**MID TERM PROJECT REPORT**

**PROJECT TITLE-** Face Mask detection and recognition using openCV

**TEAM MEMBERS-**

Sanjeev Kumar Sah (m22ma010)

Aviral Tripathi (m22ma012)

Ajjarapu Jyothi (m22cs007)

**ABSTRACT**

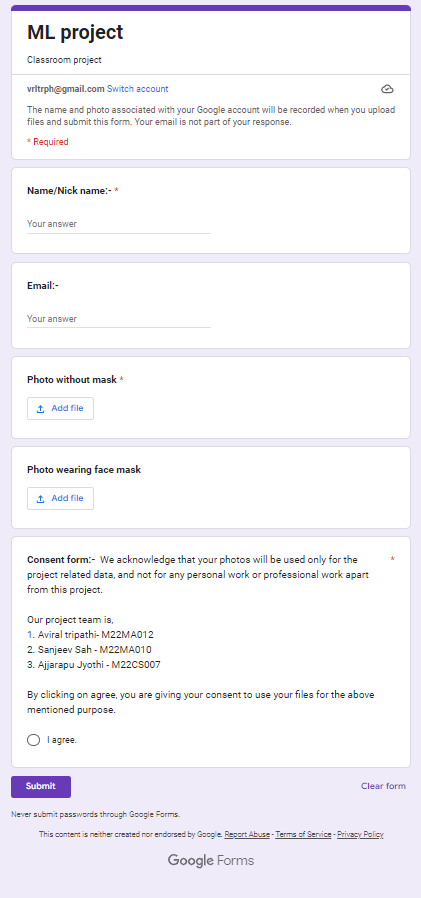
Our project is Face Mask detection.  There is a need for the Facemask now a days, as many diseases, like Coronavirus, are spreading worldwide.  Corona virus spreads through droplets that come from the infected person. Public areas have the highest transmission risk. Carrying a face mask in open areas as suggested is one of the excellent approaches to avoid getting infected. We are implementing a method that finds the face masks on humans using OpenCV. A bounding container that is drawn over the subject's face indicates whether or not the subject is wearing a mask. If a person's face is saved in the database, it can identify them if they aren't wearing face masks. We are training our model through various images of our classmates (collected through Google form, with their consent)

**INTRODUCTION**

This project is an object detection and classification problem with two different classes. We have classes like with mask and without Mask. We have collected a dataset from Google form. Our model is built to detect the faces with mask using Python, OpenCV and other ml algorithms. We will train our model using different machine learning algorithms like SVM. We will develop this model for detecting whether person is wearing a mask or not. We do this using SVM algorithm and other ML algorithms. Training the model is the first part of this project and testing based on the used algorithms is the second part. The dataset we are working on, currently consists of 50 images with 25 images containing images of people wearing masks and 25 images with people without masks. We will test the results based on the model we have trained.

**DATASET DETAILS**

We require some images to train our model, we have used Google form to collect these images. Till now we have received around 25 responses, where each response has an image with mask and an image without mask.



We have even taken the consent for protecting their data.

**METHODOLOGY**

The first step is to capture or load the image into the platform. We are importing the data and plotting the image, this is the raw data.

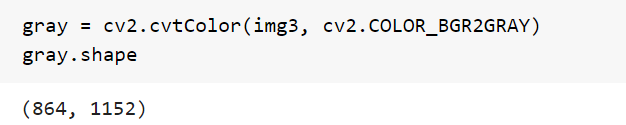
For our project we need to use all the images of the dataset and process them. For our convenience, we have used few images and processed them. After getting certain idea as to how to process the data, we will do it on the entire dataset.

The next step is to process the image. For this, we need to convert the image to gray scale. This will enhance the contrast of the image, and will be easy to process it. We can even use black and white image transition, but grayscale gives optimum visual balance, and can know which pixels are to be highlighted specifically.

Cvtcolor() is being used. This converts to grayscale.

Next step is to identify or detect the face in the complete image. For this, we use cascaadeclassifier function. And we use harrcascade function in order to detect the object. We already have face detection, eye detection and smile detection xml files available in openCV. So we are directly using those files in our project to detect eyes, face and smile components of the image.

