PROJECT REPORT

(Project Term January-May 2021)

(Capture Own Emotion With Python)

Submitted by

(Ravi Pandey)   Registration Number :11910459

Project Group Number A

Course Code INT246

Under the Guidance of

(Dr. Sagar Pande)

School of Computer Science and Engineering

DECLARATION

We hereby declare that the project work entitled (" Capture Own Emotion With Python ") is an authentic record of our own work carried out as requirements of Project for the award of B.Tech degree in Computer science from Lovely Professional University, Phagwara, under the guidance of (Dr. Sagar Pande), during August to November 2021. All the information furnished in this project report is based on our own intensive work and is genuine.

Project Group Number: A

Name of Student : Ravi Pandey
Registration Number: 11910459

(Ravi Pandey)

Date: 10-11-2021

CERTIFICATE

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. They have completed this Project under my guidance and supervision. The present work is the result of their original investigation, effort and study. No part of the work has ever been submitted for any other degree at any University. The Project is fit for the submission and partial fulfillment of the conditions for the award of B.Tech degree in Computer Science from Lovely Professional University, Phagwara.

Signature and Name of the Mentor

Designation

School of Computer Science and Engineering,
Lovely Professional University,
Phagwara, Punjab.

Date :20-11-2021

ACKNOWLEDGEMENT

I would sincerely like to thanks for the constructive criticism, support, encouragment valuable, comments, suggestions, timely helps and many innovative ideas given to me by my project supervisor Dr sagar Pande in carrying out project and the report

I must convey my gratitude to Dr sagar Pande for giving me the constant source of Inspiration and help in preparing the project ,personally correcting my work and Providing encouragement throughout the project.

I also thanks all my faculty members for steering me through the tough as well as Easy phase of the project in result oriented manner with concern attention.

INTRODUCTION

My name is Ravi Pandey. currently, I am   pursuing Btech from Lovely professional university in computer science. Here I am to tell about my project which I have completed during august-november**.** my project topic is " Capture Own Emotion With Python". in this project I have write code which when it run open camera .which scan Face of person which is front of camera.and try to detect face emotion.after detecting face emotion (like:- Angry, Disgust, Fear, Happy, Sad, Surprise and Neutral) is printed on screen.

Profile of the Problem. Rationale/Scope of the study (Problem Statement)

Emotion recognition is a method used in software that permits a program to "examine" the sentiments on a human face by utilizing sophisticated image dispensation. Firms have been testing with an amalgamation of advanced formulas with image processing practices that have materialized in the last decade to appreciate more regarding what a video or an image of an individual's face tells us concerning how they are feeling. With current innovation, emotion identification software has developed very adeptly. Moreover, its aptitude to track first facial looks for emotions like happiness, sadness, surprise, anger, etc., emotion detection software can also capture what specialists describe as "micro-expressions" or restrained cues of body language, that might reveal a person's feelings devoid of their knowledge.

Emotion recognition also concurs with other types of facial recognition technologies and bio-metric image identification. These two types of technologies can be applied in many kinds of security cases. For example, authorities can utilize emotion recognition software to further investigation efforts concerning someone at some point in an interview or interrogation. Emotion detection continues to go forward on par with other innovations such as natural language processing and these signs of progress are for the most part made probable by the blending of ever more dominant processors, the scientific growth of complex algorithms, and other associated technologies

Existing System

People were detect emotions of persons by watching theirs faces, and their body Languages.so to detect their face and bodylanguage is very difficult.

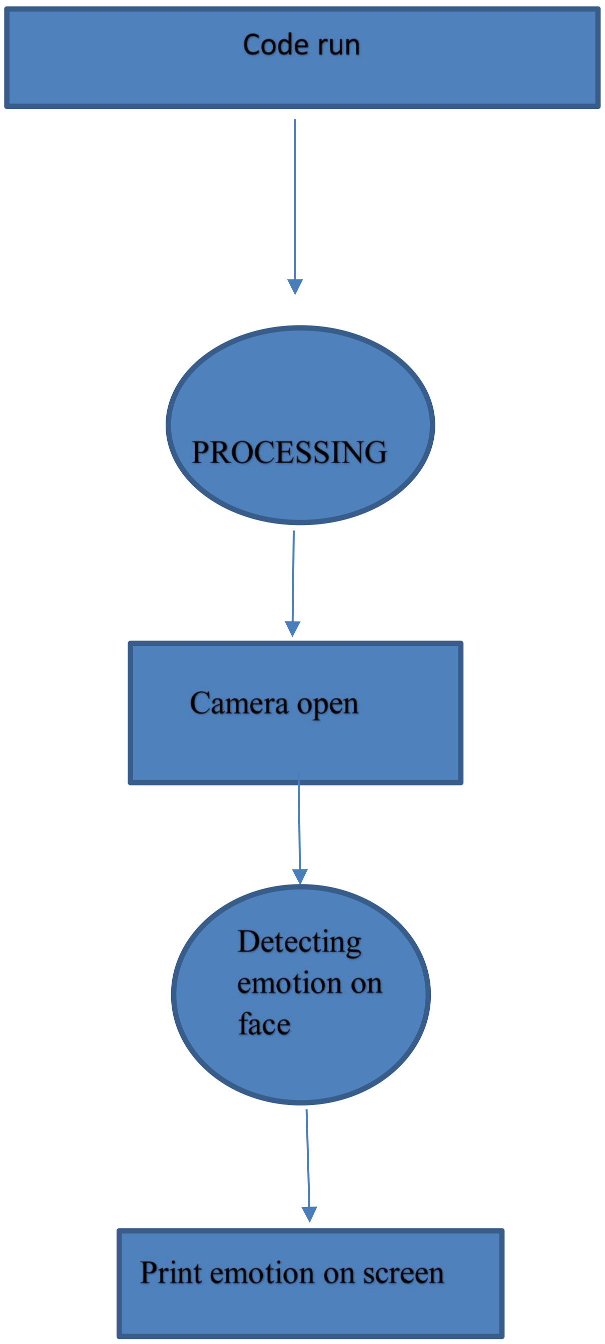
• Introduction

My system is made up of python code. which when run it try to detect the face of person.detect their emotion.and print on screen.

• Existing Software

1. Anaconda jupyter
2. Camera

- **DFD for present system**



- **What's new in the system to be developed**

The new things in system to be developed is face emotion detect by camera.

When we run the code camera will open and try to detect face which is front of camera

And capture face emotion.and then print capture emotion on screen.

The programme that I made is use to detect the emotion of person by detecting

Their face.if person is angry it print angry on screen.if person is happy it print happy

on screen.if person is surprise then print surprise on screen.if person is disgust then it

print disgust on screen.if person is sad then print sad on screen.if person is fear it print

fear on screen.if person is neutral then print neutral on screen.

• Product definition

Product definition is a critical starting point in the development of any new product. Yet for its importance, there are a number of common shortcomings to the process of product definition in many companies:

- No defined product strategy or product plan
- Lack of formal requirements as a basis for initiating product development
- Product requirements developed without true customer input
- A marketing requirement specification (MRS) that is completed late – after development is underway
- Engineering having little or no involvement in development of MRS, thereby lacking a true understanding of requirements
- An incomplete, ambiguous, or overly ambitious MRS
- Creeping elegance or a constantly evolving specification that requires increasing development scope and redesign iteration

• Feasibility Analysis

- **Project Plan**

A project plan is a formal document designed to guide the control and execution of a project. A project plan is the key to a successful project and is the most important document that needs to be created when starting any business project.

In IT, the term project plan refers to a a Gantt chart or any other document that displays project activities along a timeline. However, considering these documents alone as a project plan is inaccurate. These particular documents can be more precisely termed as project schedules, and may be considered only a part of the actual project plan.

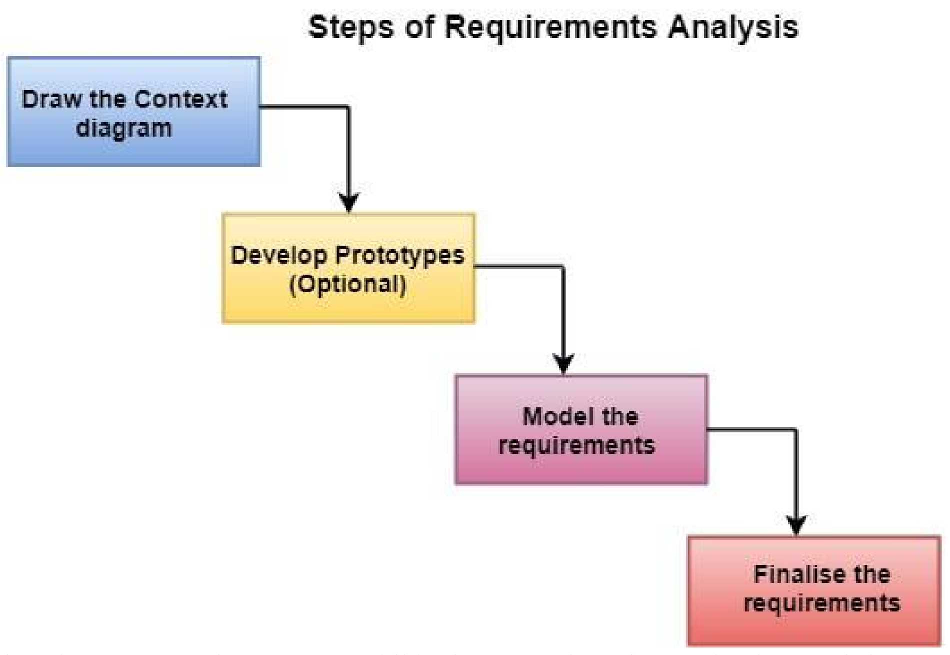
A project plan is used for the following purposes:

- To document and communicate stakeholder products and project expectations
- To control schedule and delivery
- To calculate and manage associated risks

Software Requirement Analysis

Requirement analysis is significant and essential activity after elicitation. We analyze, refine, and scrutinize the gathered requirements to make consistent and unambiguous requirements. This activity reviews all requirements and may provide a graphical view of the entire system. After the completion of the analysis, it is expected that the understandability of the project may improve significantly. Here, we may also use the interaction with the customer to clarify points of confusion and to understand which requirements are more important than others.

The various steps of requirement analysis are shown in fig:


Steps of Requirements Analysis
Draw the Context diagram
Develop Prototypes (Optional)
Model the requirements
Finalise the requirements

When it comes to the success of life there needs to be perfection and the quality of development and the strength of achievement. Success comes only when we have all the above factors, and when talking about the software development you need the analysis. Analysis of what is being introduced in the society, how is it going to work and how will the market respond to it. This analysis is really very important if you

don't want failure at the first attempt of your introduction to software. So you need software requirement analysis which will definitely lead you to the heights of perfection when your software is introduced.
In software and system engineering, requirement analysis includes task that governs the condition or requirement to meet for a new product. It includes taking account of conflicting requirements of other stakeholders.

It includes various things;

- Regular communication with the software users to know about their expectations.
- Resolution of complaints made by the user or group of users.

- Avoid feature creep.
- Keep up-to-date all the documentations from starting to present of project development.

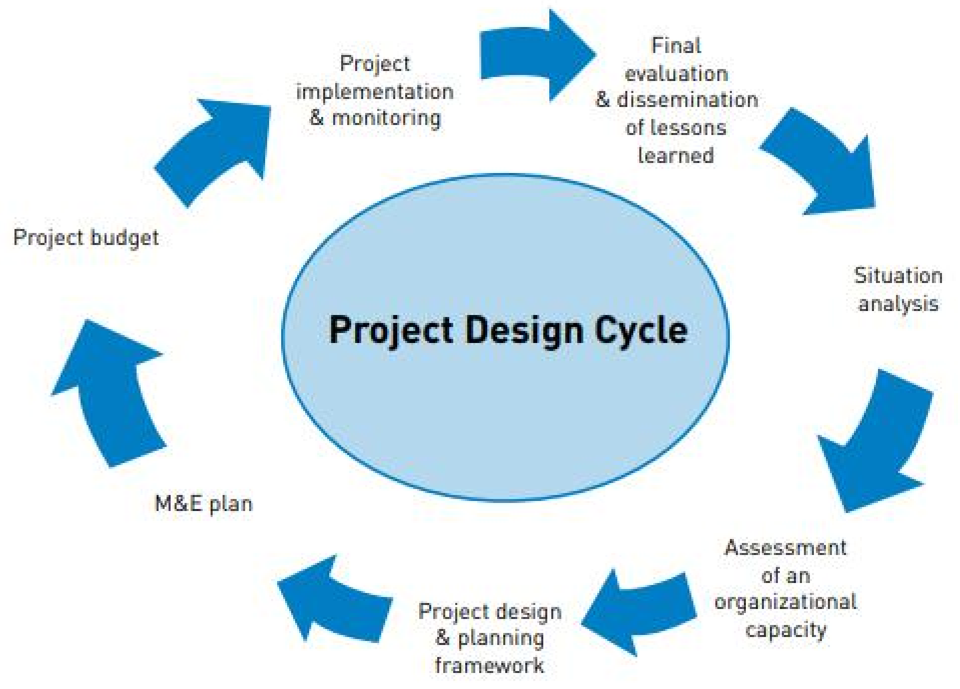
We should keep in mind that final product must include what client wants rather than afterwards making changes in the software according to users need.

It requires team work which includes hardware, software and human skills. It is very expensive and requires huge investment.

Design

- **System Design**

Figure 1: The Project Design Cycle


Project Design Cycle
Project implementation & monitoring
Final evaluation & dissemination of lessons learned
Situation analysis
Assessment of an organizational capacity
Project design & planning framework
M&E plan
Project budget

➢ Design Notations

Design notations are used when planning and should be able to communicate the purpose of a program without the need for formal code. Commonly used design notations are:

- Pseudocode
- Flow charts
- Structure chart

Pseudocode

When designing a program, it is useful to lay out how the program might work, before writing it in a programming language,.

Pseudocode is a design notation that is closely related to the logic of how a program will work. It lets you detail what your program will do without having to worry about the particular syntax of your chosen programming language.

There is no specific standard for pseudocode and programmers often have their own version. Pseudocode can look a lot like code but it does not need to be implemented as strictly.

Programming languages like python, reference language (used by the SQA) and visual basic will have specific rules around syntax and structure, whereas pseudocode gives more freedom.

