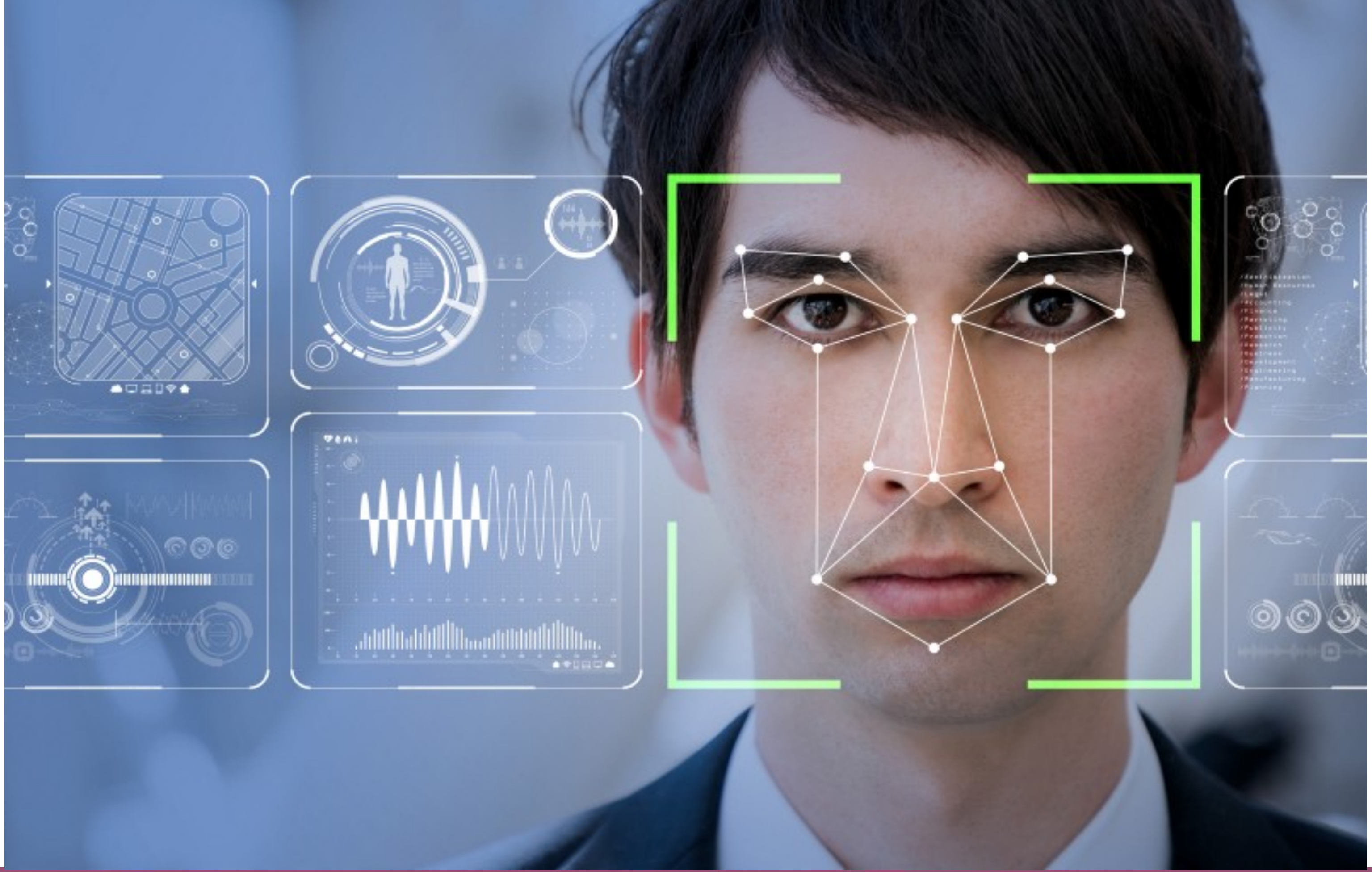
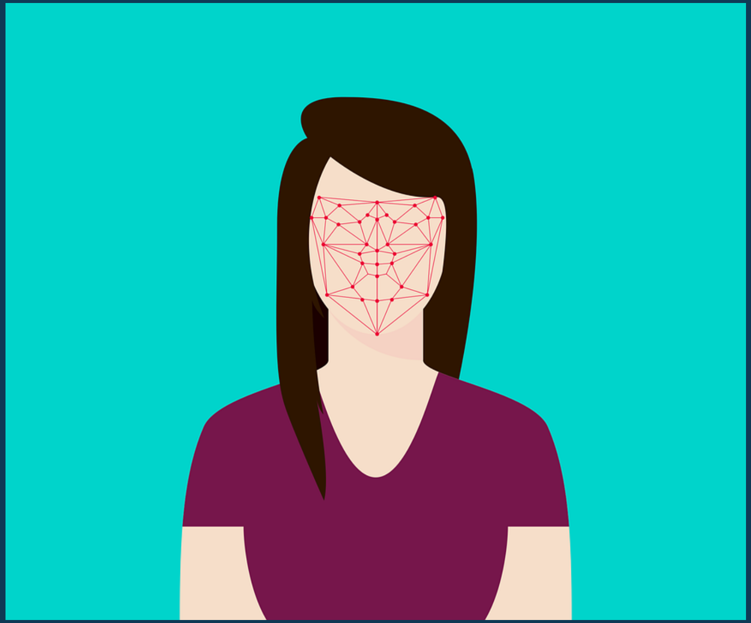


