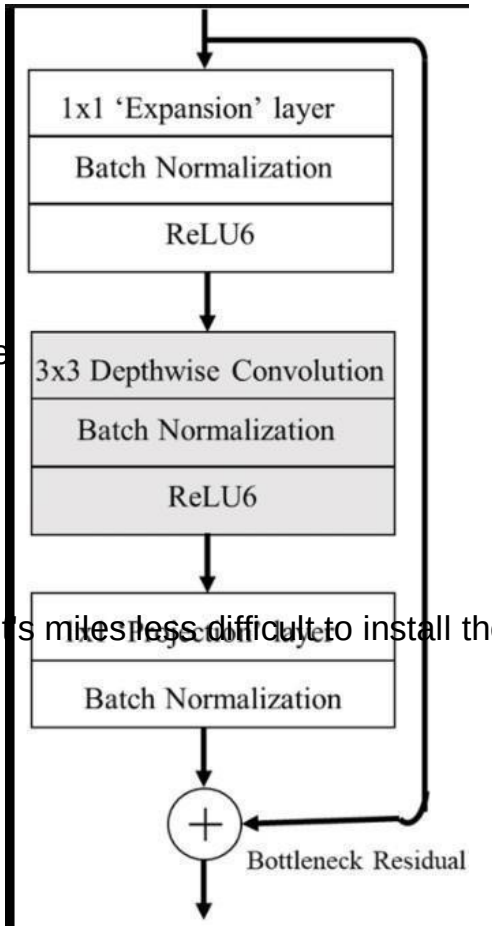
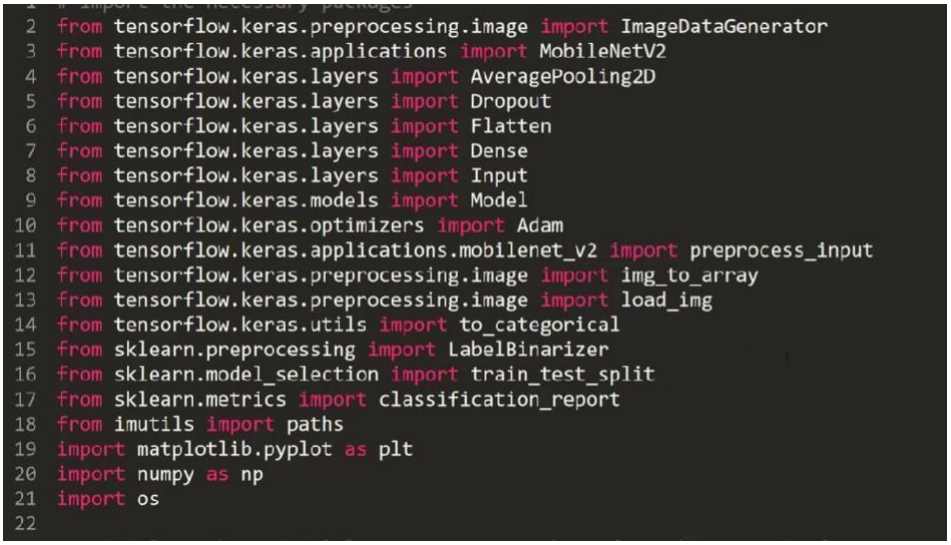


fig-11(This model uses MobileNet V2 architecture)

Project with screenshots of various stages of projects:

1 Data Preprocessing:

Before we begin the analysis, we will adjust the data to ensure accurate and correct analysis.



```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.layers import AveragePooling2D
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Input
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.utils import to_categorical
from sklearn.preprocessing import LabelBinarizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from imutils import paths
import matplotlib.pyplot as plt
import numpy as np
import os
```

fig-12(We preprocessed the data files with masks-non-face masks)

```
print("[INFO] loading images...")

data = []
labels = []

for category in CATEGORIES:
    path = os.path.join(DIRECTORY, category)
    for img in os.listdir(path):
        img_path = os.path.join(path, img)
        image = load_img(img_path, target_size=(224, 224))
        image = img_to_array(image)
        image = preprocess_input(image)

        data.append(image)
        labels.append(category)

# perform one-hot encoding on the labels
lb = LabelBinarizer()
labels = lb.fit_transform(labels)
labels = to_categorical(labels)

data = np.array(data, dtype="float32")
```

fig-13(In this fig-13 we print the datasets.)