There is no strict set of rules for pseudocode, but some of the most widely recognised are:

- `INPUT` – indicates a user will be inputting something
- `OUTPUT` – indicates that an output will appear on the screen
- `WHILE` – a loop (iteration that has a condition at the beginning)
- `FOR` – a counting loop (iteration)
- `REPEAT – UNTIL` – a loop (iteration) that has a condition at the end
- `IF – THEN – ELSE` – a decision (selection) in which a choice is made

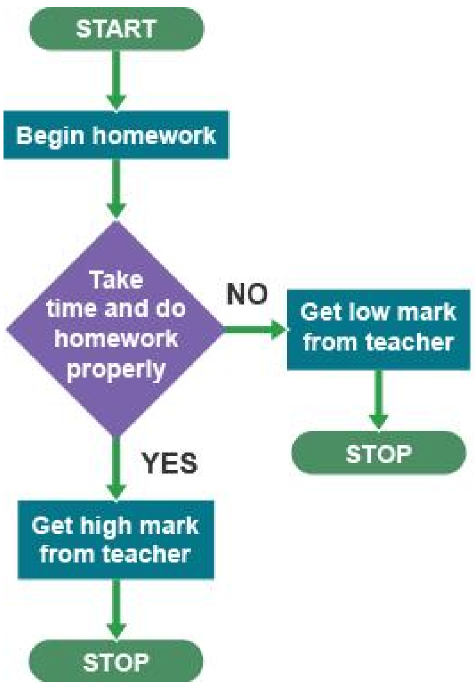
Pseudocode can be used to plan out programs. It is similar to actual code. Planning a program that asks people what their favourite subject is could look like this in pseudocode:

```
REPEAT
 OUTPUT 'What is the best subject you take?'
 INPUT user inputs the best subject they take
 STORE the user's input in the answer variable
 IF answer = 'Computer Science' THEN
  OUTPUT 'Of course it is!'
 ELSE
  OUTPUT 'Try again!'
UNTIL answer = 'Computer Science'
```

A programmer who uses pseudocode as part of their planning is able to take time to think about how their program will work, what variables they might need and what inputs and outputs there are.

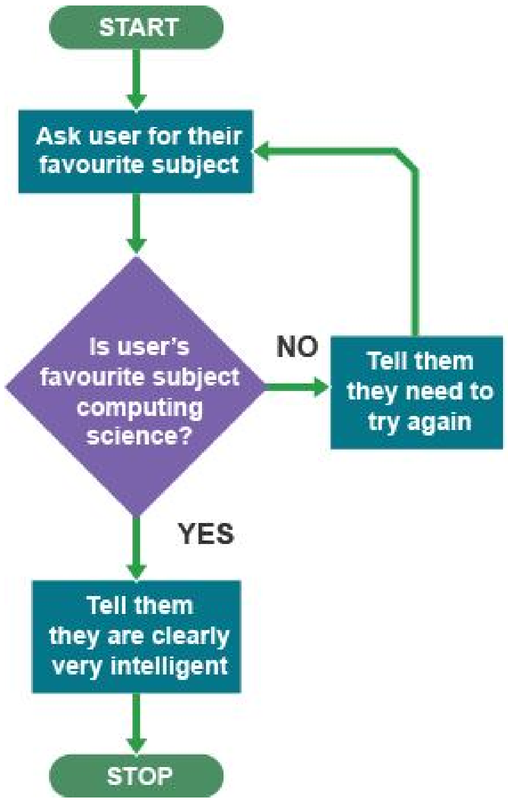
Flow charts

Flow charts show what is going on in a program and how data flows around it. Flow charts can represent everyday processes, show decisions taken and the result of these decisions.


START
Begin homework
Take time and do homework properly
NO
Get low mark from teacher
STOP
YES
Get high mark from teacher
STOP

The diamond shape explains when there is a choice to make. The flow chart shows what happens depending on the decision made at this point. Flow charts visualise the results of decisions, showing what will happen in a program, and also when, for example an if statement, is required to make a decision.

Flow charts can be used to show iteration (repeating something).


START
Ask user for their favourite subject
Is user's favourite subject computing science?
NO
Tell them they need to try again
YES
Tell them they are clearly very intelligent
STOP

In the above flow chart, the user is asked what their favourite subject is. If they answer 'Computing Science' they are told they are 'clearly very intelligent' and the program stops.

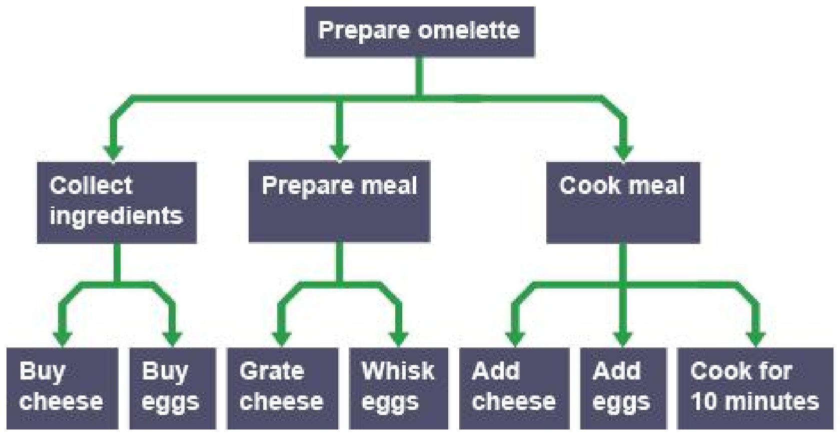
Any other answer results in the user being asked again. If the user does not enter 'Computing Science', the program will keep going round and round, asking them forever until they enter Computing Science.

Structure diagrams

Another way of representing a program design is to use a structure diagram. Structure diagrams break down a problem into smaller sections. These smaller sections can then be worked on one at a time.

This can be good for big projects where a large problem can be split into smaller tasks for separate groups, or individuals, to work on.

Below is an example of how a structure diagram might be used to break a large problem down.

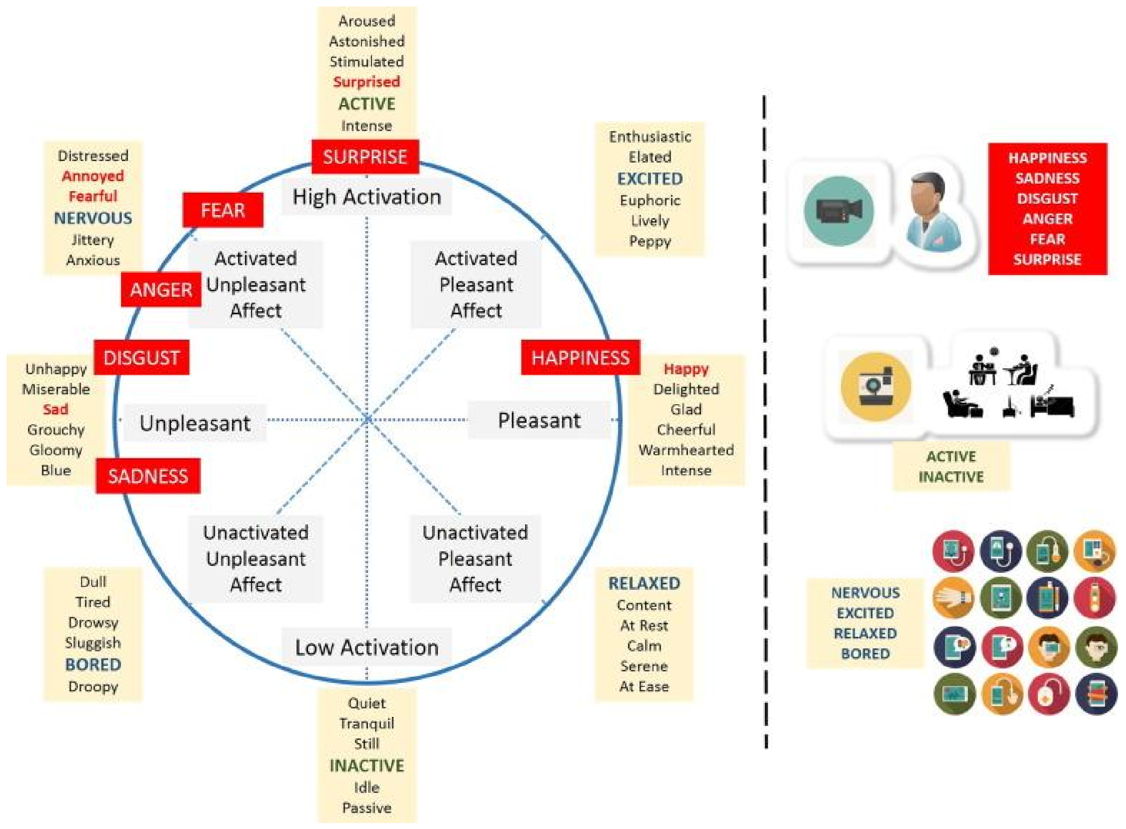


This shows how you can take a complex problem and start breaking it down into more manageable chunks.

In reality, a complex project like building a house would have many more stages, but this example shows that structure diagrams can help to break down problems when designing a program.

You would most likely use a structure diagram if you were designing a game and wanted to break down the overall design problem into individual elements.

➢ Detailed Design



➢ **Pseudo code**

```
import cv2 ### pip install opencv-python

## pip install opencv-contri-python fullpackage

#from deepface import DeepFace ## pip install deepface

path = "haarcascade_fronatalface_default.xml"

font_scale =1.5

font = cv2.FONT_HERSHEY_PLAIN


#set the rectangle background to white

rectangle_bgr = (255,255,255)

# make a block image

img = np.zeros((500,500))

# set some text

text = "some text in a box!"

#get the width and height of the text box

(text_width, text_height) = cv2.getTextSize(text, font,
fontScale=font_scale, thickness=1)[0]

# set the text start position
```

```
text_offset_x = 10

text_offset_y = img.shape[0] - 25

# make the coords of the box with a small padding of two pixels

box_coords = ((text_offset_x,text_offset_y),(text_offset_x + text_width
+2, text_offset_y - text_height - 2))

cv2.rectangle(img, box_coords[0], box_coords[1], rectangle_bgr,
cv2.FILLED)

cv2.putText(img, text, (text_offset_x, text_offset_y), font,
fontScale=font_scale, color=(0, 0, 0), thickness=1)



cap = cv2.VideoCapture(1)

# Check if the webcam is opened correctly

if not cap.isOpened():

    cap = cv2.VideoCapture(0)

if not cap.isOpened():

    raise IOError("Cannot open webcame")


while True:
    ret,frame = cap.read()

    #eye_cascade    =    cv2.CasecadeClassifier(cv2.data.haarcascades
+'haarcascades_eye.xml')

    faceCascade    =    cv2.CascadeClassifier(cv2.data.haarcascades    +
'haarcascade_frontalface_default.xml')

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    #print(faceCascade.empty())

    faces = faceCascade.detectMultiScale(gray,1.1,4)

    for x,y,w,h in faces:

        roi_gray =gray[y:y+h, x:x+w]

        roi_color =frame[y:y+h, x:x+w]

        cv2.rectangle(frame,(x,y), (x+w, y+h), (255,0,0), 2)

        facess = faceCascade.detectMultiScale(roi_gray)

        if len(facess) == 0:

            print("Face not detected")

        else:

            for (ex,ey,ew,eh) in facess:

                face_roi = roi_color[ey: ey+eh, ex:ex + ew]        ##cropping
the face
```

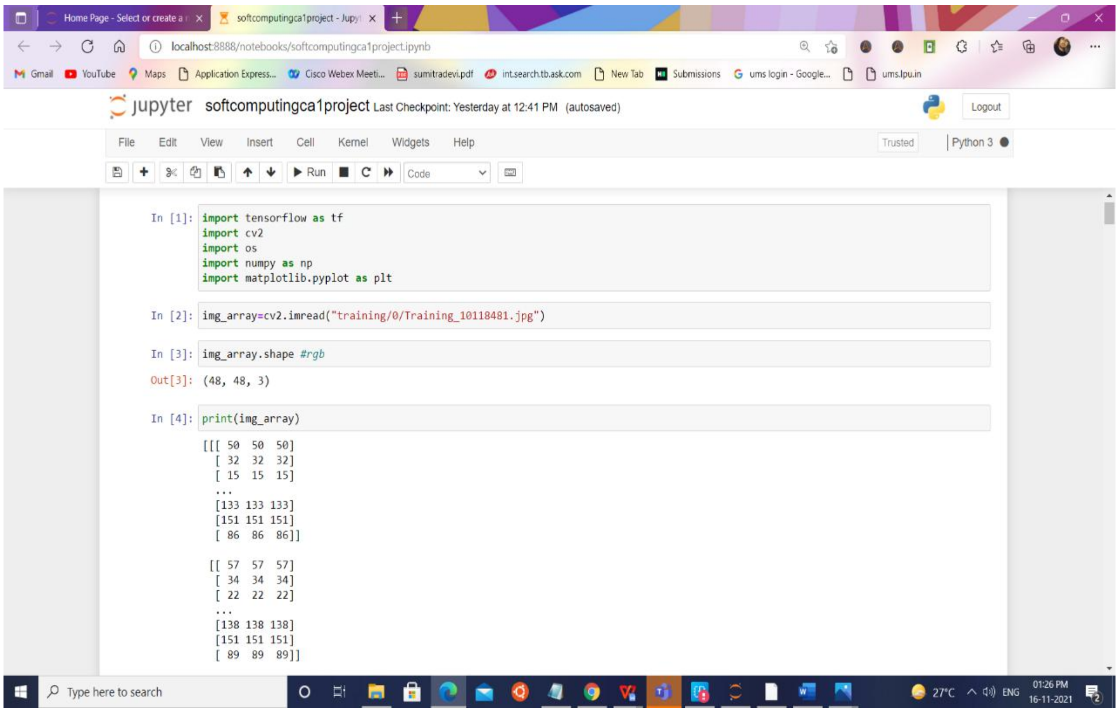
```
    final_image = cv2.resize(face_roi, (224,224))

    final_image = np.expand_dims(final_image,axis =0) ## neesd fourth
dimension

    final_image=final_image/255.0



    font = cv2.FONT_HERSHEY_SIMPLEX



    Predictions = new_model.predict(final_image)



    font_scale = 1.5

    font = cv2.FONT_HERSHEY_PLAIN



    if (np.argmax(Predictions)==0):

        status = "Angry"



        x1,y1,w1,h1 = 0,0,175,75

        # Draw black background rectangle
        cv2.rectangle(frame, (x1, x1), (x1 + w1, y1 + h1),(0,0,0), -1)

        #Add text

        cv2.putText(frame,status,    (x1    +    int(w1/10),y1    +    int(h1/2)),
    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,0,255), 2)


        cv2.putText(frame,status,(100, 150),font, 3,(0, 0, 255),2,cv2.LINE_4)


        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255))


      elif (np.argmax(Predictions)==1):

        status = "Disgust"


        x1,y1,w1,h1 = 0,0,175,75

        # Draw black background rectangle

        cv2.rectangle(frame, (x1, x1),(x1 + w1, y1 + h1), (0,0,0), -1)

        # Add text

        cv2.putText(frame,status,    (x1    +    int(w1/10),y1    +    int(h1/2)),
    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,0,255), 2)
```

```
        cv2.putText(frame,status,(100, 150),font, 3,(0, 0, 255),2,cv2.LINE_4)

        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255))

    elif (np.argmax(Predictions)== 2):
        status = "Fear"

        x1,y1,w1,h1 = 0,0,175,75
        # Draw black background rectangle
        cv2.rectangle(frame, (x1, x1), (x1+ w1, y1 + h1), (0,0,0), -1)
        # Add text
        cv2.putText(frame, status, (x1 + int(w1/10),y1 + int(h1/2)),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,0,255),2)

        cv2.putText(frame,status,(100,150),font, 3,(0, 0, 255),2,cv2.LINE_4)

        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255))
    elif (np.argmax(Predictions)==3):
        status = "Happy"

        x1,y1,w1,h1 = 0,0,175,75
        #Draw black background rectangle
        cv2.rectangle(frame, (x1, x1),(x1 + w1, y1 + h1), (0,0,0), -1)
        # Add text
        cv2.putText(frame, status, (x1 + int(w1/10),y1 + int(h1/2)) ,
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,0,255), 2)

        cv2.putText(frame,status,(100,150),font, 3,(0,0, 255),2,cv2.LINE_4)

        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255))

    elif (np.argmax(Predictions)==4):
        status = "Sad"

        x1,y1,w1,h1 = 0,0,175,75
        #Draw black background rectangle
        cv2.rectangle(frame, (x1, x1),(x1 + w1, y1 + h1), (0,0,0), -1)

        # Add text

        cv2.putText(frame, status, (x1 + int(w1/10),y1 + int(h1/2)) ,
    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,0,255), 2)


        cv2.putText(frame,status,(100,150),font, 3,(0,0, 255),2,cv2.LINE_4)


        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255))


      elif (np.argmax(Predictions)==5):

        status = "Surprise"


        x1,y1,w1,h1 = 0,0,175,75

        #Draw black background rectangle

        cv2.rectangle(frame, (x1, x1),(x1 + w1, y1 + h1), (0,0,0), -1)

        # Add text

        cv2.putText(frame, status, (x1 + int(w1/10),y1 + int(h1/2)) ,
    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,0,255), 2)
```

