# Face Mask Detection and Counting using Machine Learning

## Motivation

Today it has become mandatory for all the citizens to wear a face mask to protect themselves from COVID-19. This application can be helpful for all the shop owners, offices, banks or any public place because if anyone is not wearing a mask then he or she must not be allowed in that area. So, to take care of this problem we don't need any guard or person who keeps a watch on people. We can integrate a camera which continuously clicks pictures of humans and detect from there faces whether they are wearing a face mask or not. Before coronavirus, some people put masks to protect themselves from air pollution, while other people put face masks to hide their faces and their emotions from others.

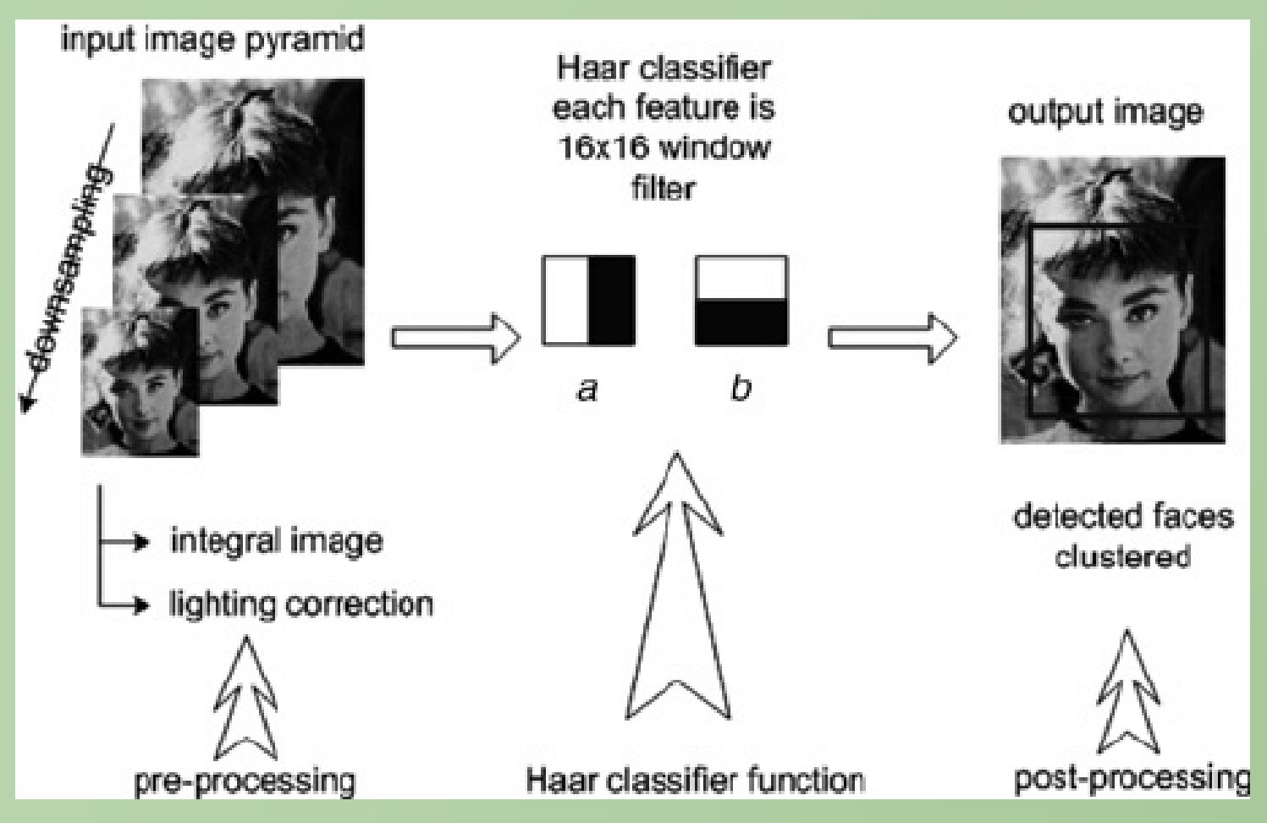
## Steps

1. **Data Collecting**: The development of the Face Mask Recognition model begins with collecting the data. The dataset train data on people who use masks and who do not. The model will differentiate between people wearing masks and not.
2. **Pre-processing:** The pre-processing phase is a phase before the training and testing of the data. There are four steps in the pre-processing which are resizing image size, converting the image to the array, pre-processing input using SVM, and the last is performing hot encoding on labels
3. **Split the Data**: After the pre-processing phase, the data is split into two batches, which are training data namely 70 percent, and the rest is testing data. Each batch is containing both of with-mask and without-mask images.
4. **Building the Model**: The next phase is building the model. There are six steps in building the model which are constructing the training image generator for augmentation, the base model with SVM, adding model parameters, compiling the model, training the model, and the last is saving the model for the future prediction process.
