FACE MASK DETECTION

Submitted by:

213244 Santra Johny

213204 Agy Sunny

Under the guidance of:

DR MANOJ KUMAR T K

TABLE OF CONTENTS

1. Abstract **3**

2. Introduction **4**

3. Problem statement **6**

4. Literature review **7**

5. Methodology **9**

5.1 Haar cascade classifier

5.2 Code

5.2.1 Code 1

5.2.2 Code 2

5.3 Theory behind implementation of code

5.4 Method of detection and classification task currently used

6. Algorithms used **19**

6.1 Logistic Regression

6.2 Support vector machine

6.3 K – nearest neighbours

6.4 Naïve Bayes

6.5 Decision Tree

6.6 Random Forest

6.7 Adaptive boosting

6.8 Gradient boosting

7. Results and evaluation **25**

8. References **27**

1. ABSTRACT

2020 was a game changing year because of the pandemic. It was hard for each one of us. Introverts enjoyed, ambiverts managed and extroverts had a really hard time. Many had issues finding a job and some had to shut down their business due to heavy loss. Apart from all these problems there was a particular statement that made many raise their eyebrows: “Wearing a mask and following proper social distancing can help in decreasing the spread of Corona virus or COVID-19”. It was seen that face masks help to reduce the spread of the virus by 17%. Why was it difficult? Humans communicate not just with their mouth but hand gestures, head movements, body movements etc are used as well. Even though the number of COVID-19 cases were rising people found it hard to comply with these rule. This affected the functionality of many organisations, schools, offices etc as they had to shut down due to the rising cases and in turn affecting their business.

Lately, researchers have found that the usage of masks has not only reduced the risk of corona virus but has also decreased the chances of being affected by diseases that are transmitted through air. It can also protect people from pollution. Some countries like Japan and Singapore have been using masks in public for a longer period of time. As of Indians we started using them in the pandemic. This has resulted in reduced number of tuberculosis patients and other respiratory viruses.

It was scientifically proven that wearing a mask can help reduce the transmission rate. With a population of 775.28 crores in the world it will be extremely hard to manually check if everyone is wearing their mask or not. There are many existing techniques to help automate mask-checking using deep learning models which is done during live security camera check. The current prominent technique uses Mask-R-CNN and ResNet as backbone. In this paper we are trying to achieve a better method with will be much easier to use.

2. INTRODUCTION

Death is scary and you cannot control it. But what if could avoid it, both yours' and the ones around you. The key to life was no medicines or machines but something as thin as a paper – a mask. Wearing a mask protects us from the virus. But living in a busy world people tend to forget it, they found it uncomfortable as they were not used to it. Making it a rule became something unavoidable. This is why there came a necessity to check if a person is wearing one or not. Doing this manually with a population of almost 755 crores is nearly impossible, hence the need of a model or an algorithm.

Machine learning, comes under artificial intelligence, has the ability to learn automatically from its past experiences. For simplicity we can consider machine learning as a combination of both mathematics and programming. Machine learning can be classified into supervised learning and unsupervised learning. Supervised learning is further classified into regression and classification. Under classification we have Logistic regression, Support Vector Machine, Decision Tree and many more.

In this project we will be focusing on supervised learning predominantly on predictive analysis. On considering face recognition, non-machine learning methods have a lower rate of prediction in comparison with machine learning techniques. Also, there are many evident results that shows machine learning techniques for facial recognition to work better on large dataset with high dimensionality compared to non-machine learning techniques. Predicting whether a person is wearing or not can be helpful for many organizations. Once someone is detected, they can be asked to wear one. This will develop a sense of responsibility in each individual. Getting someone to do this manually can be really hard as a continuous evaluation is not possible.

These machine learning models can help in the detection of masks which will in turn reduce the spread of Corona virus. Various models like Logistic regression, Gaussian Naïve Bayes, k-nearest neighbours, SVM and Random Forest are used.

3. PROBLEM STATEMENT:

Wearing a mask is far from a choice to a necessity, as evidenced by the fact that the COVID-19 cases are increasing in places where there is no rule as such. Even though vaccines are given, being cautious is important as there is no such vaccine which is hundred percent effective, turning it into a global issue. As the old saying goes: “Prevention is better than cure”. Vaccines can help reduce the chances of you being affected but you are still at risk. This is why the usage of masks is made compulsory. Doing this manually is tiresome and not practical. Machine learning techniques can be used to train the model and this information can be utilised for predicting. We focus on supervised learning under which classification models like Logistic Regression, Decision Trees, KNN, Random Forest, SVM are used for this purpose.

