

Capstone Project - 5

Face Emotion Recognition

Made By:

Puroshotam Kumar Singh

Contents

- **Project Introduction**
- **Problem Statement**
- **Methodology**
- **Dataset**
- **Problems in dataset**
- **CNN architecture**
- **Training the model**
- **Evaluation**
- **Predicted Examples**
- **Streamlit App**
- **Live Demo**
- **Challenges**
- **Conclusion**

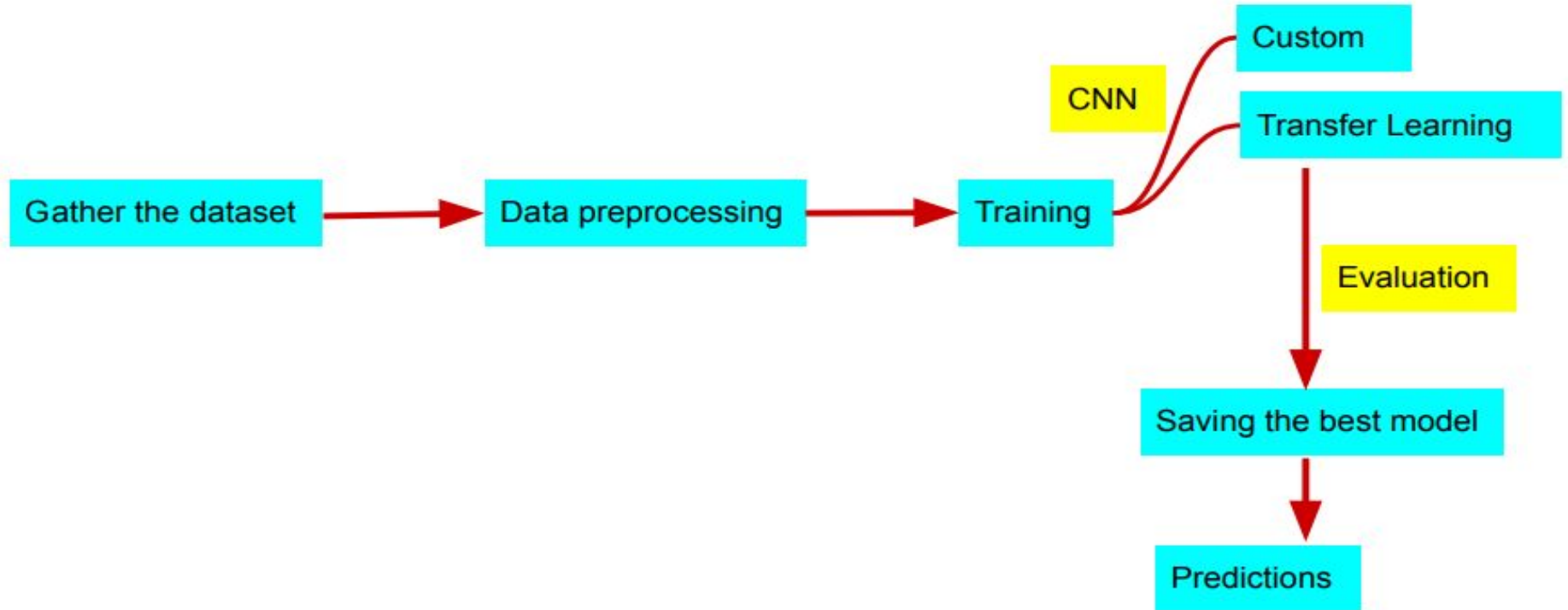
Project Introduction

- 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.
- Digital platforms to conduct live classes are proving their worth in terms of quality content and resources.
- We now have a load of data in forms of video, audio, texts.
- Using this data we can build numerous deep learning projects which will help in improving digital learning.