Facial recognition

Avoid a fight in venues with Nonlutte app

Emotion Detection APP

Easy to Implementation and accurateness much better



The emotions detected are anger, disgust, fear, happy, sad, surprised, and normal. These emotions are understood to be cross-culturally and universally communicated with particular facial expressions.

About the Project

Our model converts the video into vectors of video level feature extracts and to introduce knowledge transfer a largescale emotion-centric auxiliary image set is being used to classify the emotions.

Overview of the System

The Two main stages in the system are:

- Feature extraction
- Feature classification

The feature extraction stage involves pre-processing stages such as acquiring a sequence of images (15 frames/second) using a video camera and detecting the facial region of the image and standardizing the properties for lighting the image , this application BRIEF feature extraction is being used.

Key Benefit

Quick and easy. Check each person for less than 5 seconds

Accurate and objective. Different thresholds can be set for objective screening.

Friendly shielding. No contact and no interference.

Intelligent early warning. Real-time dynamic early warning function.

Facial Expression Recognition with Keras

Approach

- Data Collection
- Model Building
- Model Evaluation
- Testing

Data Collection

We would apply convolutional neural networks to tackle this task. And we will construct CNN with Keras using TensorFlow backend.

Dataset

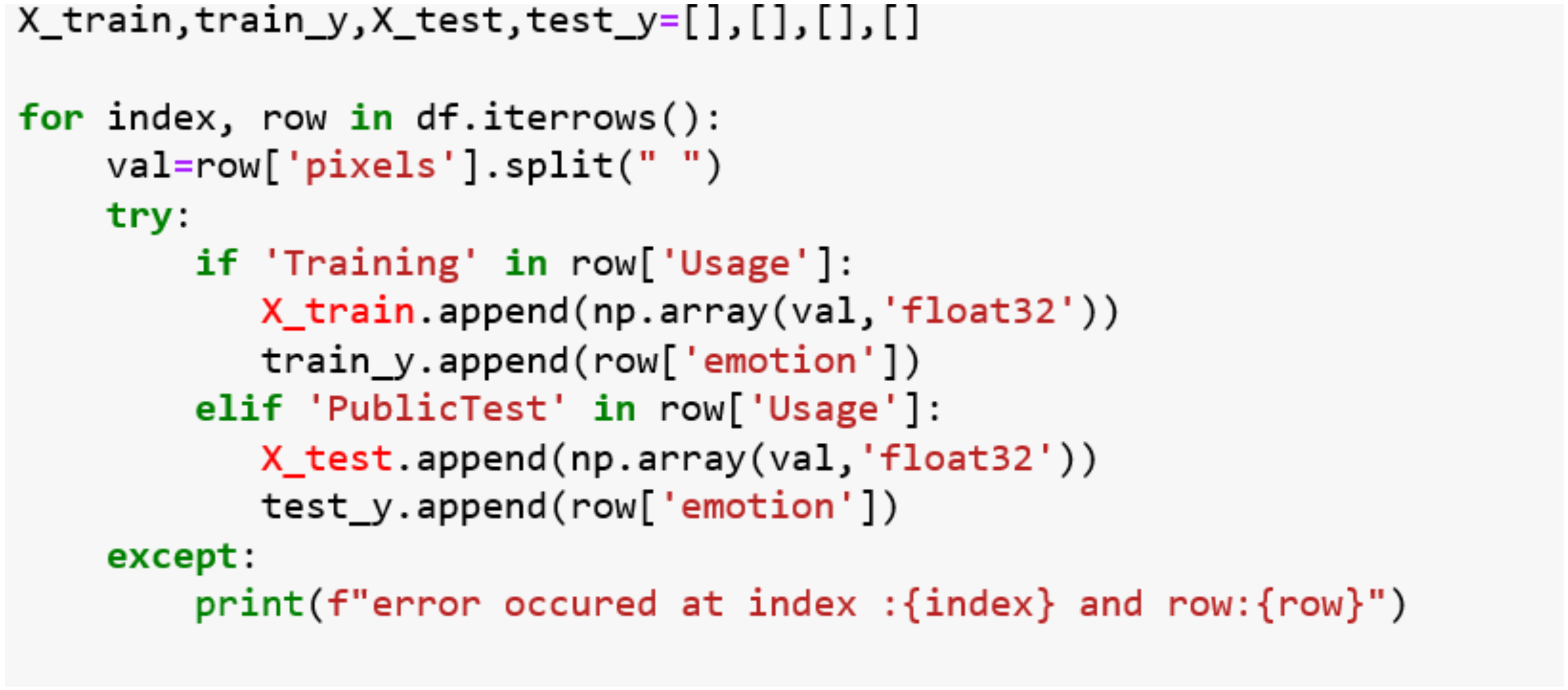
The both training and evaluation operations would be handled with Fec2013 dataset. Compressed version of the dataset takes 92 MB space whereas uncompressed version takes 295 MB space. There are 28K training and 3K testing images in the dataset. Each image was stored as 48×48 pixel. The pure dataset consists of image pixels (48×48=2304 values), emotion of each image and usage type (as train or test instance).

