Capstone Project

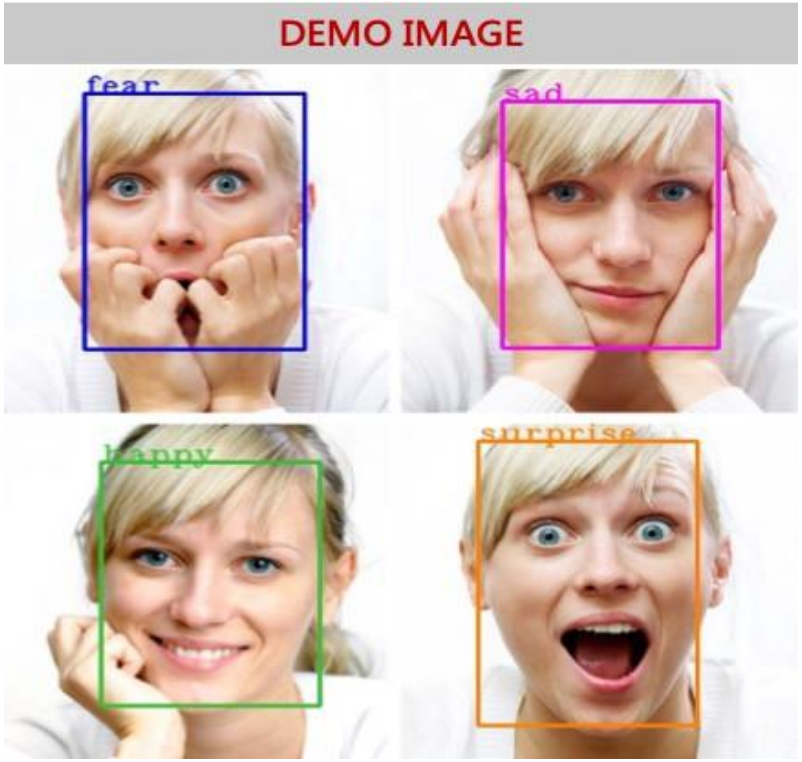
Live Class Face Emotion Recognition

- Presenting by

Bonthu Manoj Kumar Reddy

Content

- Project Introduction
- Problem Statements
- Data Summary
- Exploratory Data Analysis
- Model Formulation
- Accuracy & Performance
- Deployment
- Challenges
- Summary



Project Introduction

Live Class Monitoring System using Real time face recognition

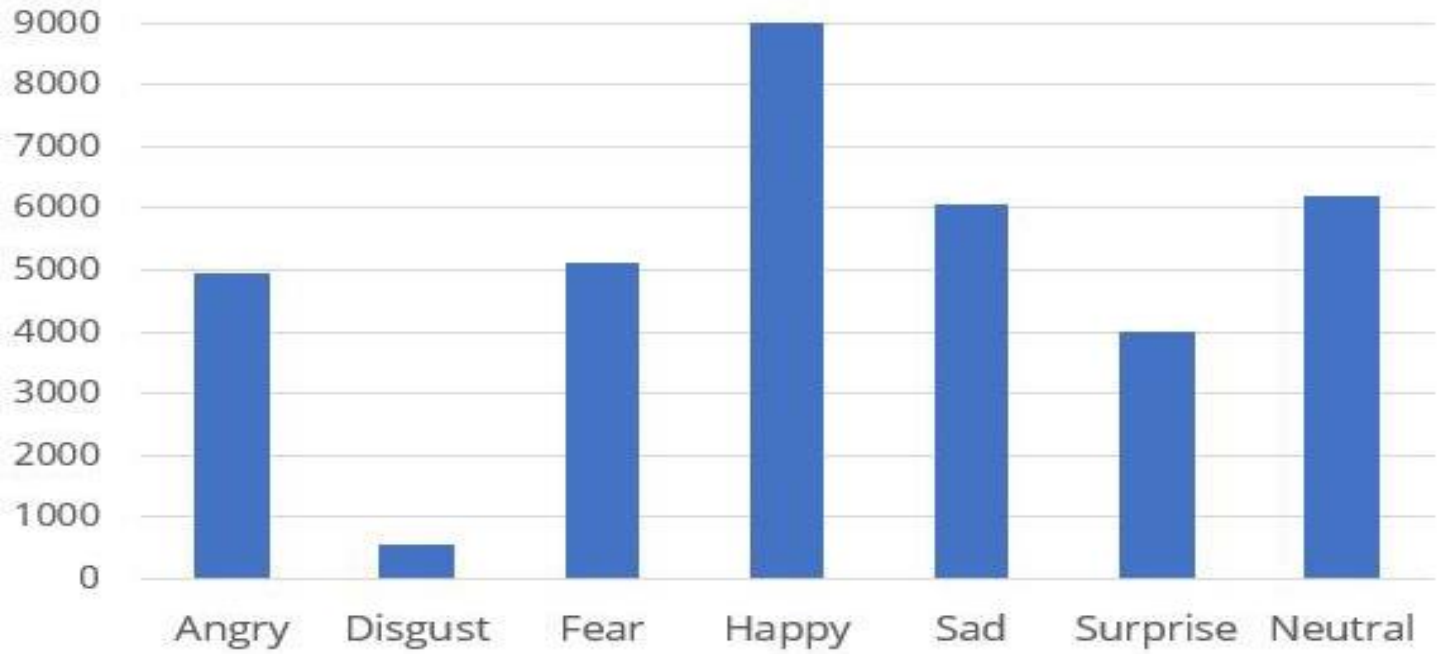
- Facial Recognition is a technology which uses biometric markers to detect emotions in human faces
- The Architecture: we use of a convolutional neural network for the image recognition task instead of regular machine learning techniques such as Decision Tree or Gradient Boosting
- Applications of facial recognition