2> MobileNet V2 :

It is a enormously green architecture that may be embedded devices with restricted computationalcapacity(Raspberry pi , Google Coral, NVIDIA Jetson Nano,and so on).

Deploying to our face mask detector to embedded gadgets should decrease the value of producing such face masks detection systems. Hence why we select to apply this architecture.

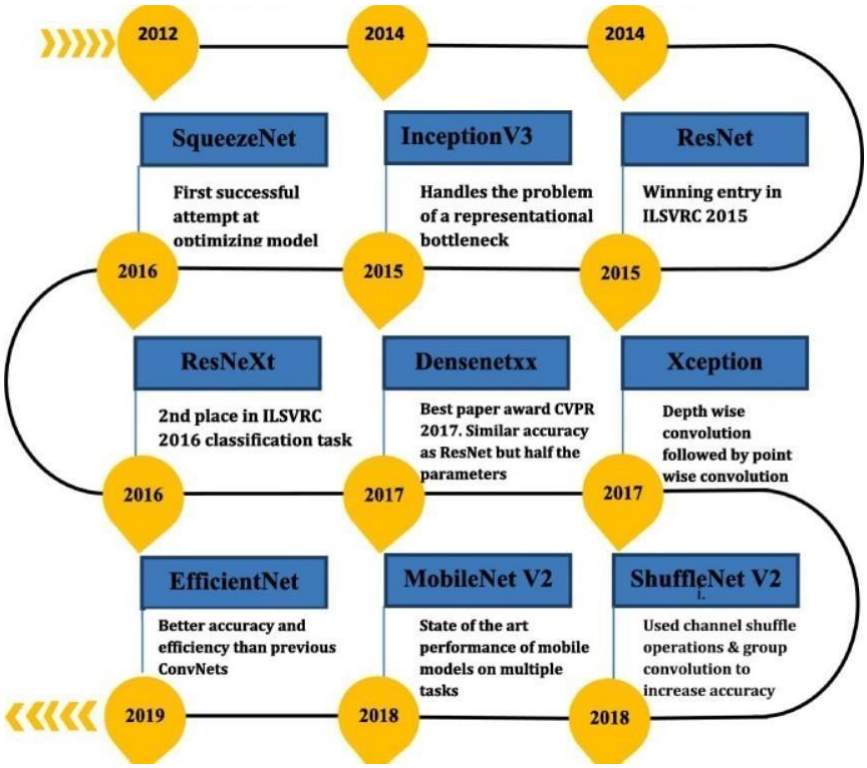


fig-15(In our model we use MobileNet V2 for the better performance in our model)

Command Line Arguments:
: Our command line arguments that are required to launch our script into a terminal.
Our command line arguments protected:
: Dataset : The course is the enter dataset of faces and faceswith masks.

: Plot: The direction to our output educate history plotn which willgenerate.

: Model: the path is the resulting serialized face mask type version.

```
print("[INFO] loading images...")
imagePaths = list(paths.list_images(args["dataset"]))
data = []
labels = []

# loop over the image paths
for imagePath in imagePaths:
    # extract the class label from the filename
    label = imagePath.split(os.path.sep)[-2]

    # load the input image (224x224) and preprocess it
    image = load_img(imagePath, target_size=(224, 224))
    image = img_to_array(image)
    image = preprocess_input(image)

    # update the data and labels lists, respectively
    data.append(image)
    labels.append(label)

# convert the data and labels to NumPy arrays
data = np.array(data, dtype="float32")
```

fig-16(In the fig, we use command line arguments for the dataset,plot& model)

Chapter 4: Performance Analysis

Tenser flow : It is a Google team-designed software library or framework for the easy execution of engineering and profound learning concepts.

The computational optimising algebra combines many mathematical expression for easy calculation.

