FACIAL RECOGNIZATION SYSTEM AND PREDICTOR

Team Members:

NADIMPALLI LAKSHMI NARASHIMHA RAJU (19BEC2247)
N.ANAND VENKATA SUBBA RAJU (19BCE0264)
P.MANOHAR VENKAT CHOWDHARY (19BCE0187)
K.V.V.M SAI DIVYESH (19BCE0841)

Course Code: CSE3013 – AI

Slot:F2

Professor: Dr.W.B.Vasantha

1. Introduction:

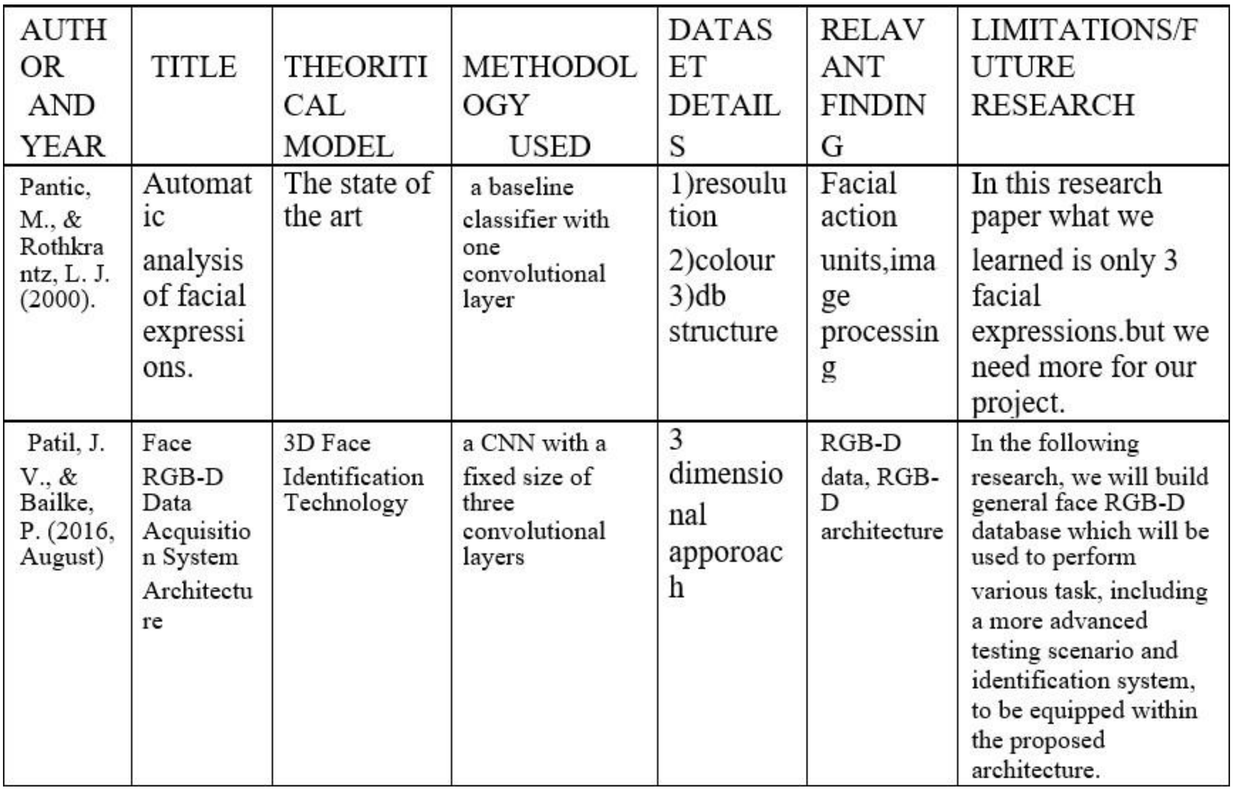
Artificial intelligence systems to recognize human emotion have attracted much research interest, and potential applications of such systems abound, spanning domains such as customer-attentive marketing, health monitoring, and emotionally intelligent robotic interfaces. In light of the important role that facial expression plays in communicating emotion in humans, there has been substantial research interest in computer vision systems to recognize human emotion. Certain facial expressions have universal meaning. Mostly in all researches, researchers identified six facial expressions that are universal across all cultures: anger, disgust, fear, happiness, sadness, and surprise. These are the same emotions that modern facial expression researchers aim to identify using computer vision. The recent success of convolutional neural networks (CNNs) in tasks such as object classification extends to the problem of facial expression recognition.

Facial expression recognition systems have attracted much research interest within the field of artificial intelligence. Many established facial expression recognition systems apply standard machine learning to extract image features, and these methods generalize poorly to previously unseen data. This project builds upon recent research to classify images of human faces into discrete emotion categories using python and anaconda atmosphere

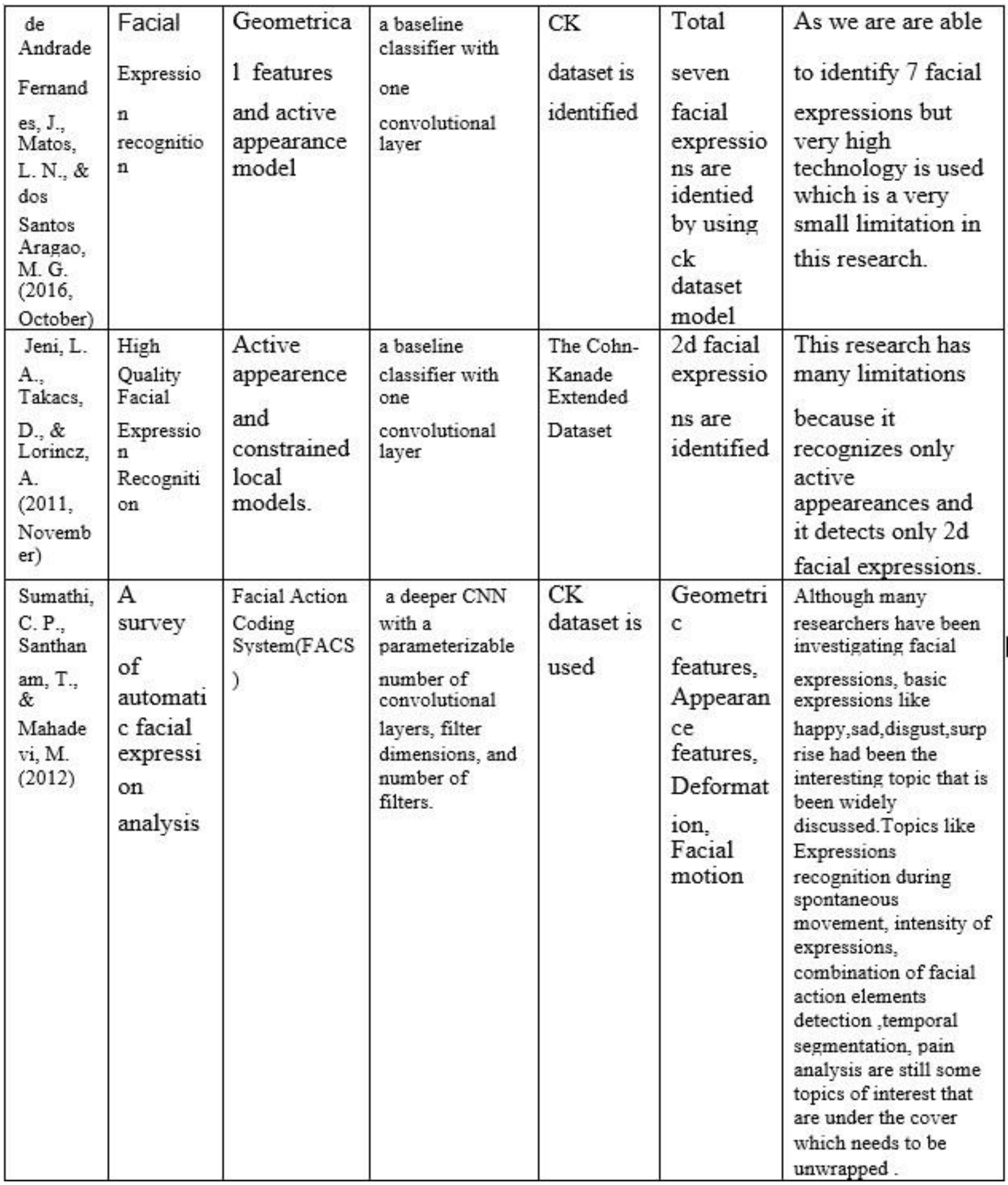
and we want to develop the project more than just detecting emotion we have done a so much detections like detecting age,nationality,gender,race and much more which is very exciting and interesting we felt that its quite awesome by just using a simple image or live vedio and detecting so many things about this random persons that's was amazing and we attracted to do it.

