# Facial Expression Recognition

By - Prakarsh , Aditi , Puspkant

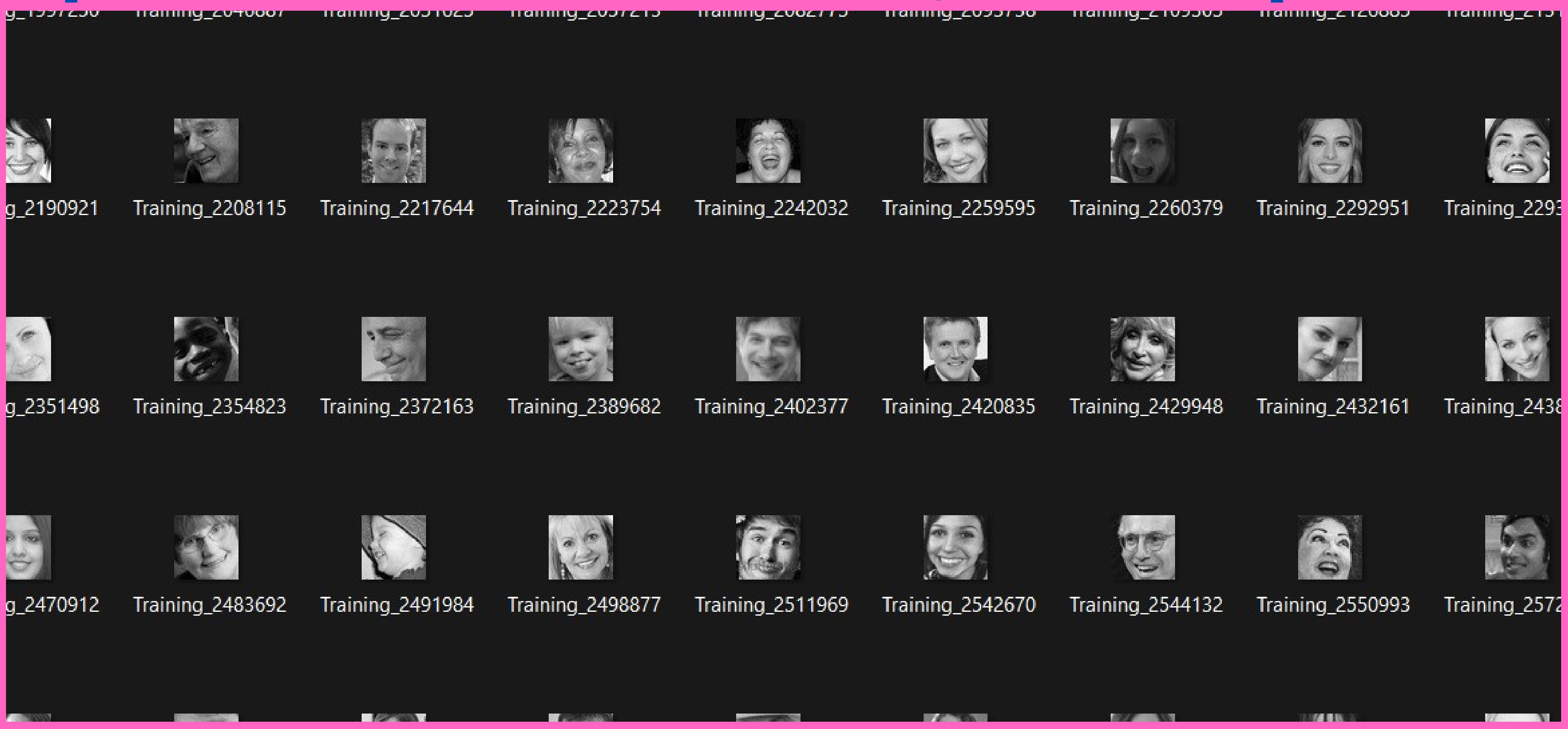
*Alliance University*

## Introduction

- Facial Emotion Recognition (FER) is the technology that analyses facial expressions from both static images and videos in order to reveal information on one's emotional state.
- The complexity of facial expressions, the potential use of the technology in any context,and the involvement of new technologies such as artificial intelligence raise significant privacy risks.

