Facial Expression Recognition

CIS 730 Term Project
Ishitaa Sayal

Introduction

- Facial emotion recognition is an AI technology used to analyze a person's face and interpret their expressions
- It conveys nonverbal communication cues that play a very important role in interpersonal relations.
- Nonverbal cues complements speech by helping the listener to understand the intended meaning of the spoken words.
- Emotion recognition can be done using texts, images, videos, speeches, and conversations.

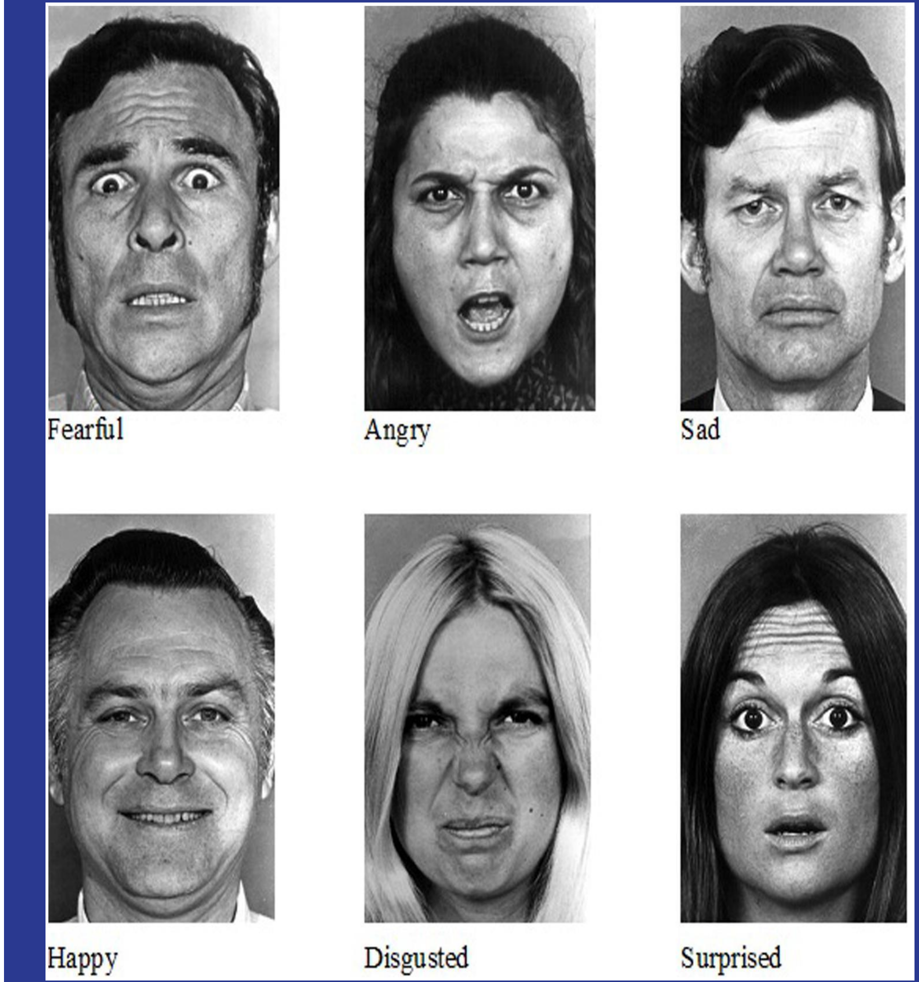
What is emotion recognition used for today ?

- Security Measure
- HR assistance
- Customer Services
- Differently abled children
- Audience engagement
- Video Game testing

Goal

- To develop an artificial agent that will detect various facial expressions in real time.

- In This project we are taking 7 universal facial expressions:
    - Neutral
    - Sad
    - Happy
    - Fear
    - Angry
    - Disgust
    - Surprised



https://algorithmia.com/blog/introduction-to-emotion-recognition

Dataset

- Source : Kaggle[1]
- It comprises pre-cropped, 48-by-48-pixel grayscale images of faces each labeled with one of the 7 emotion classes: anger, disgust, fear, happiness, sadness, surprise, and neutral.
- Dataset has a training set of 35887 facial images with facial expression labels.

