Capstone Project : 5

Face Emotion Recognition

Team Members

Atul Chouhan

Mayank Mishra

Points for Discussion

- Introduction
- Understanding Problem Statement
- Data Set Information
- Dependencies
- CNN Modelling
- Loss & Accuracy plot
- Live Demo
- Deployment
- Emotion Recognition from webapp
- Challenges
- Conclusion
- Q&A

Understanding Problem Statement

- The Indian education landscape has been undergoing rapid changes for the past 10 years owing to the advancement of web-based learning services, specifically, eLearning platforms.
- Global E-learning is estimated to witness an 8X over the next 5 years to reach USD 2B in 2021.
- India is expected to grow with a CAGR of 44% crossing the 10M users mark in 2021. Although the market is growing on a rapid scale, there are major challenges associated with digital learning when compared with brick and mortar classrooms. One of many challenges is how to ensure quality learning for students. Digital platforms might overpower physical classrooms in terms of content quality but when it comes to understanding whether students are able to grasp the content in a live class scenario is yet an open-end challenge. In a physical classroom during a lecturing teacher can see the faces and assess the emotion of the class and tune their lecture accordingly, whether he is going fast or slow. He can identify students who need special attention.
- Digital classrooms are conducted via video telephony software program (ex-Zoom) where it's not possible for medium scale class (25-50) to see all students and access the mood. Because of this drawback, students are not focusing on content due to lack of surveillance.
- While digital platforms have limitations in terms of physical surveillance but it comes with the power of data and machines which can work for you. It provides data in the form of video, audio, and texts which can be analyzed using deep learning algorithms.
- Deep learning backed system not only solves the surveillance issue, but it also removes the human bias from the system, and all information is no longer in the teacher's brain rather translated in numbers that can be analyzed and tracked.

Dataset Information

- I have built a deep learning model which detects the real time emotions of students through a webcam so that teachers can understand if students are able to grasp the topic according to students' expressions or emotions and then deploy the model. The model is trained on the FER-2013 dataset.

- This dataset consists of 35887 grayscale, 48x48 sized face images with seven emotions -angry, disgusted, fearful, happy, neutral, sad and surprised.

- Here is the dataset link:- https://www.kaggle.com/msambare/fer2013

Emotion	Number of images for Training	Number of images for Testing
angry	3995	958
disgust	436	111
fear	4097	1024
happy	7215	1774
sad	4830	1247
surprised	3171	831
neutral	4965	1233