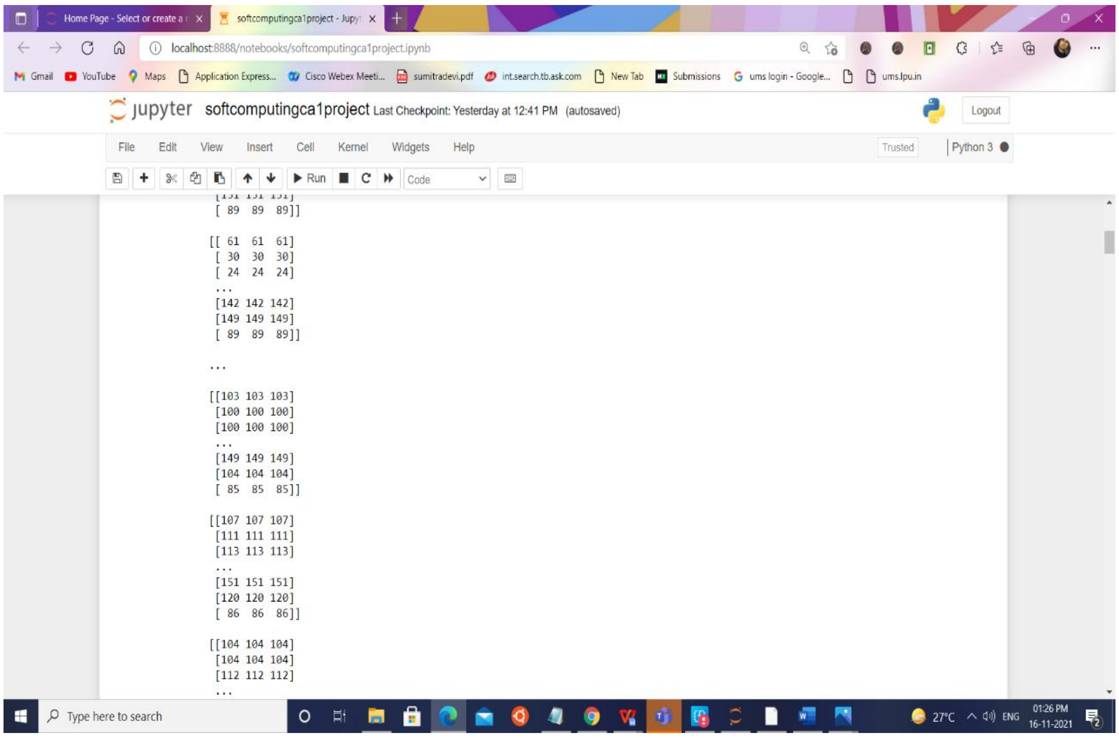
```
        cv2.putText(frame,status,(100,150),font, 3,(0,0, 255),2,cv2.LINE_4)


        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255))



    else:

        status = "Neutral"


        x1,y1,w1,h1 = 0,0,175,75

        #Draw black background rectangle

        cv2.rectangle(frame, (x1, x1),(x1 + w1, y1 + h1), (0,0,0), -1)

        # Add text

        cv2.putText(frame, status, (x1 + int(w1/10),y1 + int(h1/2)) ,
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,0,255), 2)


        cv2.putText(frame,status,(100,150),font, 3,(0,0, 255),2,cv2.LINE_4)


        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255))
```

```
# gray= cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

# print(faceCascade.empty())

# faces = faceCascade.detectMultiScale(gray,1.1,4)


# Draw a rectangle around the facesQQQQQ

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

 # cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)



# Use putText() ethod for

# inserting text on video




cv2.imshow('Face Emotion Recognition', frame)


if cv2.waitKey(2) & 0xFF == ord('q'):
        break


    cap.release()

    cv2.destroyAllWindows()
```

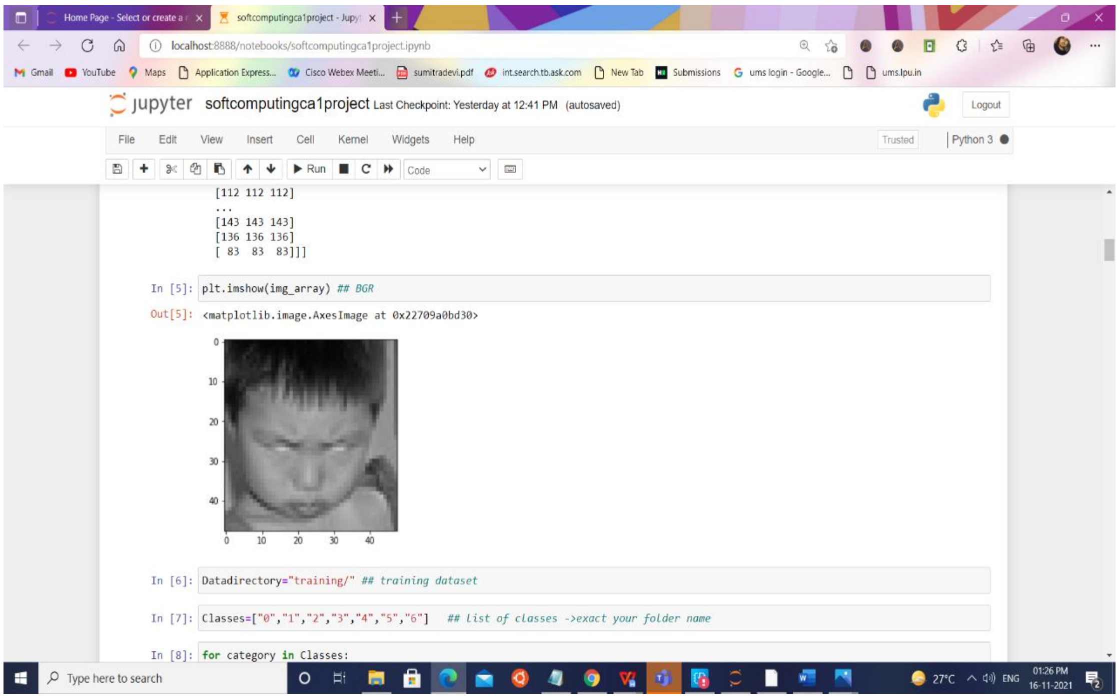
Source Code (where ever applicable) or System Snapshots


jupyter softcomputingca1project Last Checkpoint: Yesterday at 12:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
localhost:8888/notebooks/softcomputingca1project.ipynb
Type here to search
27°C ENG 01:26 PM 16-11-2021

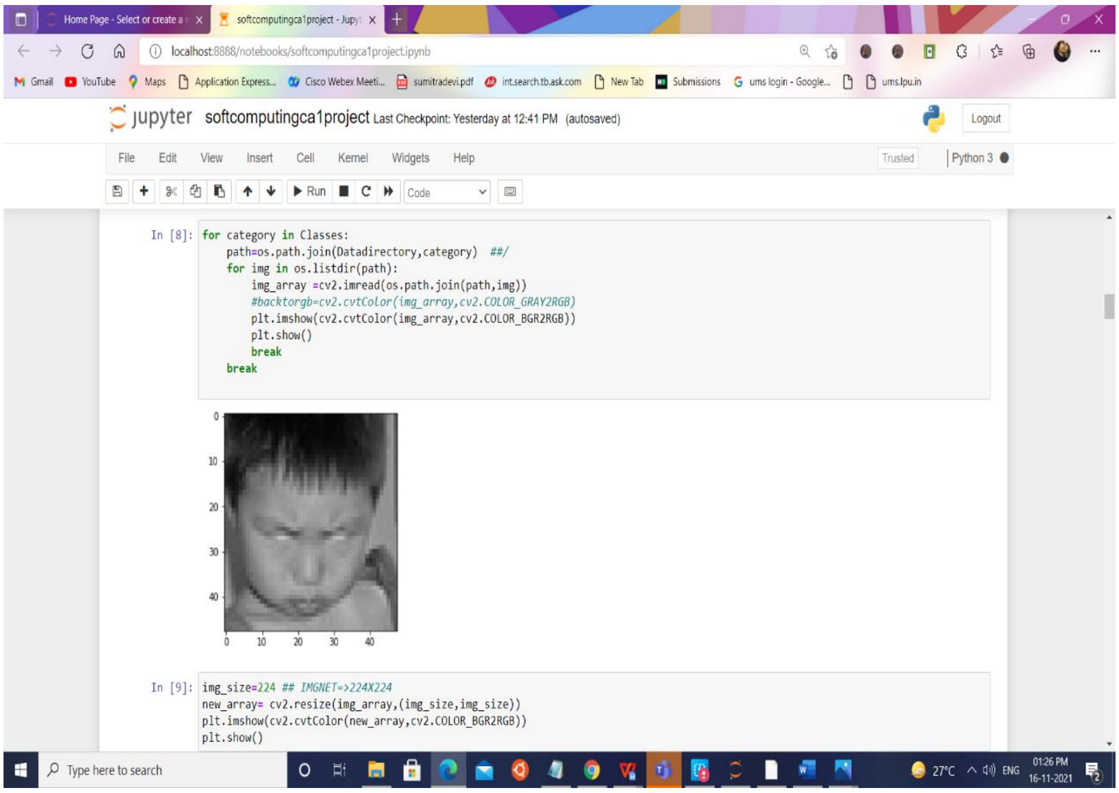
```
In [1]: import tensorflow as tf
        import cv2
        import os
        import numpy as np
        import matplotlib.pyplot as plt

In [2]: img_array=cv2.imread("training/0/Training_10118481.jpg")

In [3]: img_array.shape #rgb

Out[3]: (48, 48, 3)

In [4]: print(img_array)

        [[[ 50  50  50]
          [ 32  32  32]
          [ 15  15  15]
          ...
          [133 133 133]
          [151 151 151]
          [ 86  86  86]]

         [[ 57  57  57]
          [ 34  34  34]
          [ 22  22  22]
          ...
          [138 138 138]
          [151 151 151]
          [ 89  89  89]]
```


localhost:8888/notebooks/softcomputingca1project.ipynb
jupyter softcomputingca1project Last Checkpoint: Yesterday at 12:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search
27°C ENG
01:26 PM
16-11-2021

```
  [151 151 151]
  [ 89  89  89]]

 [[ 61  61  61]
  [ 30  30  30]
  [ 24  24  24]
  ...
  [142 142 142]
  [149 149 149]
  [ 89  89  89]]

 ...

 [[103 103 103]
  [100 100 100]
  [100 100 100]
  ...
  [149 149 149]
  [104 104 104]
  [ 85  85  85]]

 [[107 107 107]
  [111 111 111]
  [113 113 113]
  ...
  [151 151 151]
  [120 120 120]
  [ 86  86  86]]

 [[104 104 104]
  [104 104 104]
  [112 112 112]
  ...
```

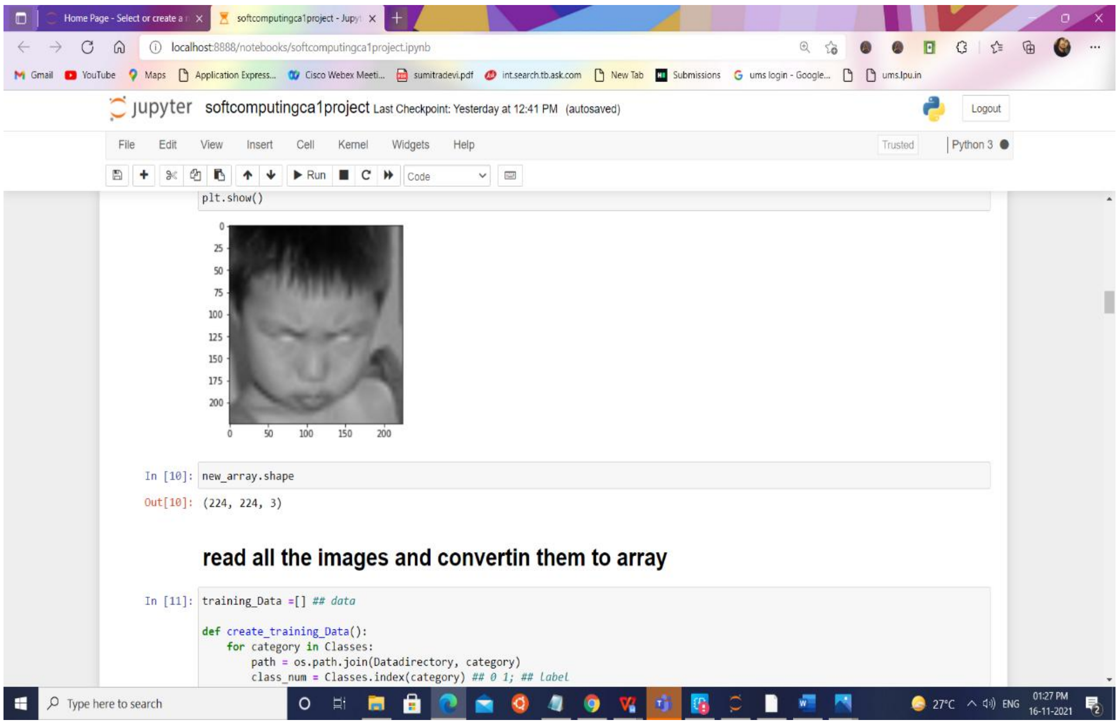

localhost:8888/notebooks/softcomputingca1project.ipynb
jupyter softcomputingca1project Last Checkpoint: Yesterday at 12:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
0
10
20
30
40
Type here to search
27°C ENG
01:26 PM
16-11-2021

```
  [112 112 112]
  ...
  [143 143 143]
  [136 136 136]
  [ 83  83  83]]]

In [5]: plt.imshow(img_array) ## BGR

Out[5]: <matplotlib.image.AxesImage at 0x22709a0bd30>

In [6]: Datadirectory="training/"  ## training dataset

In [7]: Classes=["0","1","2","3","4","5","6"]   ## list of classes ->exact your folder name

In [8]: for category in Classes:
```


jupyter softcomputingca1project Last Checkpoint: Yesterday at 12:41 PM (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
In [8]: for category in Classes:
            path=os.path.join(Datadirectory,category)  ##/
            for img in os.listdir(path):
                img_array =cv2.imread(os.path.join(path,img))
                #backtorgb=cv2.cvtColor(img_array,cv2.COLOR_GRAY2RGB)
                plt.imshow(cv2.cvtColor(img_array,cv2.COLOR_BGR2RGB))
                plt.show()
                break
            break
```