5. **Testing the Model**: To make sure the model can predict well, there are steps in testing the model. The first step is making predictions on the testing set. The result is checking the loss and accuracy while training the model.
6. **The model implemented in the video**: The video read from frame to frame, then the face detection algorithm works. If a face is detected, it proceeds to the next process. From detected frames containing faces, reprocessing will be carried out including resizing the image size, converting to the array, preprocessing input using SVM.

## Face Detection with Haar Cascade

### Haar Cascade Algorithm:

It is an Object Detection Algorithm used to identify faces in an image or a real time video. The algorithm uses edge or line detection features proposed by Viola and Jones in their research paper "Rapid Object Detection using a Boosted Cascade of Simple Features" published in 2001.



The algorithm is given a lot of positive images consisting of faces, and a lot of negative images not consisting of any face to train on them. Haar Cascade is a machine learning-based approach where a lot of positive and negative images are used to train the classifier. Positive images – These images contain the images which we want our classifier to identify. Negative Images – Images of everything else, which do not contain the object we want to detect.

### Python Pygame Library:

Python is the most popular programming language or nothing wrong to say that it is the next-generation programming language. In every emerging field in computer science, Python makes its presence actively. Python has vast libraries for various fields such as Machine Learning (Numpy, Pandas, Matplotlib), Artificial intelligence (Pytorch, TensorFlow), and Game development (Pygame, Pyglet).