Dependencies

1.Python 3

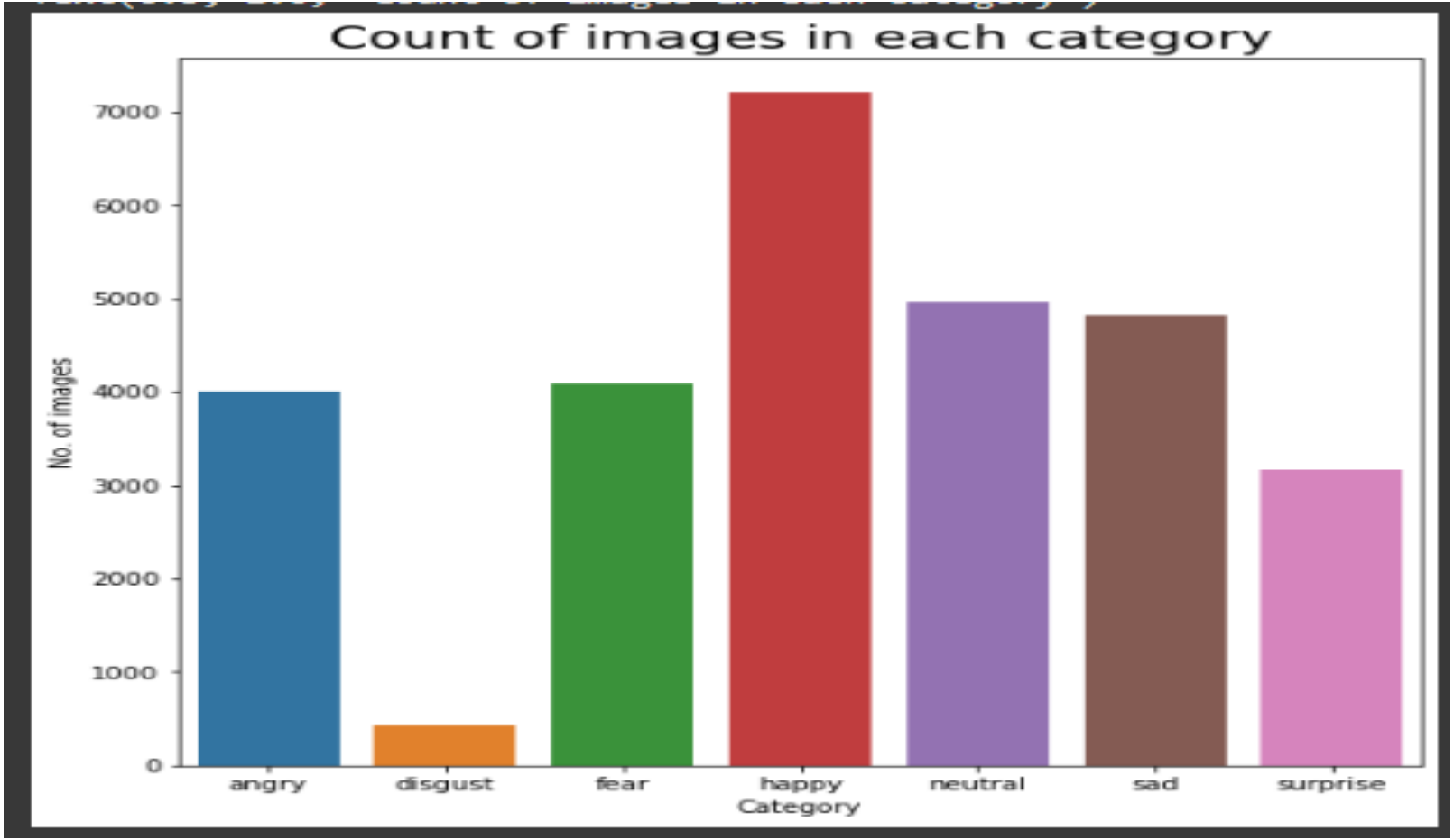
2.Tensorflow2.0

3.Streamlit

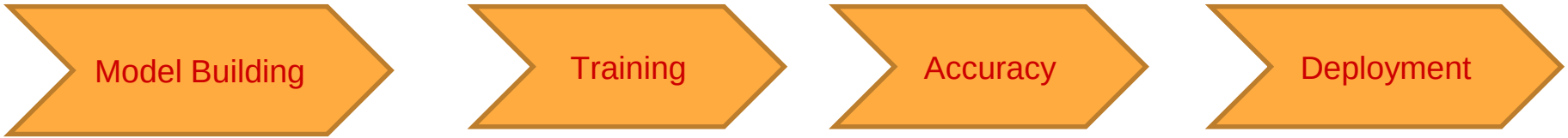
4.Streamlit-Webrtc

5.OpenCV

Count Plot of Various Emotions

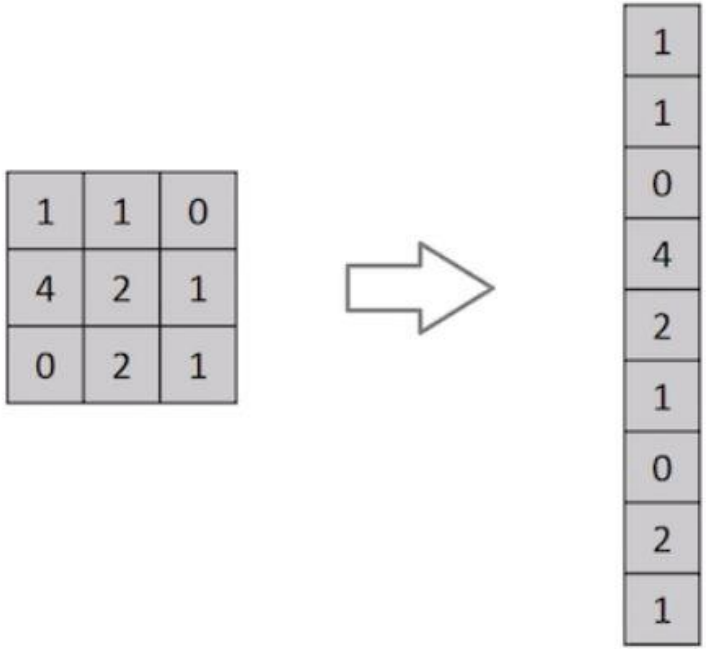


Approach Overview



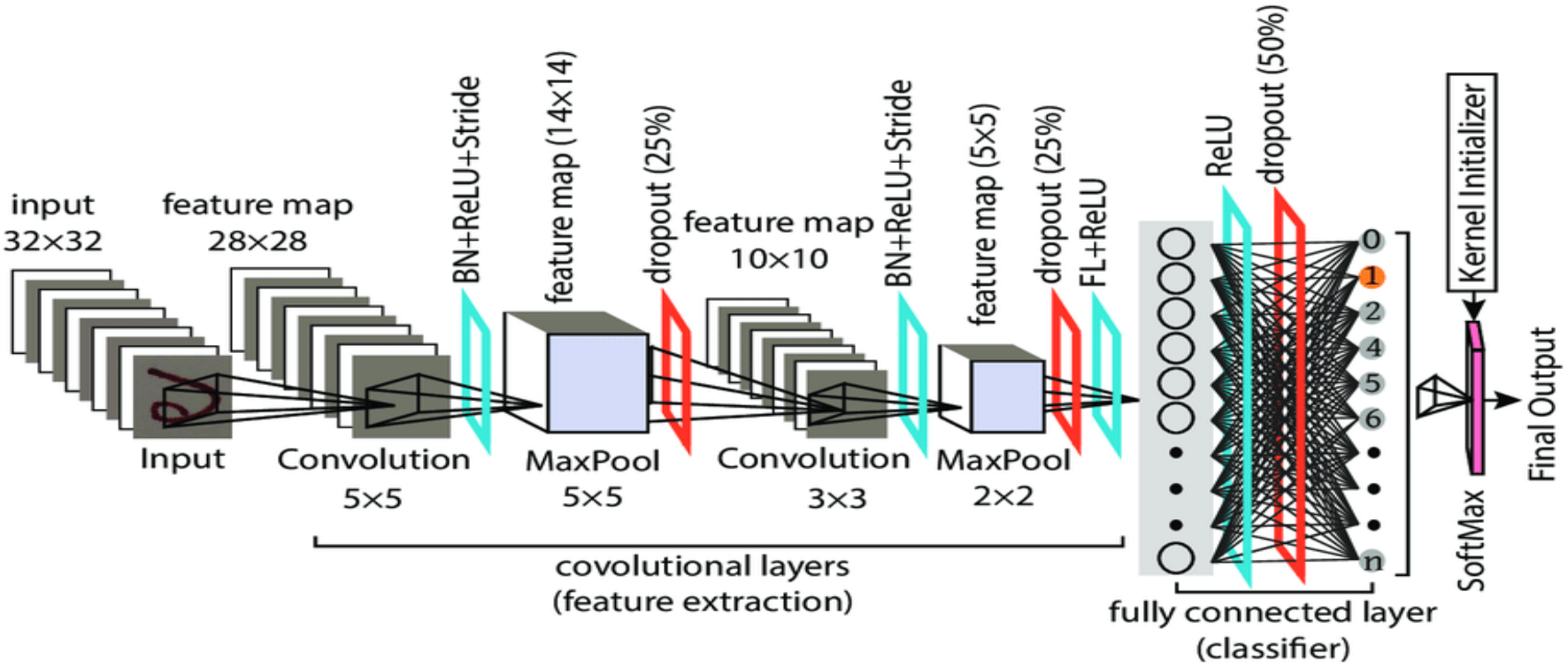
CNN (Convolutional Neural Network)

- Better option than simple neural networks.
- CNN is better as number of parameters reduces and weights can be reused.
- Made of multiple layers to capture low level features to high level features.
- Convolution layer
- Pooling layer
- Fully connected layer
- Activation layers
- Batch normalization
- Dropout layers
- Transfer learning can be applied from pre built models such as VGG, Alexnet, Resnet, Mobilenet etc.