2. Literature Review Summary Table

LITERATURE SURVEY SUMMARY TABLE



AUTHOR AND YEAR	TITLE	THEORITICAL MODEL	METHODOLOGY USED	DATASET DETAILS	RELAVANT FINDING	LIMITATIONS/FUTURE RESEARCH
Pantic, M., & Rothkrantz, L. J. (2000).	Automatic analysis of facial expressions.	The state of the art	a baseline classifier with one convolutional layer	1)resoulution 2)colour 3)db structure	Facial action units,image processing	In this research paper what we learned is only 3 facial expressions.but we need more for our project.
Patil, J. V., & Bailke, P. (2016, August)	Face RGB-D Data Acquisition System Architecture	3D Face Identification Technology	a CNN with a fixed size of three convolutional layers	3 dimensional apporoach	RGB-D data, RGB-D architecture	In the following research, we will build general face RGB-D database which will be used to perform various task, including a more advanced testing scenario and identification system, to be equipped within the proposed architecture.



de Andrade Fernandes, J., Matos, L. N., & dos Santos Aragao, M. G. (2016, October)	Facial Expression recognition	Geometrical features and active appearance model	a baseline classifier with one convolutional layer	CK dataset is identified	Total seven facial expressions are identied by using ck dataset model	As we are are able to identify 7 facial expressions but very high technology is used which is a very small limitation in this research.
Jeni, L. A., Takacs, D., & Lorincz, A. (2011, November)	High Quality Facial Expression Recognition	Active appearence and constrained local models.	a baseline classifier with one convolutional layer	The Cohn-Kanade Extended Dataset	2d facial expressions are identified	This research has many limitations because it recognizes only active appearances and it detects only 2d facial expressions.
Sumathi, C. P., Santhanam, T., & Mahadevi, M. (2012)	A survey of automatic facial expression analysis	Facial Action Coding System(FACS)	a deeper CNN with a parameterizable number of convolutional layers, filter dimensions, and number of filters.	CK dataset is used	Geometric features, Appearance features, Deformation, Facial motion	Although many researchers have been investigating facial expressions, basic expressions like happy,sad,disgust,surprise had been the interesting topic that is been widely discussed.Topics like Expressions recognition during spontaneous movement, intensity of expressions, combination of facial action elements detection ,temporal segmentation, pain analysis are still some topics of interest that are under the cover which needs to be unwrapped .

Link- https://www.researchgate.net/publication/285164623_An_Introduction_to_Convolutional_Neural_Networks
LINK-https://www.researchgate.net/publication/303741861_A_Review_of_Deep_Machine_Learning
Link- https://in.docworkspace.com/d/sIGOv9a4w0MnMhQY

Objective of the project:

Facial Expression or Facial Emotion Detector can be used basically, to find out whether a person is sad, happy, angry and other facial expressions only through his/her face. This archive can be used to carry out such a task. It basically uses your Web Camera and then identifies your expression in Real Time.

Although Humans detect and depict faces and facial expressions in a scene with little or no effort. Still, development of an automated system that accomplishes this task is a bit difficult task. There are several related problems such as detection of an image segment as a face, extraction of the facial expression information, and classification of the expression (e.g., in emotion categories). A system that performs these operations precisely and in real time would form a big step in achieving a human-like interaction between man and machine.

So, in this project we are going to see how can we proceed on solving these problems and carry forward. The capability of the human visual system with respect to these problems is also analysed. It is meant to serve as an ultimate goal and a guide for determining recommendations for development of an automatic facial expression analyser. We mainly use convolutional neural networks and deep learning techniques of artificial intelligence in this project.

OUR UNIQUE FEATURES

we have done more things other than just detecting facial expression which we can discuss further.we have used anaconda command prompt and jupyter command prompt for maintaining our project environment and very easy to run our program using python.

we are inspired from a few projects but we have improved a lot a new techniques which are more efficient and new features are also introduced

in the reference project we have learnt how to write a python code for getting the emotion of a face scanned.

here we are working out to get more than just the emotion of face.we are also trying to get more than a emotion like also predicting the gender,age,nationality,percentage breakdown of different emotion signs,race at its best and we are also trying by using live webcam and knowing the emotion of a person,if the expression of a person changes then the emotion showing in the screen also changes not only just a offline picture.

4. Innovation component in the project:

Our innovative part is our unique feature

we are working out to get more than just the emotion of face.we are also trying to get more than a emotion like also predicting the gender,age,nationality,percentage breakdown of different emotion signs,race at its best and we are also trying by using live webcam and knowing the emotion of a person,if the expression of a person changes then the emotion showing in the screen also changes not only just a offline picture.and this is how our project differs from other projects.

5. Work done and implementation

a. Methodology adapted:

First we have setup the environment perfectly and installed all the software requirements.installedopencv,deepface and haarcascade_frontalface_default.xml which are used for primarly detecting face and other applications of that are mentioned in below tools used(heading no5).and we opened jupyter notebook from anaconda command prompt and started importing all the things we have to import like opencv,deepface,facecascade.and then we also imported matlab for plotting it and the code continues and make predictions of our image(emotions,age,etc) and we made a rectangular grid later to plot the face and write the dominant emotion above the rectangular grid and for live vedio we done almost the same if the face not get detected then we return the value to print face is not detected or if the face is detected then we will make a rectangular grid around persons face and write his emotions above or side of rectangular grid,eventually if the man keep moving his face or changing his emotions then the rectangular grid also displaces along man face and eventually emotions displayed keep on changing as man changes it.

Hardware and software requirements:

SOFTWARE REQUIREMENTS:
Opencv,deepface and haarcascade_frontalface_default.xml
Word
Anaconda command prompt
Jupyter notebook
Windows

HARDWARE REQUIREMENTS:
Internet connectivity
Laptop/pc for admin
Good server
webcam

b. Dataset used:

a. Where from you are taking your dataset?

OpenCV is the huge open-source library for the computer vision, machine learning, and imageprocessing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human

We need to use the command to install OpenCV in anaconda command prompt is
1.pip install opencv-python
2.pip install opencv-contrib-python

DeepFaceis a deep learning facial recognition system created by a research group at Facebook. Itidentifies human faces in digital images. It employs a nine-layer neural network with over 120 million connection weights and was trained on four million images uploaded by Facebook users.

We need to use the command to install deepface is
1.pip install deepface

we will try to detect the face of individuals using the

haarcascade_frontalface_default.xml

Haar-cascade Detection in OpenCV OpenCValready contains many pre-trained classifiers for face, eyes,smile etc. Those XML files are stored in **opencv**/data/haarcascades/ folder.

b. Is your project based on any other reference project (Stanford Univ. or MIT)?

yes we are inspired from a few projects but we have improved a lot a new techniques which are more efficient and new features are also introduced.The projects we are inspired of the following references we provided in references column below these are few links below that will help you

- Link- https://www.researchgate.net/publication/285164623_An_Introduction_to_Convolutional_Neural_Networks
- LINK- https://www.researchgate.net/publication/303741861_A_Review_of_Deep_Machine_Learning
- Link- https://in.docworkspace.com/d/sIGOv9a4w0MnMhQY
- WE REFERRED SOME REFERENCE BOOKS ALSO FOR OUR PROJECT.THE REFERENCE BOOK LINKS ARE GIVEN BELOW-

1) Python Machine Learning Book by Sebastian Raschka.
2) Deep Learning with Python Book by François Chollet

c. How does your project differ from the reference project?*

in the reference project we have learnt how to write a python code for getting the emotion of a face scanned.

here we are working out to get more than just the emotion of face.we are also trying to get more than a emotion like also predicting the gender,age,nationality,percentage breakdown of different emotion signs,race at its best and we are also trying by using live webcam and knowing the emotion of a person,if the expression of a person changes then the emotion showing in the screen also changes not only just a offline picture.

d. Tools used:its same as dataset we used

OpenCVis the huge open-source library for the computer vision, machine learning, and imageprocessing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human

We need to use the command to install OpenCV in anaconda command prompt is
1.pip install opencv-python
2.pip install opencv-contrib-python

DeepFaceis a deep learning facial recognition system created by a research group at Facebook. Itidentifies human faces in digital images. It employs a nine-layer neural network with over 120 million connection weights and was trained on four million images uploaded by Facebook users.