4. LITERATURE REVIEW

Here, we summarize the inferences gained from previously referred research papers and thesis papers.

Vani Perumal [1] published a paper, which presents Face Recognition in Video Streams and its Application in Freedom Fighters Discovery using Machine Learning Approach. He uses the technique of Haar feature-based cascade classifiers for face recognition. Here algorithms are used for extracting features. The main objective of this paper is to identify the faces of Indian Freedom Fighters from any video stream given as an input. The training dataset contains many trainings video streams such that a particular video stream only contains images of a specific freedom fighter in different stages and different styles. He made a training dataset which embrace 25 different Indian freedom fighters. The trained faces of freedom fighters were displayed along with their names and untrained human faces were predicted as unknown.

Ren Liu and Ziang Ren [2] published a paper on Application of Yolo on Mask Detection Task. This paper has high social relevance because transmission and reduction of COVID -19 pandemic can be effectively reduced by strict mask wearing polices. Mask-R-CNN with ResNet as backbone is the current dominant mask detection method. The main idea of this paper is to replace current dominant mask detection method with more efficient and accurate method YOLO. The dataset is acquired from Kaggle, which contains 853 images belonging to 3 main classes with mask, without mask and mask worn incorrectly with their bounding boxes in PASCAL VOC format. The result was little confusing that faster R-CNN tends to perform better than YOLO, still with a little loss in accuracy YOLO has a much better running speed.

Ross Girshick [3] published a paper on Fast R-CNN. This paper highlights on a Fast Region-based Convolutional Network method (Fast R-CNN) for object detection. There are certain drawbacks for R-CNN such as Training is a multi-stage pipeline, Training is expensive in space and time and Object detection is slow. In this paper author tries to overcome the disadvantages of R-CNN along with increasing accuracy and speed through Fast R-CNN. He used three pre-trained ImageNet models that are available online. This paper proposes Fast R-CNN as a clean and fast update to R-CNN. Broadly there are two types of object detectors, those one-use sparse objects and those one-use dense objects. Sparse object proposals appear to improve detector quality. Author claims that even though it was too costly in the past, it appears to be cost effective with Fast R-CNN.

5. METHODOLOGY

In this paper, we have come up with two different codes to find if a person has worn a mask or not. We aim to help various organisations to do this task easily. We have tried to reduce the complexity of pre-existing codes and come up with a simpler method.

5.1 Haar Cascade:

The foremost step to identification starts by looking at the face. Facial recognition is found everywhere starting from security cameras to the sensor on your phone. How do you think this identification is done, considering all these similar features among human beings? This is where a haar cascade classifier comes in handy.

Haar cascade classifier or haar classifier is a machine learning program to detect various objects in a given video or image. It was first proposed by Paul Viola and Michael Jones. It requires lots of positive-images and negative-images. Positive images are those that contain images of face and the one's without it are negative image. These images are used for training and testing.

The algorithm can be divided into four steps:

(i) Finding Haar Features

(ii) Creating Integral Images

(iii) Using Adaboost

(iv) Implementing Cascading Classifiers

Once the training is done the features has to be extracted. These features are known as Haar features. These features are like convolution kernels. They obtained by subtracting the sum of white triangles from the sum of black triangles. The single value obtained from this subtraction is used for recognition. Examples

of Haar features are given below. The black region has a value of 1 and the white has value 0.