K Keras

A deep learning library

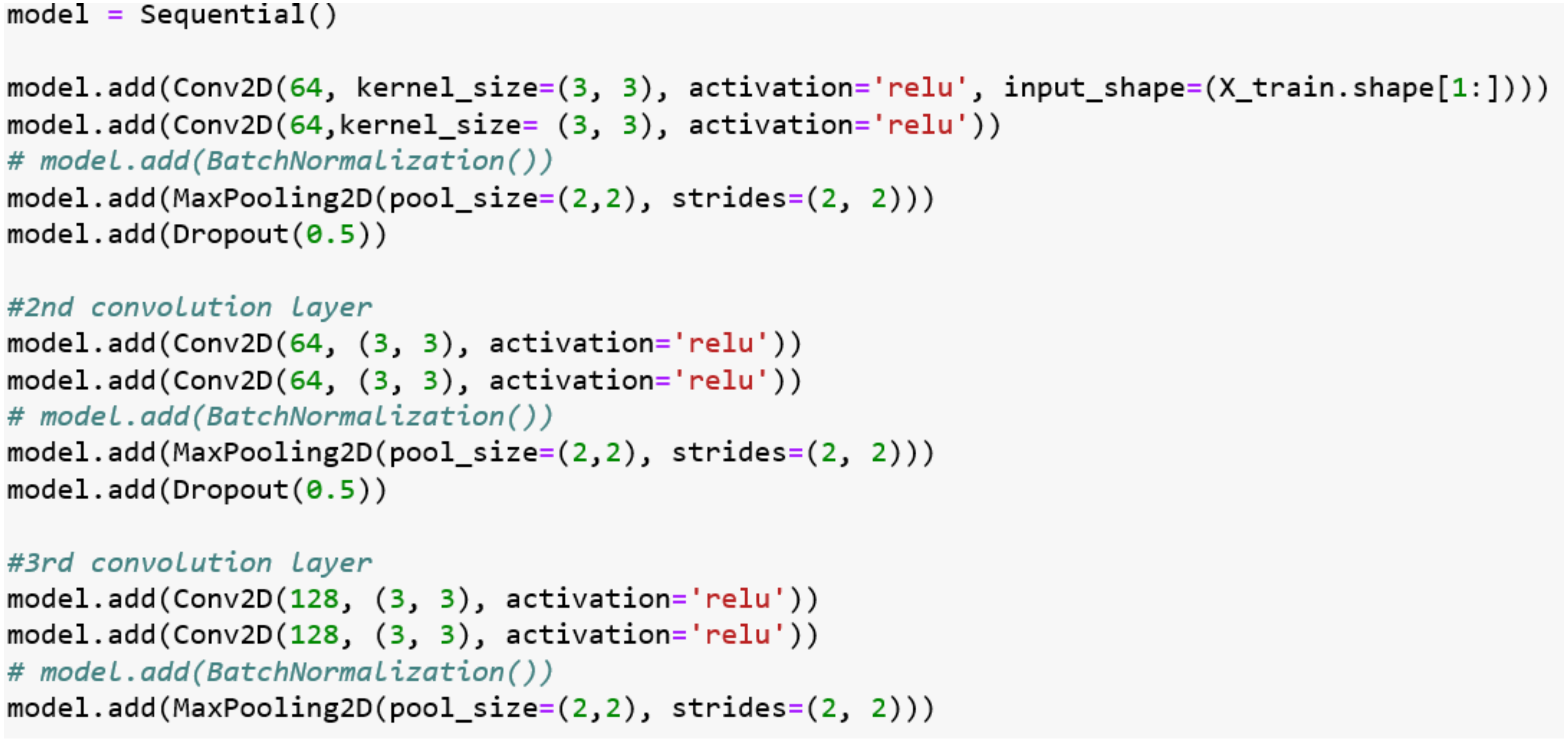
Model Building & Training

Train and test set can be stored into dedicated variables.



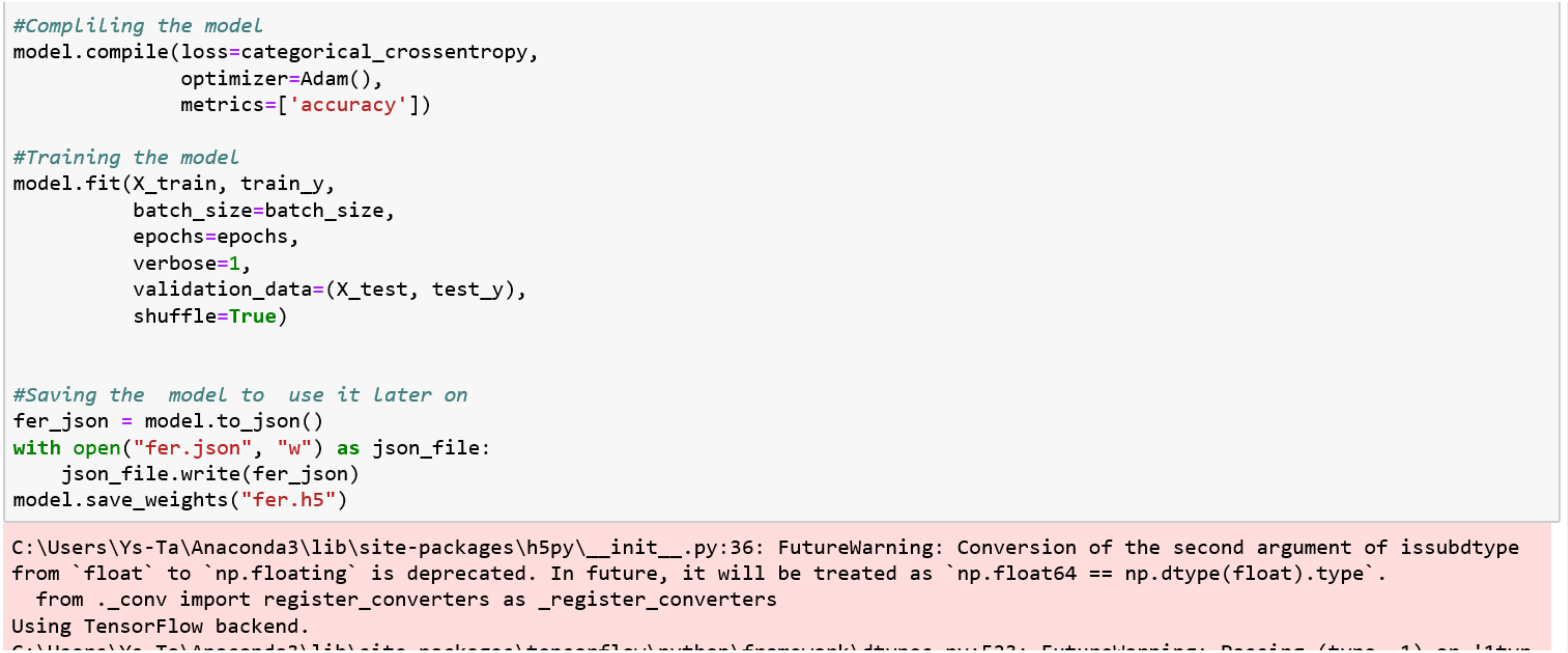
```
X_train,train_y,X_test,test_y=[],[],[],[]

for index, row in df.iterrows():
    val=row['pixels'].split(" ")
    try:
        if 'Training' in row['Usage']:
           X_train.append(np.array(val,'float32'))
           train_y.append(row['emotion'])
        elif 'PublicTest' in row['Usage']:
           X_test.append(np.array(val,'float32'))
           test_y.append(row['emotion'])
    except:
        print(f"error occured at index :{index} and row:{row}")
```