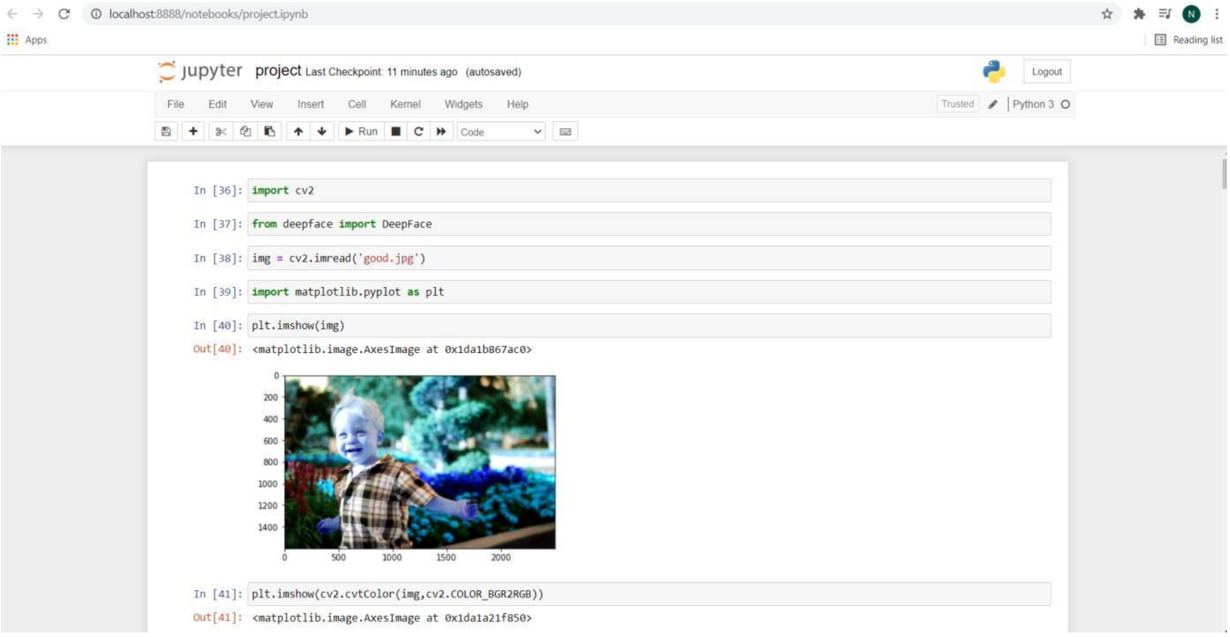
We need to use the command to install deepface is
1.pip install deepface

we will try to detect the face of individuals using the
haarcascade_frontalface_default.xml

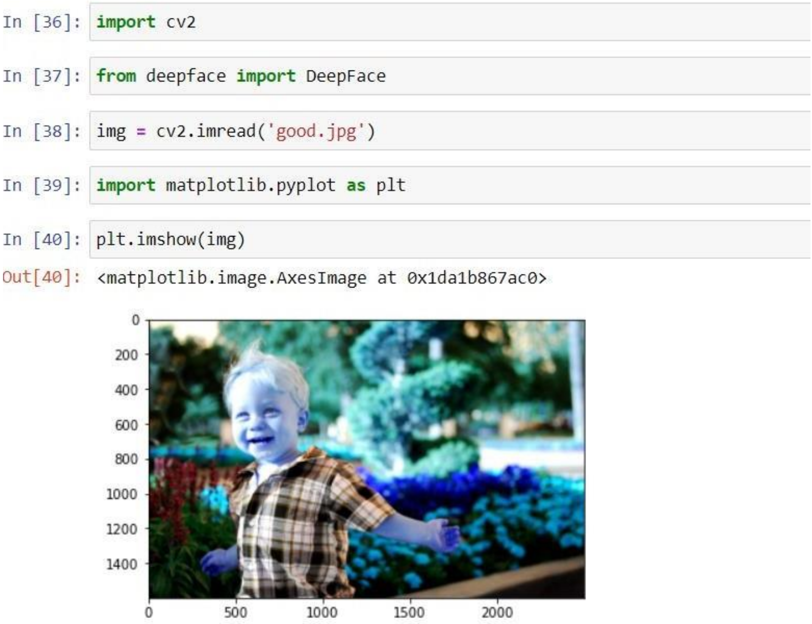
Haar-cascade Detection in OpenCV OpenCValready contains many pre-trained classifiers for face, eyes,smile etc. Those XML files are stored in **opencv**/data/haarcascades/ folder.

e. Screenshot and Demo: Each result and necessary coding part should be substantiated with related screenshot.

THIS IS HOW A JUPYTER NOTBOOK LOOKS LIKE



```
In [36]: import cv2

In [37]: from deepface import DeepFace

In [38]: img = cv2.imread('good.jpg')

In [39]: import matplotlib.pyplot as plt

In [40]: plt.imshow(img)
Out[40]: <matplotlib.image.AxesImage at 0x1da1b867ac0>

In [41]: plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))
Out[41]: <matplotlib.image.AxesImage at 0x1da1a21f850>
```



```
In [36]: import cv2

In [37]: from deepface import DeepFace

In [38]: img = cv2.imread('good.jpg')

In [39]: import matplotlib.pyplot as plt

In [40]: plt.imshow(img)
Out[40]: <matplotlib.image.AxesImage at 0x1da1b867ac0>
```

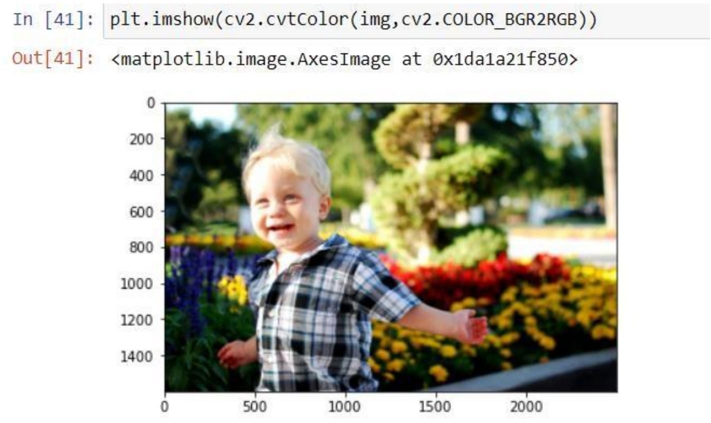
Firstly we need to install cv2 In anaconda cmdprmpt and import it

And we do this same for DeepFace also

Next download a random image from internetand read that image using cv2 as opencv identifies faces,objects and writings too.

And then import mathplot and plot the image and show the plotted image.it shows in BGR colour

As computer understands in BGR now we are converting into RGB and plotting the same image



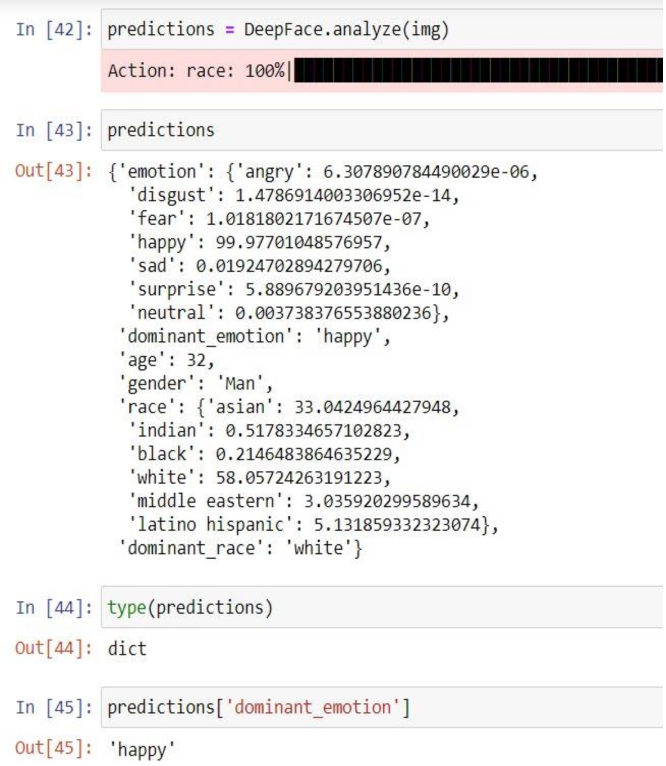
```
In [41]: plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))

Out[41]: <matplotlib.image.AxesImage at 0x1da1a21f850>
```

With the help of DeepFace we can analyze the image and can give our best predictions using it.