```
In [9]: img_size=224 ## IMGNET=>224X224
        new_array= cv2.resize(img_array,(img_size,img_size))
        plt.imshow(cv2.cvtColor(new_array,cv2.COLOR_BGR2RGB))
        plt.show()
```

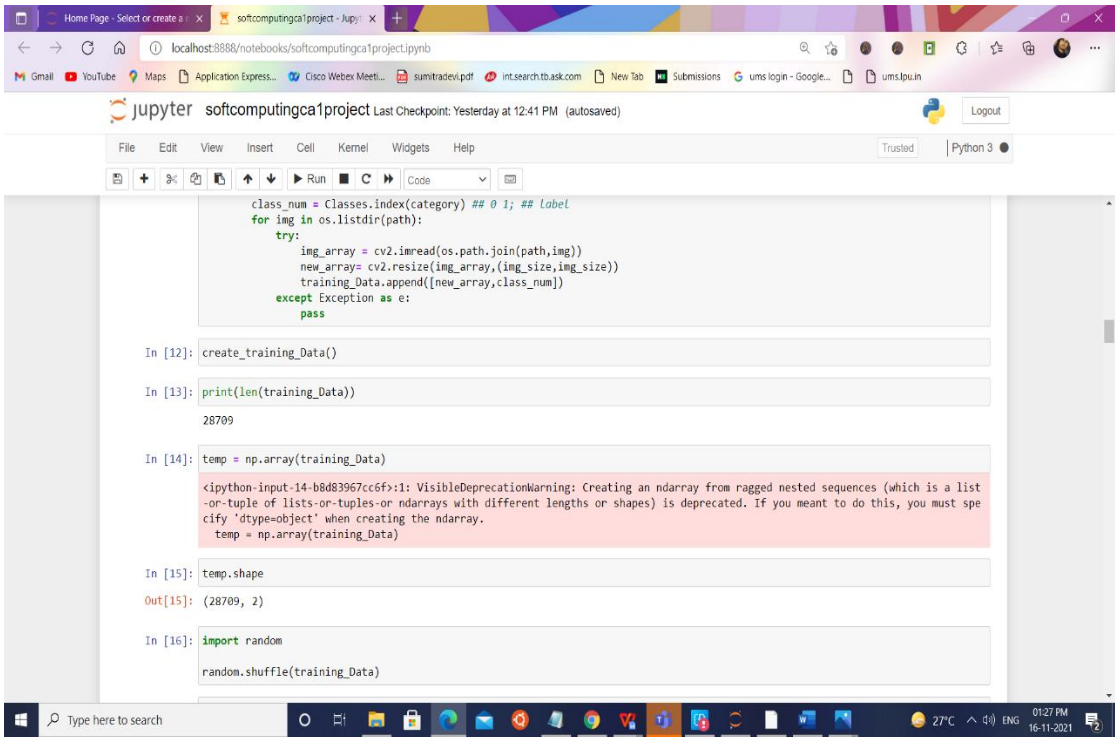

jupyter softcomputingca1project Last Checkpoint: Yesterday at 12:41 PM (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
plt.show()
```

```
In [10]: new_array.shape
Out[10]: (224, 224, 3)
```

read all the images and convertin them to array

```
In [11]: training_Data =[] ## data

         def create_training_Data():
             for category in Classes:
                 path = os.path.join(Datadirectory, category)
                 class_num = Classes.index(category) ## 0 1; ## Label
```


Home Page - Select or create a
softcomputingca1project - Jupy
localhost:8888/notebooks/softcomputingca1project.ipynb
jupyter softcomputingca1project Last Checkpoint: Yesterday at 12:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search
27°C ENG
01:27 PM
16-11-2021

```
        class_num = Classes.index(category) ## 0 1; ## label
        for img in os.listdir(path):
            try:
                img_array = cv2.imread(os.path.join(path,img))
                new_array= cv2.resize(img_array,(img_size,img_size))
                training_Data.append([new_array,class_num])
            except Exception as e:
                pass
```

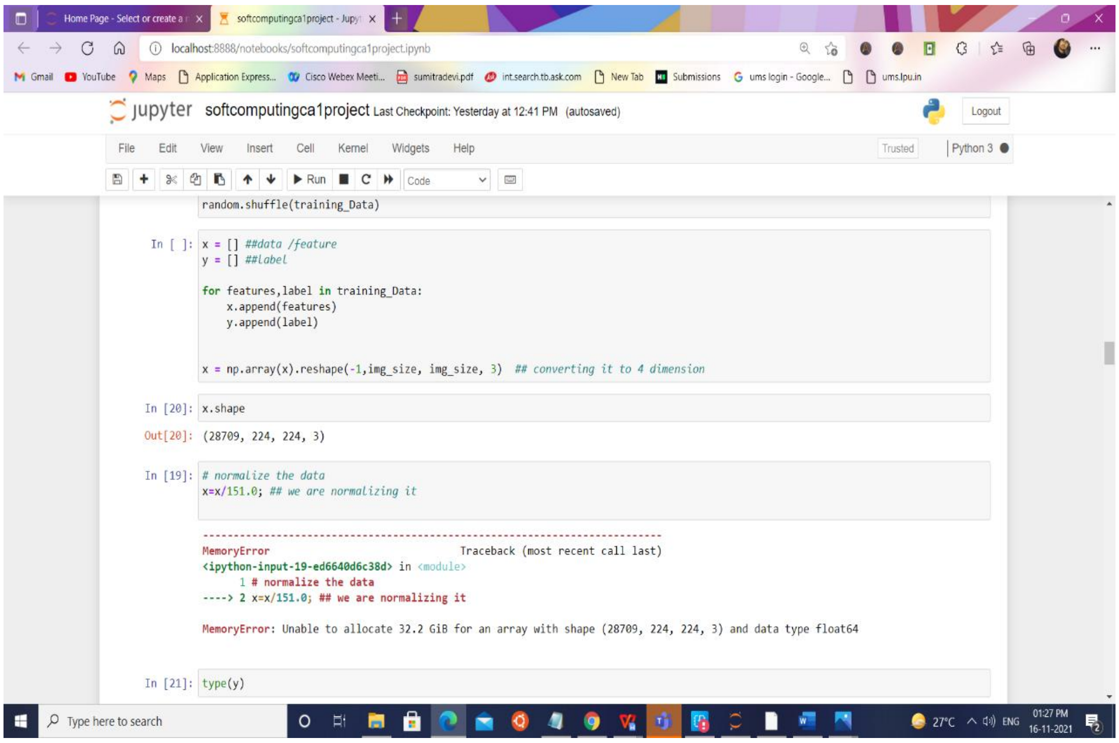
```
In [12]: create_training_Data()
```

```
In [13]: print(len(training_Data))

         28709
```

```
In [14]: temp = np.array(training_Data)

         <ipython-input-14-b8d83967cc6f>:1: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list
         -or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must spe
         cify 'dtype=object' when creating the ndarray.
           temp = np.array(training_Data)
```

```
In [15]: temp.shape
Out[15]: (28709, 2)
```

```
In [16]: import random

         random.shuffle(training_Data)
```


Home Page - Select or create a
softcomputingca1project - Jupy
localhost:8888/notebooks/softcomputingca1project.ipynb
jupyter softcomputingca1project Last Checkpoint: Yesterday at 12:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search
27°C ENG
01:27 PM
16-11-2021

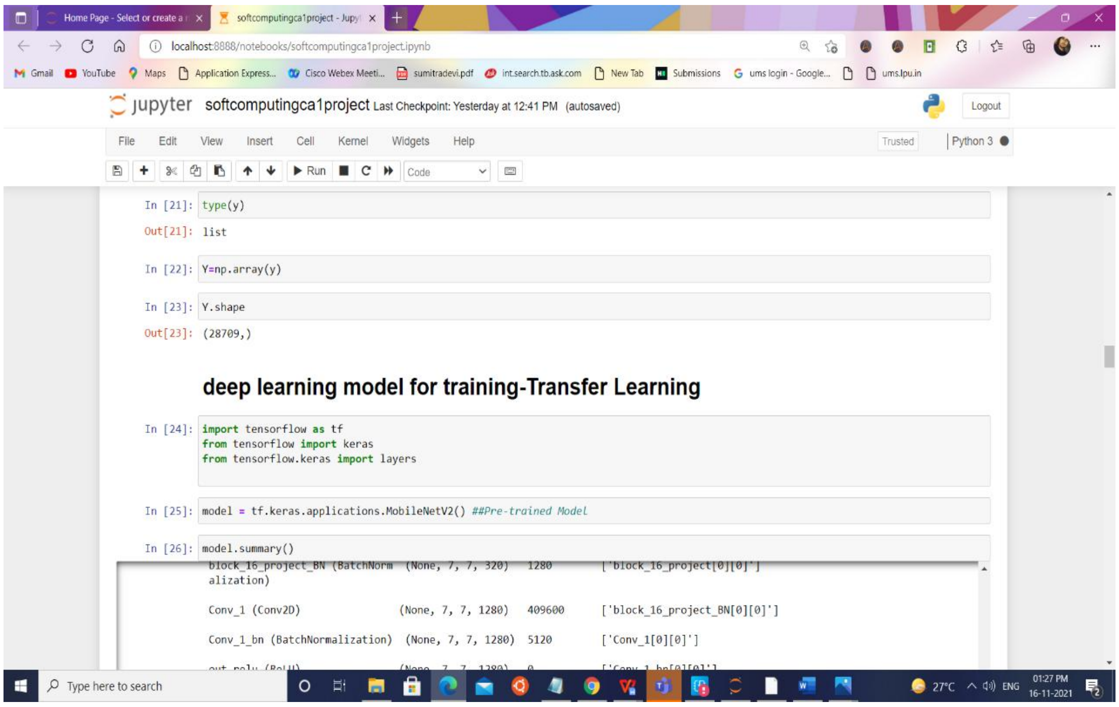
```
         random.shuffle(training_Data)
```

```
In [ ]: x = [] ##data /feature
        y = [] ##label

        for features,label in training_Data:
            x.append(features)
            y.append(label)


        x = np.array(x).reshape(-1,img_size, img_size, 3)  ## converting it to 4 dimension
```

```
In [20]: x.shape
Out[20]: (28709, 224, 224, 3)
```

```
In [19]: # normalize the data
         x=x/151.0; ## we are normalizing it

         ---------------------------------------------------------------------------
         MemoryError                               Traceback (most recent call last)
         <ipython-input-19-ed6640d6c38d> in <module>
               1 # normalize the data
         ----> 2 x=x/151.0; ## we are normalizing it

         MemoryError: Unable to allocate 32.2 GiB for an array with shape (28709, 224, 224, 3) and data type float64
```

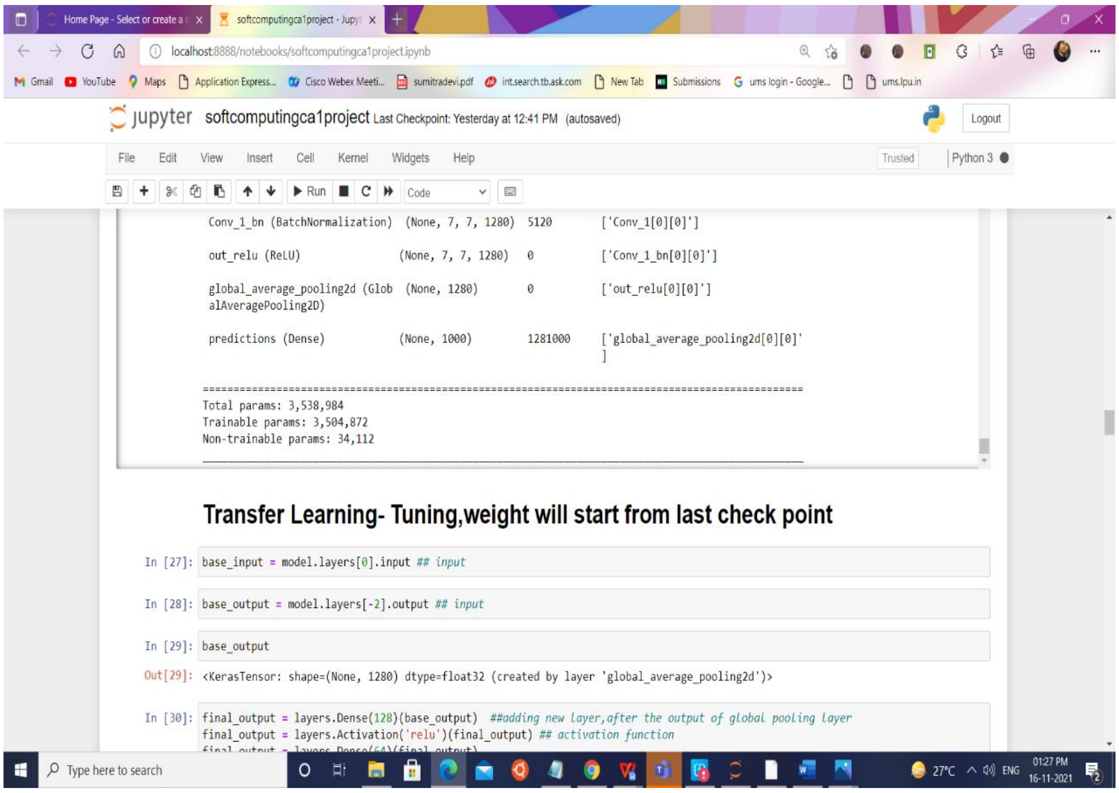
```
In [21]: type(y)
```


jupyter softcomputingca1project Last Checkpoint: Yesterday at 12:41 PM (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
In [21]: type(y)
Out[21]: list

In [22]: Y=np.array(y)