Problem statement

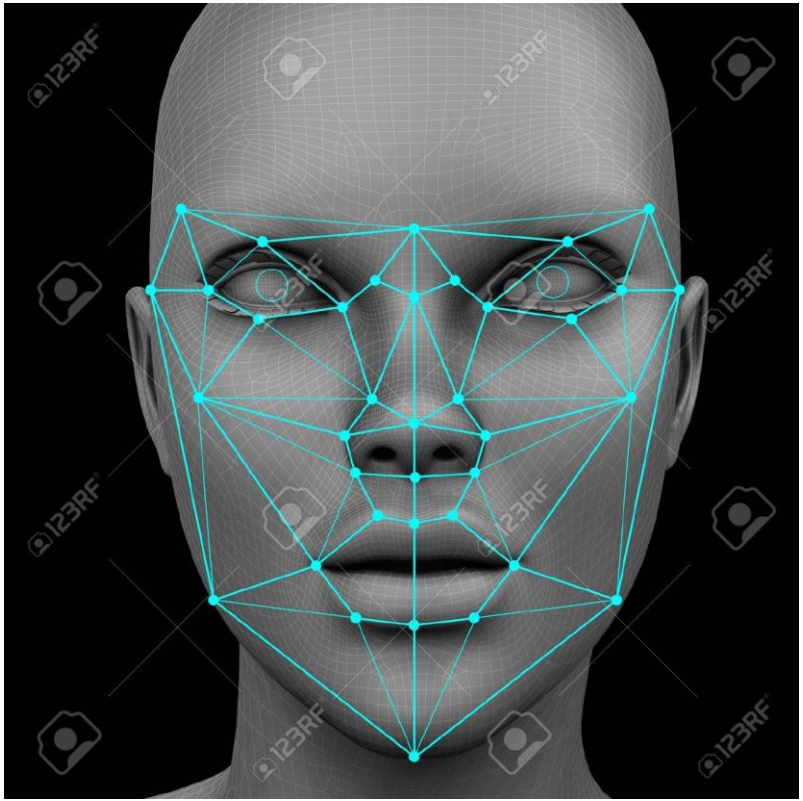
- Is data Normalized?
- Is Data Augmentation used?
- Is early stopping used?
- Is transfer learning used?
- Is model deployed? Using which web app?