Important features of Tenser Flow:

: It contain a feature that easily define, optimises and calculate mathematical expression.

: It includes the deep neural networks and machineteaching.

: It also includes highly scable computing feature with different data sets.
: TensorFlow uses automated management, GPUcomputing.

Keras:

It is a development kit which offered an artificial neural network python interface. The TensorFlowlibrary interface is Keras. Keras has supported several backends until the version 2.3 such as Tenseerflow, Microsoft Cognitive toolkit,Theano and PlaidML.Conceived to allowed for the fast experiment with profound neural networks.

:It is user-friendly,modular and expandable.

:It was develop as part of research effort of "ONEIROS" project and its principal author

:For the import of different image preprocessing features, we used the Tenserflow library.

The Tenserflow Keras interface is also used for MobileNets and MTCNN algorithms.

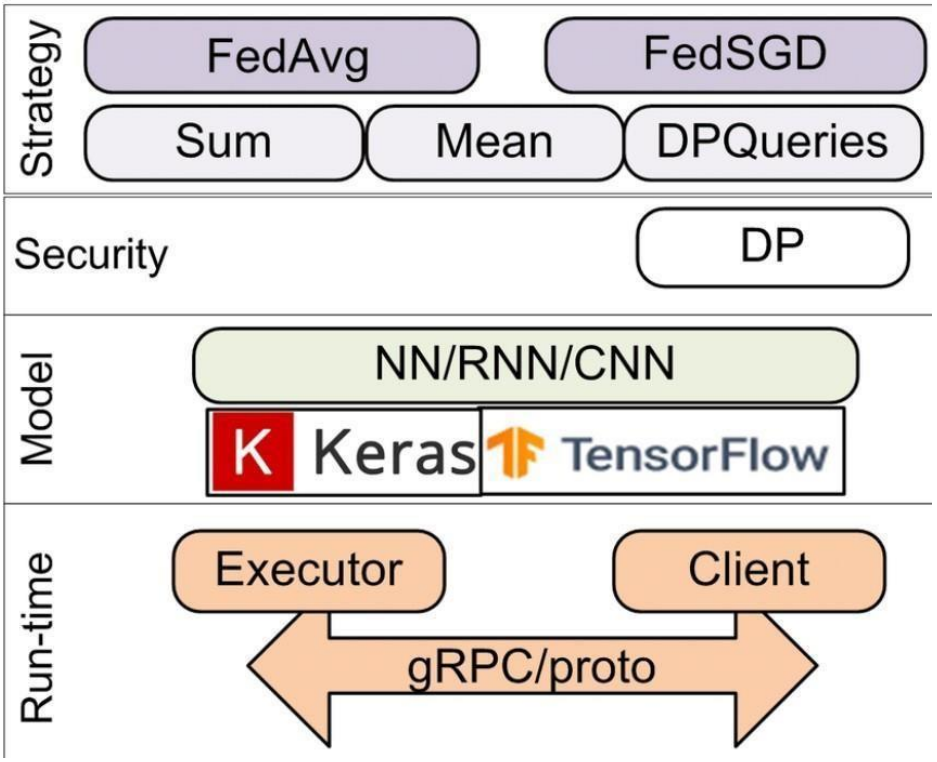


fig-17(It shows working of Keras & tenserflow ,its runtime executor and strategy.)

Fine tuning of Pretrained model

In this work, we can achieve facemask detection by adeep neural network because of its good performance than others classification algos.
It is is expensive.

Because is a time consumed task and required the high power to trains with the networks faster.