CNN structure



```
model = Sequential()

model.add(Conv2D(64, kernel_size=(3, 3), activation='relu', input_shape=(X_train.shape[1:])))
model.add(Conv2D(64,kernel_size= (3, 3), activation='relu'))
# model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2), strides=(2, 2)))
model.add(Dropout(0.5))

#2nd convolution layer
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(Conv2D(64, (3, 3), activation='relu'))
# model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2), strides=(2, 2)))
model.add(Dropout(0.5))

#3rd convolution layer
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(Conv2D(128, (3, 3), activation='relu'))
# model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2), strides=(2, 2)))
```

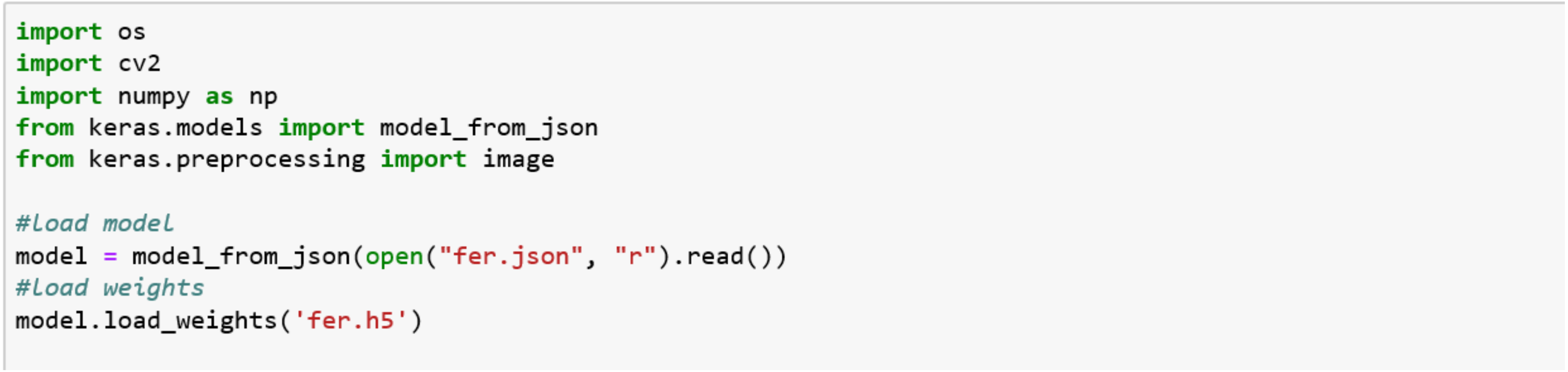
Model Saving



```
#Compliling the model
model.compile(loss=categorical_crossentropy,
              optimizer=Adam(),
              metrics=['accuracy'])

#Training the model
model.fit(X_train, train_y,
          batch_size=batch_size,
          epochs=epochs,
          verbose=1,
          validation_data=(X_test, test_y),
          shuffle=True)


#Saving the  model to  use it later on
fer_json = model.to_json()
with open("fer.json", "w") as json_file:
    json_file.write(fer_json)
model.save_weights("fer.h5")
```

```
C:\Users\Ys-Ta\Anaconda3\lib\site-packages\h5py\__init__.py:36: FutureWarning: Conversion of the second argument of issubdtype
from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 == np.dtype(float).type`.
  from ._conv import register_converters as _register_converters
Using TensorFlow backend.
C:\Users\Ys-Ta\Anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:523: FutureWarning: Passing (type, 1) or '1typ
```