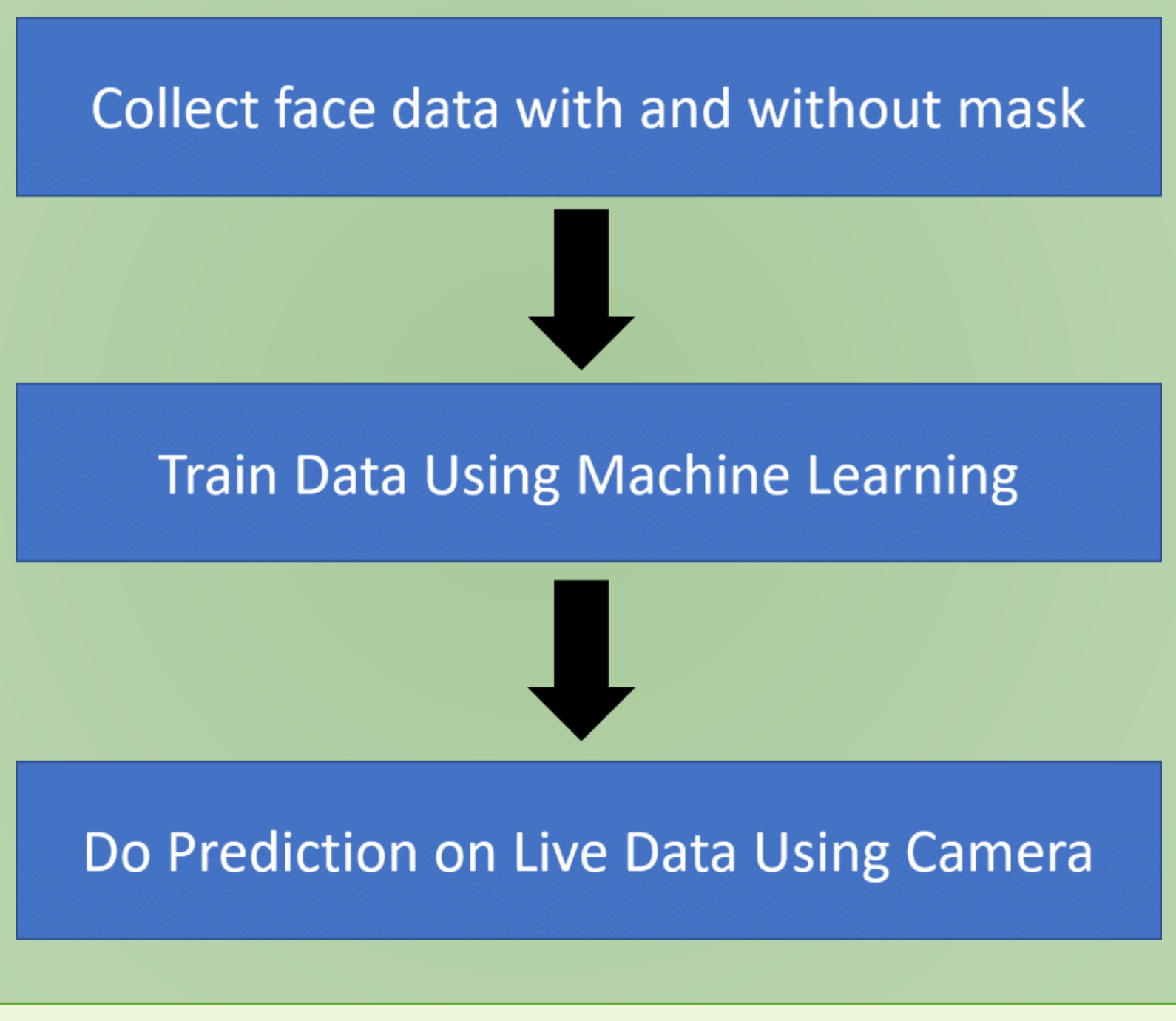
- Pygame is a cross-platform set of Python modules which is used to create video games.
- It consists of computer graphics and sound libraries designed to be used with the Python programming language.
- Pygame was officially written by Pete Shinners to replace PySDL.

## Face Detection Process

Face Detection using haarcascade frontal face default classifier mentioned which is frequently used step for its high detection rate, accuracy and fast processing speed for complex input. Face detection is method that is highly responsible for human face identifying and existence of objects of a with in a small unit which are presents on it. As process of this considered as a method of image processing to identify face from live video via webcam which is record frame by frame.