In [23]: Y.shape
Out[23]: (28709,)
```

deep learning model for training-Transfer Learning

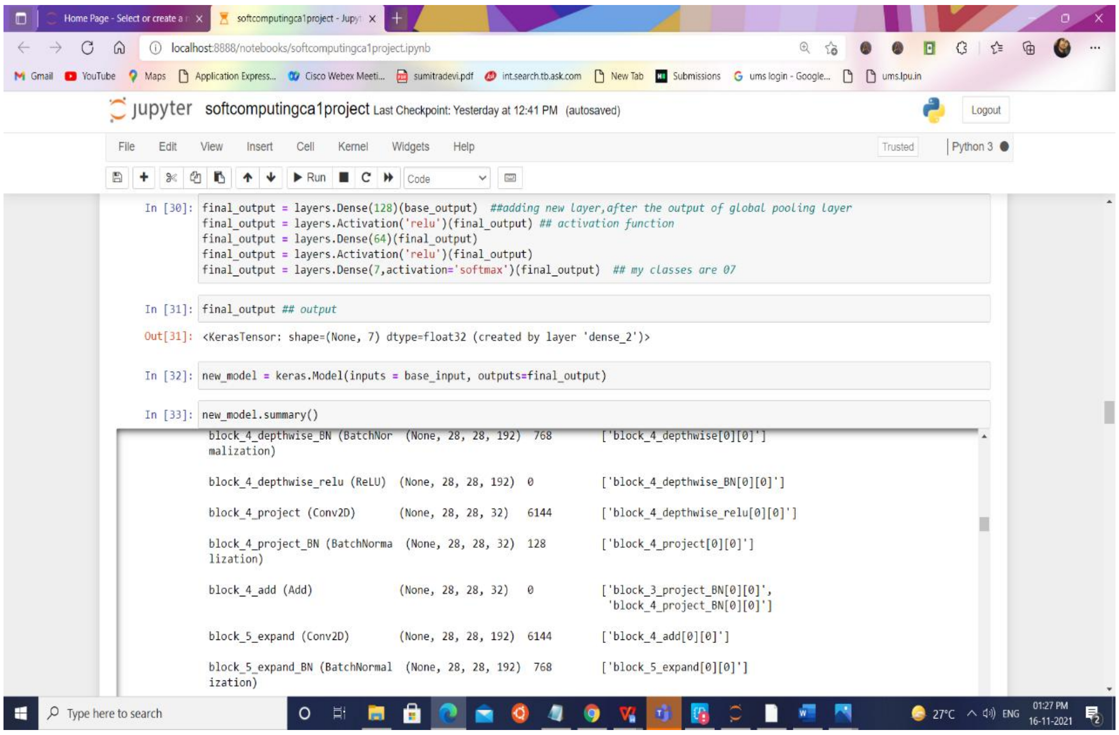
```
In [24]: import tensorflow as tf
         from tensorflow import keras
         from tensorflow.keras import layers

In [25]: model = tf.keras.applications.MobileNetV2() ##Pre-trained Model

In [26]: model.summary()
          block_16_project_BN (BatchNorm  (None, 7, 7, 320)   1280        ['block_16_project[0][0]']
          alization)

          Conv_1 (Conv2D)                (None, 7, 7, 1280)  409600      ['block_16_project_BN[0][0]']

          Conv_1_bn (BatchNormalization)  (None, 7, 7, 1280)  5120        ['Conv_1[0][0]']

          out_relu (ReLU)                (None, 7, 7, 1280)  0           ['Conv_1_bn[0][0]']
```


jupyter softcomputingca1project Last Checkpoint: Yesterday at 12:41 PM (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
          Conv_1_bn (BatchNormalization)  (None, 7, 7, 1280)  5120        ['Conv_1[0][0]']

          out_relu (ReLU)                (None, 7, 7, 1280)  0           ['Conv_1_bn[0][0]']

          global_average_pooling2d (Glob  (None, 1280)        0           ['out_relu[0][0]']
          alAveragePooling2D)

          predictions (Dense)            (None, 1000)        1281000     ['global_average_pooling2d[0][0]'
                                                                          ]

         ==================================================================================================
         Total params: 3,538,984
         Trainable params: 3,504,872
         Non-trainable params: 34,112
         __________________________________________________________________________________________________
```

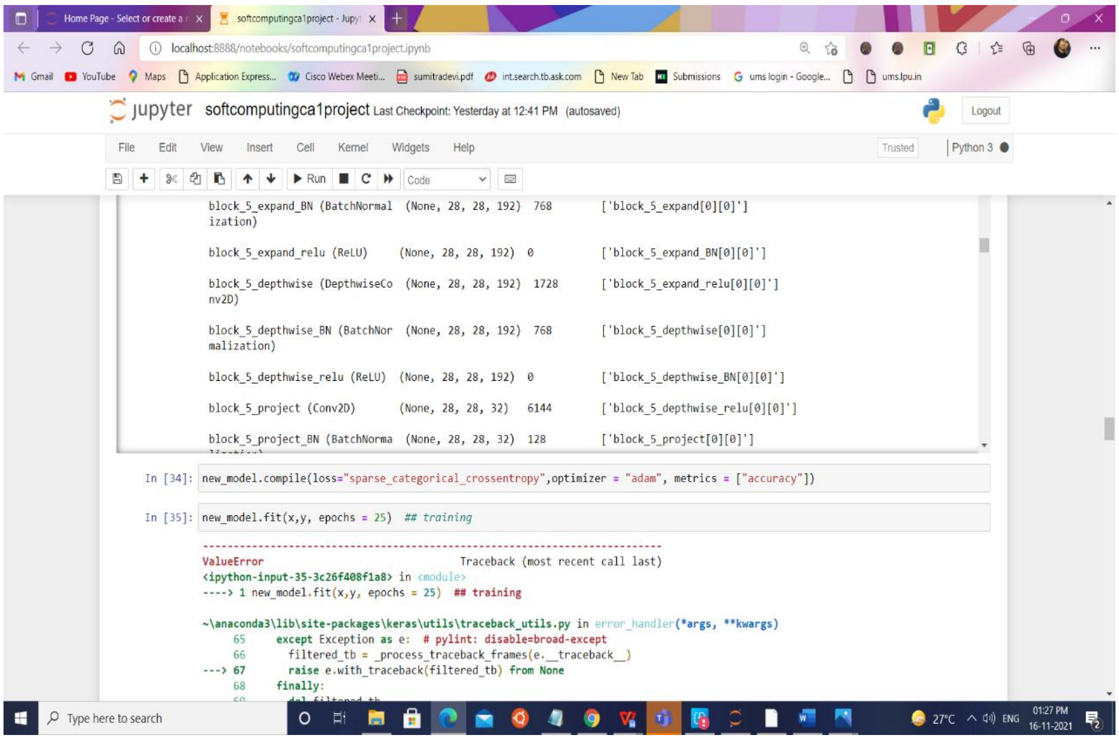
Transfer Learning- Tuning,weight will start from last check point

```
In [27]: base_input = model.layers[0].input ## input

In [28]: base_output = model.layers[-2].output ## input

In [29]: base_output
Out[29]: <KerasTensor: shape=(None, 1280) dtype=float32 (created by layer 'global_average_pooling2d')>

In [30]: final_output = layers.Dense(128)(base_output)  ##adding new layer,after the output of global pooling layer
         final_output = layers.Activation('relu')(final_output) ## activation function
         final_output = layers.Dense(64)(final_output)
```



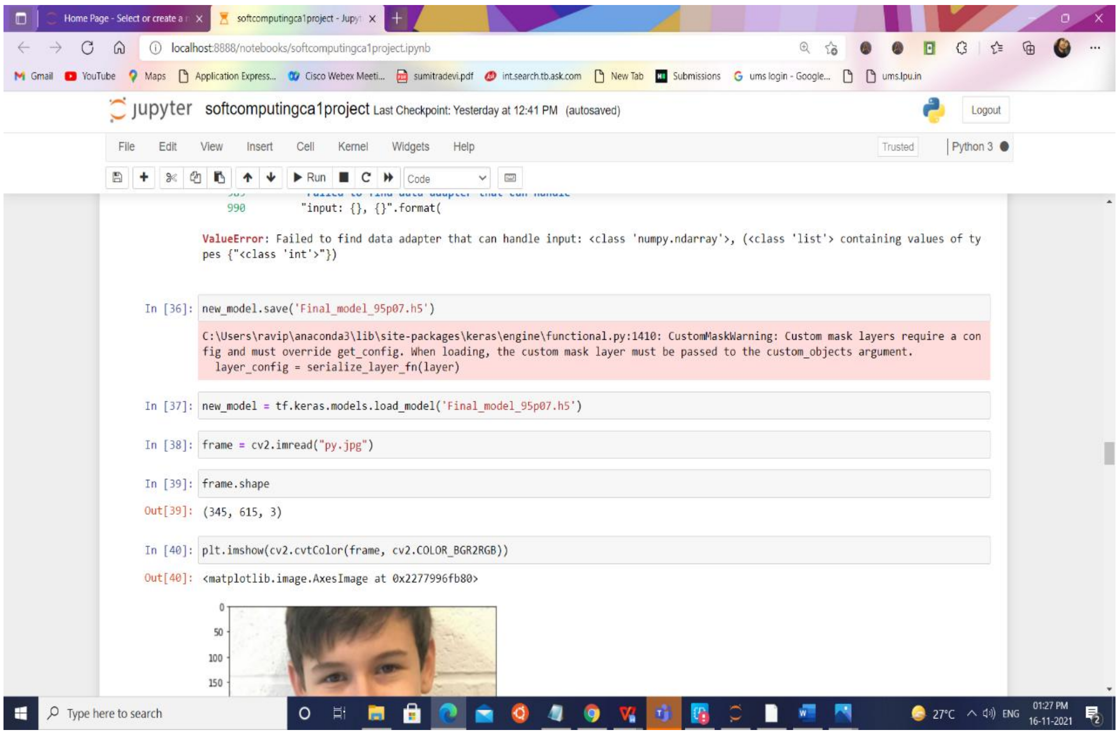
```
In [30]: final_output = layers.Dense(128)(base_output)  ##adding new layer,after the output of global pooling layer
         final_output = layers.Activation('relu')(final_output) ## activation function
         final_output = layers.Dense(64)(final_output)
         final_output = layers.Activation('relu')(final_output)
         final_output = layers.Dense(7,activation='softmax')(final_output)  ## my classes are 07

In [31]: final_output ## output

Out[31]: <KerasTensor: shape=(None, 7) dtype=float32 (created by layer 'dense_2')>

In [32]: new_model = keras.Model(inputs = base_input, outputs=final_output)

In [33]: new_model.summary()
```

```
 block_4_depthwise_BN (BatchNor  (None, 28, 28, 192)  768         ['block_4_depthwise[0][0]']
 malization)

 block_4_depthwise_relu (ReLU)  (None, 28, 28, 192)  0           ['block_4_depthwise_BN[0][0]']

 block_4_project (Conv2D)       (None, 28, 28, 32)   6144        ['block_4_depthwise_relu[0][0]']

 block_4_project_BN (BatchNorma  (None, 28, 28, 32)  128         ['block_4_project[0][0]']
 lization)

 block_4_add (Add)              (None, 28, 28, 32)   0           ['block_3_project_BN[0][0]',
                                                                  'block_4_project_BN[0][0]']

 block_5_expand (Conv2D)        (None, 28, 28, 192)  6144        ['block_4_add[0][0]']

 block_5_expand_BN (BatchNormal  (None, 28, 28, 192)  768         ['block_5_expand[0][0]']
 ization)
```

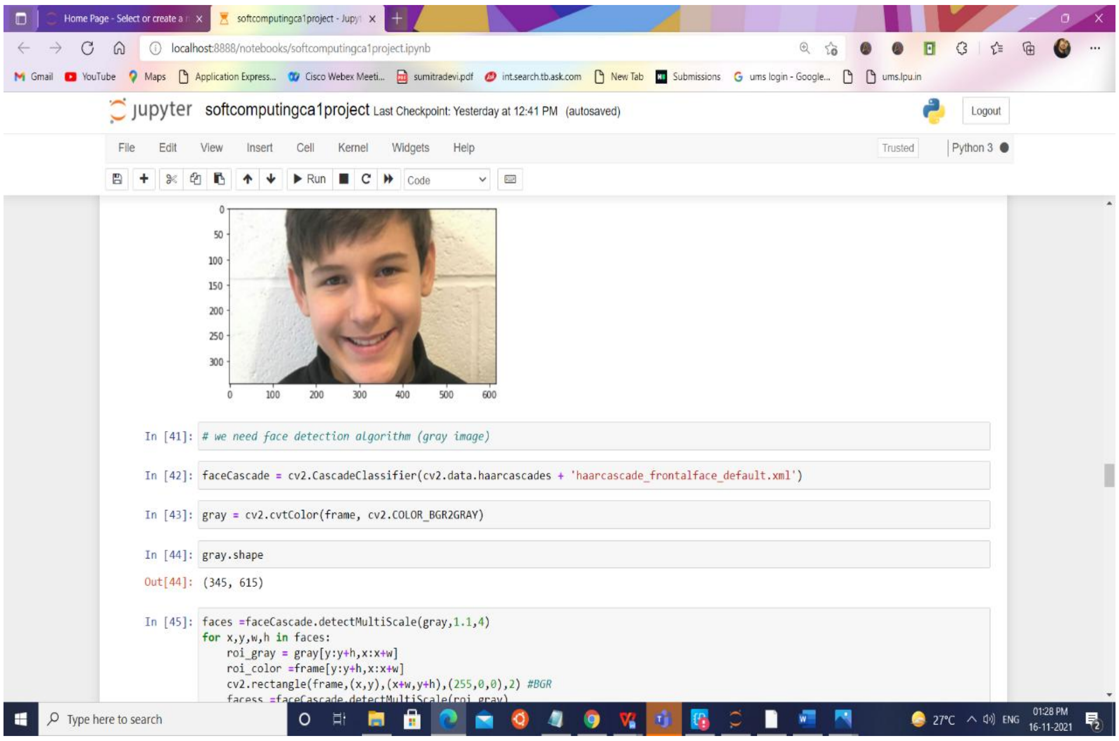


```
 block_5_expand_BN (BatchNormal  (None, 28, 28, 192)  768         ['block_5_expand[0][0]']
 ization)

 block_5_expand_relu (ReLU)     (None, 28, 28, 192)  0           ['block_5_expand_BN[0][0]']

 block_5_depthwise (DepthwiseCo  (None, 28, 28, 192)  1728        ['block_5_expand_relu[0][0]']
 nv2D)

 block_5_depthwise_BN (BatchNor  (None, 28, 28, 192)  768         ['block_5_depthwise[0][0]']
 malization)

 block_5_depthwise_relu (ReLU)  (None, 28, 28, 192)  0           ['block_5_depthwise_BN[0][0]']

 block_5_project (Conv2D)       (None, 28, 28, 32)   6144        ['block_5_depthwise_relu[0][0]']

 block_5_project_BN (BatchNorma  (None, 28, 28, 32)  128         ['block_5_project[0][0]']
```