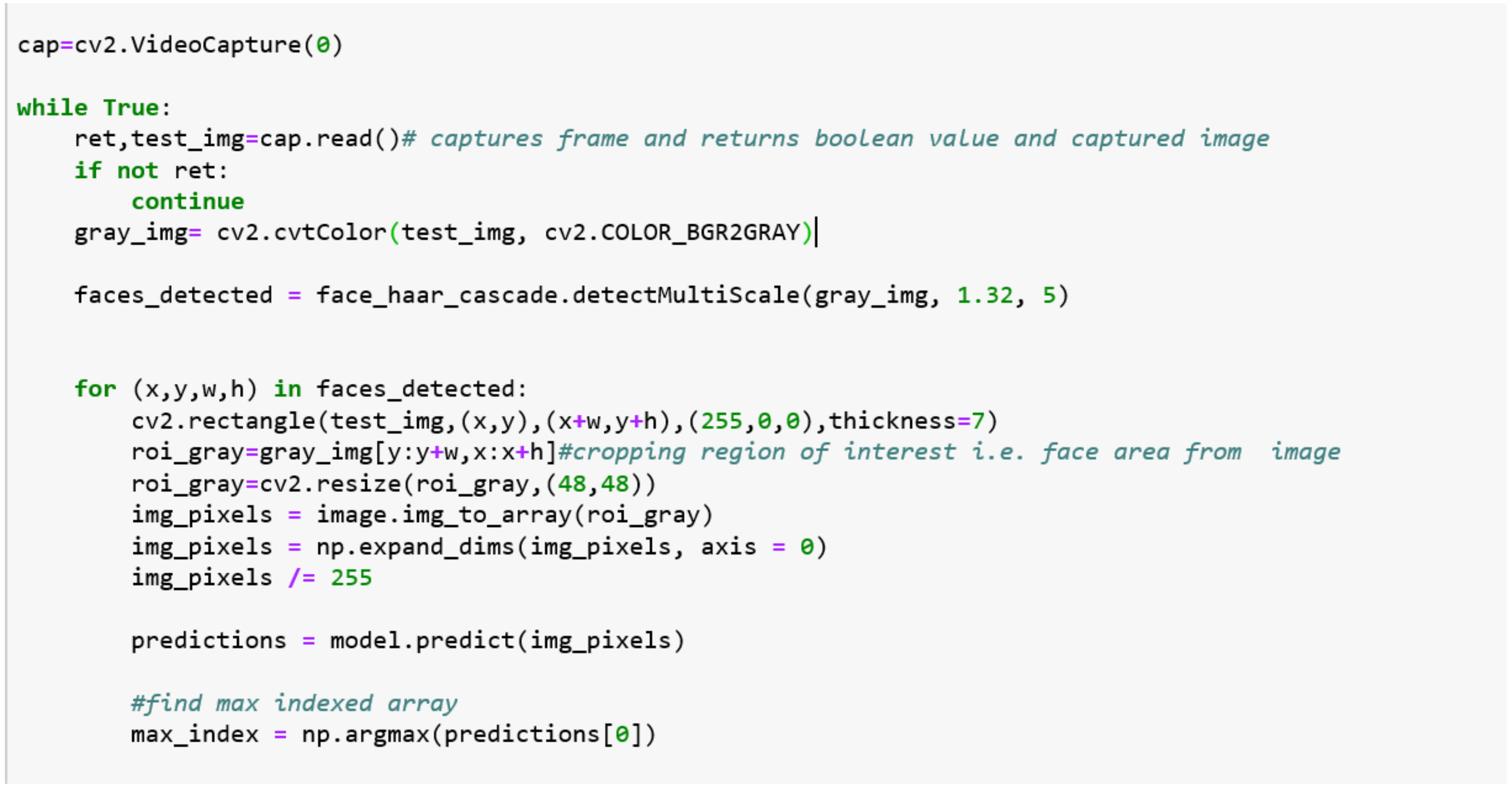
Load Model



```python
import os
import cv2
import numpy as np
from keras.models import model_from_json
from keras.preprocessing import image

#load model
model = model_from_json(open("fer.json", "r").read())
#load weights
model.load_weights('fer.h5')
```