Data Summary

- This dataset consists of 35887 grayscale,
- 48 x 48 sized face images with seven emotions.
- File size : 54MB
- link : https://www.kaggle.com/jonathanoheix/face-expression-recognition-dataset



Face embeddings



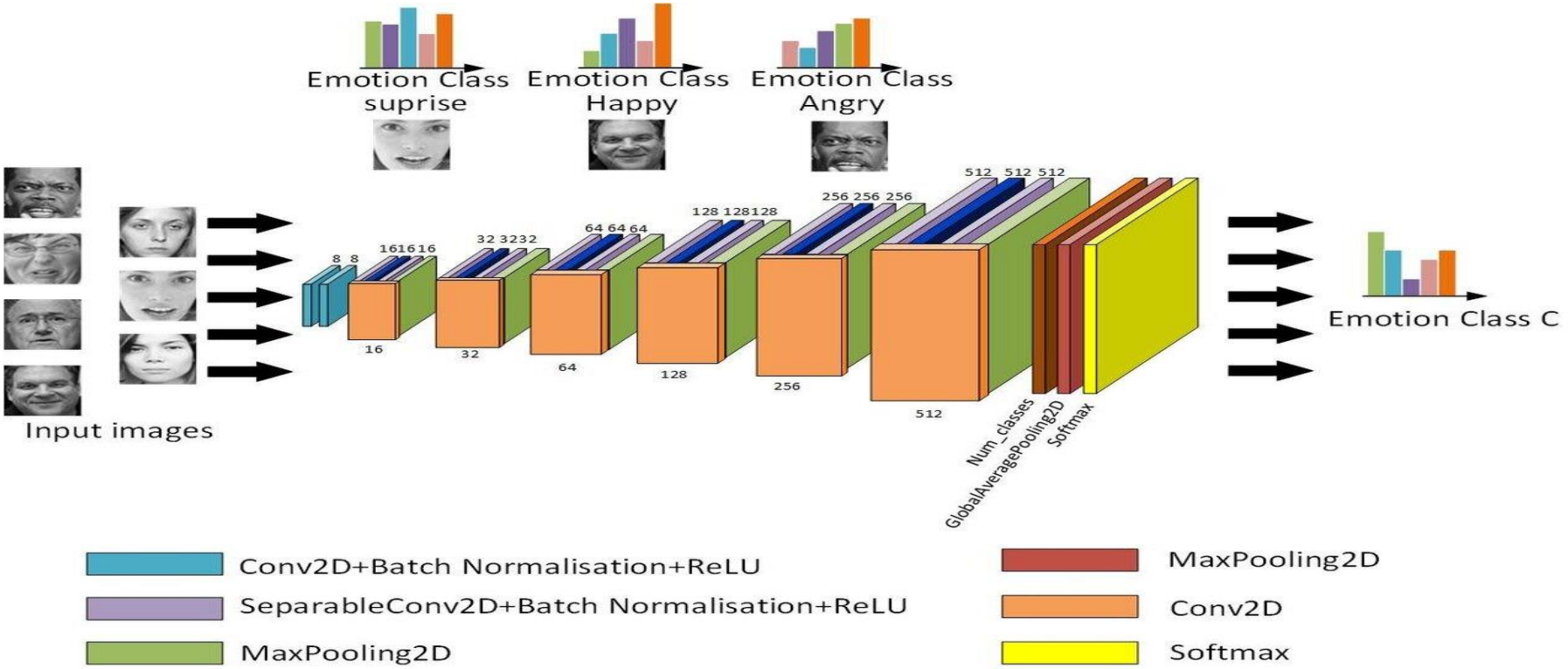
By creating face embeddings you are converting a face image into numerical data. That data is then represented as a vector in a latent semantic space. The closer the embeddings are to each other in the latent space, the more likely they are of the same person.

Exploratory data analysis

Plotting the Happy Faces in Training

Model Building

Convolution Neural Network



Hyper tuning parameters

Batch Size : It refers to the number of training examples utilized in one iteration. higher batch sizes leads to lower asymptotic test accuracy

#Batch size = 128

Learning rate in Adam : the learning rate is a tuning parameter in an optimization algorithm that determines the step size at each iteration while moving toward a minimum of a loss function.