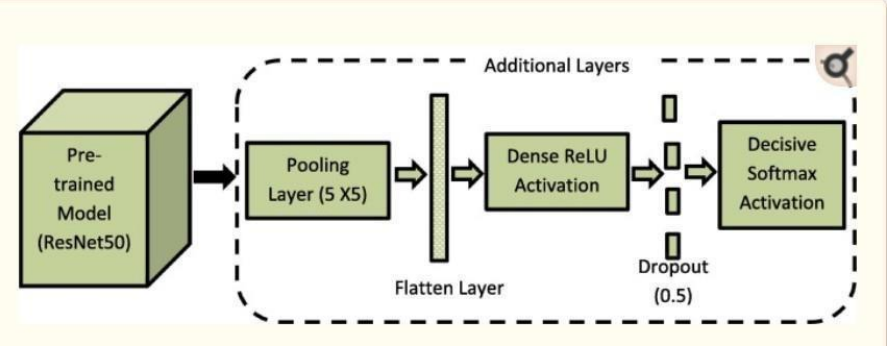


fig-18(This shows how we done fine tuning in our model)

Supervises the pre training with the domain specific fine tuning

Firstly removed inherit biases presentin the datasets and executes itwith supervisedlearning over a specific balanced datasets.

This formula used for computing the the imbalance ratio:

$$\rho = \frac{Count\left(majority\left(D_i\right)\right)}{Count\left(minority\left(D_i\right)\right)}$$

Real time analysis flowchart:

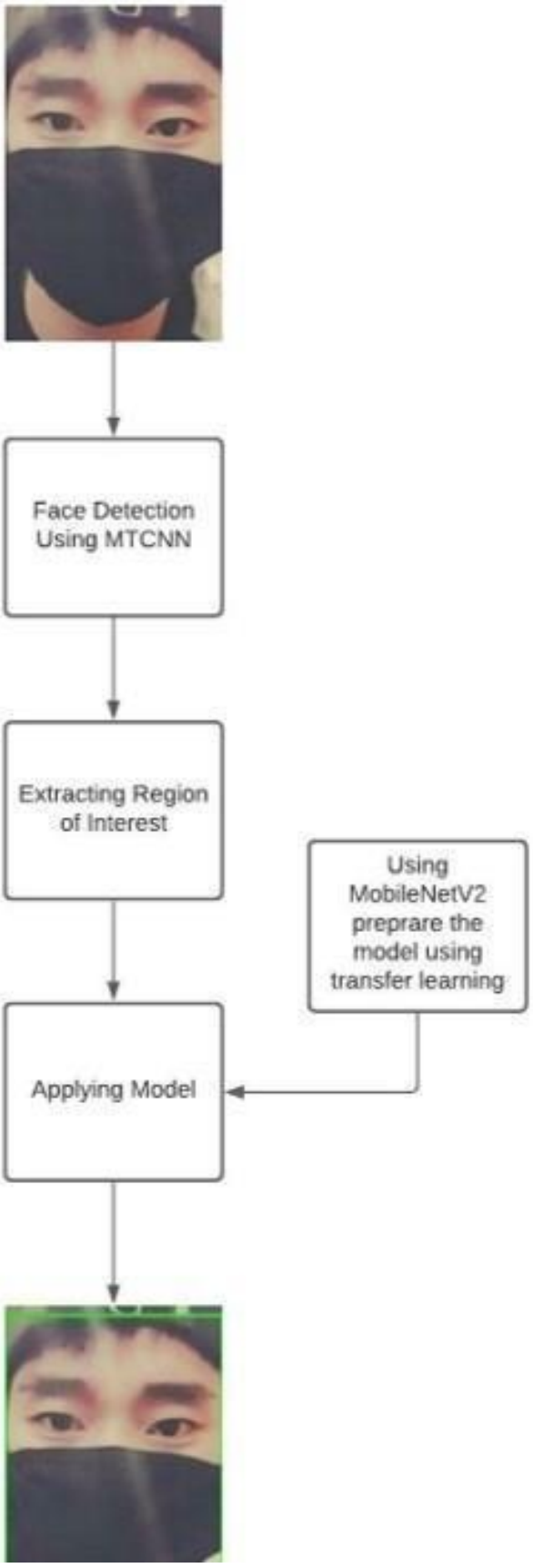


fig-19(This is the real time analysis it shows our model works and how its done.)