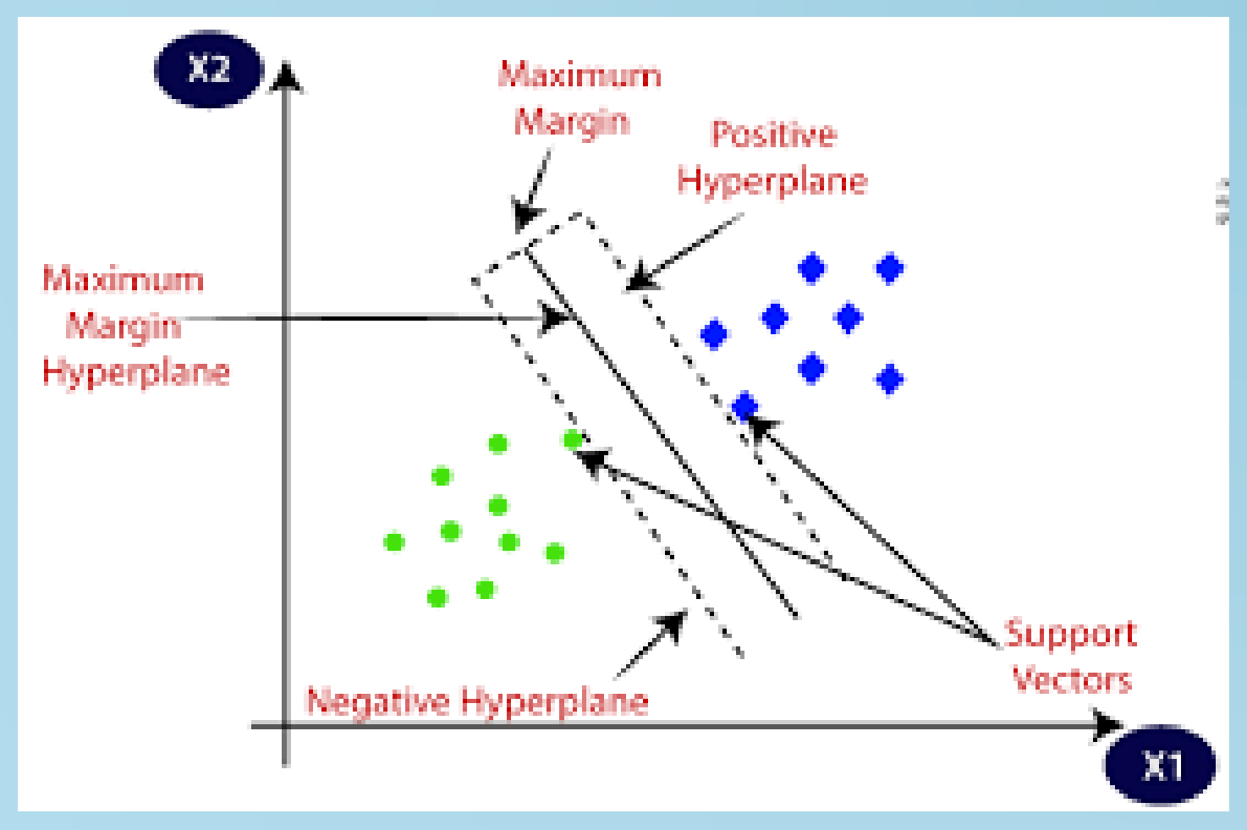
We approach face cascade first in the code. After the face are located, after that we will call our solution that draw a one rectangle around them so after that we know what the machine locates and where. Sometime Machine can create mistakes, but our goals should be to teach the machine get the most and best result way so that prediction is more accurate and truer. In final step we will export our result image to detect like how many faces founded, and also at same time console side it shows same information and face detection live result. At conclusion it extracts a feature from the video and find all the detected faces and identify them by without error.

## Flow Diagram


Collect face data with and without mask
Train Data Using Machine Learning
Do Prediction on Live Data Using Camera