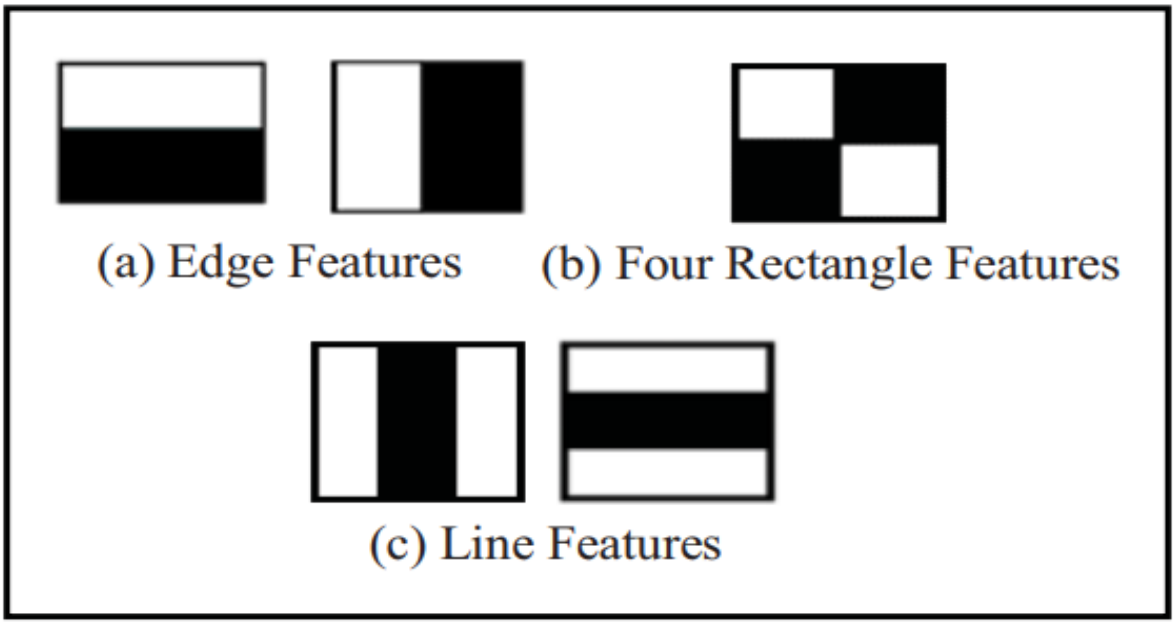


Fig.5.1

Pre-processing steps involves:

Before the application of cascade classifier, the input image or video is divided into frames each of which is resized into a specific size. Then each frame is converted from RGB to Gray.

Once the gray scaling is done, you'll have a matrix like structure with each pixel of a certain value ranging from 0 to 1. You take a rectangular portion and do the calculation mentioned above. If the image has a line that separates dark pixels on the top and light pixels on the bottom, then the haar values is closer to 1. This means that there is an edge that was detected and by looking at the Fig.5.1. we can say it is a line feature. Other haar features will detect other edges on other parts of the image which results in the traversing of the haar feature through the entire

image. It traverses from top left to bottom right, during which all possible sizes of the haar feature will be applied.
The first set of two rectangular features finds out the edges in either a horizontal direction or a vertical direction.
The second set of three rectangular features if there is a darker region surrounded by a lighter region or vice-versa
The third set of four rectangular features finds out the change in pixel intensities along the diagonal.

To find the best Haar feature a boosting technique called Adaboost is used. Haar-cascade classifier is quite fast but not so accurate compared to the modern techniques but yet is simple to use. The open-cv python library contains a repository that has various haar cascades such as face detection, mouth detection, eye detection etc. We are using “haarcascade_frontalface_default.xml”. This file is downloaded and loaded using cv2.CascadeClassifier.

5.2 .1 CODE 1:

For the process of visualising the images open CV is taken into account.

The module import name of open CV is cv2. It is a python library which helps in visualising and reading of images. It mainly deals with computer vision problems.

Function used are:

cv2.imread() - loads image from the specified path.

- If the image cannot be found it returns an empty matrix.

cv2.imshow() - used to display an image in the window

cv2.waitKey() - used to display the image for a millisecond

- If '0' is given as a parameter the image gets displayed until a specified key is pressed.

cv2.destroyAllWindows()

- It closes all windows after exiting the code.

cv2.COLOR_BGR2GRAY

- It is used to convert the normal RGB colour to gray colour.

cv2.CascadeClassifier(<path of the haar-cascade xml file>)

- It is used to load any haar-cascade xml file.

detectMultiScale - It detects various objects with the help of the xml file and returns a rectangular box on the

positions of faces.

for (x,y,w,h) in faces:

- x, y, w, h will loop through each rectangle (detects the face).
- x is the horizontal initial position and y the vertical initial position.
- w and h are width and height respectively.