Using OpenCV

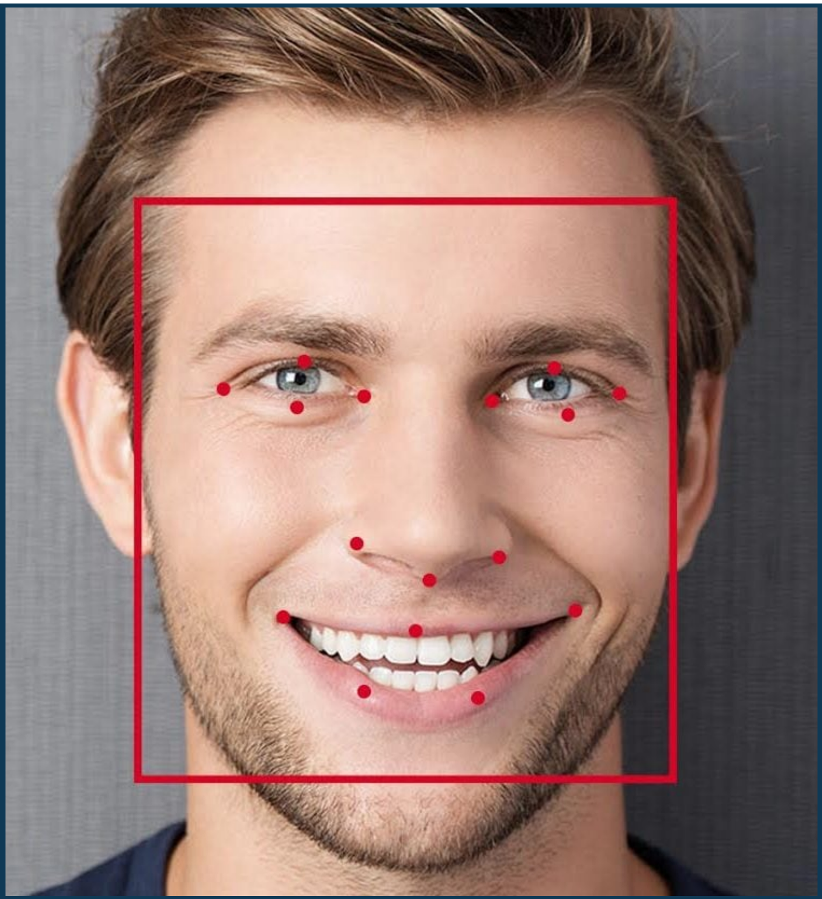
Read an image

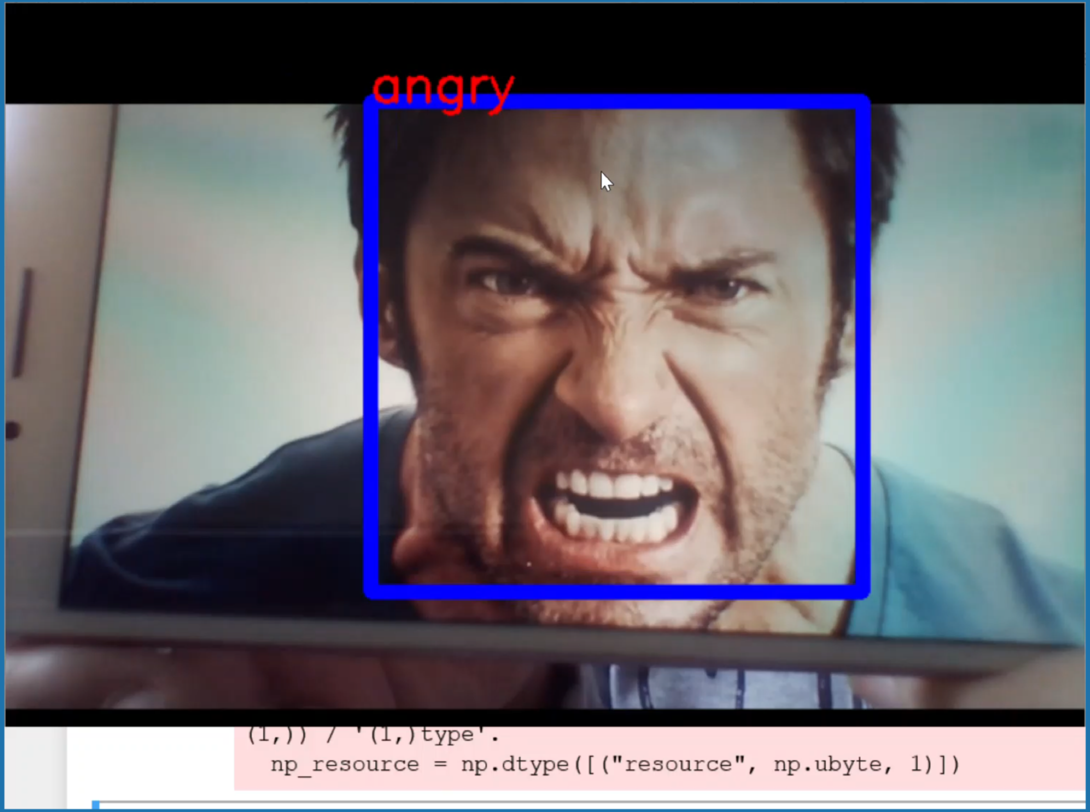


```
cap=cv2.VideoCapture(0)

while True:
    ret,test_img=cap.read()# captures frame and returns boolean value and captured image
    if not ret:
        continue
    gray_img= cv2.cvtColor(test_img, cv2.COLOR_BGR2GRAY)

    faces_detected = face_haar_cascade.detectMultiScale(gray_img, 1.32, 5)


    for (x,y,w,h) in faces_detected:
        cv2.rectangle(test_img,(x,y),(x+w,y+h),(255,0,0),thickness=7)
        roi_gray=gray_img[y:y+w,x:x+h]#cropping region of interest i.e. face area from  image
        roi_gray=cv2.resize(roi_gray,(48,48))
        img_pixels = image.img_to_array(roi_gray)
        img_pixels = np.expand_dims(img_pixels, axis = 0)
        img_pixels /= 255

        predictions = model.predict(img_pixels)

        #find max indexed array
        max_index = np.argmax(predictions[0])
```