We are now ready ready to compile and train our face mask detector:

```
# compile our model
print("[INFO] compiling model...")
opt = Adam(lr=INIT_LR, decay=INIT_LR / EPOCHS)
model.compile(loss="binary_crossentropy", optimizer=opt,
    metrics=["accuracy"])

# train the head of the network
print("[INFO] training head...")
H = model.fit(
    aug.flow(trainX, trainY, batch_size=BS),
    steps_per_epoch=len(trainX) // BS,
    validation_data=(testX, testY),
    validation_steps=len(testX) // BS,
    epochs=EPOCHS)
```

fig-20(we comple our model and find the accuracy and print the training head.)

We will evaluate the resulting model on test set:

```
print("[INFO] evaluating network...")
predIdxs = model.predict(testX, batch_size=BS)

# for each image in the testing set we need to find the index of the
# label with corresponding largest predicted probability
predIdxs = np.argmax(predIdxs, axis=1)

# show a nicely formatted classification report
print(classification_report(testY.argmax(axis=1), predIdxs,
    target_names=lb.classes_))

# serialize the model to disk
print("[INFO] saving mask detector model...")
model.save(args["model"], save_format="h5")
```

fig-21(we print each image in the testing sets and serialize the model to disk.)

Our last step is to plot our accuracy and loss curves:

```
# plot the training loss and accuracy
N = EPOCHS
plt.style.use("ggplot")
plt.figure()
plt.plot(np.arange(0, N), H.history["loss"], label="train_loss")
plt.plot(np.arange(0, N), H.history["val_loss"], label="val_loss")
plt.plot(np.arange(0, N), H.history["accuracy"], label="train_acc")
plt.plot(np.arange(0, N), H.history["val_accuracy"], label="val_acc")
plt.title("Training Loss and Accuracy")
plt.xlabel("Epoch #")
plt.ylabel("Loss/Accuracy")
plt.legend(loc="lower left")
plt.savefig(args["plot"])
```

fig-22(We plot our training loss and accuracy)

ACCURACY:

- Training and Testing accuracy VS EPOCHS

fig-23(This is the final result of our training and accuracy test)

Chapter 5: Conclusion

We can use face mask detection for the purpose of people to enforced people to wears a mask in the public platforms, social gathering events. In the end its only our loved ones health and to protect them from this pandemic.

Applications:

Because of coronavirus,we can use face mask detection

to control th spread of coronavirus. As a technology advanceswe try to give the cost effective and accurate diagonosis.

Limitation of the Project:

We tried several machine learning methods in the project, but we still can't obtain very good accuracy and due to lack of data.

Future Work:

Finally, this works will be opened for the future planned for the countries to apply the face mask detection.In the public platforms, to get control in the spread of the covid. This system integrates with the highly resolutions surveillance devices.This system is a great for the future.

References:

1. World Health Organization et al. Coronavirus disease 2019 (covid-19): situation report, 96. 2020. - Google Search. (n.d.). https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200816-covid-19-sitrep-209.pdf?sfvrsn=5dde1ca2_2.

2. DataSet Kaggle:
https://www.kaggle.com/omkargurav/face-mask-dataset

3. Garcia Godoy L.R. Facial protection for healthcare workers during pandemics: a scoping review, BMJ. *Glob. Heal.* 2020;5(5) doi: 10.1136/bmjgh-2020-002553. [PMC free article] [PubMed] [CrossRef] [Google Scholar]

4. Eikenberry S.E. To mask or not to mask: Modeling the potential for face mask use by the general public to curtail the COVID-19 pandemic. *Infect. Dis. Model.* 2020;5:293–308. doi: 10.1016/j.idm.2020.04.001. [PMC free article] [PubMed] [CrossRef] [Google Scholar]