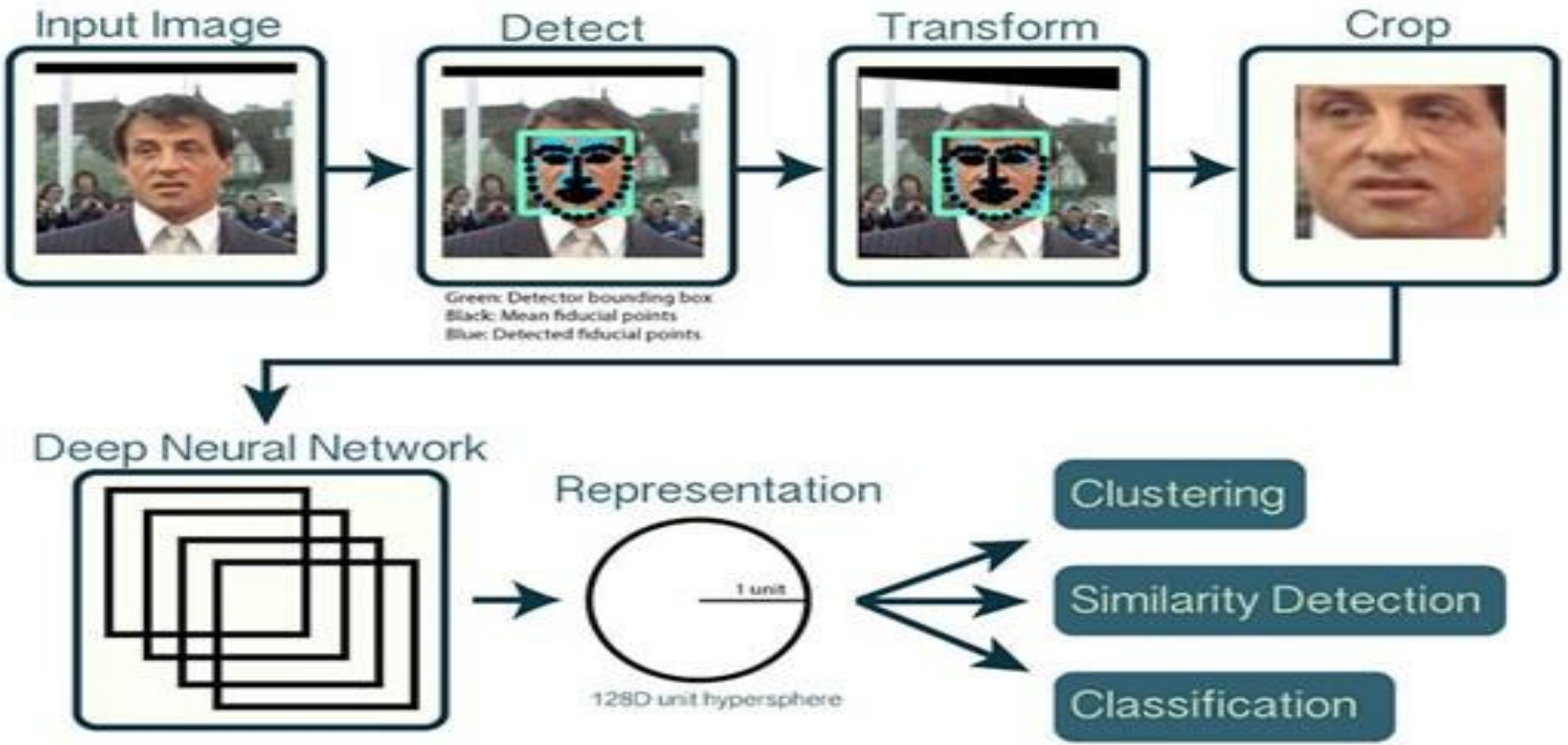


Flattening of a 3×3 image matrix into a 9×1 vector

CNN Architecture



OpenCV Python



Data Augmentation

AI

- The performance of deep learning neural networks often improves with the amount of data available.
- Data augmentation is a technique to artificially create new training data from existing training data.
- Variations of the training set images that are likely to be seen by the model

```
rescale=1./255,
rotation_range=30,
shear_range=0.3,
zoom_range=0.3,
width_shift_range=0.4,
height_shift_range=0.4,
horizontal_flip=True,
fill_mode='nearest')
```

Model Hyperparameters

Batch Size: The batch size is a hyperparameter that defines the number of samples to work through before updating the internal model parameters.

```
# Number of classes of emotion
num_emotion = 7

# Image Size
img_array = 48,48

# define batch Size
batch_size = 148
```

Epoch: The number of epochs is a hyperparameter that defines the number times that the learning algorithm will work through the entire training dataset.