## Support vector machine

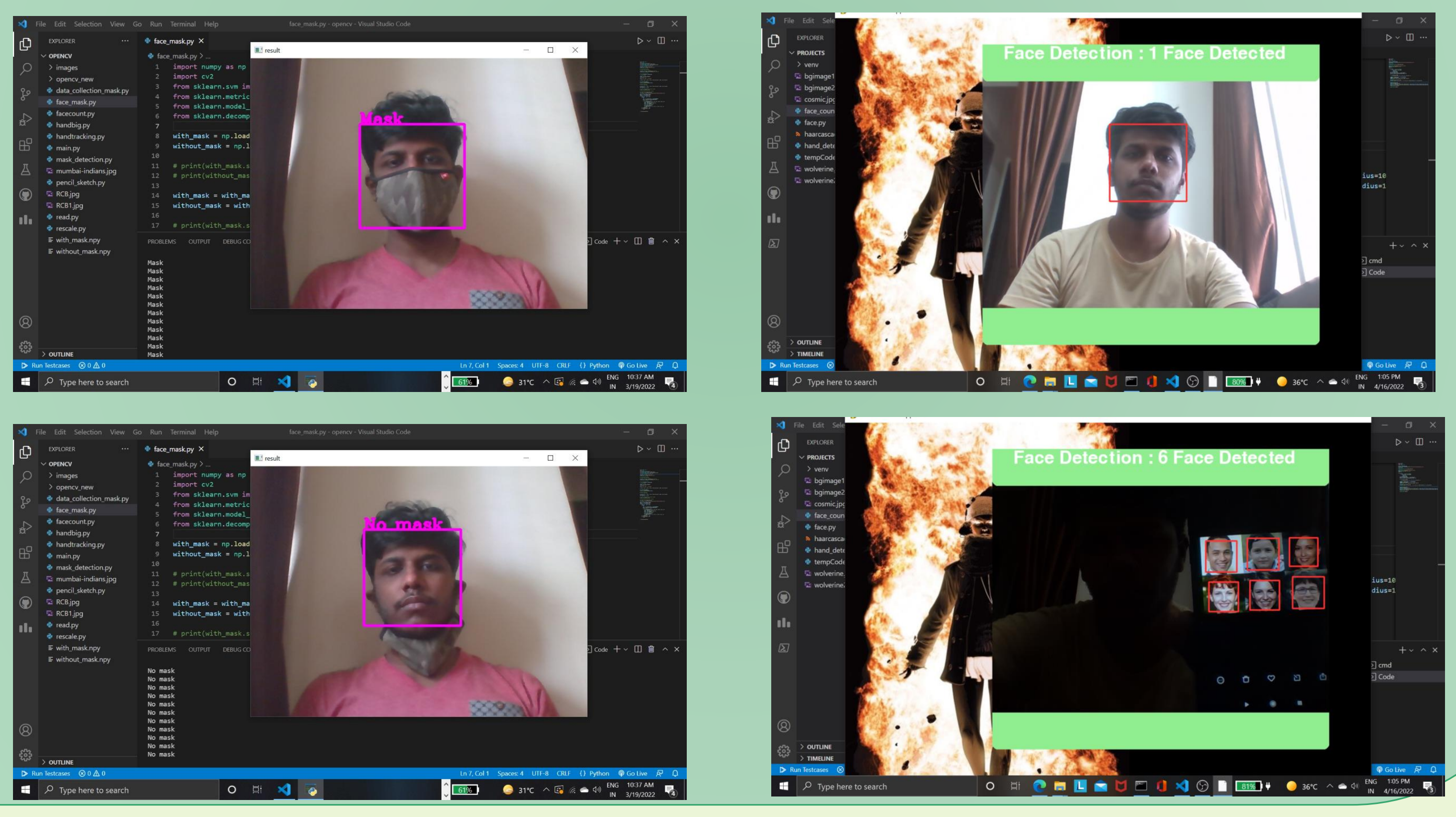
"Support Vector Machine" (SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is a number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well (look at the below snapshot).



SVM is a very good algorithm for doing classification. It's a supervised learning algorithm that is mainly used to classify data into different classes. SVM trains on a set of label data. SVM works relatively well when there is a clear margin of separation between classes. SVM is more effective in high dimensional spaces. SVM is effective in cases where the number of dimensions is greater than the number of samples. SVM is relatively memory efficient.

## Results & Discussion

The results of the proposed model can be analyzed:
1) Initially the person is detected with a camera for the wearing of mask.
2) Person is detected as no mask.
3) In this case, the algorithm calculated person count as 1.
4) And in last image, the algorithm calculated person count as 6.


Mask
Face Detection : 1 Face Detected
No mask
Face Detection : 6 Face Detected

**Made by**:

| | |
|---|---|
| 1. Ruturaj Javeri | - 309 [ENTC] |
| 2. Abhishek Sandhan | - 321 [ENTC] |
| 3. Rushikesh Karanjkar | - 313 [ETX] |

**References:**

- A. Malakar, A. Kumar and S. Majumdar, "Detection of Face Mask in Real-time using Convolutional Neural Networks and Open-CV," *2021 2nd International Conference for Emerging Technology (INCET)*, 2021, pp. 1-5, doi: 10.1109/INCET51464.2021.9456415.
- https://towardsdatascience.com/face-detection-with-haar-cascade-727f68dafd08
- https://brain-mentors.com/face-mask-detection-using-opencv-in-python/
- https://www.javatpoint.com/pygame#:~:text=Pygame%20is%20a%20cross%2Dplatform,Pete%20Shinners%20to%20replace%20PySDL