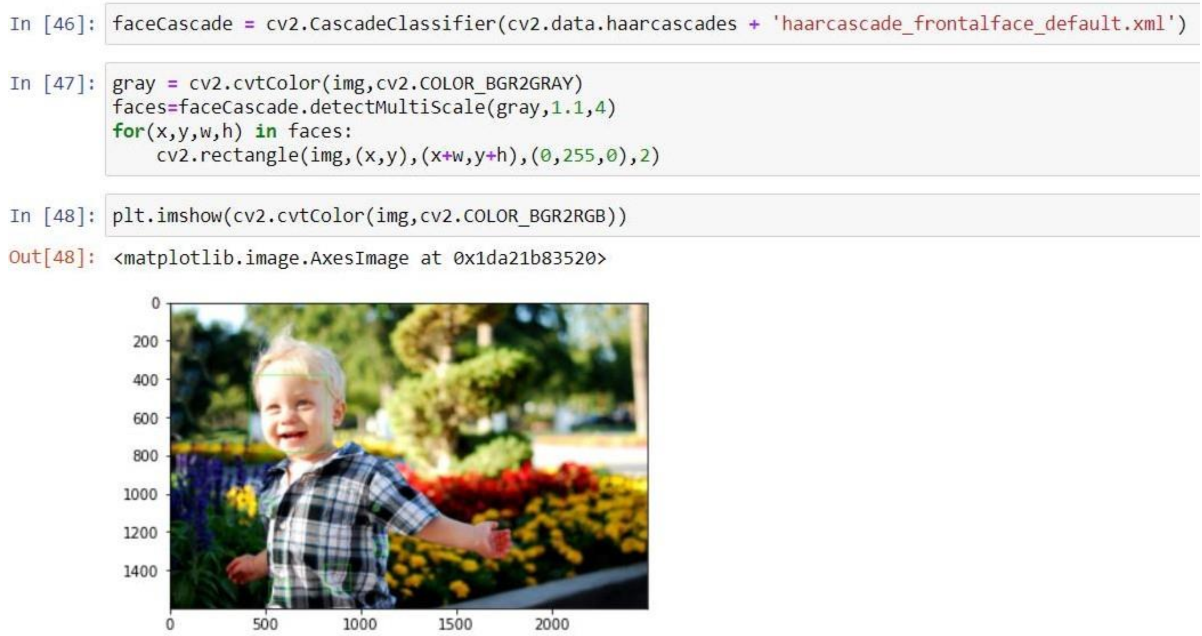
Here we are predicting various characteristics of a person in the image where deepface help us to do it very easy

In this pic we can see he was 99.977 happy so his dominant emotion is happy

Predictions will be stored as dict which is dictonary “{}”;



```
In [42]: predictions = DeepFace.analyze(img)

         Action: race: 100%|████████████████████████████████████

In [43]: predictions

Out[43]: {'emotion': {'angry': 6.307890784490029e-06,
           'disgust': 1.4786914003306952e-14,
           'fear': 1.0181802171674507e-07,
           'happy': 99.97701048576957,
           'sad': 0.01924702894279706,
           'surprise': 5.889679203951436e-10,
           'neutral': 0.003738376553880236},
          'dominant_emotion': 'happy',
          'age': 32,
          'gender': 'Man',
          'race': {'asian': 33.0424964427948,
           'indian': 0.5178334657102823,
           'black': 0.2146483864635229,
           'white': 58.05724263191223,
           'middle eastern': 3.035920299589634,
           'latino hispanic': 5.131859332323074},
          'dominant_race': 'white'}

In [44]: type(predictions)

Out[44]: dict

In [45]: predictions['dominant_emotion']

Out[45]: 'happy'
```

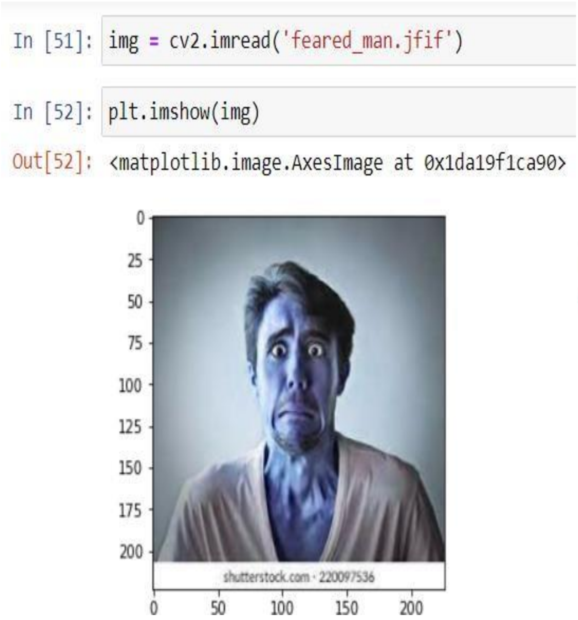


```
In [46]: faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')

In [47]: gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
         faces=faceCascade.detectMultiScale(gray,1.1,4)
         for(x,y,w,h) in faces:
             cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)

In [48]: plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))

Out[48]: <matplotlib.image.AxesImage at 0x1da21b83520>
```

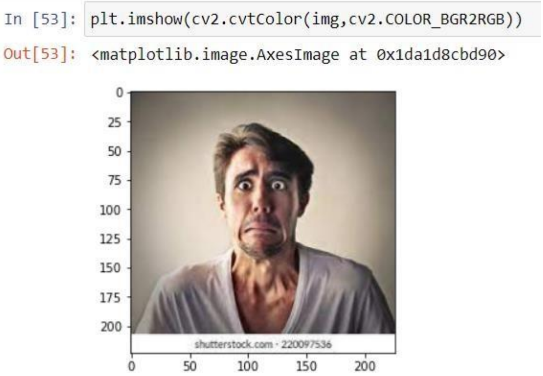
Observe carefully here there is greenbox plotted around the face of a child,haar is the famous algorithm to detect face but not the emotion it’s a part of faceCascade(which we get by importing cv2).

We are just trying to detect the face and draw a green rectangle and we set the values as (0,255,0) Which is red is 0,green is 255 the highest,blue is 0 ,the plot and show the image.

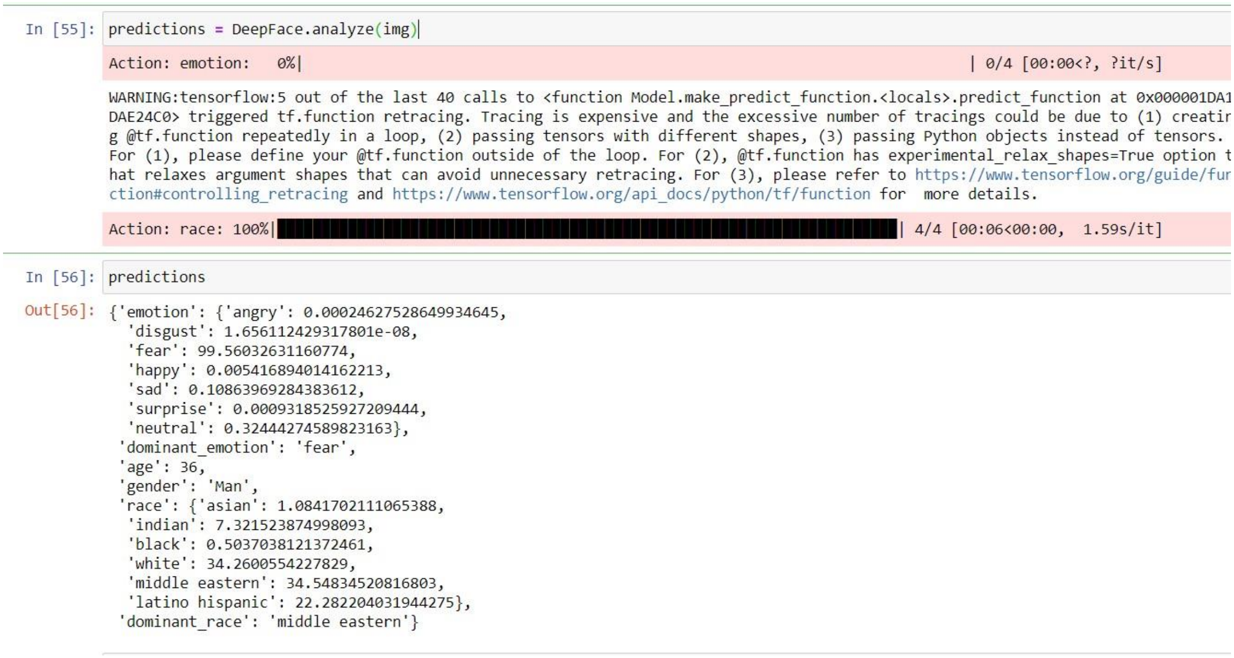
we also used another emotion called fear