Learning rate=0.0001

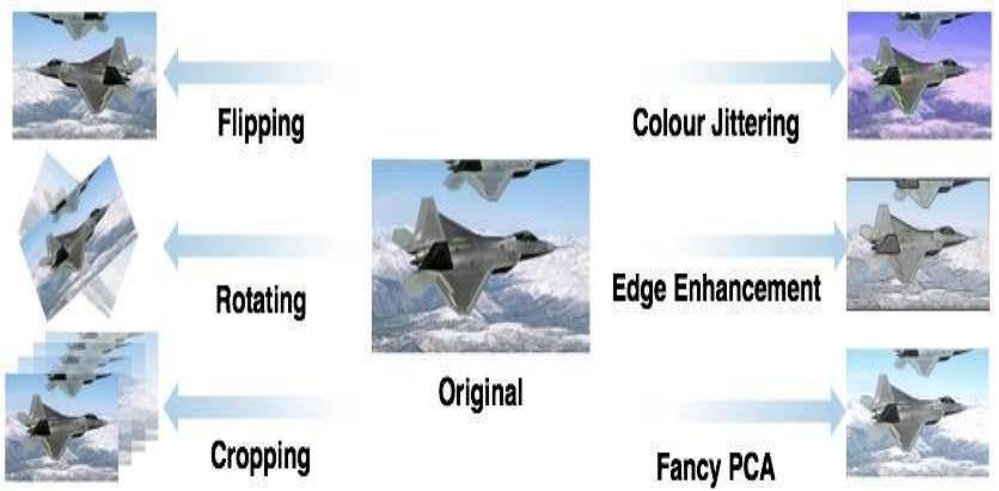
Epochs : In terms of artificial neural networks, an epoch refers to one cycle through the full training dataset

#Epochs = 150

By doing these tuning and with the help of GPU our validation accuracy improved by 4%

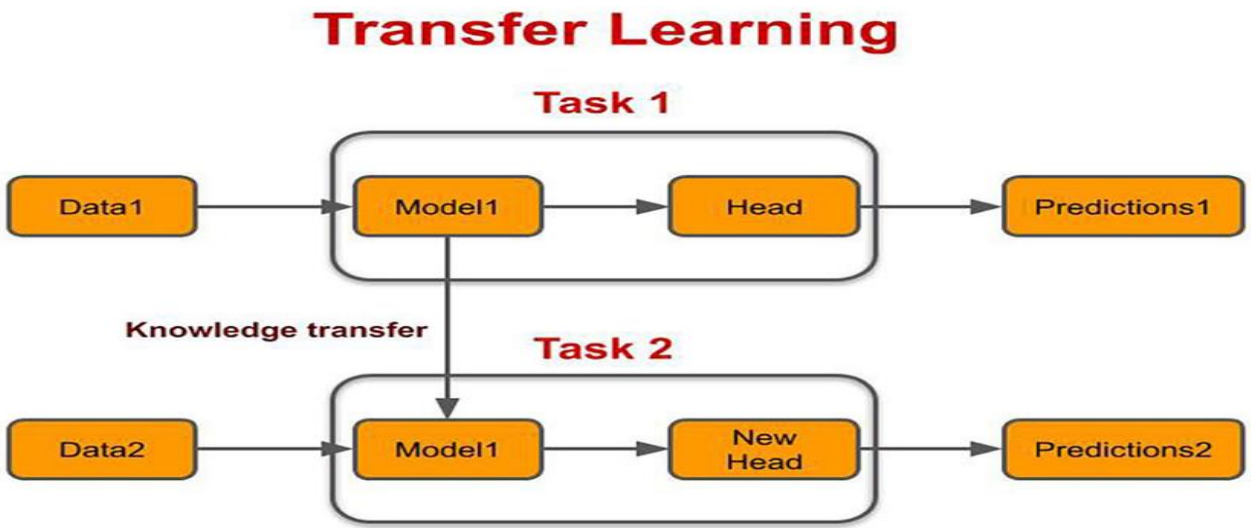
Data Augmentation

Data augmentation is a technique to artificially create new training data from existing training data. This is done by applying domain-specific techniques to examples from the training data that create new and different training examples