Problem Statement

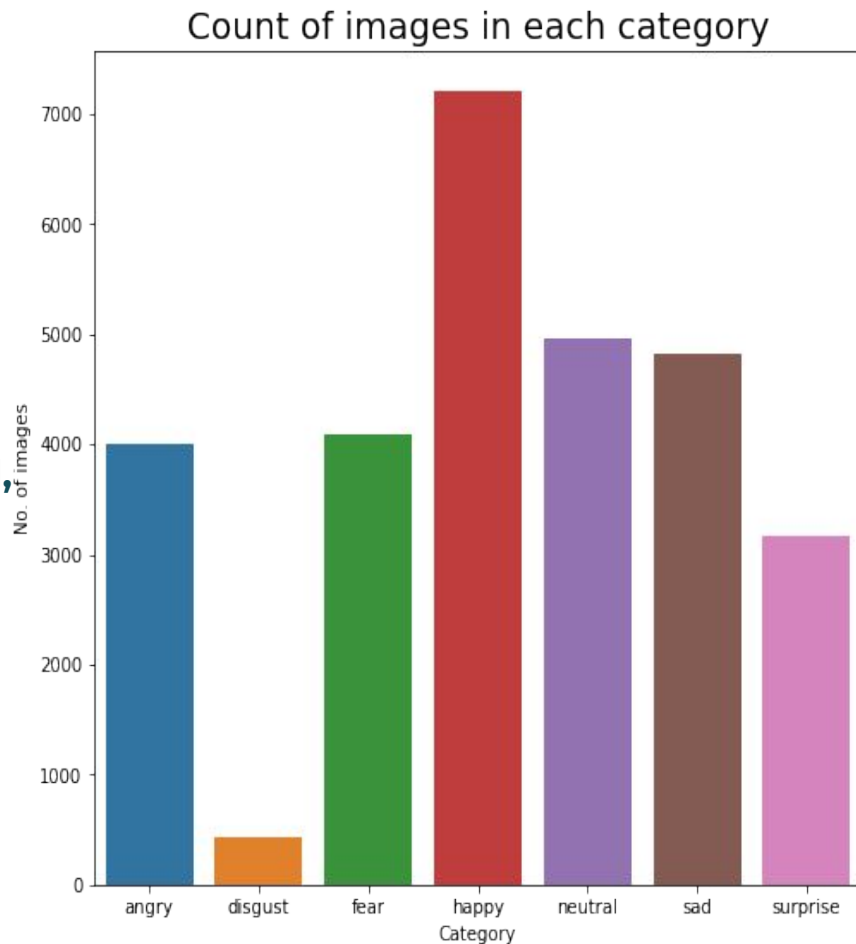
- In E-learning the quality of learning is compromised as teacher is not able to understanding whether students are able to grasp the content in a live class as teacher cant see the faces and assess the emotion of the class .
- Also scale of class is large in online classes.
- Because of this drawback, students are not focusing on content due to lack of surveillance
- We have build a deep learning model which detect the real time emotions of students through webcam ,which will understand if students are able to grasp the topic according to their expressions or emotions , and then deploy the model.