Libraries used are:

i. cv2
ii. matplotlib
iii. numpy
iv. sklearn

THEORY BEHIND IMPLEMENTATION OF CODE:

Basic pre-processing on the data is done like loading a random image and displaying it. The image is loaded into a variable called 'img'. By importing matplotlib.pyplot we display the image (plt.imshow(img)).This is demonstrated in the figure given below:

```
In [2]: img=cv2.imread(r"C:\Users\santr\PycharmProjects\MINIPROJECT\RESOURCES\masked elon.jpg")

In [3]: img.shape
Out[3]: (900, 1600, 3)

In [4]: import matplotlib.pyplot as plt

In [5]: plt.imshow(img)
Out[5]: <matplotlib.image.AxesImage at 0x1e73a6e1850>
```

Two .npy file were created namingly “with mask.npy” and “without mask.npy”. It is then assigned to variables with_mask and without_mask respectively. Here instead of using a dataset which contains random images of people with mask and without mask we use the webcam to capture various images (length set to 200) of the person with mask and then another set of 200 images without mask. These images are then saved into the .npy files respectively, giving you a total of 400 images of which 200 are with mask and 200 without. On looking at the shape of with_mask you get (200, 50, 50, 3). This is of dimension 4 and hence we reshape it to 2-D.

In order to train this dataset, we need an independent variable and a dependant variable. The independent variable(x) is formed by concatenating both with_mask and without_mask. The dependant variable y is created using an array of 0’s and 1’s of which 0 represents with mask and 1 represent without mask. Using train_test_split the data is split for training and testing. Principal component analysis is used to reduce the dimension of both x_train and x_test. Various models like SVM, logistic regression, K-nn, Naïve Bayes, Decision Tree, Random Forest, Gradient Boost and Adaboost are used. The model which shows highest accuracy is chosen for prediction. The detected face from the webcam is given as the input to the model which predicts if the person is wearing a mask or not.

5.2.2 CODE 2:

Function used are:

cv2.imread() - loads image from the specified path.

- If the image cannot be found it returns an empty matrix.

cv2.imshow() - used to display an image in the window

cv2.waitKey() - used to display the image for a millisecond

- If '0' is given as a parameter the image gets displayed until a specified key is pressed.

cv2.destroyAllWindows()

- It closes all windows after exiting the code.

cv2.COLOR_BGR2GRAY

- It is used to convert the normal RGB colour to gray colour.

cv2.CascadeClassifier(<path of the haar-cascade xml file>)

- It is used to load any haar-cascade xml file.

os.path.join() - It is used to concatenate two paths.

os.listdir() - It is used to convert the concatenated path in the form of a list.

np.array - to convert the image to an array format.

Libraries used are:

i. cv2
ii. matplotlib
iii. numpy
iv. sklearn
v. os

5.3 THEORY BEHIND IMPLEMENTATION OF CODE:

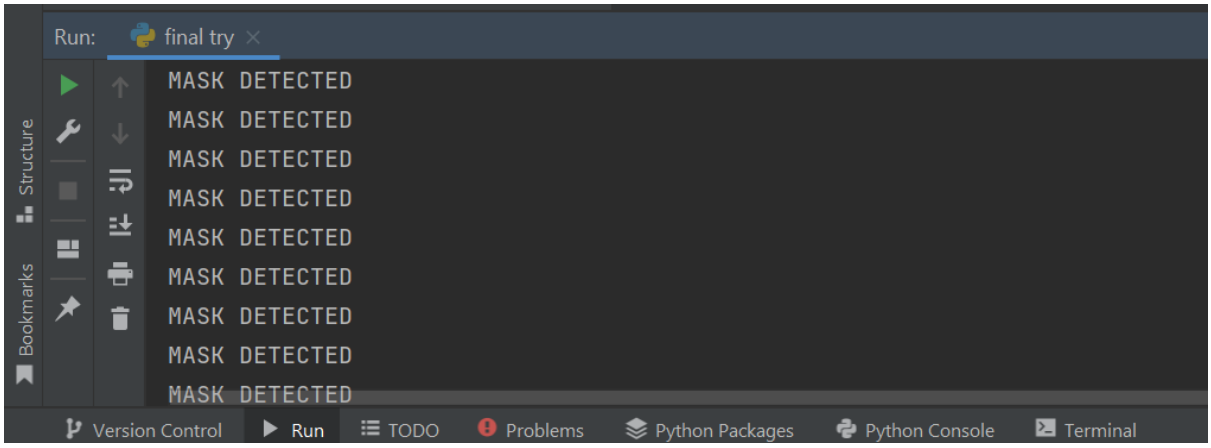
5.3.1 Dataset:

The dataset is downloaded from Kaggle which contains two folder namingly "wearing mask" and "not_wearing_mask". The "wearing mask" folder contains 1415 images of various people wearing mask and "not_wearing_mask" contains 407 images of people not wearing a mask.