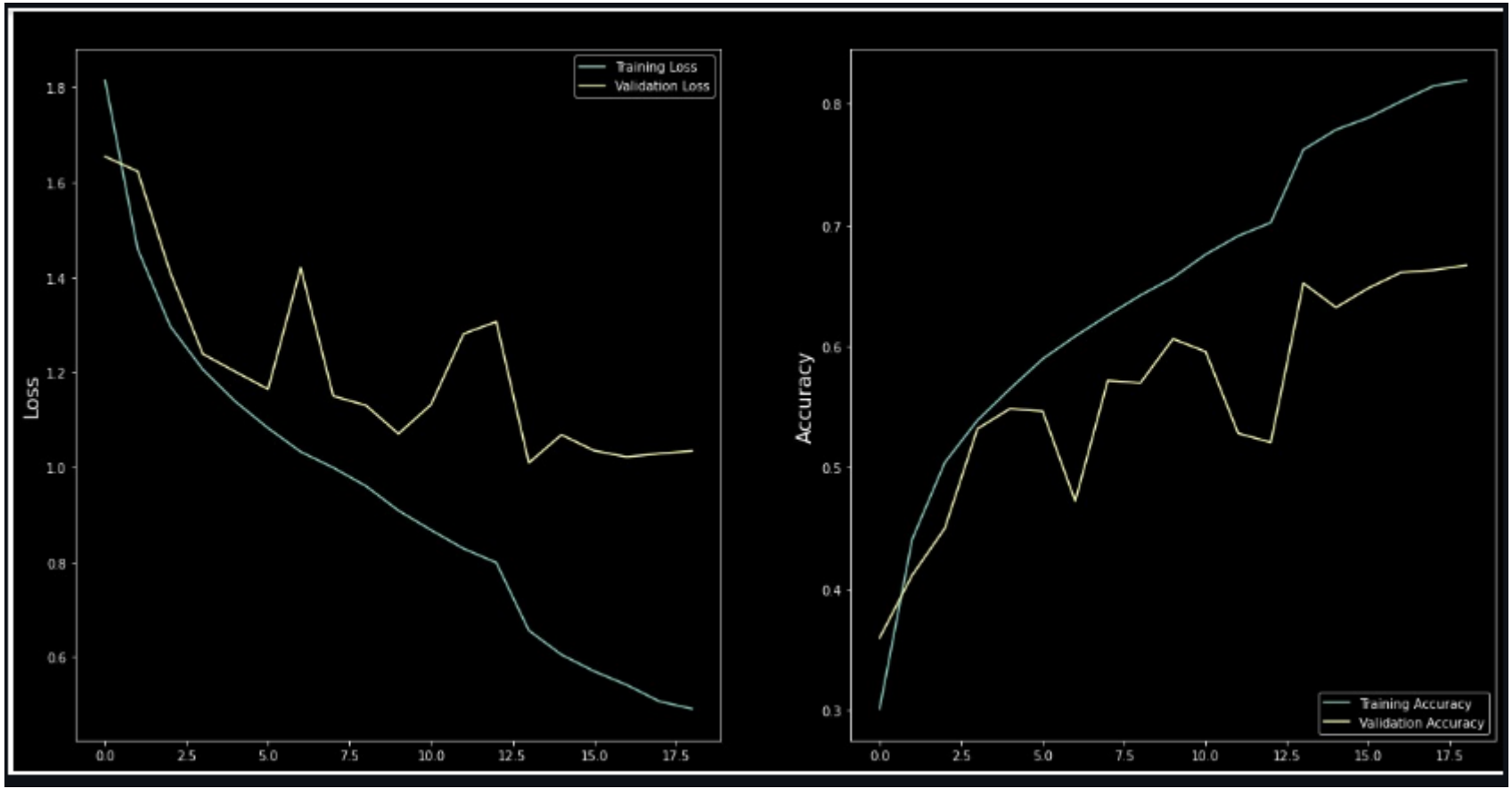

```
datagen_train= ImageDataGenerator(rescale = 1.0/255.0,
                                  width_shift_range = 0.1,
                                   height_shift_range = 0.1
                                   rotation_range = 20,
                                   horizontal_flip = True)
```

Model Building



- Transfer learning is a technique whereby a neural network model is first trained on a problem similar to the problem that is being solved.
- Example take a model trained on a large dataset and transfer its knowledge to a smaller dataset
- We used MobileNet as transfer learning technique which is the state of the art model and It gave us accuracy about 78%.

