

```
In [1]:  ▶ import cv2
```

```
In [6]:  ▶ from deepface import DeepFace
```

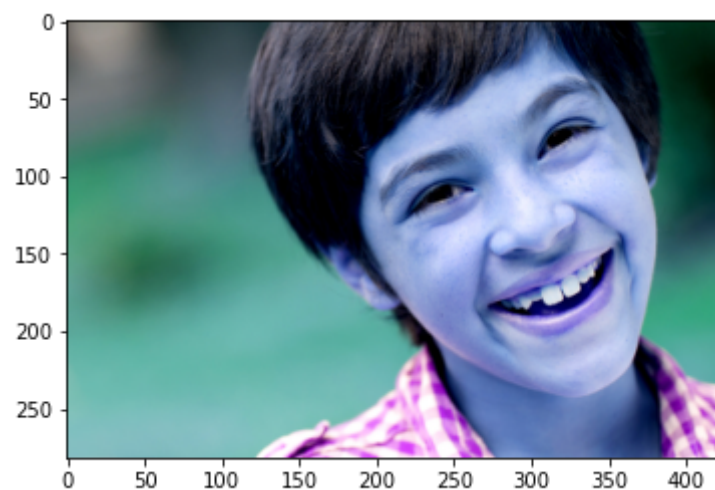
```
Directory C:\Users\rvsan /.deepface created  
Directory C:\Users\rvsan /.deepface/weights created
```

```
In [2]:  ▶ img=cv2.imread('happyboy.jpg')
```

```
In [3]:  ▶ import matplotlib.pyplot as plt
```

```
In [4]:  ▶ plt.imshow(img)
```

```
Out[4]: <matplotlib.image.AxesImage at 0x2c2ccca5a30>
```



```
In [5]: ▶ plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
```

```
Out[5]: <matplotlib.image.AxesImage at 0x2c2cd44d820>
```




```
In [9]: ▶ type(predictions)
```

```
Out[9]: dict
```

```
In [11]: ▶ predictions['dominant_emotion']
```

```
Out[11]: 'happy'
```

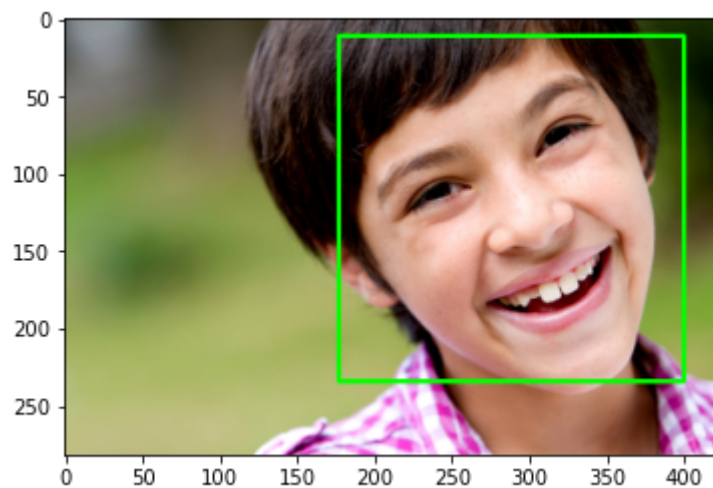
```
In [12]: ▶ faceCascade= cv2.CascadeClassifier(cv2.data.harcascades+'haarcascade_frontalface_default.xml')
```

```
In [13]: ▶ 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 [14]: ▶ plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
```

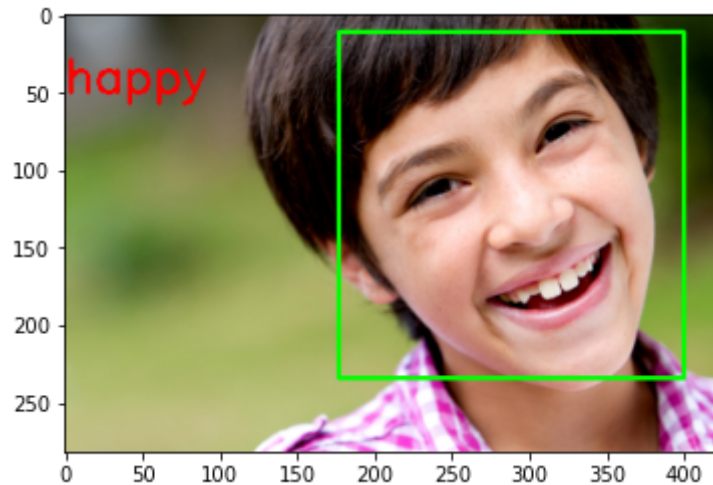
```
Out[14]: <matplotlib.image.AxesImage at 0x2c2905909d0>
```



```
In [15]: ▶ font=cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(img,
            predictions['dominant_emotion'],
            (0,50),
            font,1,
            (0,0,255),
            2,
            cv2.LINE_4);
```

```
In [16]: ▶ plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
```

```
Out[16]: <matplotlib.image.AxesImage at 0x2c290605610>
```



```
In [57]: ▶ img=cv2.imread('woman.jpg')
```

```
In [58]: ▶ plt.imshow(img)
```

```
Out[58]: <matplotlib.image.AxesImage at 0x2c29bd745b0>
```