## About Dataset

- The data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centred and occupies about the same amount of space in each image.
- The task is to categorize each face based on the emotion shown in the facial expression into one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Neutral, 5=Sad, 6=Surprise). The training set consists of 28,709 examples and the public test set consists of 3,589 examples



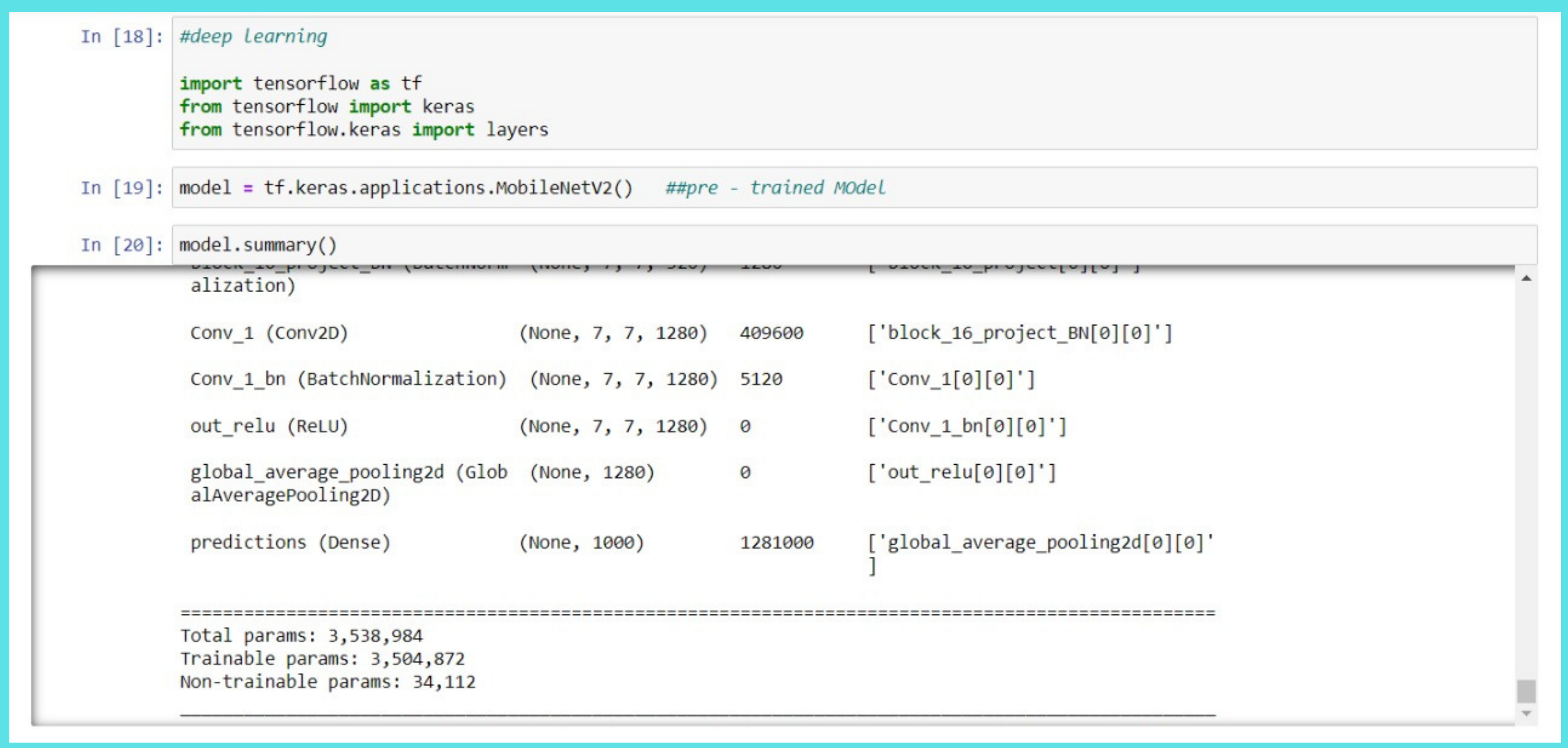
## Libraries Used:

- Numpy
- Pandas
- Sckit Learn
- Seaborn
- Tensorflow
- Keras
- OpenCV - V2

## Model Used:

**MobileNet-v2 -**

- is a convolutional neural network that is 53 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database [1].
- As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224.