5.3.2 Implementation:

After loading the necessary libraries, the path of the folder containing the dataset is assigned to a variable 'DIRECTORY' and the two folder names of with and without mask are put into a list 'CATEGORIES'. An empty list 'data' is created. We then initiate a loop traversing through the 'CATEGORIES' in which first, the path inside the "DIRECTORY" and CATEGORIES are concatenated. Then labels is created in which the index of the traversing variable is appended (that is 0 for with mask and 1 for without). It is then converted to a list and traversed through. After which each image is read and resized to a definite size and appended to the empty list 'data' along with the label. Then we create two lists x,y in which the features and labels in data are appended. After reshaping x, we use x and y for training and splitting and is fit into a model. The model we used is Logistic Regression. A webcam is used to capture the image which is in turn given as an

input to the model. This is then used for prediction. A prediction of this sought is obtained after wearing a mask:



5.4) Method of detection and classification task currently used:

The most common and comparatively efficient method used in real time mask detection is using:

(i) Fast R-CNN (Region-based Convolutional Network method)

(ii) YOLO

5.4.1) Fast R-CNN (Region-based Convolutional Network method):

Fast R-CNN was developed from pre-existing R-CNN. Region-based Convolutional Network method is a highly accurate method used for object detection with high accuracy. It uses a deep ConvNet to classify objects. What is ConvNet? ConvNet is nothing but convolutional neural network. It is a feed-forward neural network that is used to analyse visual images by processing data with grid-like topology. In ConvNet every input image is converted to an array format of pixel values. It uses the concept of convolution operation.

For example, let's understand the convolution operation on two matrices A and B.

Let A = [8,4,6,8,7,3,6] and B = [1,3,2]

In convolution operation, the arrays are multiplied element-wise, and the product is summed to create a new array, which represents a*b.

A & B	Multiply the array element wise	Sum the products.	A * B
A = [8,4,6,8,7,3,6] B = [1,3,2]	[8, 12, 12]	32	[32,]
A = [8,4,6,8,7,3,6] B = [1,3,2]	[4, 18, 16]	38	[32, 38]

This process continuous until all convolutional operations are done.

A fast R-CNN takes the entire image as the input also a set of object proposals (the model segments the image and assumes the object is inside that segment). It processes the entire image by using convolutional and pooling layers. The image is then resized which cause in losing some data. After which the CNN features are calculated and then the object is identified. This process are really very expensive and is slow – the frames per second performance is slow when it comes to live detection.

5.4.2) YOLOV3

YOLOV3 model was used to overcome these drawbacks, that is, to obtain higher accuracy and faster output in detection. The paper that used this method focused on mask detection – those who did not wear mask and those who wore their mask incorrectly was also spotted. It was much faster in comparison with the first case.

6) ALGORITHMS USED:

6.1 LOGISTIC REGRESSION

Logistic Regression is one of the machine learning models which comes under supervised learning. It is applicable for both regression and classification problems but it mainly focuses on classification. It is one of the most commonly used discriminative model for classification. Finding the maximum-likelihood linear classifier using the logistic model is called logistic regression. The hypothesis class associated with logistic regression is the composition of a sigmoid function. The name “sigmoid” means “S-shaped,” referring to the plot of this function. We get decision boundary directly in logistic regression. For example, if the classes are overlapping then logistic regression will tend to locate the decision boundary in an area where classes are maximally overlapping.

In contrast to linear regression, which predicts if something is continuous like height, logistic regression determines whether something is true or untrue. The categorical dependent variable is predicted using this approach, using the set of independent factors that has been provided. The outcome must be a categorical or discrete value since the model predicts the output of a categorical dependent variable. It has options like True or False, 0 or 1, and Yes or No.

The model we used is Logistic Regression. A webcam is used to capture the image which is in turn given as an input to the model. This is then used for prediction. Then the accuracy is calculated.

6.2 SUPPORT VECTOR MACHINE

Support vector machine is one of the standard tools for machine learning algorithms, the svm can be used for classification and regression problems, however it is most frequently employed for classification issues. It is applied for

learning linear predictors in high dimensional feature spaces. The high dimensionality of the feature space creates problems with sample complexity and computational complexity.

Recent developments in statistical learning theory are used to construct SVM. The goal of SVM is to identify the optimum decision boundary or line to divide n-dimensional space into classes so that fresh data points can be classified into the appropriate category. This judgement boundary is known as a hyperplane.