[1] https://www.kaggle.com/sionehoghen/facial-expression

Methodology

- Performing data processing and data augmentation to make data more robust
  - Resizing the images
  - Random flip
  - Random rotation
  - Random zoom

- Creating a Convolution Neural Network Model (CNN model)
  - Creating EfficientNetB2 model

- Training and testing the data, which is done in two phases
  - Phase I: entire data is trained
  - Phase II: we freeze the initial layers and perform training in last few layers

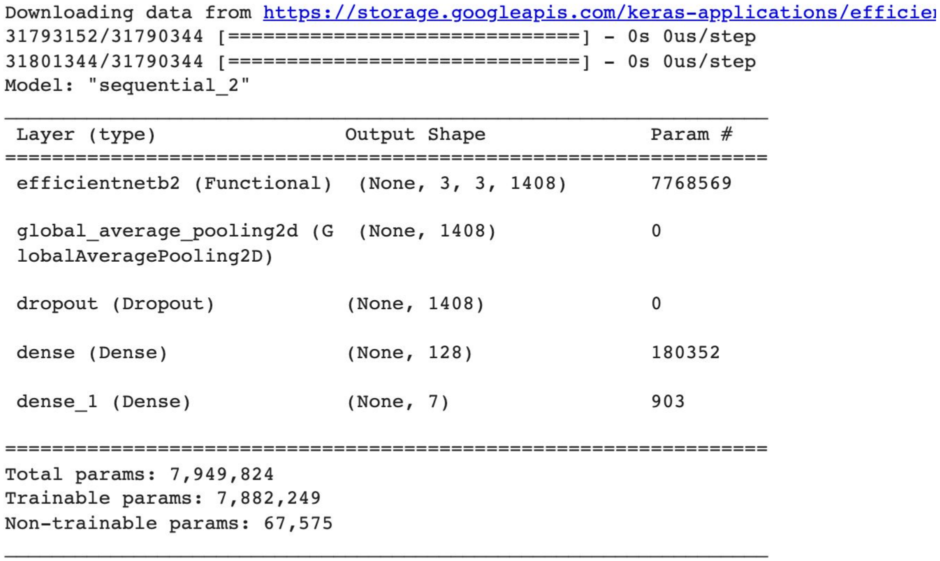
EfficientNetB2

- As the name suggest efficientnet models are more efficient and produce better results as compared to results produced by state-of-art-models.

- Generally, the models are made to wide, deep, or with a very high resolution

- Increasing these characteristics helps the model initially but it quickly saturates and the model made just has more parameters and is therefore not efficient.

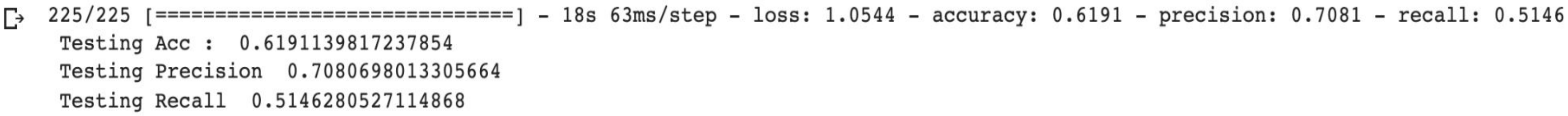
- In EfficientNet they are scaled in a more principled way i.e. gradually everything is increased. Therefore making these models more efficient as compared to other models.



```
Downloading data from https://storage.googleapis.com/keras-applications/efficie
31793152/31790344 [==============================] - 0s 0us/step
31801344/31790344 [==============================] - 0s 0us/step
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 efficientnetb2 (Functional)  (None, 3, 3, 1408)       7768569

 global_average_pooling2d (G  (None, 1408)             0
 lobalAveragePooling2D)

 dropout (Dropout)           (None, 1408)              0

 dense (Dense)               (None, 128)               180352

 dense_1 (Dense)             (None, 7)                 903

=================================================================
Total params: 7,949,824
Trainable params: 7,882,249
Non-trainable params: 67,575
_________________________________________________________________
```