Methodology



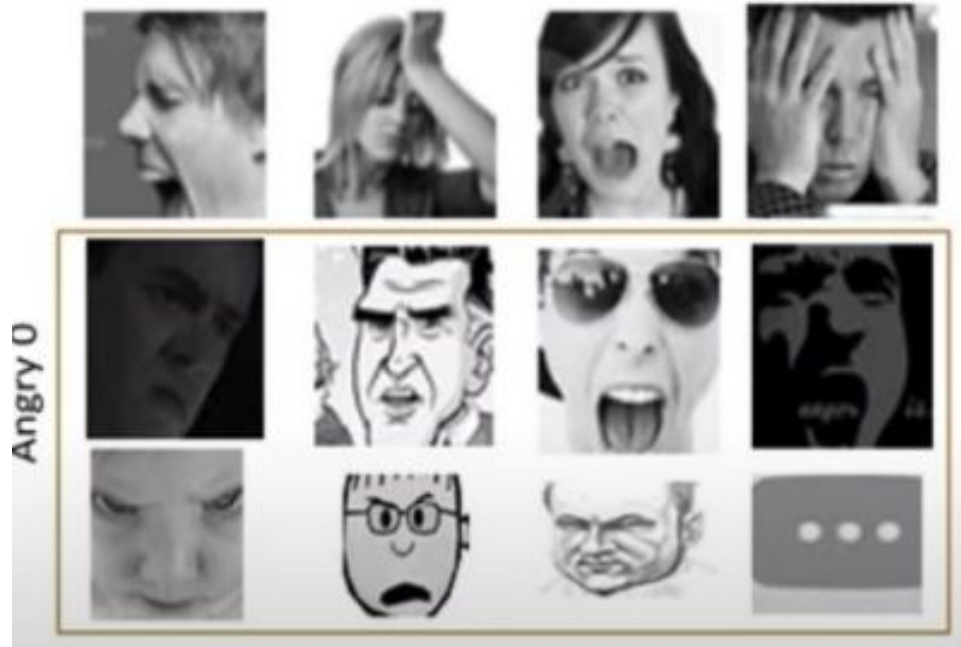
Dataset

- The model has been trained on “FER - 2013” dataset which was published on international conference on machine Learning(ICML).
- Consists of 35887 grayscale, 48x48 sized face images with seven emotions.
- Emotions are angry, disgust, fear, happy, neutral, sad, surprise.
- Disgust has very less examples, so model might not perform good on disgust images.



Problems in Dataset

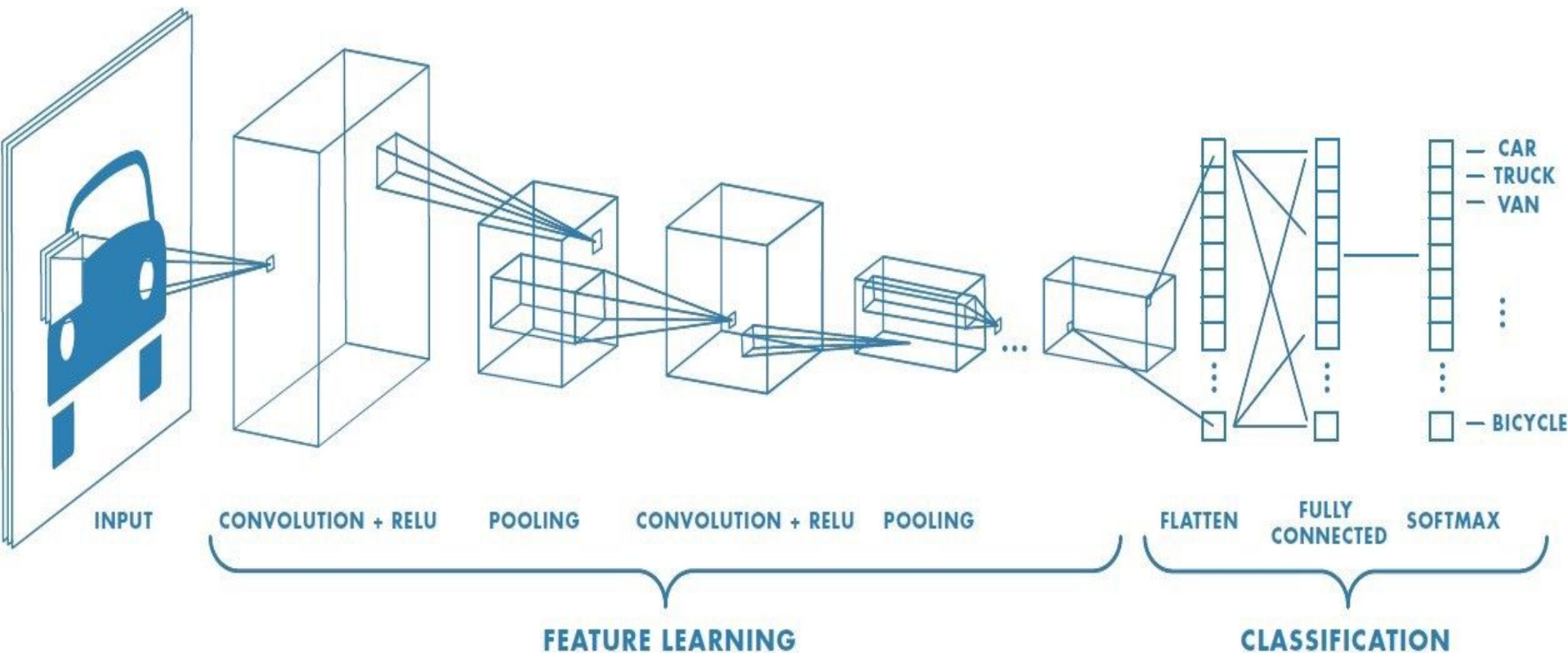
- Imbalance classes
- Intra class variation
- Occlusion
- Contrast
- Sunglasses
- Outliers



CNN Architecture

- 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 simple features to complex 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.