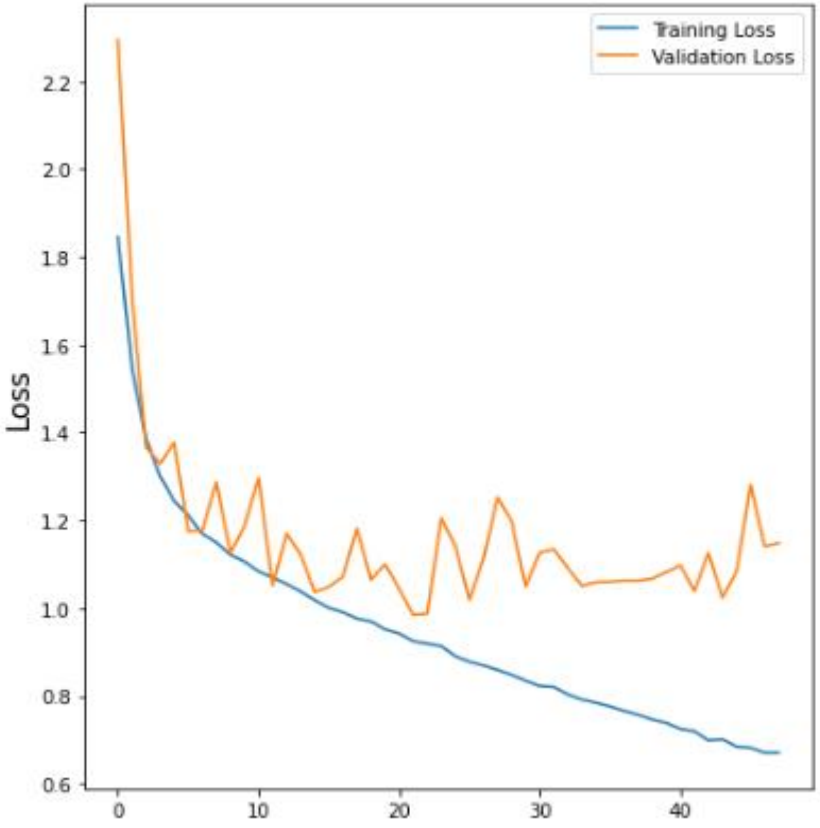
```
# Number of epochs
epochs = 45
```

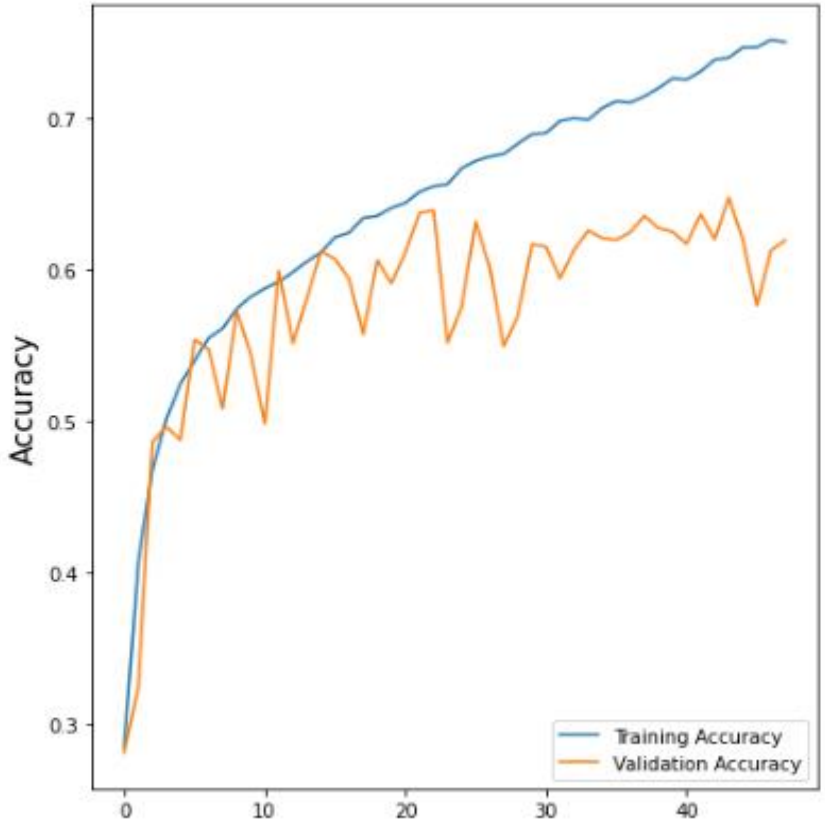
Model Training

- Training set accuracy is 77%
- Validation set accuracy is 67% at 45 epochs

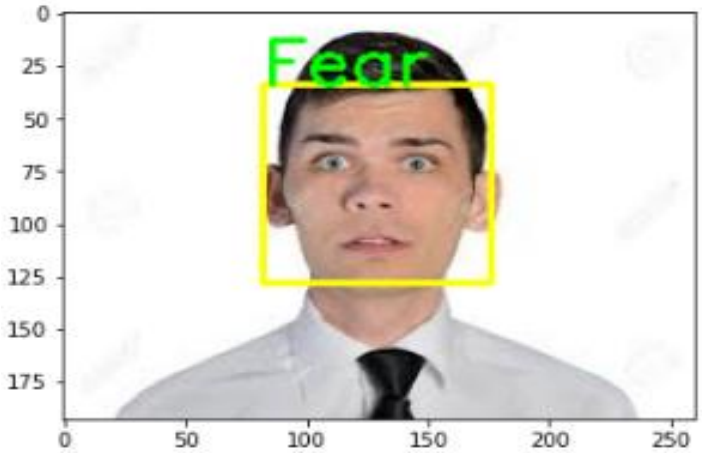
```
Epoch 00041: val_accuracy did not improve from 0.63876
Epoch 42/48
225/225 [==============================] - 33s 146ms/step - loss: 0.7198 - accuracy: 0.7302 - val_loss: 1.0397 - val_accuracy: 0.6360

Epoch 00042: val_accuracy did not improve from 0.63876
Epoch 43/48
225/225 [==============================] - 33s 146ms/step - loss: 0.6990 - accuracy: 0.7380 - val_loss: 1.1258 - val_accuracy: 0.6197

Epoch 00043: val_accuracy did not improve from 0.63876
Epoch 44/48
225/225 [==============================] - 33s 148ms/step - loss: 0.7015 - accuracy: 0.7394 - val_loss: 1.0239 - val_accuracy: 0.6471

Epoch 00044: val_accuracy improved from 0.63876 to 0.64712, saving model to model.h5
Epoch 45/48
225/225 [==============================] - 33s 146ms/step - loss: 0.6849 - accuracy: 0.7460 - val_loss: 1.0837 - val_accuracy: 0.6191

Epoch 00045: val_accuracy did not improve from 0.64712
Epoch 46/48
225/225 [==============================] - 33s 147ms/step - loss: 0.6824 - accuracy: 0.7462 - val_loss: 1.2808 - val_accuracy: 0.5759

Epoch 00046: val_accuracy did not improve from 0.64712
Epoch 47/48
225/225 [==============================] - 33s 148ms/step - loss: 0.6710 - accuracy: 0.7511 - val_loss: 1.1408 - val_accuracy: 0.6120

Epoch 00047: val_accuracy did not improve from 0.64712
Epoch 48/48
225/225 [==============================] - 33s 146ms/step - loss: 0.6713 - accuracy: 0.7497 - val_loss: 1.1485 - val_accuracy: 0.6190

Epoch 00048: val accuracy did not improve from 0.64712
```