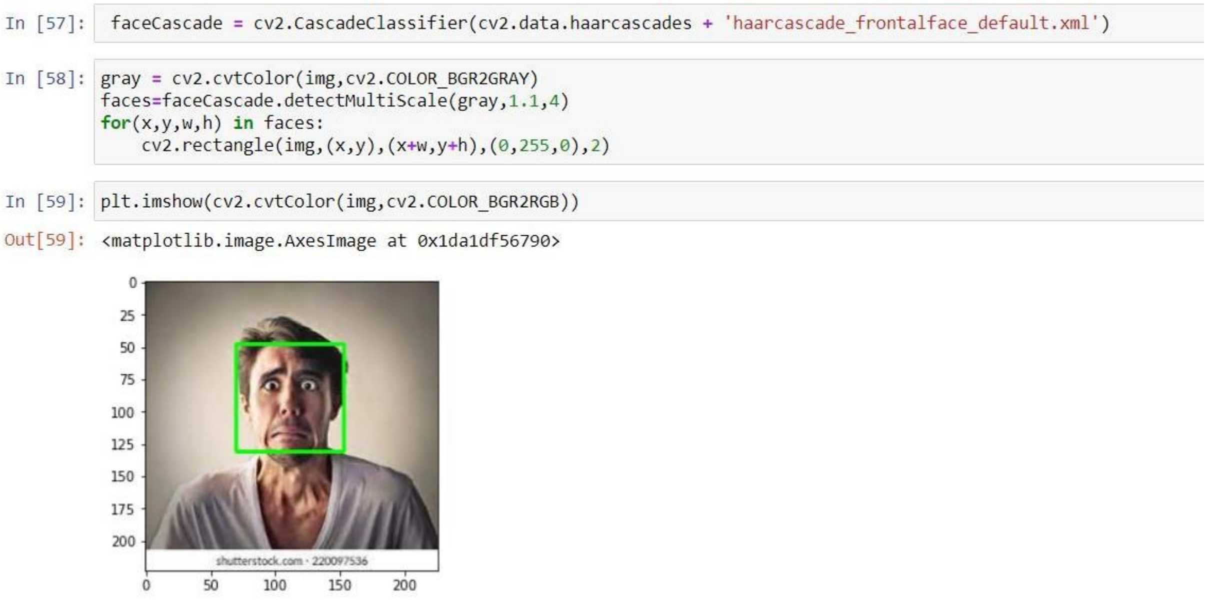
```
In [51]: img = cv2.imread('feared_man.jfif')

In [52]: plt.imshow(img)

Out[52]: <matplotlib.image.AxesImage at 0x1da19f1ca90>
```



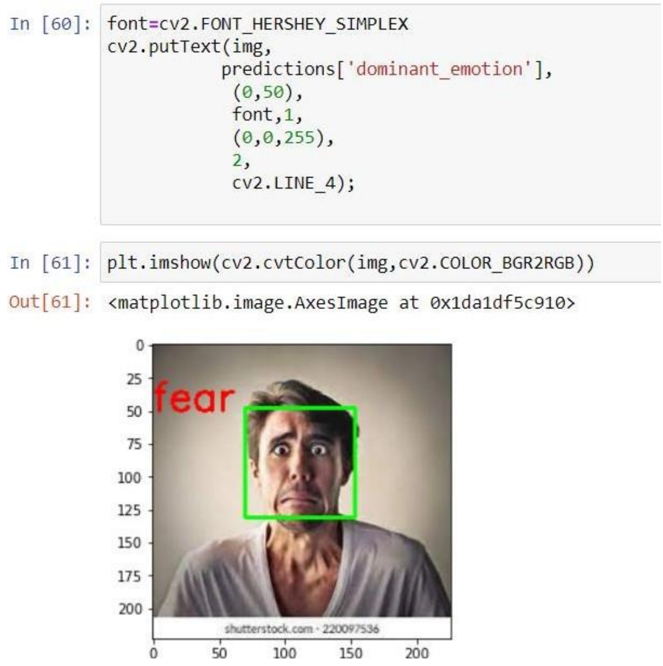
```
In [53]: plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))

Out[53]: <matplotlib.image.AxesImage at 0x1da1d8cbd90>
```



```
In [55]: predictions = DeepFace.analyze(img)

         Action: emotion:   0%|                                                          | 0/4 [00:00<?, ?it/s]

         WARNING:tensorflow:5 out of the last 40 calls to <function Model.make_predict_function.<locals>.predict_function at 0x000001DA1
         DAE24C0> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creatir
         g @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors.
         For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option t
         hat relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/fur
         ction#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for  more details.

         Action: race: 100%|██████████████████████████████████████████████████| 4/4 [00:06<00:00,  1.59s/it]

In [56]: predictions

Out[56]: {'emotion': {'angry': 0.00024627528649934645,
           'disgust': 1.656112429317801e-08,
           'fear': 99.56032631160774,
           'happy': 0.005416894014162213,
           'sad': 0.10863969284383612,
           'surprise': 0.0009318525927209444,
           'neutral': 0.32444274589823163},
          'dominant_emotion': 'fear',
          'age': 36,
          'gender': 'Man',
          'race': {'asian': 1.0841702111065388,
           'indian': 7.321523874998093,
           'black': 0.5037038121372461,
           'white': 34.2600554227829,
           'middle eastern': 34.54834520816803,
           'latino hispanic': 22.282204031944275},
          'dominant_race': 'middle eastern'}
```



```
In [57]:  faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')

In [58]: gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
         faces=faceCascade.detectMultiScale(gray,1.1,4)
         for(x,y,w,h) in faces:
             cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)

In [59]: plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))

Out[59]: <matplotlib.image.AxesImage at 0x1da1df56790>
```



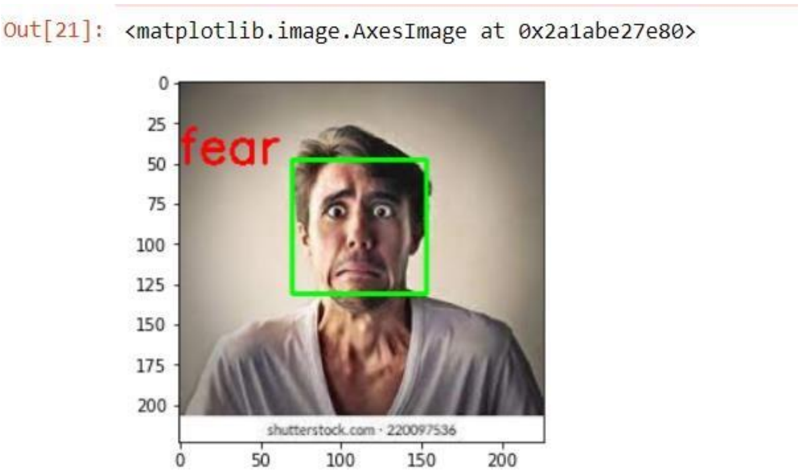
```
In [60]: font=cv2.FONT_HERSHEY_SIMPLEX
         cv2.putText(img,
                     predictions['dominant_emotion'],
                      (0,50),
                      font,1,
                      (0,0,255),
                      2,
                      cv2.LINE_4);

In [61]: plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))

Out[61]: <matplotlib.image.AxesImage at 0x1da1df5c910>
```

HERE YOU CAN SEE I HAVE HIGHIGHTED THE BOX WITH GREEN AND EMOTION WITH RED.

CODE FOR DIRECT IMPLEMATION OF DOMINANT EMOTION AND PLOTTING FACE.