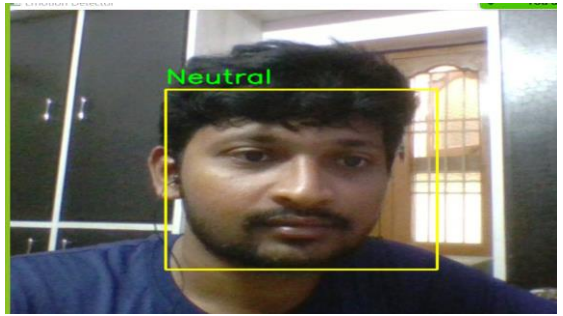
Loss and Accuracy plot

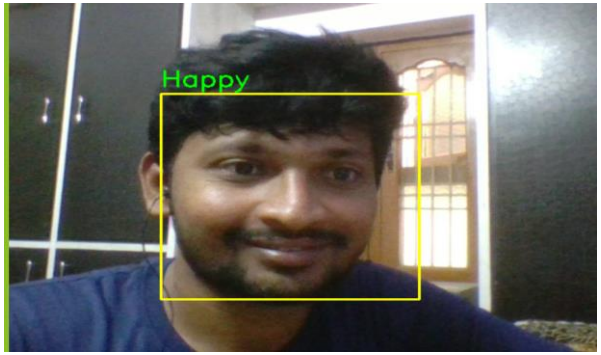


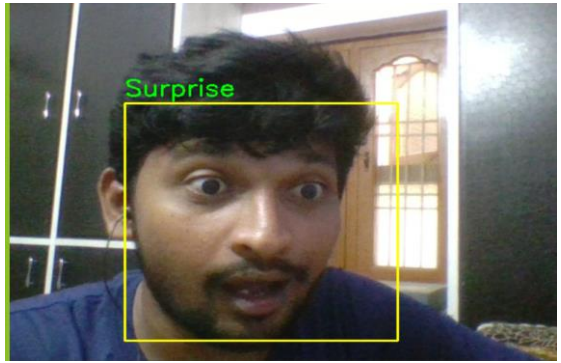
Accuracy

```
Epoch 00142: val_accuracy improved from 0.68480 to 0.69247, saving model to ./modelbestweights.h5
Epoch 143/150
225/225 [==============================] - 45s 198ms/step - loss: 0.5605 - accuracy: 0.7880 - val_loss: 1.0344 - val_accurac
y: 0.6730
```

Test results of Open-CV in local machine







Deployment

- I have created two patterns for detecting and predicting single faces and as well as multiple faces using OpenCV video capture in local machine

- I have deployed model in streamlit web app :
  https://share.streamlit.io/bonthumanojkumarreddy/live-face-emotion-recognition/main

Challenges Faced

- Choosing the dataset was difficult without Kaggle it saved most of the time.
- To Run CNN models we wasted to Days to improve accuracy without GPU later we came to How much time we can save with extra GPU's we accessed it though Kaggle Notebooks.
- Tuning for the best weights was also a challenge initially default values got around 65% accuracy later we improved by increase to epochs to 150, batch size = 128, using Adaptive learning rate optimization algorithm =0.0001.
- Continuous Runtime and RAM Crash due to large dataset.
- GCP, AWS and Azure could platforms were not able to deploy due to credit card details requirement, instead we did in Heroku.

Summary

- I used CNN sequential model with 4 Convolution Layers with relu as activation function and 2 Fully connected layers passed through softmax function
- I initially got a validation accuracy around 64% with batch size 64, and epochs 50.
- By Hyper tunning the model to 128 batch size, Epochs =150 with Adaptive learning rate = 0.0001 reach validation accuracy of 69% at 138 Epoch
- I run the best model weights h5 file to OpenCV it worked with all 7 emotions.
- Transfer learning has best accuracy over 78%
- I have deployed this model in streamlit webapp which worked fine


Q&A


thank
you