CNN Architecture



Training the model

- **Training set accuracy : 63%**
- **Validation set accuracy : 60% at 19 epochs**

Epoch 00013: val_accuracy did not improve from 0.58261

Epoch 14/19

225/225 [=====] - 1441s 6s/step - loss: 1.0425 - accuracy: 0.6030 - val_loss: 1.1115 - val_accuracy: 0.5790

Epoch 00014: val_accuracy did not improve from 0.58261

Epoch 15/19

225/225 [=====] - 1438s 6s/step - loss: 1.0204 - accuracy: 0.6127 - val_loss: 1.1132 - val_accuracy: 0.5818

Epoch 00015: val_accuracy did not improve from 0.58261

Epoch 16/19

225/225 [=====] - 1437s 6s/step - loss: 1.0092 - accuracy: 0.6183 - val_loss: 1.2976 - val_accuracy: 0.5244

Epoch 00016: val_accuracy did not improve from 0.58261

Epoch 17/19

225/225 [=====] - 1429s 6s/step - loss: 0.9957 - accuracy: 0.6227 - val_loss: 1.0559 - val_accuracy: 0.6053

Epoch 00017: val_accuracy improved from 0.58261 to 0.60532, saving model to model.h5

Epoch 18/19

225/225 [=====] - 1425s 6s/step - loss: 0.9814 - accuracy: 0.6306 - val_loss: 1.1072 - val_accuracy: 0.5887