Appendices:

```python
# import the necessary packages
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.layers import AveragePooling2D
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Input
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.utils import to_categorical
from sklearn.preprocessing import LabelBinarizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from imutils import paths
import matplotlib.pyplot as plt
import numpy as np
import os

# initialize the initial learning rate, number of epochs to train for,
# and batch size
INIT_LR = 1e-4
EPOCHS = 20
BS = 32
```

```
28
29  DIRECTORY = r"C:\Mask Detection\CODE\Face-Mask-Detection-master\dataset"
30  CATEGORIES = ["with_mask", "without_mask"]
31
32  # grab the list of images in our dataset directory, then initialize
33  # the list of data (i.e., images) and class images
34  print("[INFO] loading images...")
35
36  data = []
37  labels = []
38
39  for category in CATEGORIES:
40      path = os.path.join(DIRECTORY, category)
41      for img in os.listdir(path):
42          img_path = os.path.join(path, img)
43          image = load_img(img_path, target_size=(224, 224))
44          image = img_to_array(image)
45          image = preprocess_input(image)
46
47          data.append(image)
48          labels.append(category)
49
50  # perform one-hot encoding on the labels
51  lb = LabelBinarizer()
52  labels = lb.fit_transform(labels)
53  labels = to_categorical(labels)
54
```

```
data = np.array(data, dtype="float32")
labels = np.array(labels)

(trainX, testX, trainY, testY) = train_test_split(data, labels,
        test_size=0.20, stratify=labels, random_state=42)

# construct the training image generator for data augmentation
aug = ImageDataGenerator(
        rotation_range=20,
        zoom_range=0.15,
        width_shift_range=0.2,
        height_shift_range=0.2,
        shear_range=0.15,
        horizontal_flip=True,
        fill_mode="nearest")

# load the MobileNetV2 network, ensuring the head FC layer sets are
# left off
baseModel = MobileNetV2(weights="imagenet", include_top=False,
        input_tensor=Input(shape=(224, 224, 3)))

# construct the head of the model that will be placed on top of the
# the base model
headModel = baseModel.output
headModel = AveragePooling2D(pool_size=(7, 7))(headModel)
headModel = Flatten(name="flatten")(headModel)
headModel = Dense(128, activation="relu")(headModel)
headModel = Dropout(0.5)(headModel)
headModel = Dense(2, activation="softmax")(headModel)

# place the head FC model on top of the base model (this will become
# the actual model we will train)
model = Model(inputs=baseModel.input, outputs=headModel)
```

```
# loop over all layers in the base model and freeze them so they will
# *not* be updated during the first training process
for layer in baseModel.layers:
	layer.trainable = False

# compile our model
print("[INFO] compiling model...")
opt = Adam(lr=INIT_LR, decay=INIT_LR / EPOCHS)
model.compile(loss="binary_crossentropy", optimizer=opt,
	metrics=["accuracy"])

# train the head of the network
print("[INFO] training head...")
H = model.fit(
	aug.flow(trainX, trainY, batch_size=BS),
	steps_per_epoch=len(trainX) // BS,
	validation_data=(testX, testY),
	validation_steps=len(testX) // BS,
	epochs=EPOCHS)

# make predictions on the testing set
print("[INFO] evaluating network...")
predIdxs = model.predict(testX, batch_size=BS)

# for each image in the testing set we need to find the index of the
# label with corresponding largest predicted probability
predIdxs = np.argmax(predIdxs, axis=1)

# show a nicely formatted classification report
print(classification_report(testY.argmax(axis=1), predIdxs,
	target_names=lb.classes_))

# serialize the model to disk
print("[INFO] saving mask detector model...")
model.save("mask_detector.model", save_format="h5")

```

```
124
125 # plot the training loss and accuracy
126 N = EPOCHS
127 plt.style.use("ggplot")
128 plt.figure()
129 plt.plot(np.arange(0, N), H.history["loss"], label="train_loss")
130 plt.plot(np.arange(0, N), H.history["val_loss"], label="val_loss")
131 plt.plot(np.arange(0, N), H.history["accuracy"], label="train_acc")
132 plt.plot(np.arange(0, N), H.history["val_accuracy"], label="val_acc")
133 plt.title("Training Loss and Accuracy")
134 plt.xlabel("Epoch #")
135 plt.ylabel("Loss/Accuracy")
136 plt.legend(loc="lower left")
137 plt.savefig("plot.png")
```

```python
# import the necessary packages
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
from imutils.video import VideoStream
import numpy as np
import imutils
import time
import cv2
import os

def detect_and_predict_mask(frame, faceNet, maskNet):
        # grab the dimensions of the frame and then construct a blob
        # from it
        (h, w) = frame.shape[:2]
        blob = cv2.dnn.blobFromImage(frame, 1.0, (224, 224),
                (104.0, 177.0, 123.0))

        # pass the blob through the network and obtain the face detections
        faceNet.setInput(blob)
        detections = faceNet.forward()
        print(detections.shape)

        # initialize our list of faces, their corresponding locations,
        # and the list of predictions from our face mask network
        faces = []
        locs = []
        preds = []