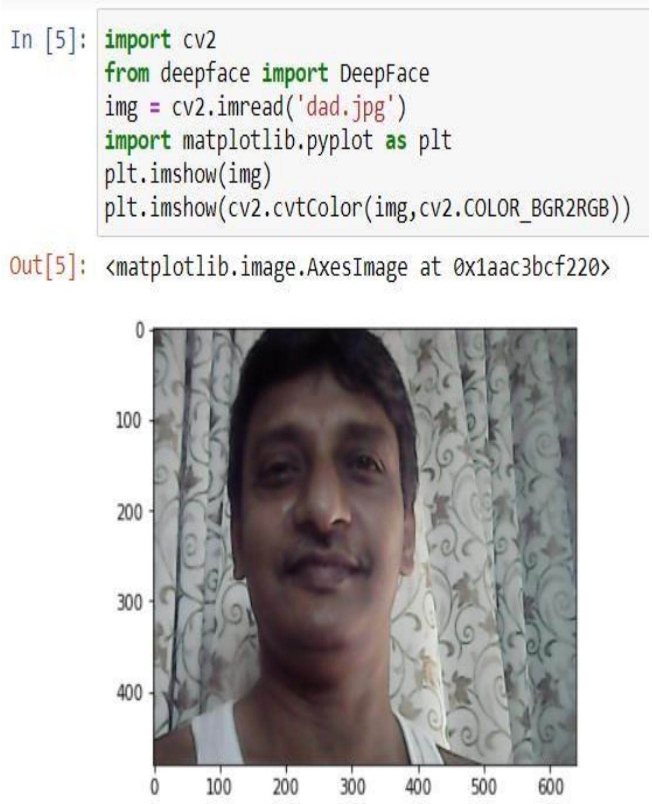
```
import cv2
from deepface import DeepFace
img = cv2.imread('feared_man.jfif')
import matplotlib.pyplot as plt
plt.imshow(img)
plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))
predictions = DeepFace.analyze(img)
predictions
faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
faces=faceCascade.detectMultiScale(gray,1.1,4)
for(x,y,w,h) in faces:
   cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)

plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))
font=cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(img,
          predictions['dominant_emotion'],
           (0,50),
           font,1,
           (0,0,255),
           2,
           cv2.LINE_4);
plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))
```

OUTPUT=

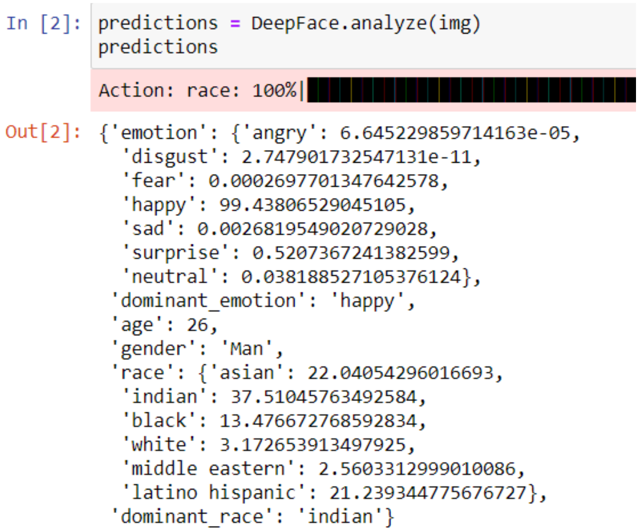


```
Out[21]: <matplotlib.image.AxesImage at 0x2a1abe27e80>
```

EXAMPLE OF MY dad PIC IN MY COMPUTER.

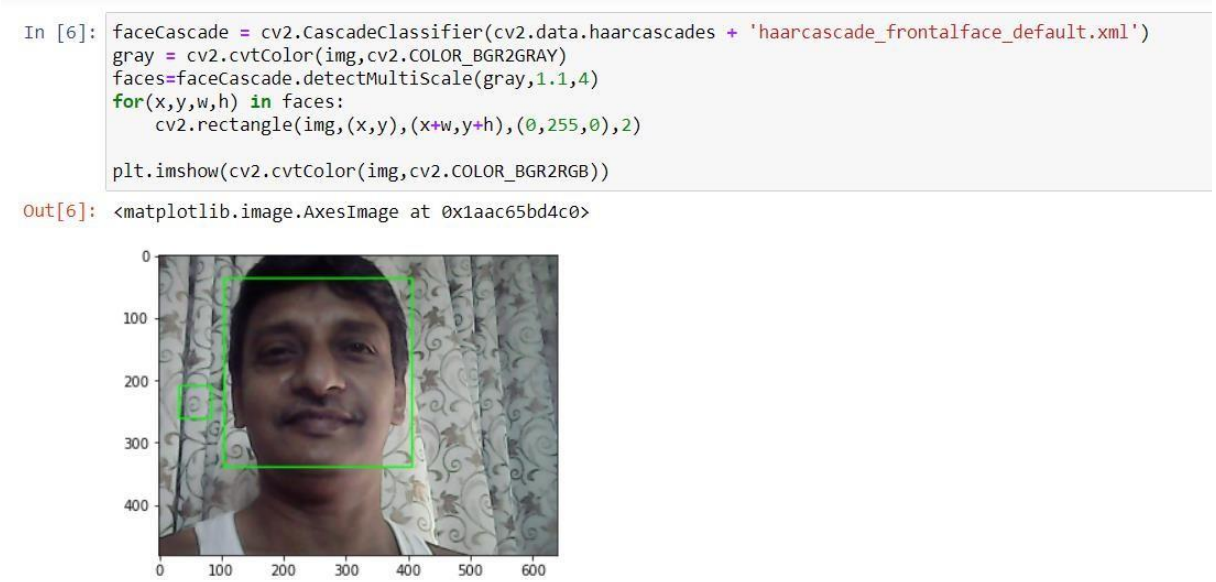


```
In [5]: import cv2
        from deepface import DeepFace
        img = cv2.imread('dad.jpg')
        import matplotlib.pyplot as plt
        plt.imshow(img)
        plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))

Out[5]: <matplotlib.image.AxesImage at 0x1aac3bcf220>
```

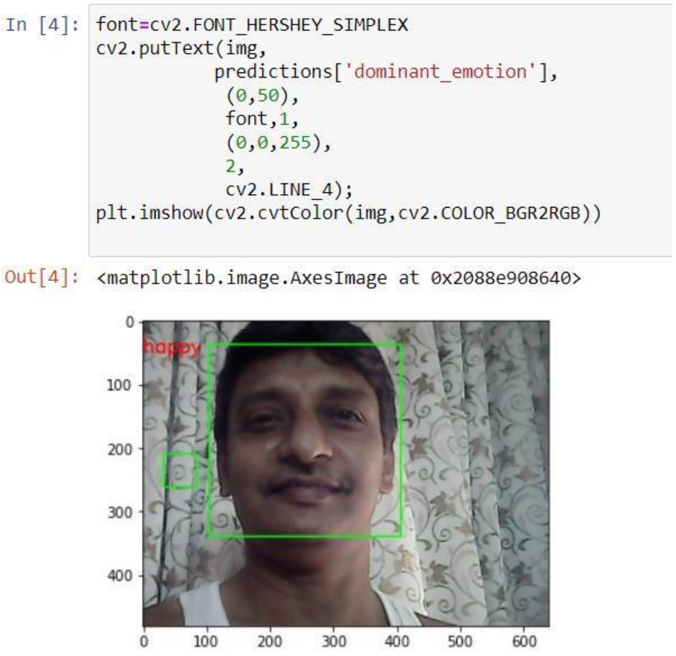


```
In [2]: predictions = DeepFace.analyze(img)
        predictions

        Action: race: 100%|

Out[2]: {'emotion': {'angry': 6.645229859714163e-05,
          'disgust': 2.747901732547131e-11,
          'fear': 0.0002697701347642578,
          'happy': 99.43806529045105,
          'sad': 0.0026819549020729028,
          'surprise': 0.5207367241382599,
          'neutral': 0.038188527105376124},
         'dominant_emotion': 'happy',
         'age': 26,
         'gender': 'Man',
         'race': {'asian': 22.04054296016693,
          'indian': 37.51045763492584,
          'black': 13.476672768592834,
          'white': 3.172653913497925,
          'middle eastern': 2.5603312999010086,
          'latino hispanic': 21.239344775676727},
         'dominant_race': 'indian'}
```

In predictions you can see its predicting as he was an Indian and his emotion divisions and age etc



```
In [6]: faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
        gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
        faces=faceCascade.detectMultiScale(gray,1.1,4)
        for(x,y,w,h) in faces:
            cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)

        plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))

Out[6]: <matplotlib.image.AxesImage at 0x1aac65bd4c0>
```