Epoch 00018: val_accuracy did not improve from 0.60532

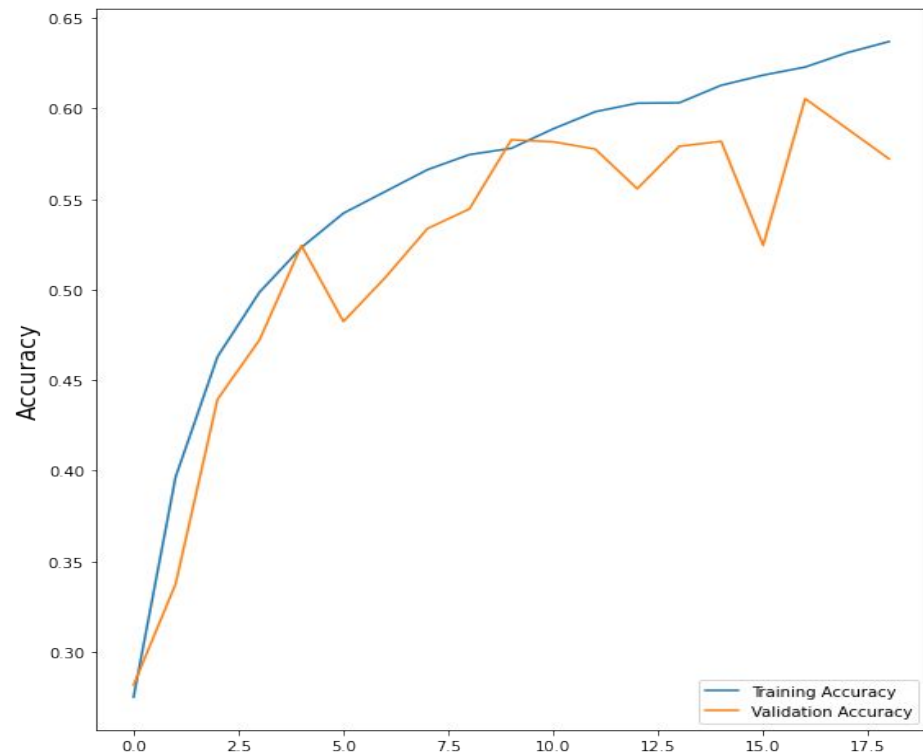
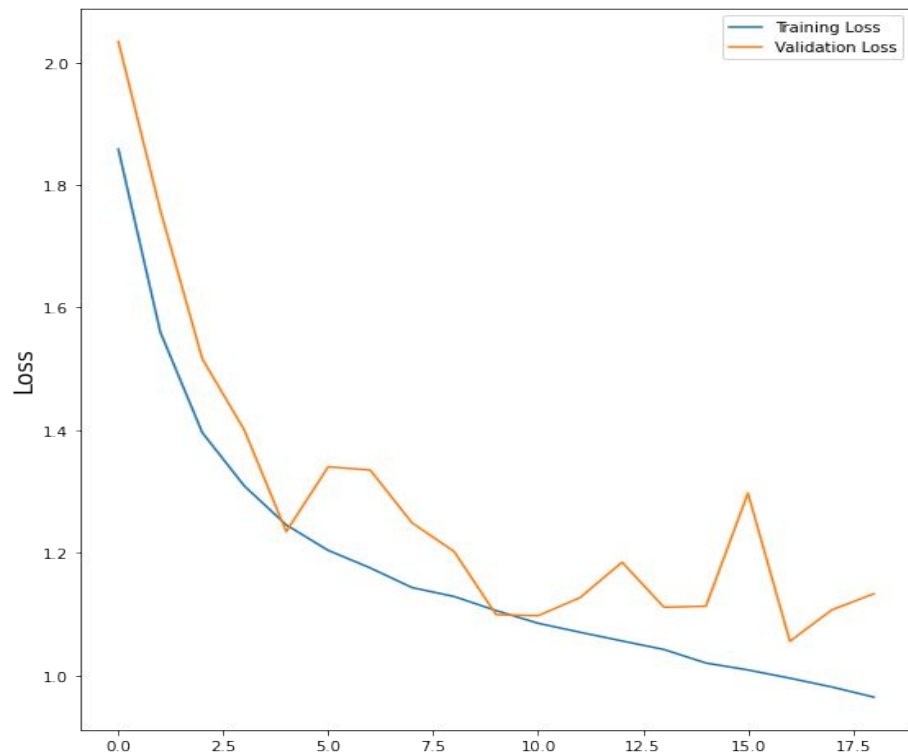
Epoch 19/19

225/225 [=====] - 1429s 6s/step - loss: 0.9649 - accuracy: 0.6368 - val_loss: 1.1334 - val_accuracy: 0.5720

Epoch 00019: val_accuracy did not improve from 0.60532

Evaluation

Optimizer : Adam



Predicted Examples



Streamlit App

Streamlit App

- **Streamlit-webrtc helps to deal with real-time video streams.**
- **Streamlit doesn't provide live capture feature itself, instead use a third party API that's why the video is taking a bit time for loading.**
- **Image captured from webcam is sent to VideoTransformer function to detect the emotion.**
- **Then this app was deployed on streamlit cloud.**
- **It provides end to end solution.**

Streamlit App(Contd.)

Face Emotion Detection App

Made by- Puroshotam Singh

Real time face emotion detection



START

SELECT DEVICE

Face Emotion Detection App

Made by- Puroshotam Singh

Please upload an image to detect the emotion.

Make sure to upload an image of a face to get positive result.

Upload image



Drag and drop file here

Limit 200MB per file • JPEG, PNG, JPG

Browse files

No image uploaded yet

Live Demo

Challenges

- Tried Batch Gradient Descent but failed due to low compute resources.
- Overfitting occurred on training set more often. So, applied dropout.
- Training took so much time just to train simple CNN model due to less computational powers.
- Streamlit app is slow on cloud due to free account and less resources.

Conclusion

- This emotion recognition model will be helpful for teachers to teach in online mode.
- Model gave an accuracy of 63% for training set and 60% for test set.
- Accuracy can be further improved if we have more data, more compute resources, trying different combinations of CNN layers, Transfer learning etc.
- Frontend of model was made using flask and Streamlit and ran successfully locally.
- App can run smoothly on cloud if compute resources are better.

Thank You

Q & A