Evaluation

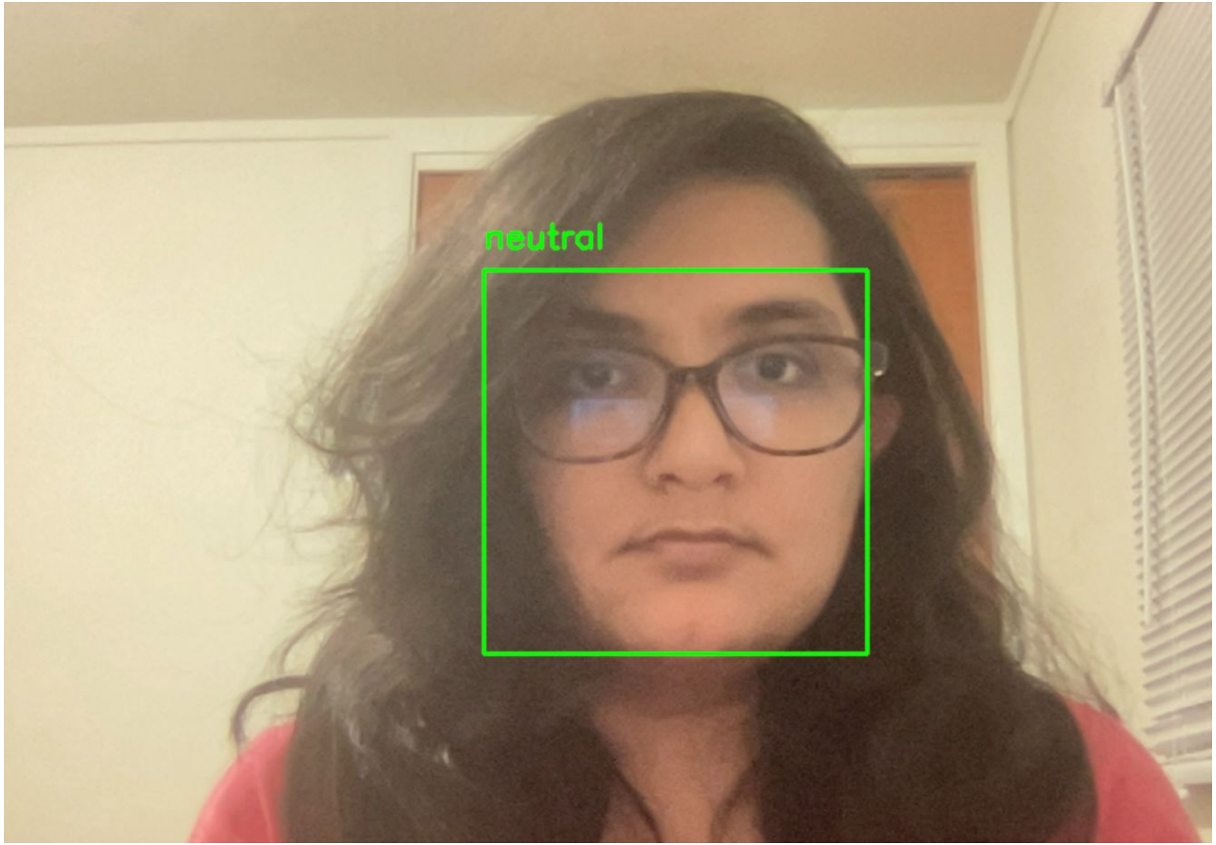
- Precision , Recall, Accuracy



```
225/225 [==============================] - 18s 63ms/step - loss: 1.0544 - accuracy: 0.6191 - precision: 0.7081 - recall: 0.5146
 Testing Acc :  0.6191139817237854
 Testing Precision  0.7080698013305664
 Testing Recall  0.5146280527114868
```

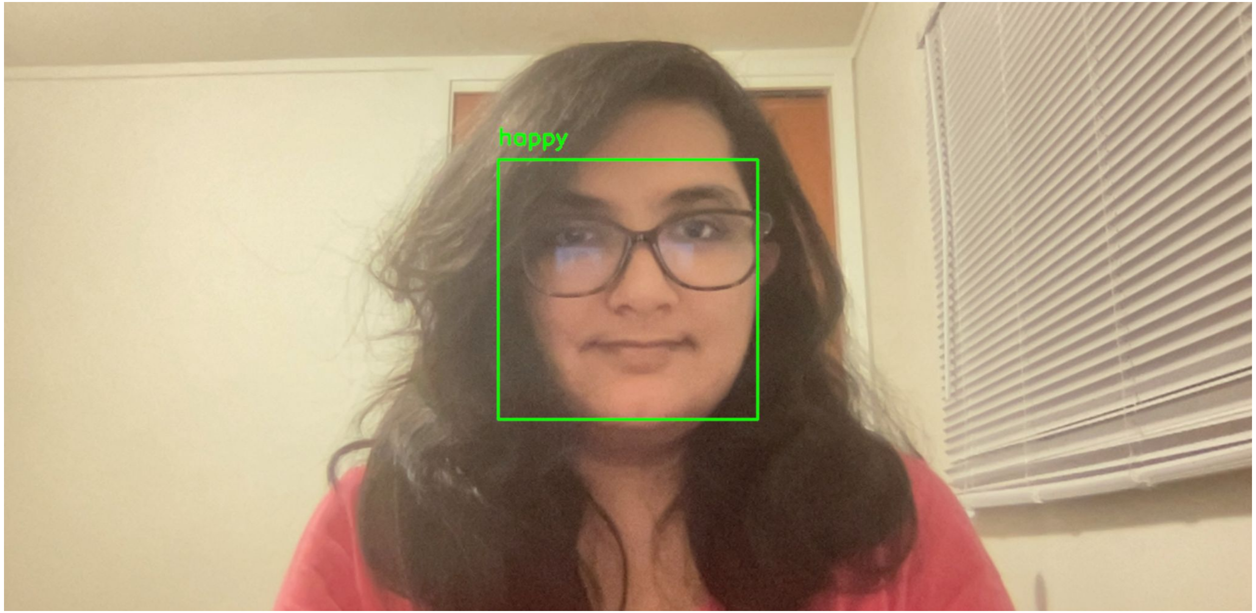
- As we can see accuracy is around 60 percent, as it is a very noisy dataset