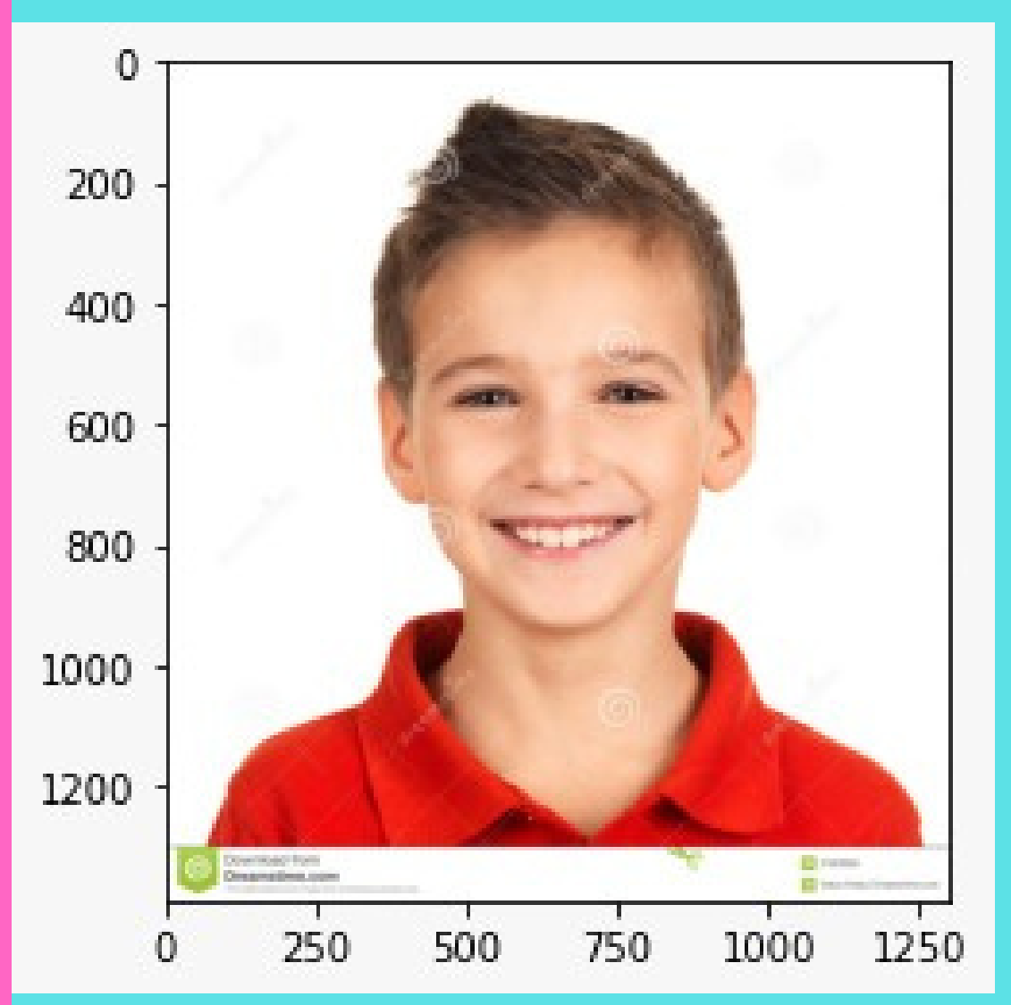


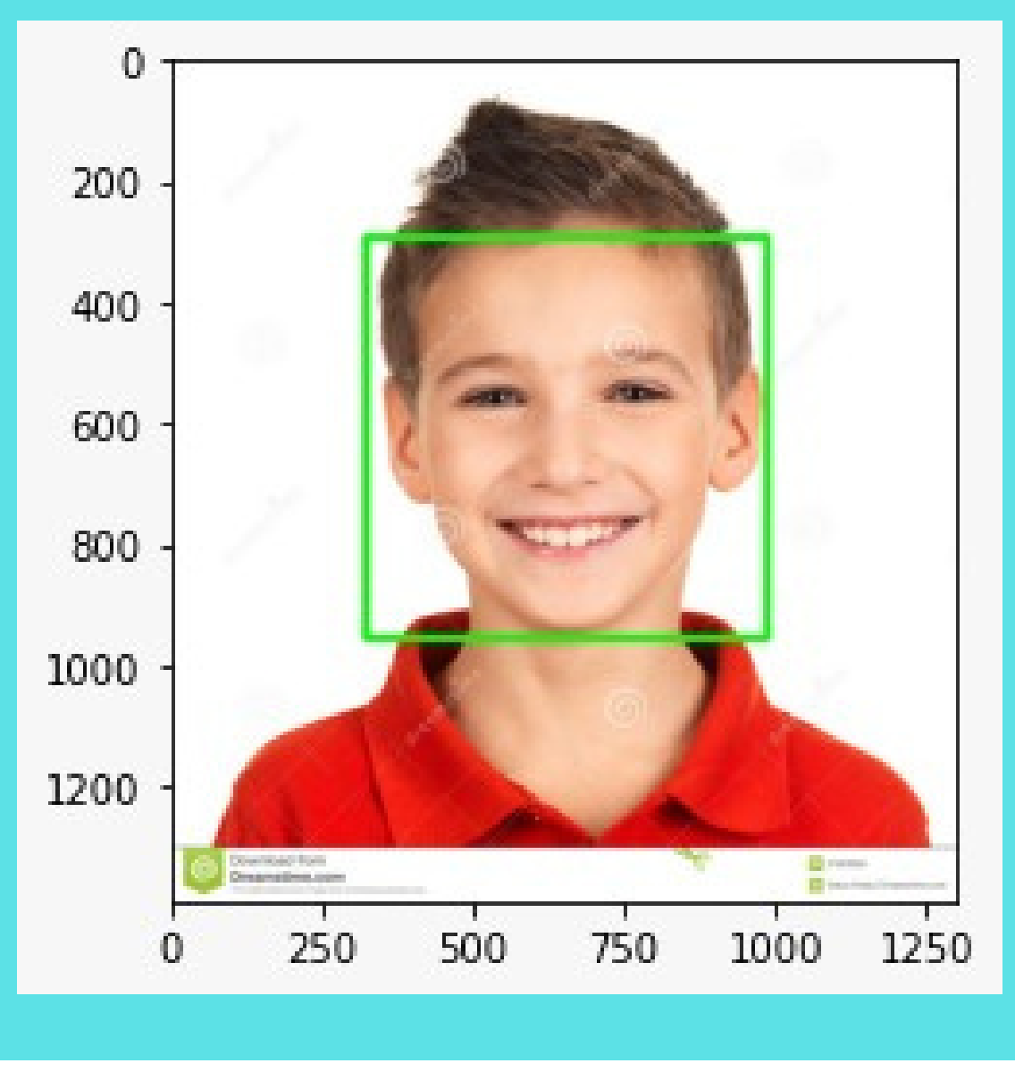
```
In [18]: #deep learning

         import tensorflow as tf
         from tensorflow import keras
         from tensorflow.keras import layers

In [19]: model = tf.keras.applications.MobileNetV2()    ##pre - trained MOdel

In [20]: model.summary()

 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]']

 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
__________________________________________________________________________________________________
```