Real time solution

Besides, we can apply emotion analysis on a video streaming or web cam capturing. We try to act all emotion classes. As seen, this implementation runs very fast.





```
(1,)) / '(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])
```

Resource

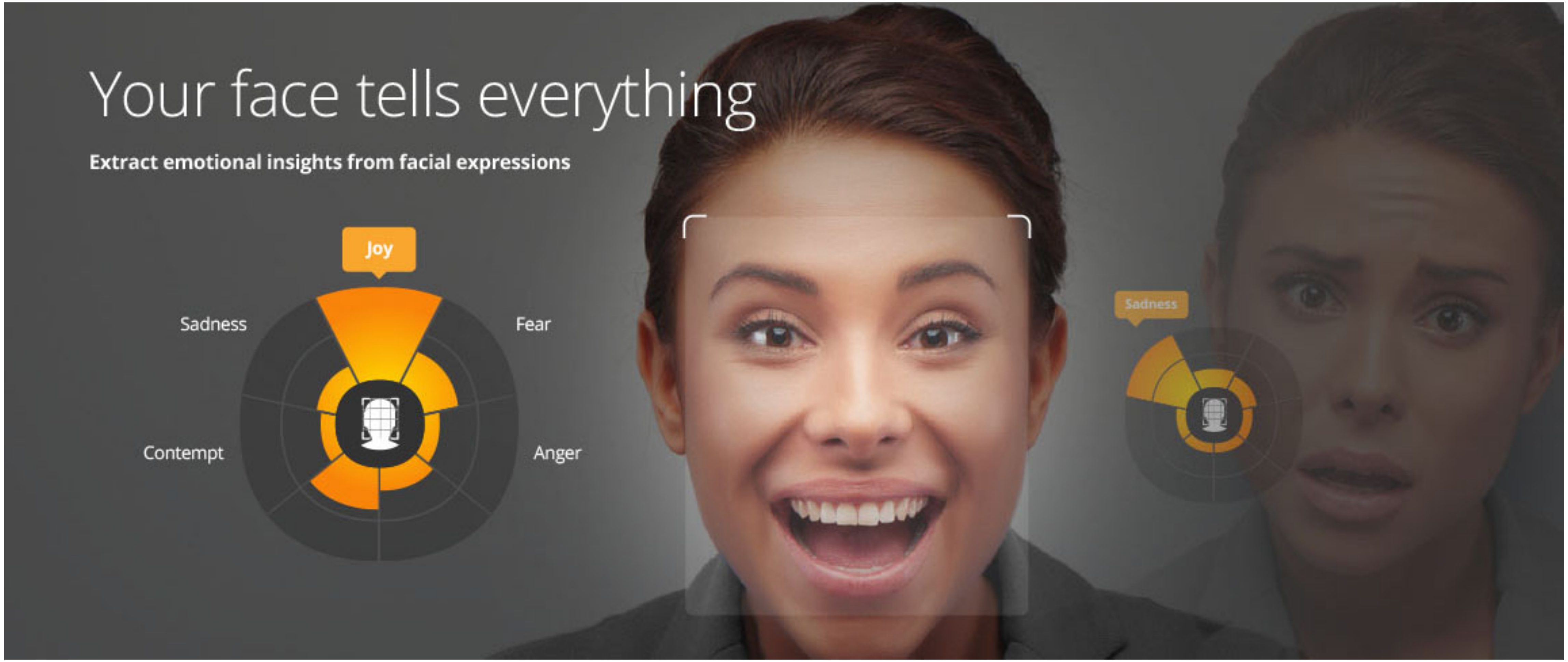
https://github.com/BishalLakha/Facial-Expression-Recognition-with-Keras

https://sefiks.com/2018/01/01/facial-expression-recognition-with-keras/

https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_gui/py_image_display/py_image_display.html

https://www.youtube.com/watch?v=DtBu1u5aBsc

Thank you


Your face tells everything
Extract emotional insights from facial expressions
Joy
Sadness
Fear
Contempt
Anger
Sadness