Code:





We can train the data using this function where we can use both positive and negative set of examples to detect the faces.

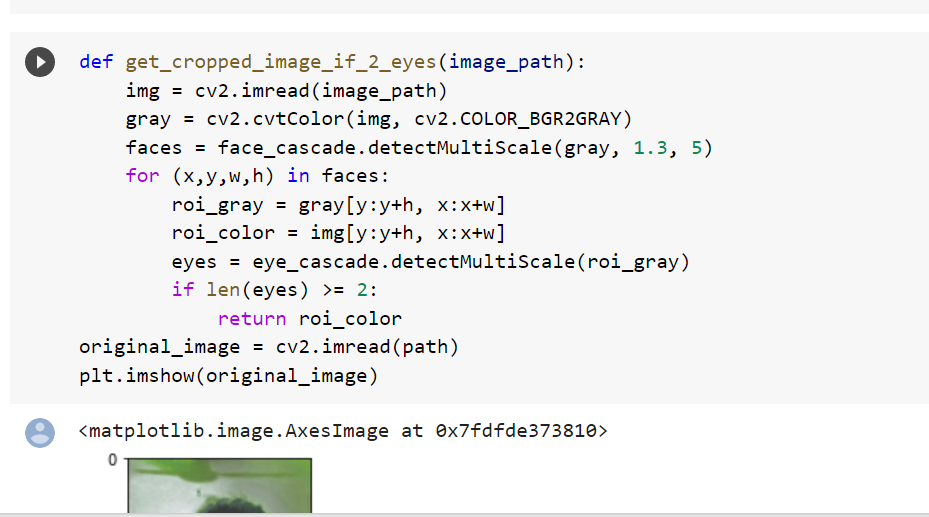
Now, after detecting the objects, we may get different sizes of the targeted sub-image. In order to process those different sizes, we use detect\_ multiscale function. The output for this function will be the list.

We can see the output of this by taking various values and applying rectangle() on it. This will show the image where the target is detected and drawn with a rectangle.

Now, we are done with preprocessing. So we will be taking the cropped image where the targets are specified. For this, we are checking if there are two eyes in the image otherwise reject the image. If number of eyes >=2 proceed further, else reject this image

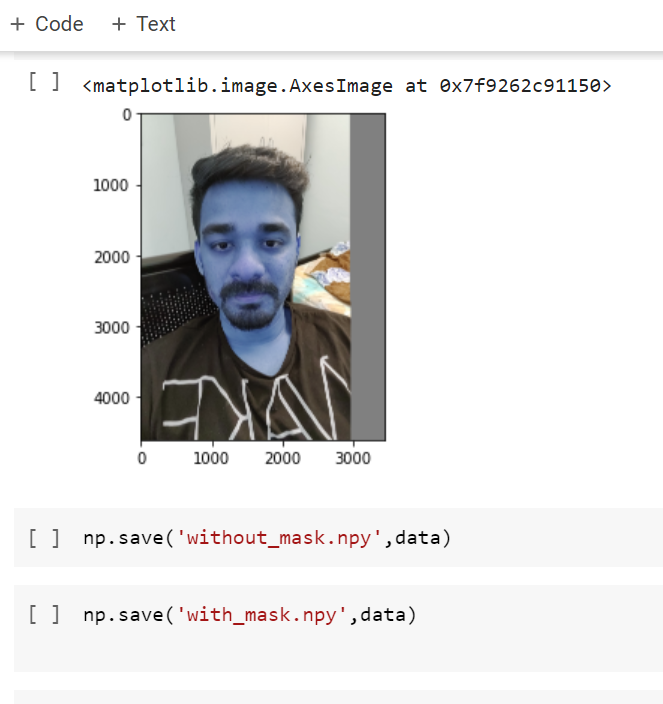
Resizing these cropped images. And adding all of them to a separate dataset array.