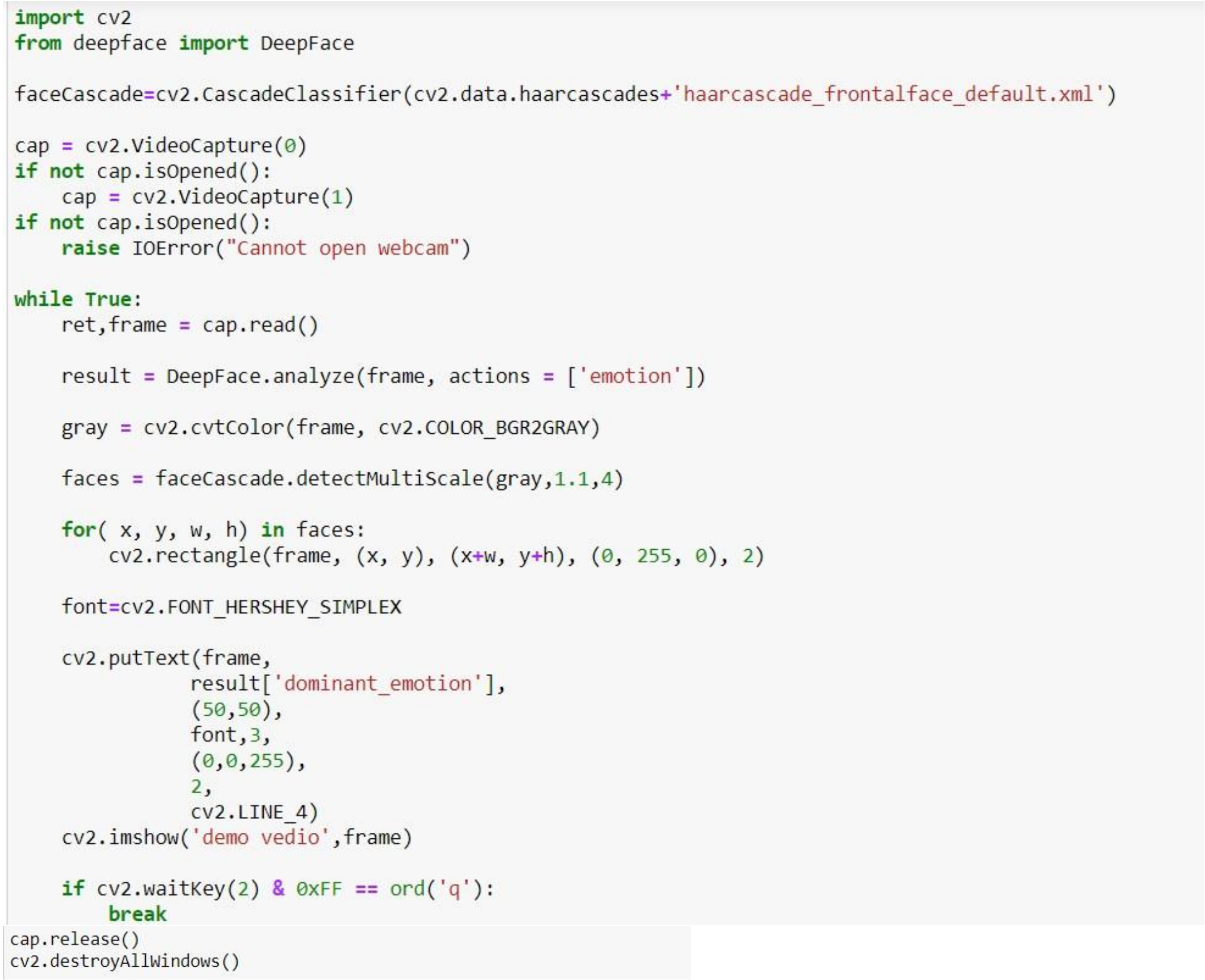


```
In [4]: font=cv2.FONT_HERSHEY_SIMPLEX
        cv2.putText(img,
                    predictions['dominant_emotion'],
                     (0,50),
                     font,1,
                     (0,0,255),
                     2,
                     cv2.LINE_4);
        plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))

Out[4]: <matplotlib.image.AxesImage at 0x2088e908640>
```

You can see its detecting the peron is happy

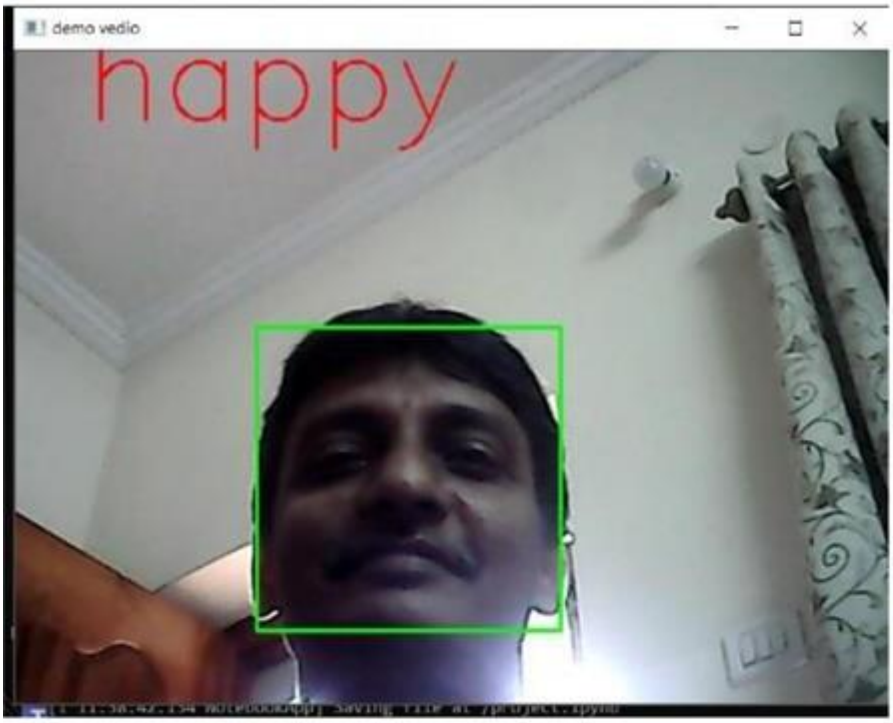
CODE FOR LIVE DEMO VEDIO..

```
import cv2
from deepface import DeepFace

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

cap = cv2.VideoCapture(0)
if not cap.isOpened():
   cap = cv2.VideoCapture(1)
if not cap.isOpened():
   raise IOError("Cannot open webcam")

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

   result = DeepFace.analyze(frame, actions = ['emotion'])

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

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

for( x, y, w, h) in faces:
      cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

   font=cv2.FONT_HERSHEY_SIMPLEX

   cv2.putText(frame,
            result['dominant_emotion'],
            (50,50),
            font,3,
            (0,0,255),
            2,
            cv2.LINE_4)
    cv2.imshow('demo vedio',frame)

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

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

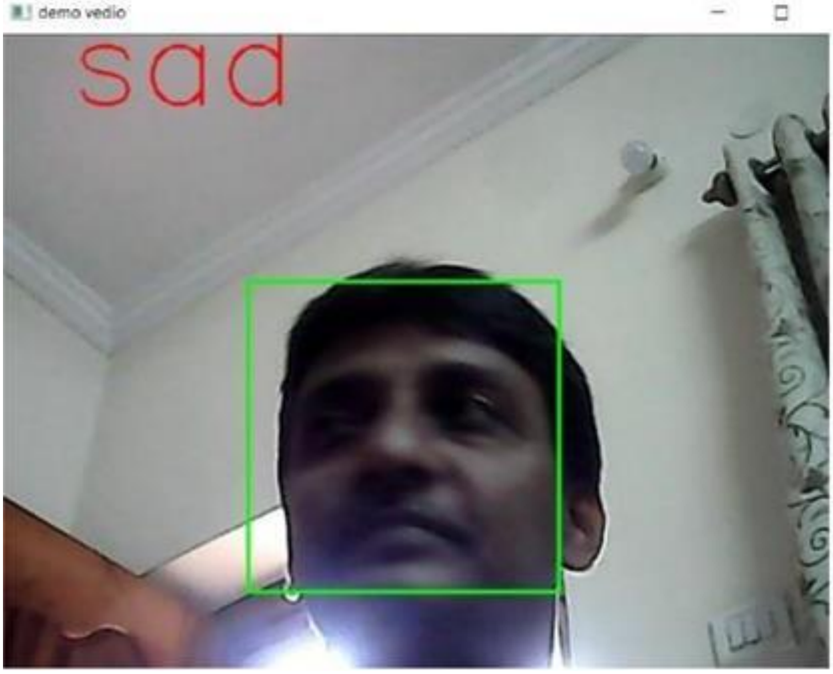


```
import cv2
from deepface import DeepFace

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

cap = cv2.VideoCapture(0)
if not cap.isOpened():
    cap = cv2.VideoCapture(1)
if not cap.isOpened():
    raise IOError("Cannot open webcam")

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

    result = DeepFace.analyze(frame, actions = ['emotion'])

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

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

    for( x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

    font=cv2.FONT_HERSHEY_SIMPLEX

    cv2.putText(frame,
                result['dominant_emotion'],
                (50,50),
                font,3,
                (0,0,255),
                2,
                cv2.LINE_4)
    cv2.imshow('demo vedio',frame)