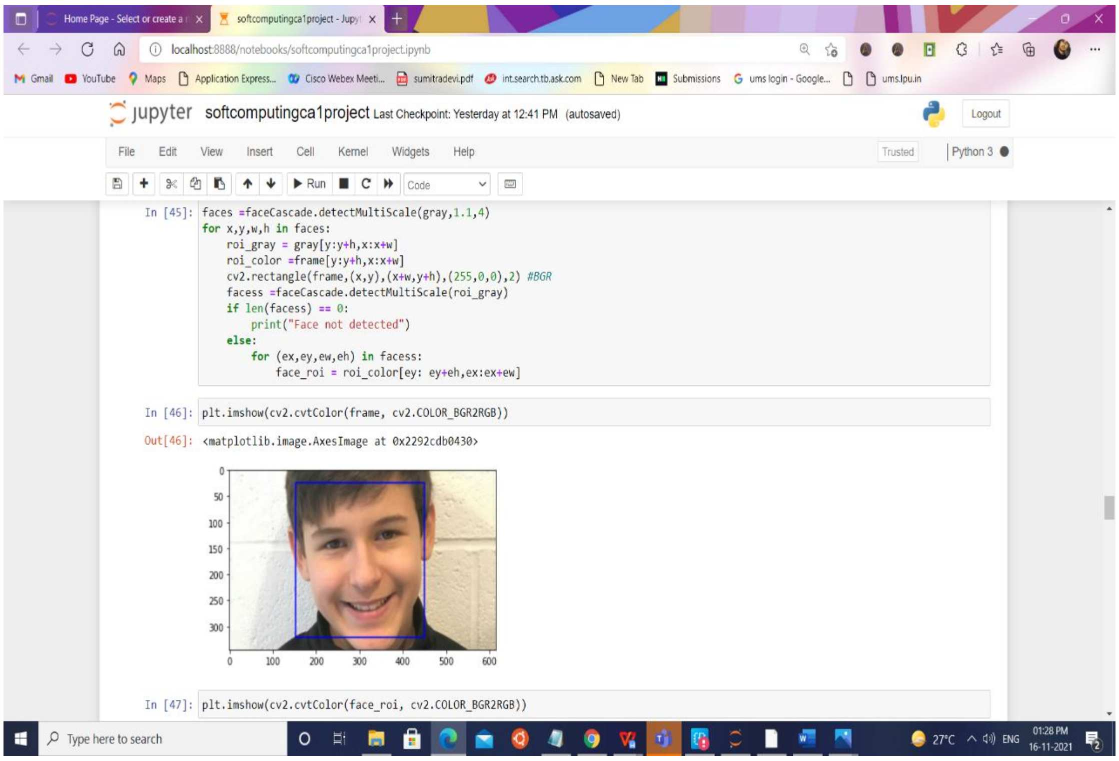
```
In [34]: new_model.compile(loss="sparse_categorical_crossentropy",optimizer = "adam", metrics = ["accuracy"])

In [35]: new_model.fit(x,y, epochs = 25)  ## training

         ---------------------------------------------------------------------------
         ValueError                                Traceback (most recent call last)
         <ipython-input-35-3c26f408f1a8> in <module>
         ----> 1 new_model.fit(x,y, epochs = 25)  ## training

         ~\anaconda3\lib\site-packages\keras\utils\traceback_utils.py in error_handler(*args, **kwargs)
              65     except Exception as e:  # pylint: disable=broad-except
              66       filtered_tb = _process_traceback_frames(e.__traceback__)
         ---> 67       raise e.with_traceback(filtered_tb) from None
              68     finally:
```


Home Page - Select or create a
softcomputingca1project - Jupyter
localhost:8888/notebooks/softcomputingca1project.ipynb
jupyter softcomputingca1project Last Checkpoint: Yesterday at 12:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search
27°C ENG
01:27 PM
16-11-2021

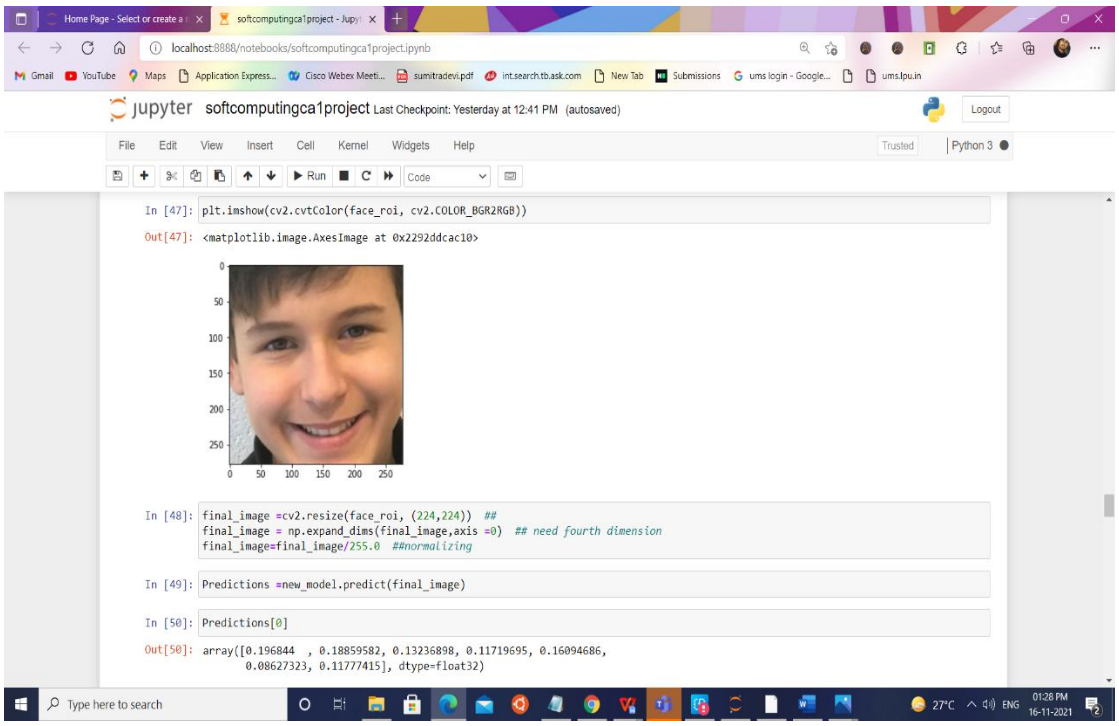
```
    990         "input: {}, {}".format(

ValueError: Failed to find data adapter that can handle input: <class 'numpy.ndarray'>, (<class 'list'> containing values of ty
pes {"<class 'int'>"})

In [36]: new_model.save('Final_model_95p07.h5')

C:\Users\ravip\anaconda3\lib\site-packages\keras\engine\functional.py:1410: CustomMaskWarning: Custom mask layers require a con
fig and must override get_config. When loading, the custom mask layer must be passed to the custom_objects argument.
  layer_config = serialize_layer_fn(layer)

In [37]: new_model = tf.keras.models.load_model('Final_model_95p07.h5')

In [38]: frame = cv2.imread("py.jpg")

In [39]: frame.shape

Out[39]: (345, 615, 3)

In [40]: plt.imshow(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))

Out[40]: <matplotlib.image.AxesImage at 0x2277996fb80>
```


Home Page - Select or create a
softcomputingca1project - Jupyter
localhost:8888/notebooks/softcomputingca1project.ipynb
jupyter softcomputingca1project Last Checkpoint: Yesterday at 12:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
0
50
100
150
200
250
300
0
100
200
300
400
500
600
Type here to search
27°C ENG
01:28 PM
16-11-2021

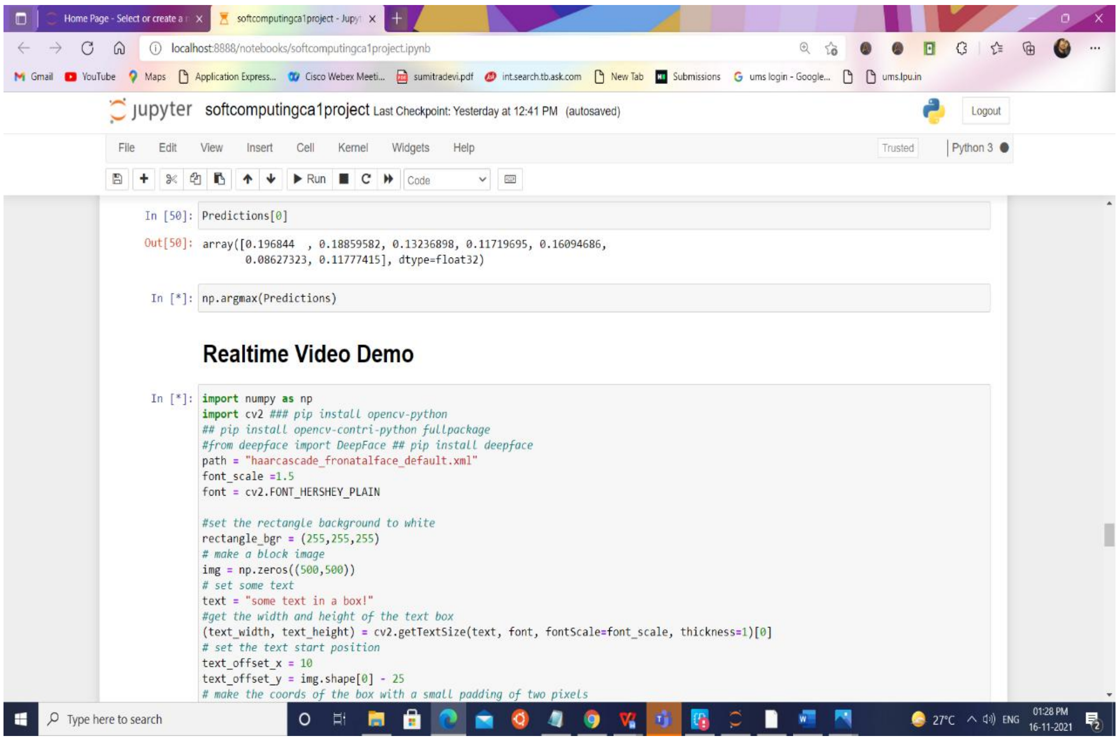
```
In [41]: # we need face detection algorithm (gray image)

In [42]: faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')

In [43]: gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

In [44]: gray.shape

Out[44]: (345, 615)

In [45]: faces =faceCascade.detectMultiScale(gray,1.1,4)
         for x,y,w,h in faces:
             roi_gray = gray[y:y+h,x:x+w]
             roi_color =frame[y:y+h,x:x+w]
             cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2) #BGR
```


jupyter softcomputingca1project Last Checkpoint: Yesterday at 12:41 PM (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Logout

```
In [45]: faces =faceCascade.detectMultiScale(gray,1.1,4)
         for x,y,w,h in faces:
             roi_gray = gray[y:y+h,x:x+w]
             roi_color =frame[y:y+h,x:x+w]
             cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2) #BGR
             facess =faceCascade.detectMultiScale(roi_gray)
             if len(facess) == 0:
                 print("Face not detected")
             else:
                 for (ex,ey,ew,eh) in facess:
                     face_roi = roi_color[ey: ey+eh,ex:ex+ew]

In [46]: plt.imshow(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))

Out[46]: <matplotlib.image.AxesImage at 0x2292cdb0430>

In [47]: plt.imshow(cv2.cvtColor(face_roi, cv2.COLOR_BGR2RGB))
```


jupyter softcomputingca1project Last Checkpoint: Yesterday at 12:41 PM (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Logout

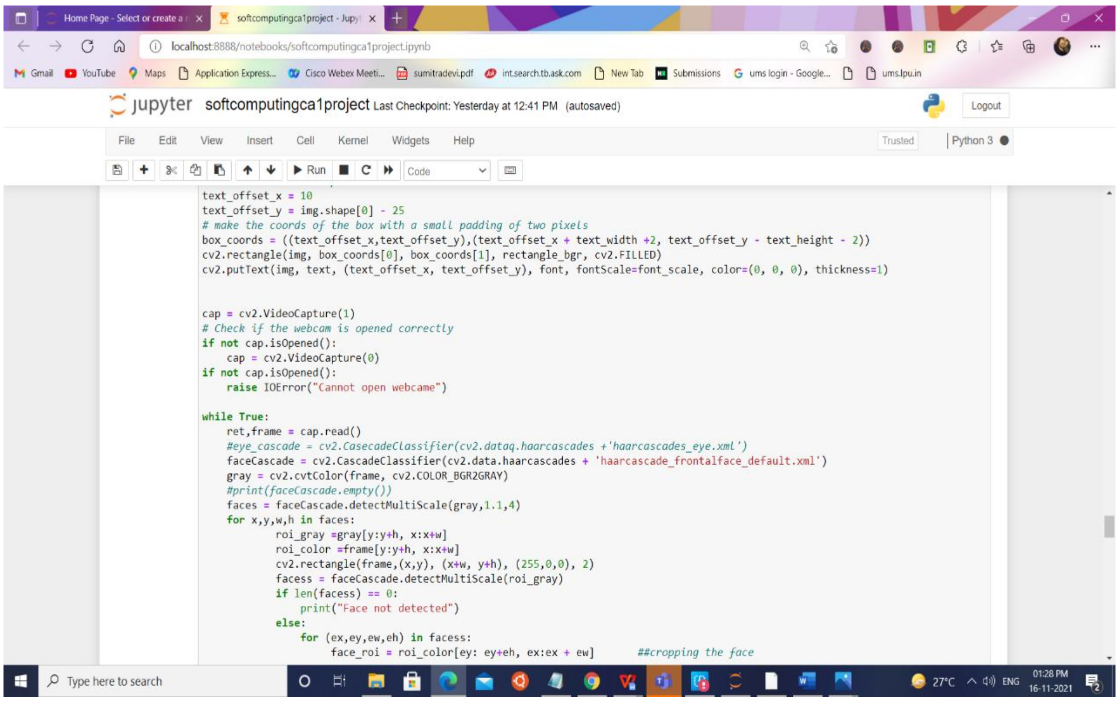
```
In [47]: plt.imshow(cv2.cvtColor(face_roi, cv2.COLOR_BGR2RGB))

Out[47]: <matplotlib.image.AxesImage at 0x2292ddcac10>

In [48]: final_image =cv2.resize(face_roi, (224,224))  ##
         final_image = np.expand_dims(final_image,axis =0)  ## need fourth dimension
         final_image=final_image/255.0  ##normalizing

In [49]: Predictions =new_model.predict(final_image)

In [50]: Predictions[0]

Out[50]: array([0.196844  , 0.18859582, 0.13236898, 0.11719695, 0.16094686,
                0.08627323, 0.11777415], dtype=float32)
```