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

```
Out[59]: <matplotlib.image.AxesImage at 0x2c29bdbb670>
```




```
Action: emotion: 0%| | 0/4 [00:00<?, ?i
t/s]

1/1 [=====] - 0s 41ms/step

Action: age: 25%| | 1/4 [00:00<00:01, 2.75
it/s]

1/1 [=====] - 2s 2s/step

Action: gender: 50%| | 2/4 [00:02<00:03, 1.57
s/it]

1/1 [=====] - 1s 1s/step

Action: race: 75%| | 3/4 [00:03<00:01, 1.37
s/it]

1/1 [=====] - 0s 288ms/step

Action: race: 100%| | 4/4 [00:04<00:00, 1.06
s/it]
```

In [61]: ► predictions

```
Out[61]: {'emotion': {'angry': 0.0005878078705087106,
  'disgust': 9.66969992870277e-06,
  'fear': 28.556080374484086,
  'happy': 6.365207620683802e-05,
  'sad': 71.44274541394385,
  'surprise': 8.760931646466004e-14,
  'neutral': 0.0005187974547879185},
  'dominant_emotion': 'sad',
  'region': {'x': 153, 'y': 86, 'w': 354, 'h': 354},
  'age': 33,
  'gender': 'Woman',
  'race': {'asian': 2.575255371630192,
  'indian': 0.9512495249509811,
  'black': 0.23985877633094788,
  'white': 45.996177196502686,
  'middle eastern': 7.729353755712509,
  'latino hispanic': 42.50810742378235},
  'dominant_race': 'white'}
```

In []: ►

In []: ►

In []: ►

In []: ►

In []: ►

In []: ►

In []: ▶

In []: ▶

In []: ▶

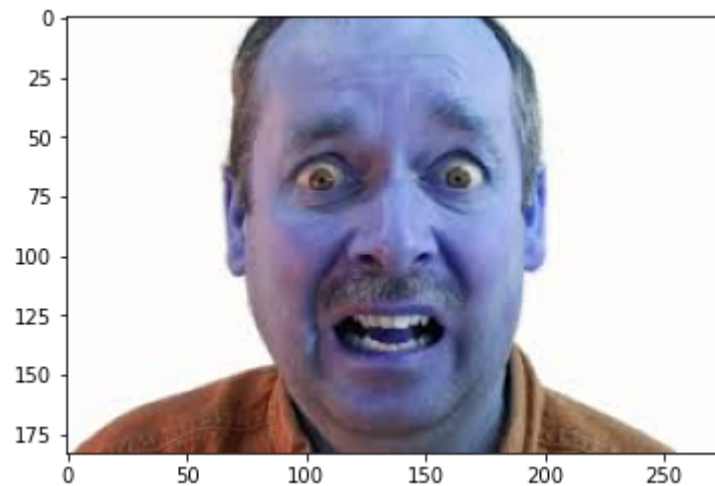
In []: ▶

In []: ▶

In [41]: ▶ `img=cv2.imread('fearedman.jpg')`

In [42]: ▶ `plt.imshow(img)`

Out[42]: <matplotlib.image.AxesImage at 0x2c28ade9040>



```
In [43]: ▶ plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
```

```
Out[43]: <matplotlib.image.AxesImage at 0x2c28ae0bb80>
```



```
Action: emotion: 0%|
t/s]

1/1 [=====] - 0s 61ms/step

Action: age: 25%|
it/s]

1/1 [=====] - 1s 869ms/step

Action: gender: 50%|
t/s]

1/1 [=====] - 2s 2s/step

Action: race: 75%|
s/it]

1/1 [=====] - 3s 3s/step

Action: race: 100%|
s/it]
```

In [45]: ► predictions

```
Out[45]: {'emotion': {'angry': 0.8368918672204018,
  'disgust': 2.2489819784254905e-06,
  'fear': 99.03684854507446,
  'happy': 0.025895604630932212,
  'sad': 8.239829440981339e-06,
  'surprise': 0.1003512879833579,
  'neutral': 4.857352787677148e-13},
  'dominant_emotion': 'fear',
  'region': {'x': 64, 'y': 10, 'w': 146, 'h': 146},
  'age': 29,
  'gender': 'Man',
  'race': {'asian': 4.951405897736549,
  'indian': 9.23803523182869,
  'black': 8.942444622516632,
  'white': 26.732921600341797,
  'middle eastern': 18.011222779750824,
  'latino hispanic': 32.1239709854126},
  'dominant_race': 'latino hispanic'}
```

real time demo

In [46]: ►

In [47]: ►

In [48]: ►

```
In [1]: ▶ import cv2
from deepface import DeepFace
faceCascade= cv2.CascadeClassifier(cv2.data.harcascades+'haarcascade_frontalface_default.xml')
cap=cv2.VideoCapture(1)
if not cap.isOpened():
    cap=cv2.VideoCapture(0)
if not cap.isOpened():
    raise IOError("Cannot open camera")

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 video',frame)

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

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

1/1 [=====] - 1s 545ms/step

1/1 [=====] - 0s 49ms/step

```
1/1 [=====] - 0s 47ms/step
1/1 [=====] - 0s 50ms/step
1/1 [=====] - 0s 63ms/step
1/1 [=====] - 0s 41ms/step
1/1 [=====] - 0s 55ms/step
1/1 [=====] - 0s 37ms/step
1/1 [=====] - 0s 47ms/step
1/1 [=====] - 0s 46ms/step
1/1 [=====] - 0s 45ms/step
1/1 [=====] - 0s 49ms/step
1/1 [=====] - 0s 50ms/step
1/1 [=====] - 0s 45ms/step
1/1 [=====] - 0s 50ms/step
1/1 [=====] - 0s 34ms/step
1/1 [=====] - 0s 32ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 24ms/step
```

In []: ▶