SVM can be classified into two linear SVM and non-linear SVM. Linear SVM functions for data that can be linearly separated in the feature space, it cannot be applied in many real-world circumstances. Despite this, it is the simplest algorithm to comprehend and serves as the foundation for the more complex Support Vector Machines. We use nonlinear SVM if we are unable to divide our dataset into two classes using a single straight line.

The model we used is SVM. A webcam is used to capture the image which is in turn given as an input to the model. This is then used for prediction. Then the accuracy is calculated.

6.3 K -NEAREST NEIGHBOUR

One of the fundamental Machine Learning algorithms that utilises the Supervised Learning technique is K-Nearest Neighbour. The k-NN rule is a very simple learning algorithm that relies on the assumption that “things that look alike must be alike.

Nearest Neighbour is a learning-by-memorization. The goal is to memorize the training set and then to predict the label of any new instance on the basis of the labels of its closest neighbours in the training set. Such an approach is justified on the grounds that nearby points are likely to have the same label because the

features that are used to describe the domain points are relevant to their labelling's.

In the K-NN technique, the input values are passed through the trained prior model to identify the item or predict the value. There are certain rules for selecting k in K-NN, Usually, k is chosen to be odd. There is no particular algorithm to decide the value of k other than trial and error. Large values of k smoothen the data but when the data is small choice of k shouldn't be large as it may out vote the data. Small values of k such as1 and 2 can be noisy and maybe affected by outliers

It requires the entire training data set to be stored, and at test time, we need to scan the entire data set in order to find the neighbours. So, it is also known as lazy learning algorithm.

The model we used is K-NN. A webcam is used to capture the image which is in turn given as an input to the model. This is then used for prediction. Then the accuracy is calculated.

6.4 NAÏVE BAYES CLASSIFIER

It is a supervised learning algorithm that relies on the Bayes theorem to solve classification problems. A well-known example of how generating assumptions and parameter estimations can make the learning process simpler. One of the best classification algorithms, because it is relatively easy to use and produces quick predictions on datasets.

It predicts on the basis of probability of given object so this classifier is also known as probabilistic classifier. It is naïve because we make the generative assumption that, the features are independent of each other.

It uses conditional probability and the Bayes theorem to operate. The following is the formula for the Bayes theorem:

$$P(A/B) = P(B/A)\ P(A) / P(B)$$ ----- Probability of A given B

Here, we want to determine the probability of a specific condition given a specific condition.

In Naïve Bayes, we have a dataset with pre-existing information, the input data is then compared with this dataset and then classified.

The model we used is Naïve Bayes Classifier. A webcam is used to capture the image which is in turn given as an input to the model. This is then used for prediction. Then the accuracy is calculated.

6.4 DECISION TREE ALGORITHM:

It is a supervised learning techniques which is used for both classification and regression problems. Just like the name it has tree like structure consisting of root node, parent node, leaf node etc. Splitting happens when a decision has to be made. The decisions are usually of the type YES/NO or can be considered as if-else statements. At the beginning stage the root contains the entire dataset. The decision of splitting of a node is done with the help of various algorithms. Decision tree uses the CART algorithm. It can be considered as a set of all possible solution to the proposed problem.

6.5 RANDOM FOREST

It can be considered as a combination of many decision trees. It comes under ensemble learning and uses the method of bagging on the dataset. All these

decision trees come up with a prediction class, of which the one that satisfies maximum available conditions is chosen. The beauty of this model lies in the fact the error made by one tree does not affect its neighbours. That is, every tree has got their own workspace. But as a whole they work together for better precision and stability. Since each tree is independent of its neighbour, they have very low correlation. In decision tree, when you have to split, we go for the factor or feature that causes a split between the left node and right node. Whereas in random forest, it can only choose from the available subset of features. It uses bagging and randomly choosing the training set to get trees with low correlation.

6.6 ADABOOST

It is also known as adaptive boosting algorithm. It is an ensemble method. The algorithm uses decision tree but with just one split. It uses the concept of weights by assigning weights to all datapoints and higher values are given to the one's that are wrongly classified. Those weights having higher weights are taken into consideration for the next model. This process continuous until error is minimised.