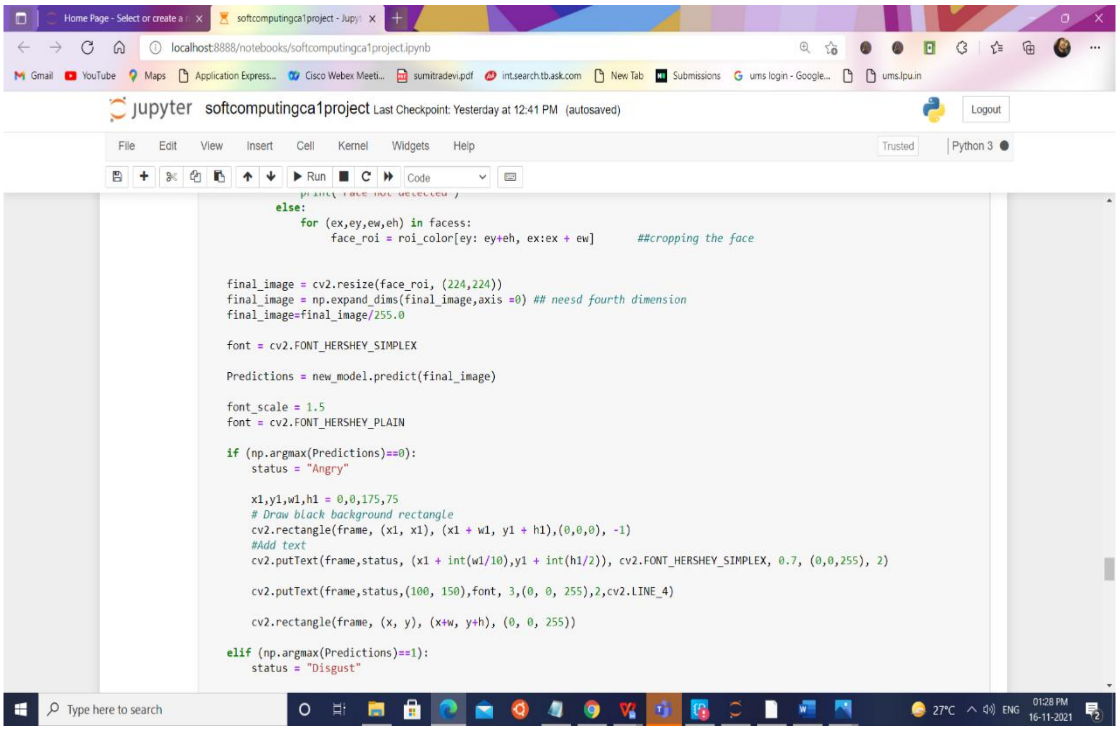
```
In [50]: Predictions[0]

Out[50]: array([0.196844  , 0.18859582, 0.13236898, 0.11719695, 0.16094686,
                0.08627323, 0.11777415], dtype=float32)

In [*]: np.argmax(Predictions)
```

Realtime Video Demo

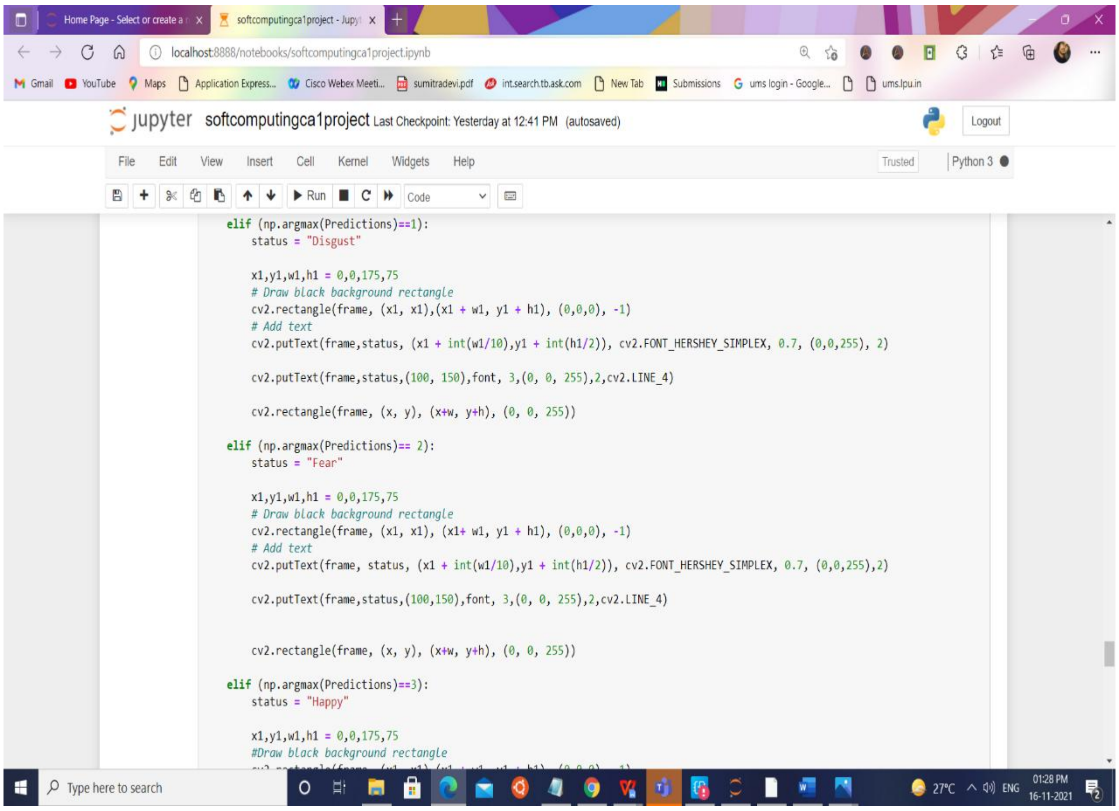
```
In [*]: import numpy as np
        import cv2 ### pip install opencv-python
        ## pip install opencv-contri-python fullpackage
        #from deepface import DeepFace ## pip install deepface
        path = "haarcascade_fronatalface_default.xml"
        font_scale =1.5
        font = cv2.FONT_HERSHEY_PLAIN

        #set the rectangle background to white
        rectangle_bgr = (255,255,255)
        # make a block image
        img = np.zeros((500,500))
        # set some text
        text = "some text in a box!"
        #get the width and height of the text box
        (text_width, text_height) = cv2.getTextSize(text, font, fontScale=font_scale, thickness=1)[0]
        # set the text start position
        text_offset_x = 10
        text_offset_y = img.shape[0] - 25
        # make the coords of the box with a small padding of two pixels
```



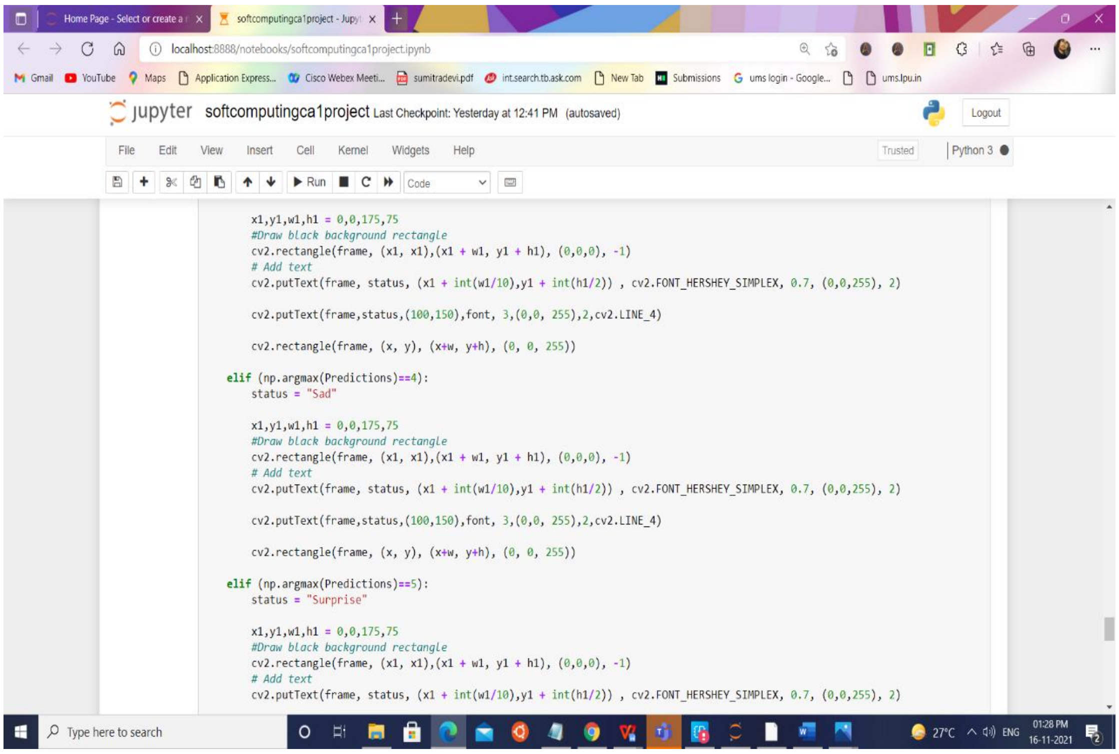
```
text_offset_x = 10
text_offset_y = img.shape[0] - 25
# make the coords of the box with a small padding of two pixels
box_coords = ((text_offset_x,text_offset_y),(text_offset_x + text_width +2, text_offset_y - text_height - 2))
cv2.rectangle(img, box_coords[0], box_coords[1], rectangle_bgr, cv2.FILLED)
cv2.putText(img, text, (text_offset_x, text_offset_y), font, fontScale=font_scale, color=(0, 0, 0), thickness=1)


cap = cv2.VideoCapture(1)
# Check if the webcam is opened correctly
if not cap.isOpened():
    cap = cv2.VideoCapture(0)
if not cap.isOpened():
    raise IOError("Cannot open webcame")

while True:
    ret,frame = cap.read()
    #eye_cascade = cv2.CasecadeClassifier(cv2.dataq.haarcascades +'haarcascades_eye.xml')
    faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    #print(faceCascade.empty())
    faces = faceCascade.detectMultiScale(gray,1.1,4)
    for x,y,w,h in faces:
            roi_gray =gray[y:y+h, x:x+w]
            roi_color =frame[y:y+h, x:x+w]
            cv2.rectangle(frame,(x,y), (x+w, y+h), (255,0,0), 2)
            facess = faceCascade.detectMultiScale(roi_gray)
            if len(facess) == 0:
                print("Face not detected")
            else:
                for (ex,ey,ew,eh) in facess:
                    face_roi = roi_color[ey: ey+eh, ex:ex + ew]       ##cropping the face
```


jupyter softcomputingca1project Last Checkpoint: Yesterday at 12:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

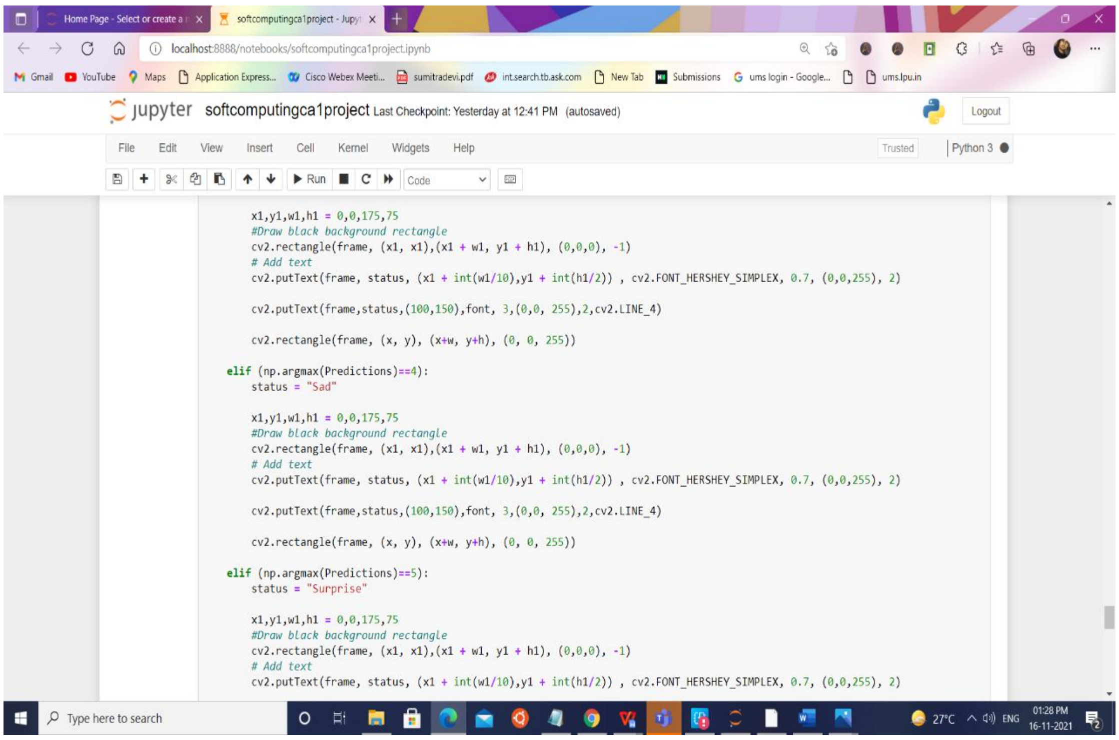
```
        else:
            for (ex,ey,ew,eh) in facess:
                face_roi = roi_color[ey: ey+eh, ex:ex + ew]      ##cropping the face


final_image = cv2.resize(face_roi, (224,224))
final_image = np.expand_dims(final_image,axis =0) ## neesd fourth dimension
final_image=final_image/255.0

font = cv2.FONT_HERSHEY_SIMPLEX

Predictions = new_model.predict(final_image)

font_scale = 1.5
font = cv2.FONT_HERSHEY_PLAIN

if (np.argmax(Predictions)==0):
    status = "Angry"

    x1,y1,w1,h1 = 0,0,175,75
    # Draw black background rectangle
    cv2.rectangle(frame, (x1, x1), (x1 + w1, y1 + h1),(0,0,0), -1)
    #Add text
    cv2.putText(frame,status, (x1 + int(w1/10),y1 + int(h1/2)), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,0,255), 2)

    cv2.putText(frame,status,(100, 150),font, 3,(0, 0, 255),2,cv2.LINE_4)

    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255))

elif (np.argmax(Predictions)==1):
    status = "Disgust"
```


jupyter softcomputingca1project Last Checkpoint: Yesterday at 12:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