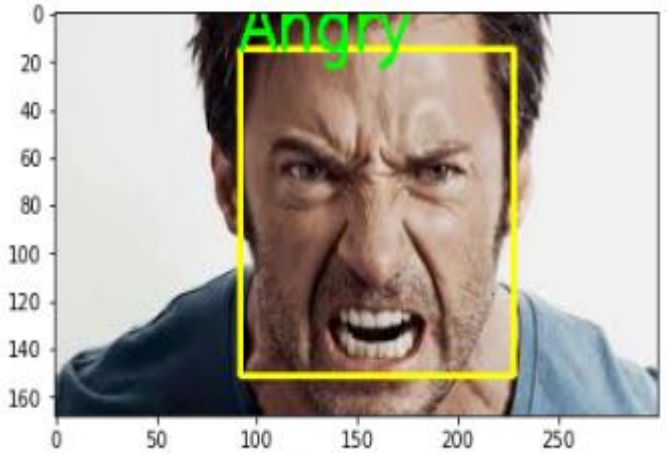
Evaluation

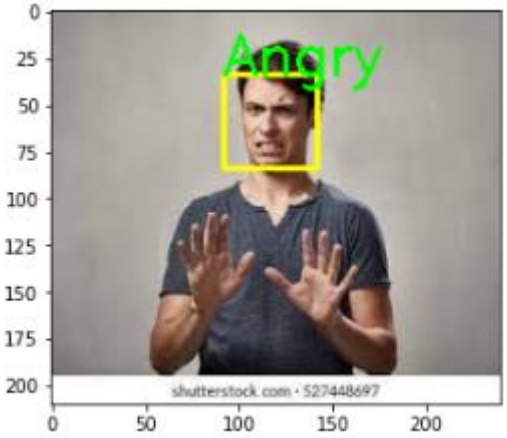

Training Loss
Validation Loss
Loss
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0
10
20
30
40


Accuracy
0.7
0.6
0.5
0.4
0.3
Training Accuracy
Validation Accuracy
0
10
20
30
40

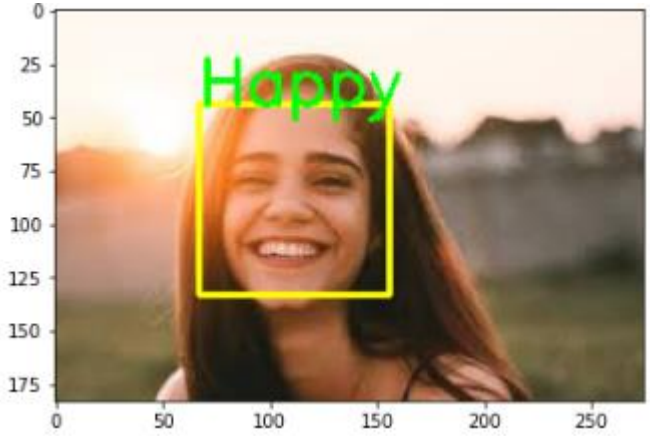
Demo Predictions

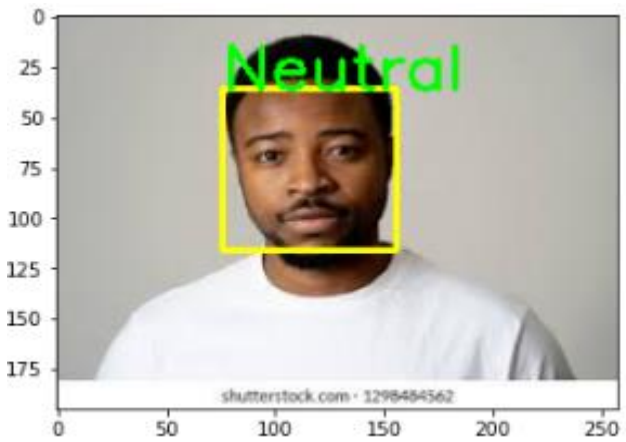






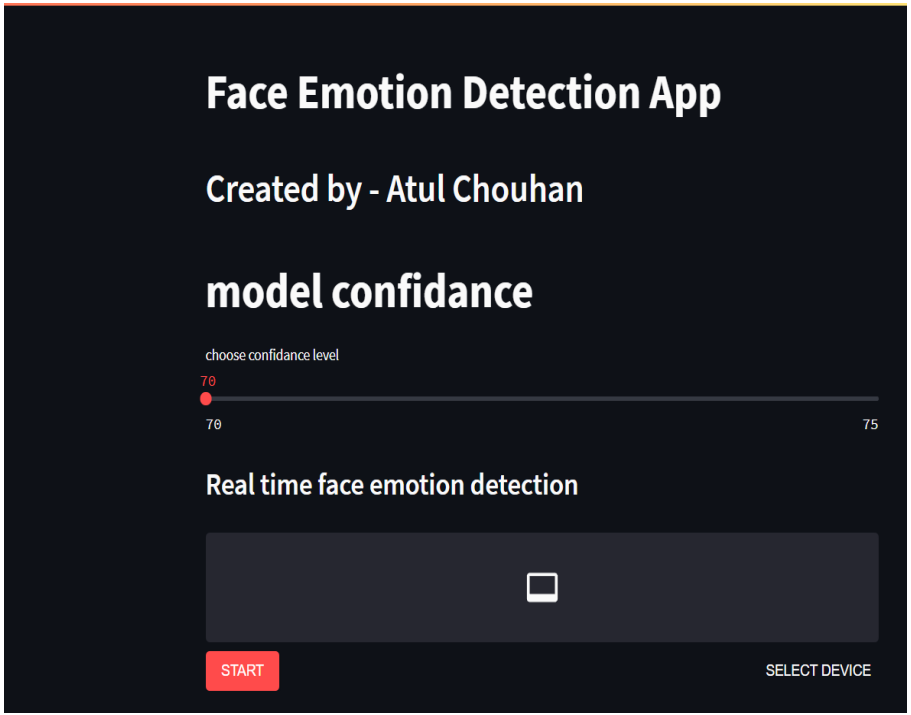
Demo Predictions(Contd.)