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

OUTPUTS FOR LIVE VEDIO EMOTION DETECTOR=



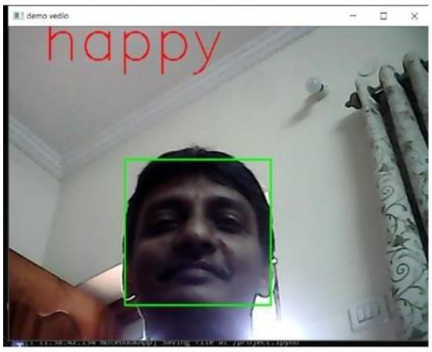
```
[I 11:38:42.134 NotebookApp] Saving file at /project.ipynb
```

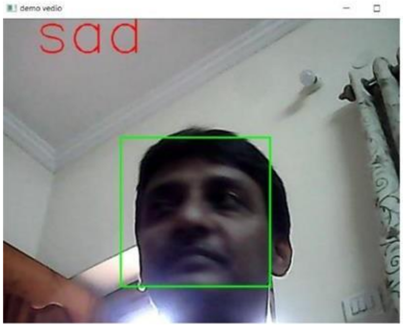


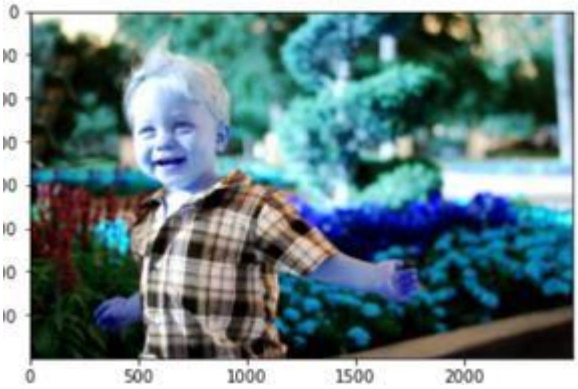
6. Results and discussion

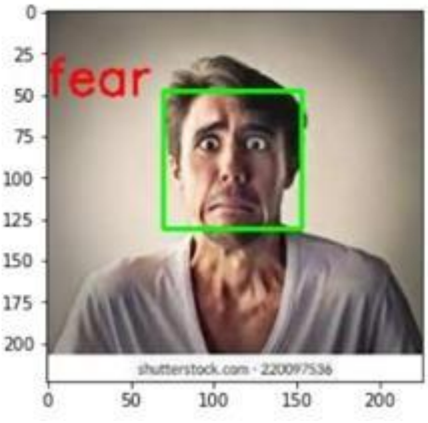
Results I have already included above and discussed now we will again make some discussions about it.

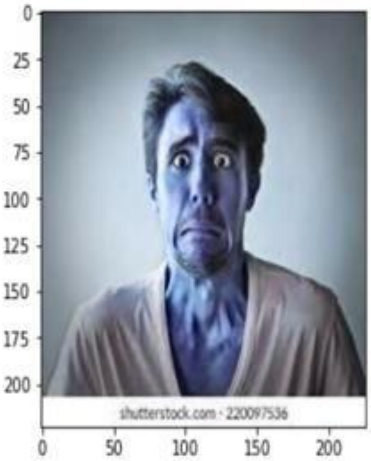
Results we obtained.

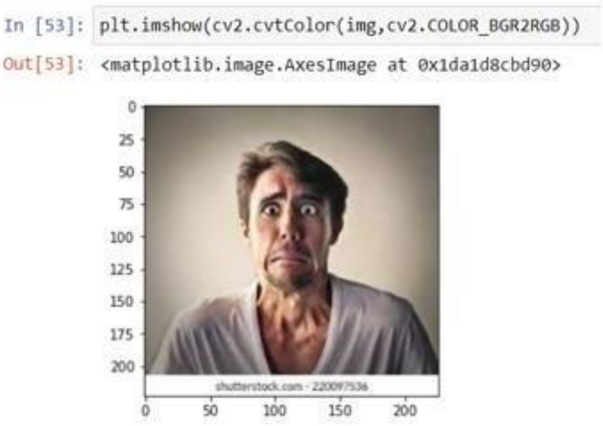






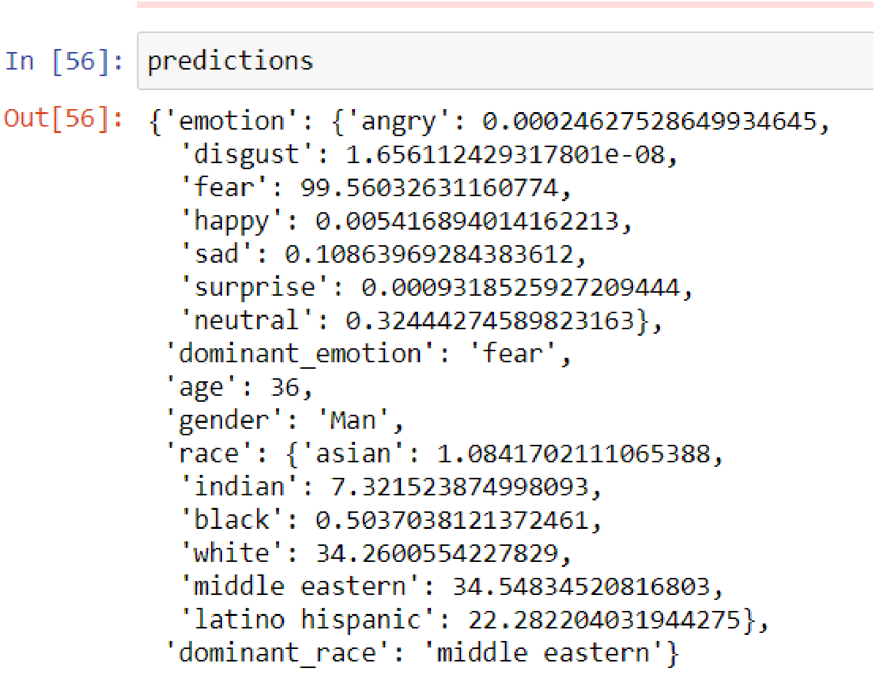






```
In [53]: plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))
Out[53]: <matplotlib.image.AxesImage at 0x1da1d8cbd90>
```

We can obtain results like this for various examples and here we can see the we are able to detect the dominant emotion and face by plotting it and also displaying the dominant emotion along with plotting the face in the picture and we also predicted the various characteristics of a person just by have his image.



```
In [56]: predictions

Out[56]: {'emotion': {'angry': 0.00024627528649934645,
           'disgust': 1.656112429317801e-08,
           'fear': 99.56032631160774,
           'happy': 0.005416894014162213,
           'sad': 0.10863969284383612,
           'surprise': 0.0009318525927209444,
           'neutral': 0.3244427458982316З},
          'dominant_emotion': 'fear',
          'age': 36,
          'gender': 'Man',
          'race': {'asian': 1.0841702111065388,
           'indian': 7.321523874998093,
           'black': 0.5037038121372461,
           'white': 34.2600554227829,
           'middle eastern': 34.54834520816803,
           'latino hispanic': 22.282204031944275},
          'dominant_race': 'middle eastern'}
```

You can see above about the predictions of the pic.

7. References

REFRENCES

[1] Pantic, M., &Rothkrantz, L. J. (2000). Automatic analysis of facial expressions: The state of the art. IEEE Transactions on Pattern Analysis & Machine Intelligence, (12), 1424-1445

. [2] Patil, J. V., &Bailke, P. (2016, August). Real time facial expression recognition using RealSense camera and ANN. In 2016 International Conference on Inventive Computation Technologies (ICICT) (Vol. 2, pp. 1-6). IEEE.

[3] de Andrade Fernandes, J., Matos, L. N., & dos Santos Aragao, M. G. (2016, October). Geometrcal approaches for facial expression recognition using support vector machines. In 2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI) (pp. 347-354). IEEE.

[4] Jeni, L. A., Takacs, D., &Lorincz, A. (2011, November). High quality facial expression recognition in video streams using shape related information only. In 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops) (pp. 2168- 2174). IEEE.

[5] Sumathi, C. P., Santhanam, T., & Mahadevi, M. (2012). Automatic facial expression analysis a survey. International Journal of Computer Science and Engineering Survey, 3(6),
[6] Link- https://www.researchgate.net/publication/285164623_An_Introduction_to_Convolutional_Neural_Networks
[5] LINK- https://www.researchgate.net/publication/303741861_A_Review_of_Deep_Machine_Learning

[6] Link- https://in.docworkspace.com/d/sIGOv9a4w0MnMhQY