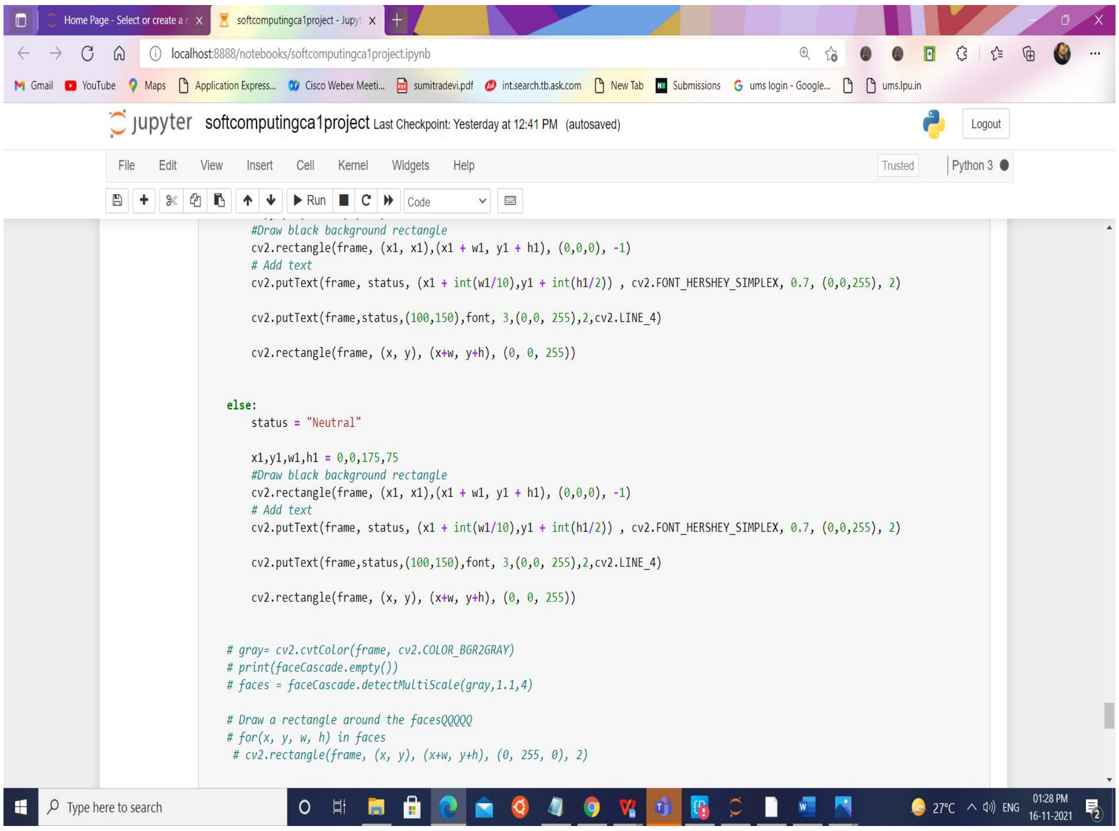
```
elif (np.argmax(Predictions)==1):
    status = "Disgust"

    x1,y1,w1,h1 = 0,0,175,75
    # Draw black background rectangle
    cv2.rectangle(frame, (x1, x1),(x1 + w1, y1 + h1), (0,0,0), -1)
    # Add text
    cv2.putText(frame,status, (x1 + int(w1/10),y1 + int(h1/2)), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,0,255), 2)

    cv2.putText(frame,status,(100, 150),font, 3,(0, 0, 255),2,cv2.LINE_4)

    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255))

elif (np.argmax(Predictions)== 2):
    status = "Fear"

    x1,y1,w1,h1 = 0,0,175,75
    # Draw black background rectangle
    cv2.rectangle(frame, (x1, x1), (x1+ w1, y1 + h1), (0,0,0), -1)
    # Add text
    cv2.putText(frame, status, (x1 + int(w1/10),y1 + int(h1/2)), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,0,255),2)

    cv2.putText(frame,status,(100,150),font, 3,(0, 0, 255),2,cv2.LINE_4)


    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255))

elif (np.argmax(Predictions)==3):
    status = "Happy"

    x1,y1,w1,h1 = 0,0,175,75
    #Draw black background rectangle
```


jupyter softcomputingca1project Last Checkpoint: Yesterday at 12:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

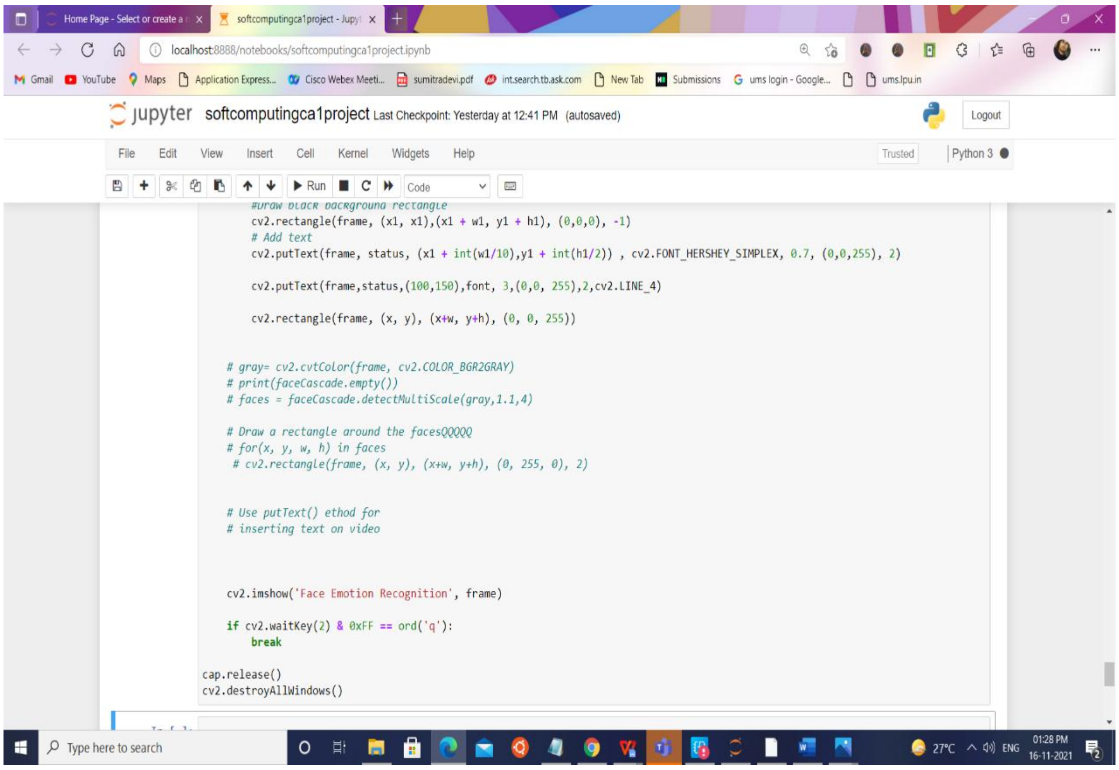
```
        x1,y1,w1,h1 = 0,0,175,75
        #Draw black background rectangle
        cv2.rectangle(frame, (x1, x1),(x1 + w1, y1 + h1), (0,0,0), -1)
        # Add text
        cv2.putText(frame, status, (x1 + int(w1/10),y1 + int(h1/2)) , cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,0,255), 2)

        cv2.putText(frame,status,(100,150),font, 3,(0,0, 255),2,cv2.LINE_4)

        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255))

    elif (np.argmax(Predictions)==4):
        status = "Sad"

        x1,y1,w1,h1 = 0,0,175,75
        #Draw black background rectangle
        cv2.rectangle(frame, (x1, x1),(x1 + w1, y1 + h1), (0,0,0), -1)
        # Add text
        cv2.putText(frame, status, (x1 + int(w1/10),y1 + int(h1/2)) , cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,0,255), 2)

        cv2.putText(frame,status,(100,150),font, 3,(0,0, 255),2,cv2.LINE_4)

        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255))

    elif (np.argmax(Predictions)==5):
        status = "Surprise"

        x1,y1,w1,h1 = 0,0,175,75
        #Draw black background rectangle
        cv2.rectangle(frame, (x1, x1),(x1 + w1, y1 + h1), (0,0,0), -1)
        # Add text
        cv2.putText(frame, status, (x1 + int(w1/10),y1 + int(h1/2)) , cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,0,255), 2)
```


jupyter softcomputingca1project Last Checkpoint: Yesterday at 12:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
        x1,y1,w1,h1 = 0,0,175,75
        #Draw black background rectangle
        cv2.rectangle(frame, (x1, x1),(x1 + w1, y1 + h1), (0,0,0), -1)
        # Add text
        cv2.putText(frame, status, (x1 + int(w1/10),y1 + int(h1/2)) , cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,0,255), 2)

        cv2.putText(frame,status,(100,150),font, 3,(0,0, 255),2,cv2.LINE_4)

        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255))

    elif (np.argmax(Predictions)==4):
        status = "Sad"

        x1,y1,w1,h1 = 0,0,175,75
        #Draw black background rectangle
        cv2.rectangle(frame, (x1, x1),(x1 + w1, y1 + h1), (0,0,0), -1)
        # Add text
        cv2.putText(frame, status, (x1 + int(w1/10),y1 + int(h1/2)) , cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,0,255), 2)

        cv2.putText(frame,status,(100,150),font, 3,(0,0, 255),2,cv2.LINE_4)

        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255))

    elif (np.argmax(Predictions)==5):
        status = "Surprise"

        x1,y1,w1,h1 = 0,0,175,75
        #Draw black background rectangle
        cv2.rectangle(frame, (x1, x1),(x1 + w1, y1 + h1), (0,0,0), -1)
        # Add text
        cv2.putText(frame, status, (x1 + int(w1/10),y1 + int(h1/2)) , cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,0,255), 2)
```

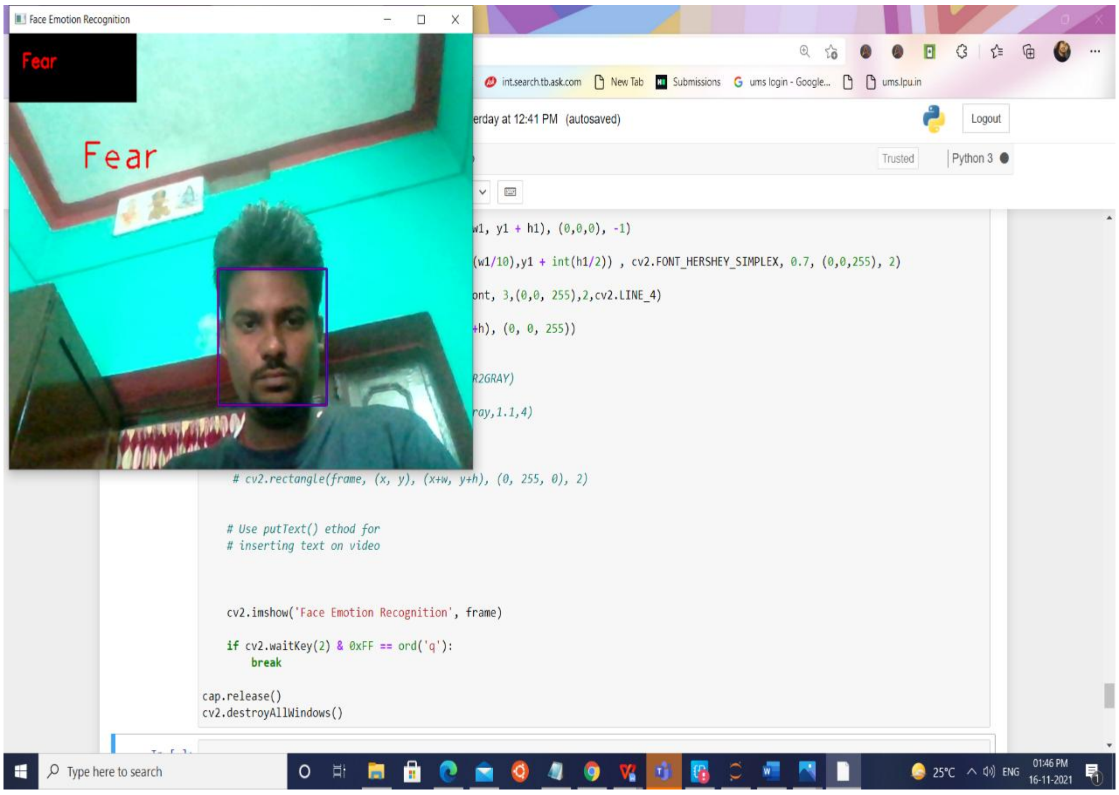

Home Page - Select or create a
softcomputingca1project - Jupyter
localhost:8888/notebooks/softcomputingca1project.ipynb
jupyter softcomputingca1project Last Checkpoint: Yesterday at 12:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search
27°C ENG
01:28 PM
16-11-2021

```
            #Draw black background rectangle
            cv2.rectangle(frame, (x1, x1),(x1 + w1, y1 + h1), (0,0,0), -1)
            # Add text
            cv2.putText(frame, status, (x1 + int(w1/10),y1 + int(h1/2)) , cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,0,255), 2)

            cv2.putText(frame,status,(100,150),font, 3,(0,0, 255),2,cv2.LINE_4)

            cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255))


        else:
            status = "Neutral"

            x1,y1,w1,h1 = 0,0,175,75
            #Draw black background rectangle
            cv2.rectangle(frame, (x1, x1),(x1 + w1, y1 + h1), (0,0,0), -1)
            # Add text
            cv2.putText(frame, status, (x1 + int(w1/10),y1 + int(h1/2)) , cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,0,255), 2)

            cv2.putText(frame,status,(100,150),font, 3,(0,0, 255),2,cv2.LINE_4)

            cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255))


        # gray= cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        # print(faceCascade.empty())
        # faces = faceCascade.detectMultiScale(gray,1.1,4)

        # Draw a rectangle around the facesQQQQQ
        # for(x, y, w, h) in faces
         # cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
```


Home Page - Select or create a
softcomputingca1project - Jupyter
localhost:8888/notebooks/softcomputingca1project.ipynb
jupyter softcomputingca1project Last Checkpoint: Yesterday at 12:41 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Type here to search
27°C ENG
01:28 PM
16-11-2021

```
            #Draw black background rectangle
            cv2.rectangle(frame, (x1, x1),(x1 + w1, y1 + h1), (0,0,0), -1)
            # Add text
            cv2.putText(frame, status, (x1 + int(w1/10),y1 + int(h1/2)) , cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,0,255), 2)

            cv2.putText(frame,status,(100,150),font, 3,(0,0, 255),2,cv2.LINE_4)

            cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255))


        # gray= cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        # print(faceCascade.empty())
        # faces = faceCascade.detectMultiScale(gray,1.1,4)

        # Draw a rectangle around the facesQQQQQ
        # for(x, y, w, h) in faces
         # cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)


        # Use putText() ethod for
        # inserting text on video



        cv2.imshow('Face Emotion Recognition', frame)

        if cv2.waitKey(2) & 0xFF == ord('q'):
            break

cap.release()
cv2.destroyAllWindows()
```

Realtime Video Demo



```
w1, y1 + h1), (0,0,0), -1)

(w1/10),y1 + int(h1/2)) , cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,0,255), 2)

ont, 3,(0,0, 255),2,cv2.LINE_4)

+h), (0, 0, 255))


R2GRAY)

ray,1.1,4)


     # cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)


    # Use putText() ethod for
    # inserting text on video



    cv2.imshow('Face Emotion Recognition', frame)

    if cv2.waitKey(2) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

Githublink:-

https://github.com/ravipandey9973/softcomputingprojectwork

Bibliography

- https://machinelearningmastery.com/how-to-perform-face-detection-with-classical-and-deep-learning-methods-in-python-with-keras/
- https://realpython.com/face-recognition-with-python/
- https://cloud.google.com/vision/docs/face-tutorial