Deployment

- Built a streamlit python script.
- Made a docker image.
- Deployed streamlit cloud.
- Provides end to end solution.

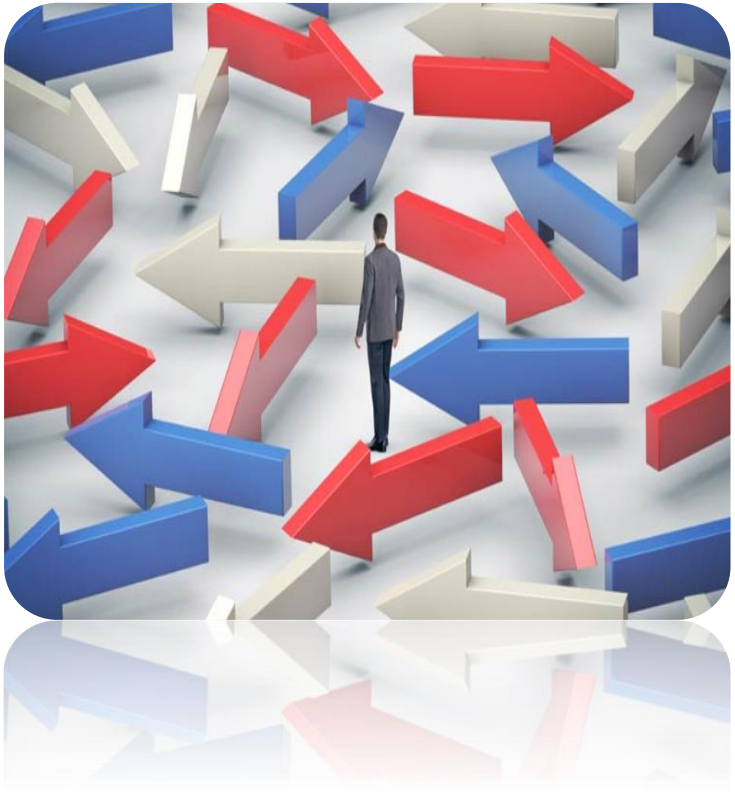


Webapp Link: https://share.streamlit.io/mmishra1092/face-emotion-recognition/main/app.py

Live Demo

Challenges

- Collecting the data.
- We got only 38k images for training due to which our accuracy be affected.
- For achieving good accuracy we need more training data.
- Training model on 38k images taking too much time. So, we took the help of kaggle notebook for training model.
- Deployment has been one of the most challenging thing in this project.
- Tried Batch Gradient Descent but failed due to low compute resources.
- Overfitting occurred on training set more often. Applied dropout.
- Too many errors while experimenting real time. Learnt from every errors and improved.
- Gathering codes for real time demo on streamlit app was haunting as local code did not run on cloud servers.
- Streamlit app is still slow due to free account and less resources



Overall Conclusion

- At the end we got an accuracy of around 77% on the train set and 67% on the test set. Considering less amount of data to train and less computation power, this can be considered a worth model.
- As always the performance of the model is not limited to these numbers. It can improve if we use more deeper networks, transfer learning, tunings etc. But at the cost of more time and more computation power.
- Since the dataset is Imbalance model has more number of happy image in the dataset the model is slightly bias.
- We build the model using Tensorflow library then created a docker image.
- Created web-app using Streamlit library and deployed our model on the streamlit cloud platform.
- The model has been trained with only 38k input images. In future ,gathering more input images could improve the performance of the model.
- Optimal Hyper parameters tuning could improve the accuracy of the model.

Thank You