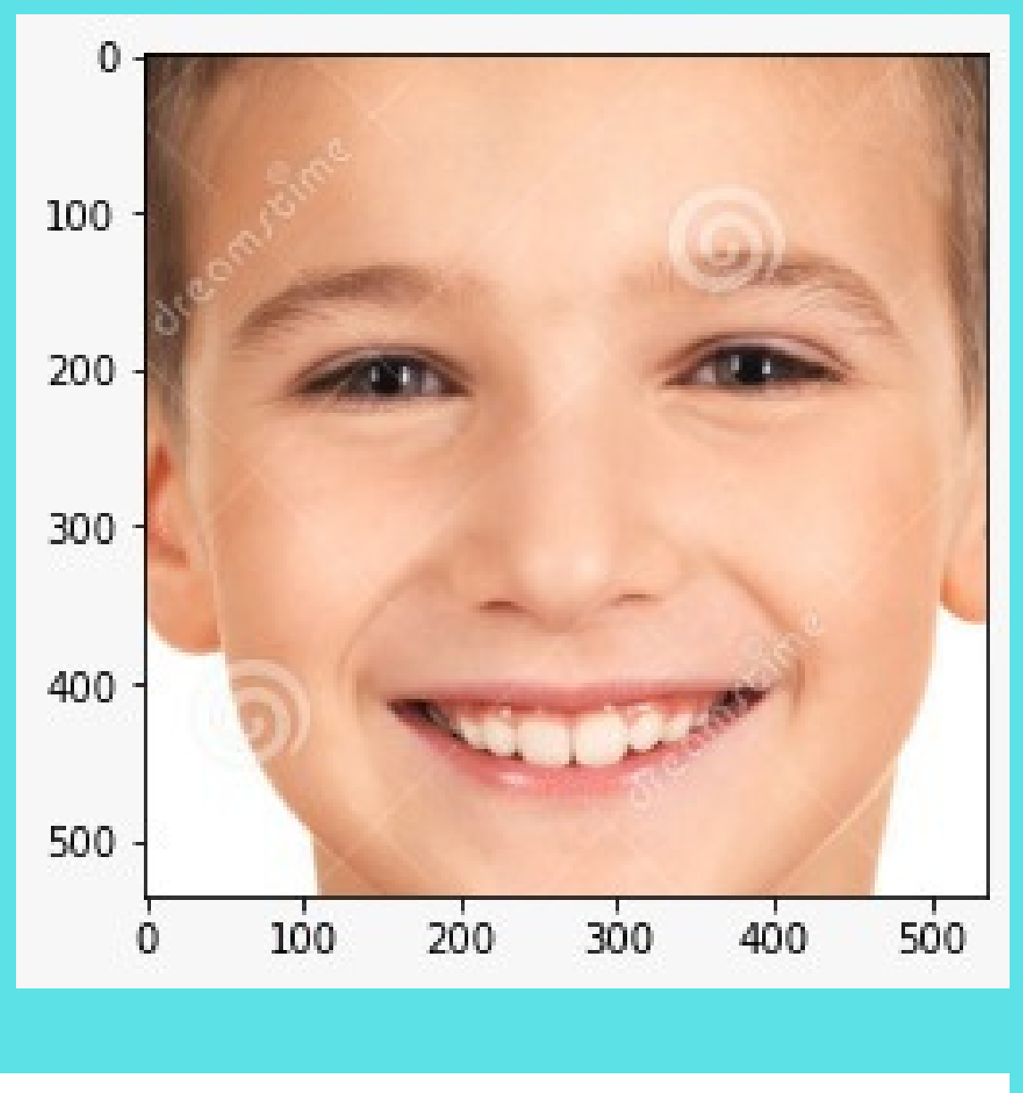
## Haar Cascade

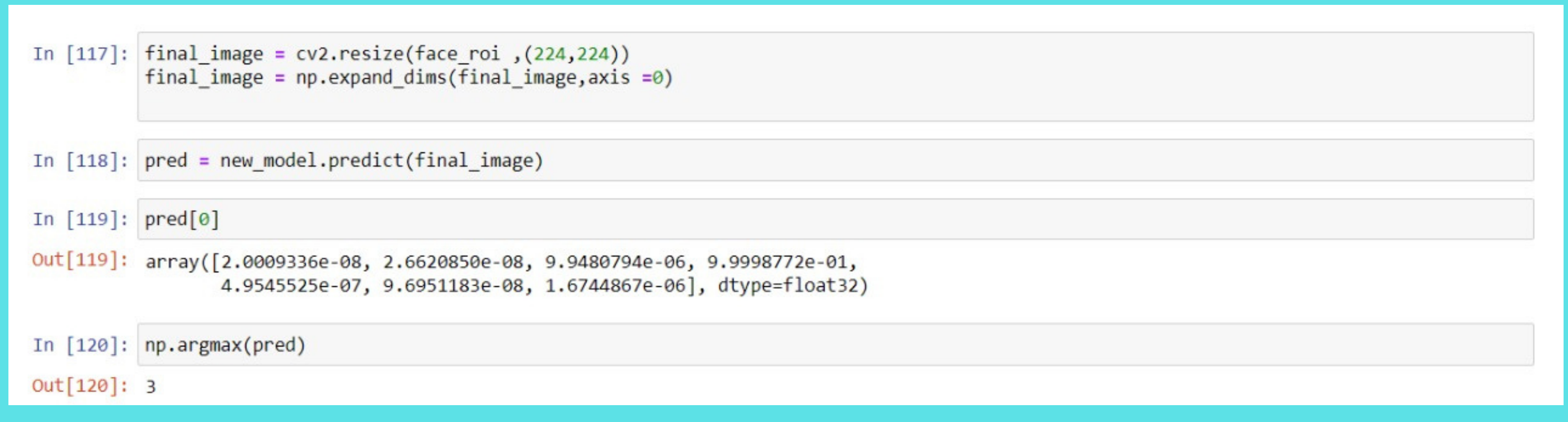
- Haar Cascade is an algorithm that is used to detect a face quickly and in real-time. At the same time, CNN utilizes the convolution process by moving a convolution (filter) kernel of a specific size to the next image from the result of multiplying the image with the filter used.
- haarcascade_frontalface_default.xml

## Results









```
In [117]: final_image = cv2.resize(face_roi ,(224,224))
          final_image = np.expand_dims(final_image,axis =0)

In [118]: pred = new_model.predict(final_image)

In [119]: pred[0]

Out[119]: array([2.0009336e-08, 2.6620850e-08, 9.9480794e-06, 9.9998772e-01,
                 4.9545525e-07, 9.6951183e-08, 1.6744867e-06], dtype=float32)

In [120]: np.argmax(pred)

Out[120]: 3
```

- We extracted specific labels, converted images to arrays, preprocess our dataset .We trained our model on a train data and achieved the accuracy of 93.04% and a loss value of 1.971.