```

```
        # loop over the detections
        for i in range(0, detections.shape[2]):
                # extract the confidence (i.e., probability) associated with
                # the detection
                confidence = detections[0, 0, i, 2]

                # filter out weak detections by ensuring the confidence is
                # greater than the minimum confidence
                if confidence > 0.5:
                        # compute the (x, y)-coordinates of the bounding box for
                        # the object
                        box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
                        (startX, startY, endX, endY) = box.astype("int")

                        # ensure the bounding boxes fall within the dimensions of
                        # the frame
                        (startX, startY) = (max(0, startX), max(0, startY))
                        (endX, endY) = (min(w - 1, endX), min(h - 1, endY))

                        # extract the face ROI, convert it from BGR to RGB channel
                        # ordering, resize it to 224x224, and preprocess it
                        face = frame[startY:endY, startX:endX]
                        face = cv2.cvtColor(face, cv2.COLOR_BGR2RGB)
                        face = cv2.resize(face, (224, 224))
                        face = img_to_array(face)
                        face = preprocess_input(face)

                        # add the face and bounding boxes to their respective
                        # lists
                        faces.append(face)
                        locs.append((startX, startY, endX, endY))

```

```
        # only make a predictions if at least one face was detected
        if len(faces) > 0:
                # for faster inference we'll make batch predictions on *all*
                # faces at the same time rather than one-by-one predictions
                # in the above `for` loop
                faces = np.array(faces, dtype="float32")
                preds = maskNet.predict(faces, batch_size=32)

        # return a 2-tuple of the face locations and their corresponding
        # locations
        return (locs, preds)

# load our serialized face detector model from disk
prototxtPath = r"face_detector\deploy.prototxt"
weightsPath = r"face_detector\res10_300x300_ssd_iter_140000.caffemodel"
faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)

# load the face mask detector model from disk
maskNet = load_model("mask_detector.model")

# initialize the video stream
print("[INFO] starting video stream...")
vs = VideoStream(src=0).start()

```

```
# loop over the frames from the video stream
while True:
	# grab the frame from the threaded video stream and resize it
	# to have a maximum width of 400 pixels
	frame = vs.read()
	frame = imutils.resize(frame, width=400)

	# detect faces in the frame and determine if they are wearing a
	# face mask or not
	(locs, preds) = detect_and_predict_mask(frame, faceNet, maskNet)

	# loop over the detected face locations and their corresponding
	# locations
	for (box, pred) in zip(locs, preds):
		# unpack the bounding box and predictions
		(startX, startY, endX, endY) = box
		(mask, withoutMask) = pred

		# determine the class label and color we'll use to draw
		# the bounding box and text
		label = "Mask" if mask > withoutMask else "No Mask"
		color = (0, 255, 0) if label == "Mask" else (0, 0, 255)

		# include the probability in the label
		label = "{}: {:.2f}%".format(label, max(mask, withoutMask) * 100)

```

```
111
112                 # display the label and bounding box rectangle on the output
113                 # frame
114                 cv2.putText(frame, label, (startX, startY - 10),
115                         cv2.FONT_HERSHEY_SIMPLEX, 0.45, color, 2)
116                 cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)
117
118         # show the output frame
119         cv2.imshow("Frame", frame)
120         key = cv2.waitKey(1) & 0xFF
121
122         # if the `q` key was pressed, break from the loop
123         if key == ord("q"):
124                 break
125
126 # do a bit of cleanup
127 cv2.destroyAllWindows()
128 vs.stop()
```

```
tensorflow>=1.15.2
keras==2.3.1
imutils==0.5.3
numpy==1.18.2
opencv-python==4.2.0.*
matplotlib==3.2.1
scipy==1.4.1
```