Code:

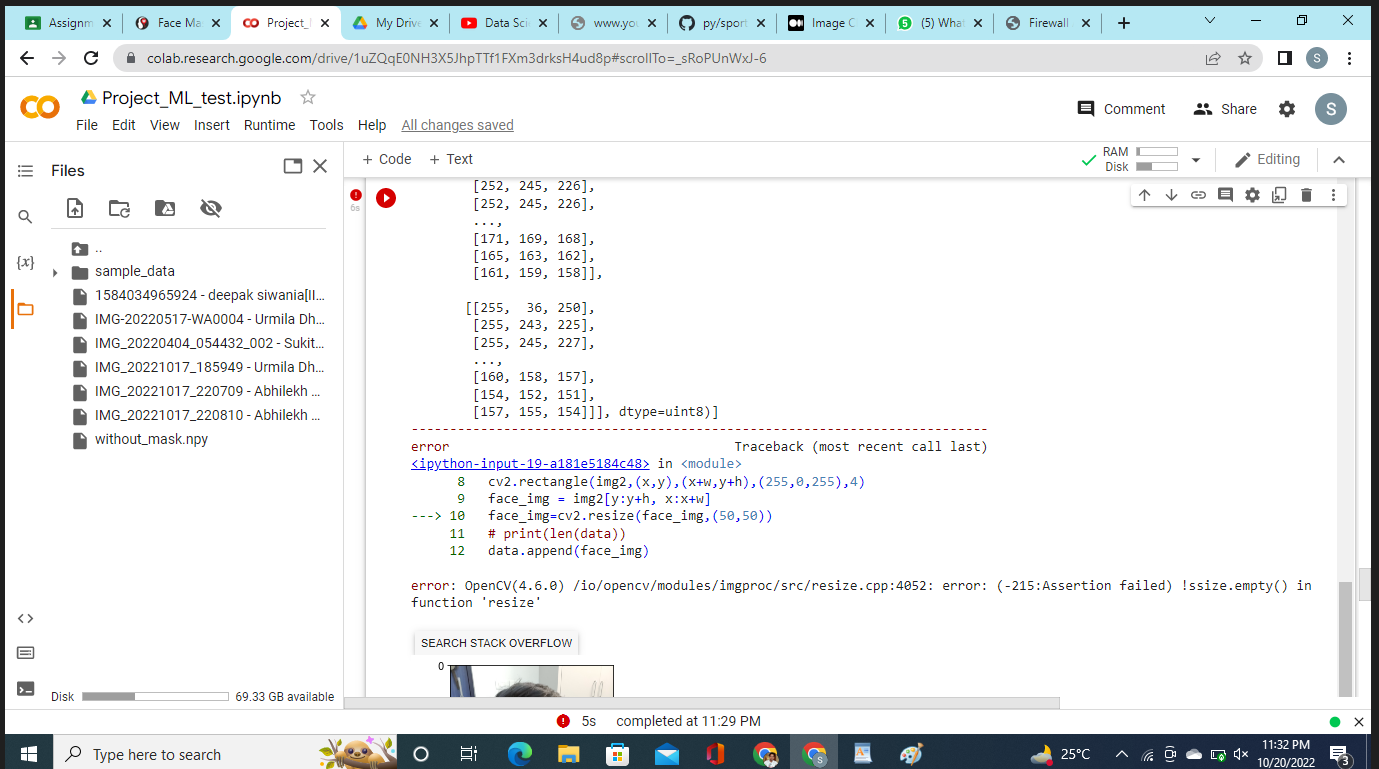


We are creating an array for the cropped image. Converting it to numpy array for easy calculation.

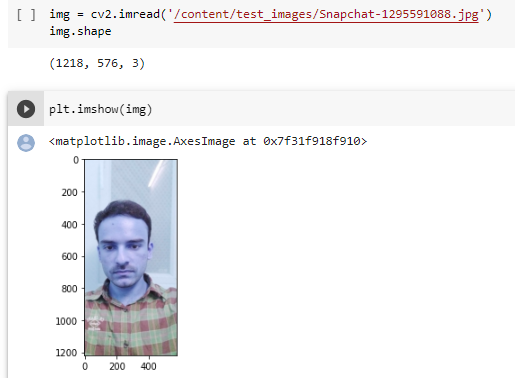
We are saving images with mask and without mask separately. And converting them to numpy. And assigning the labels to the datasets. Like, with mask –“1” and without mask –“0”.