6.7 GRADIENT BOOSTING

Why do we use different algorithms in predicting something? It is mainly because we want to find the best model in terms of speed and accuracy. Gradient boosting is a method which has high accuracy and is relatively faster. It uses the method of boosting. It is used for predicting both categorical and continuous target variable. Here we have many so-called weak learners which are asked to come up with a strong learner. The weak learners are basically the decision trees. They are connected in series. Apart from random forest, they are related. The error captured by the first tree is rectified using the second and so on. The problem is

that one tree has to wait for the previous tree to gather information and rectify the mistake making it a bit slower. Also, any number of trees can be added and the new trees does not affect the results of the previous ones.

7)RESULTS AND EVALUATION:

CODE 1:

Different algorithms such as SVM, Logistic Regression, K-nn, Naïve Bayes, Decision Tree, Random Forest, Gradient boost and Adaboost are used to detect if a person is wearing a mask or not.

After training the dataset, the accuracy of each model is calculated and the one with highest accuracy is chosen for prediction. If the person on the video appears to wear a mask it will show the output as ‘mask detected’ else ‘no mask detected’.

The results of a trial are given below in Fig.7.1:

```
In [190]: faceCascade=cv2.CascadeClassifier(r"C:\Users\santr\Downloads\haarcascade_frontalface_default.xml")
video_cap = cv2.VideoCapture(0)
data = []
while True:
    success, img = video_cap.read()
    if success:
        faces = faceCascade.detectMultiScale(img, 1.1, 4)

        for (x, y, w, h) in faces:
            cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
            face_image=img[y:y+h, x:x+w, :]
            face_image=cv2.resize(face_image,(50,50))
            face_image=face_image.reshape(1,-1)
            face_image = p.transform(face_image)
            pred = knn.predict(face_image)
            l = LABELS[int(pred)]
            print(l)
        cv2.imshow("Resultant", img)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
cv2.destroyAllWindows()
video_cap.release()

MASK DETECTED
MASK DETECTED
MASK DETECTED
```

Fig.7.1.

CODE 2:

Here we have focused on one particular algorithm, which is, Logistic Regression. The model gave us an accuracy of 76.71%. The result of a trail of a person wearing a mask is given below in Fig 7.2.

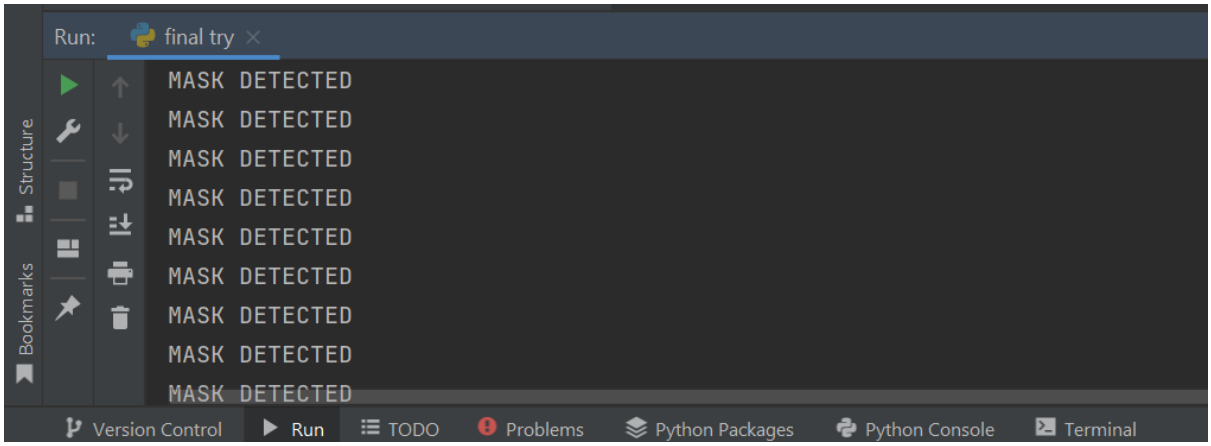


Fig.7.2.

EVALUATION:

From of evaluation, we found the second code to be better as it has higher accuracy in detection. We used Logistic Regression for this evaluation.

8. REFERENCES

[1] Face Recognition in Video Streams and its Application in Freedom Fighters Discovery - A Machine Learning Approach, Vani Perumal, IT Department University of Technology and Applied Sciences.

[2] Application of Yolo on Mask Detection Task, Ren Liu and Ziang Ren, Georgia Institute of Technology.

[3] Fast R-CNN, Ross Girshick, Microsoft Research.