Expression Recognition Results (1)



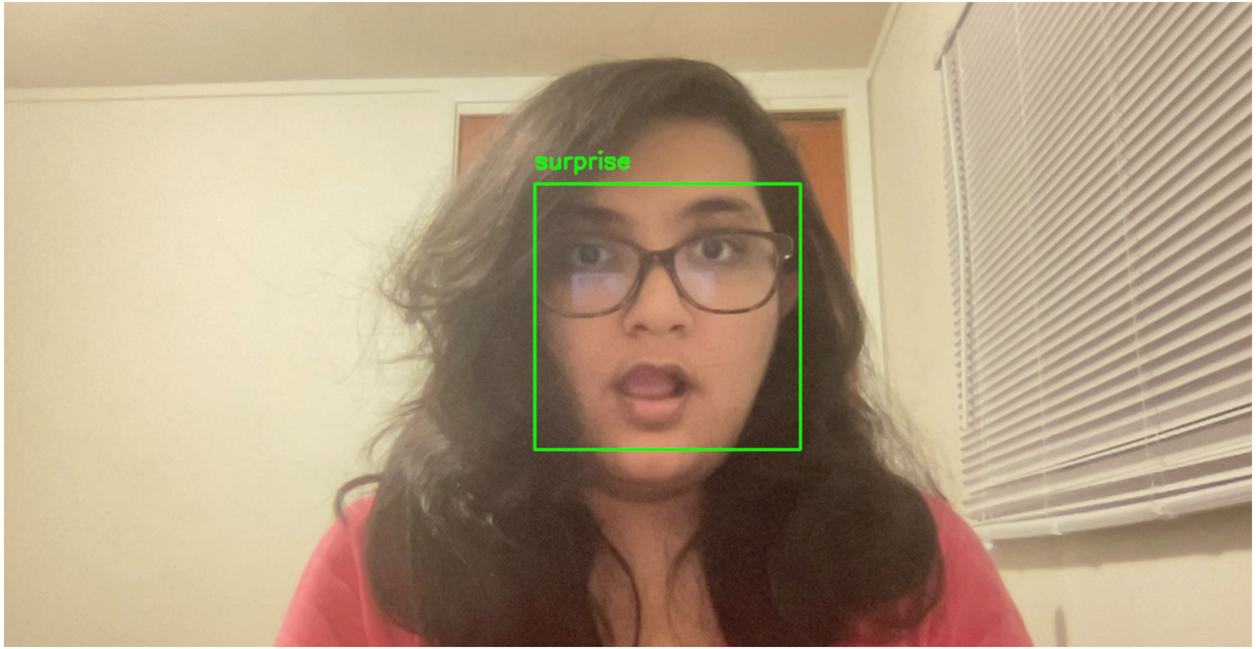
Agent detecting neutral expressions

Expression Recognition Results (2)



Agent detecting happy expressions

Expression Recognition Results (3)



Agent detecting surprise expressions

Tools Used

- Python
- Tensorflow
- Keras
- Open CV
- Google Colab
- Jupyter Notebook

https://www.google.com/

Various Approaches to Improve Training (1)

- Performing face detection and cropping face images in our training dataset.
  - Although we have closely cropped faces in our dataset, they are very noisy.
  - By filtering out all images which were not detected by face detection algorithm we can make our model more robust.

Various Approaches to Improve Training (2 & 3)

- Increase Number of phases and Decrease number of epochs per phase while training.
  - **For example:** You can train a model using a phase 1 with 6 epochs repeated by phase 2 with 4 epochs for 2 times

- Try out a better version of EfficientNet.
  - There are even better versions of EfficientNet such as B3 to B7.
  - We have used EfficientNetB2 just to keep it less computationally intensive.
  - So that it can be executed even on Low hardware specification. .

Future Plans

- Perform more data augmentation to improve the accuracy of the model
- Convert it into a live project for a good social cause.

DEMO

THANK YOU