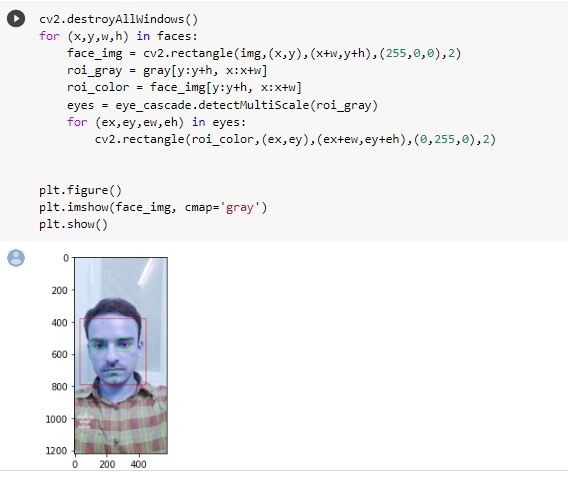


Now the dataset is ready for model training. We can train the model using different machine learning algorithms like SVM, logistic, linear regressions, decision trees etc. For now, we are trying using SVM algorithm. We are getting some errors related to this work.

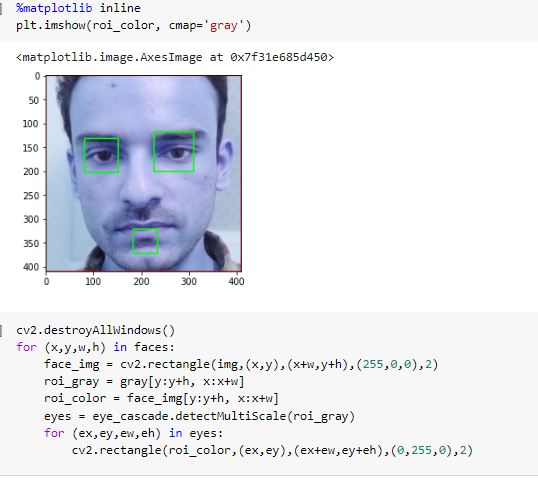


**EXPERIMENTS AND RESULTS**

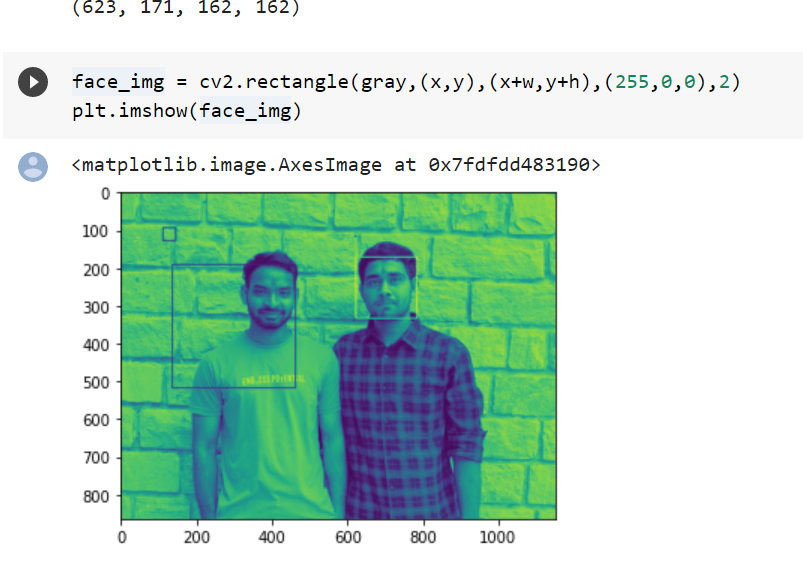




****

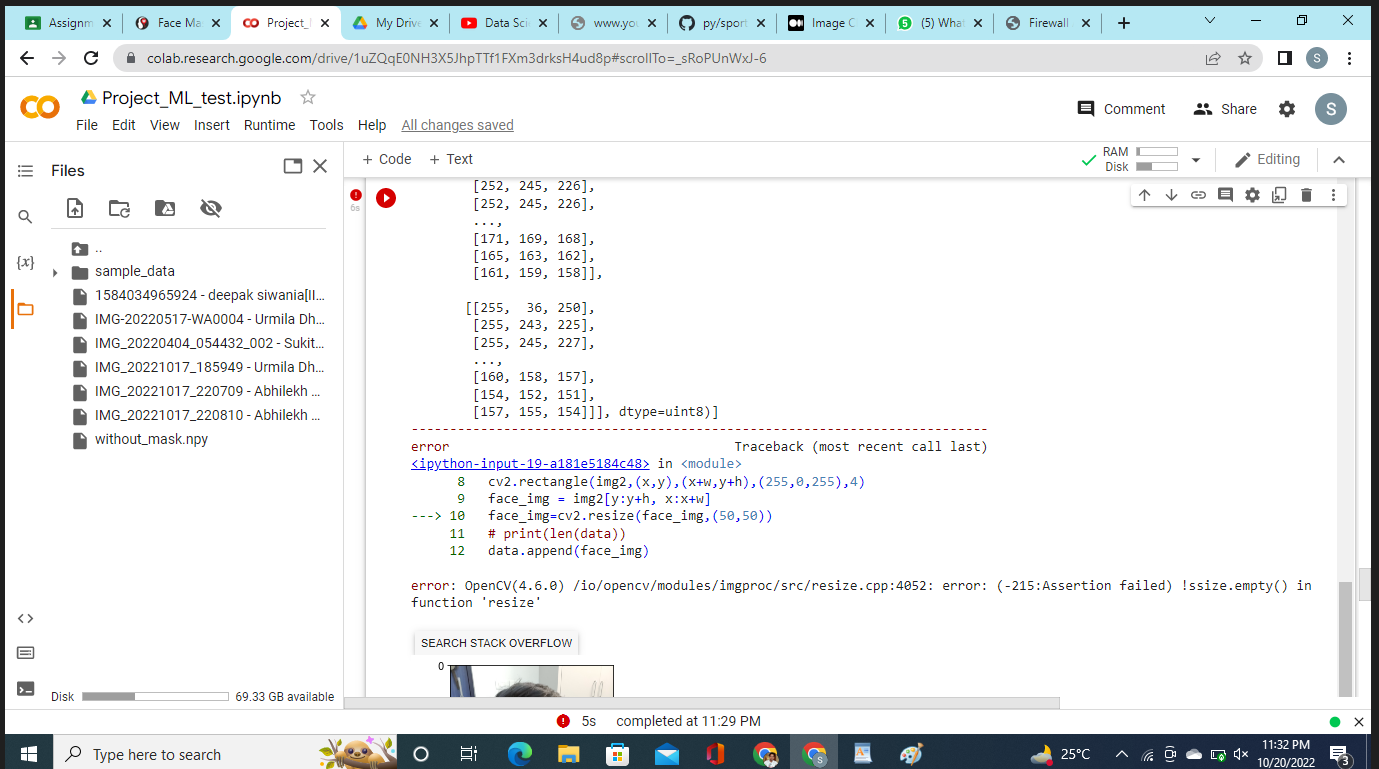
****

When we try to detect faces in the images having multiple faces, it’s not detecting properly. This is the example.



It’s not detecting properly when we have a single image but having dark spots.





**CONCLUSION**

We trained our model for one image as of now. We will do it using all the images of the dataset. We found, if the image has two or more faces, those are not getting cropped properly and showing errors. So we will work on it in the upcoming days. And we will try recognizing the names of the people. We will work on all these tasks in the upcoming days.

**FUTURE SCOPE**

We can integrate our project with Face detection where the person who isn’t wearing a mask, gets prompted through mail or whatsapp or even a simple message. We can directly use our methodology to detect the faces, considering it to be a face detection technology. Face detection technology can be used to enable real-time surveillance and tracking of people in a variety of industries, such as security, biometrics, law enforcement, entertainment, and personal safety.