# Capstone Project

## Live Class Monitoring System (Face Emotion Recognition)

Akash Salmuthe

# Content

- Introduction
- Problem Statement
- Data Summary
- Pipeline
- Model Overview
- Model Evaluation
- RTFED
- Deployment
- Challenges
- Conclusion

# Introduction

- Facial Emotion Recognition is a way of identifying the current emotional state of an individual to observer.

- Facial expressions can display personal emotions and indicate an individual's intentions within a social situation.

- Facial expressions and other gestures convey nonverbal communication cues that play an important role in interpersonal relations.

- **Product Development:** Observing users interaction while interacting with a brand or a product helps the company to assess the effectiveness of any business product.

# Problem Statement

- Indian education system is moving towards e-learning platforms.
- Digital learning is going to increase in future , but there are some challenges
- In physical class teacher can access the faces and emotions of each student but in digital class its not possible.
- Lack of surveillance, Lack of attention
- We will solve the above-mentioned challenge by applying deep learning algorithms to live video data. The solution to this problem is by recognizing facial emotions.

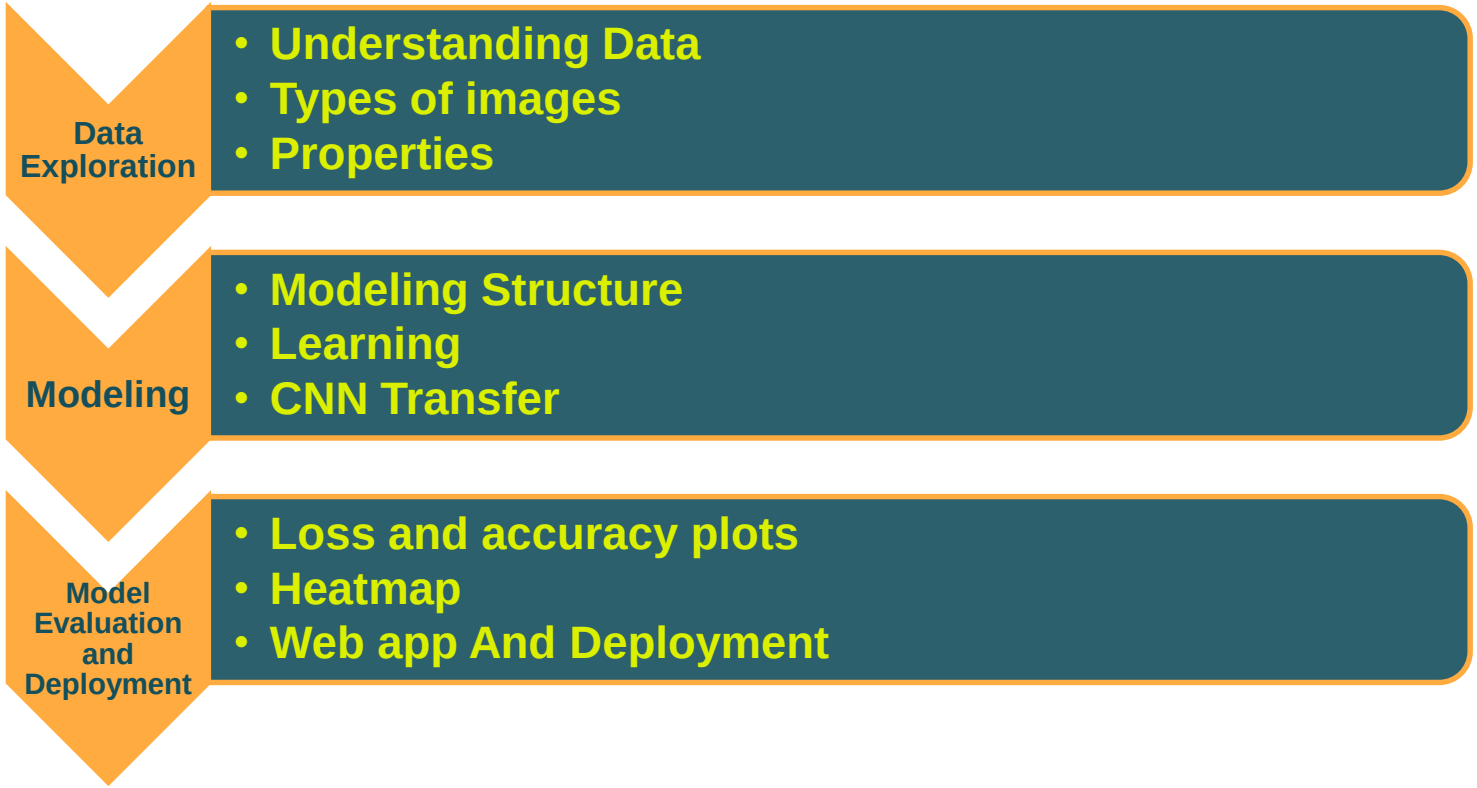
# Data Summary

- The model is trained on the FER-2013 dataset .This dataset consists of 35887 grayscale, 48x48 sized face images with 7 emotions -angry, disgusted, fear, happy, neutral, sad and surprised.

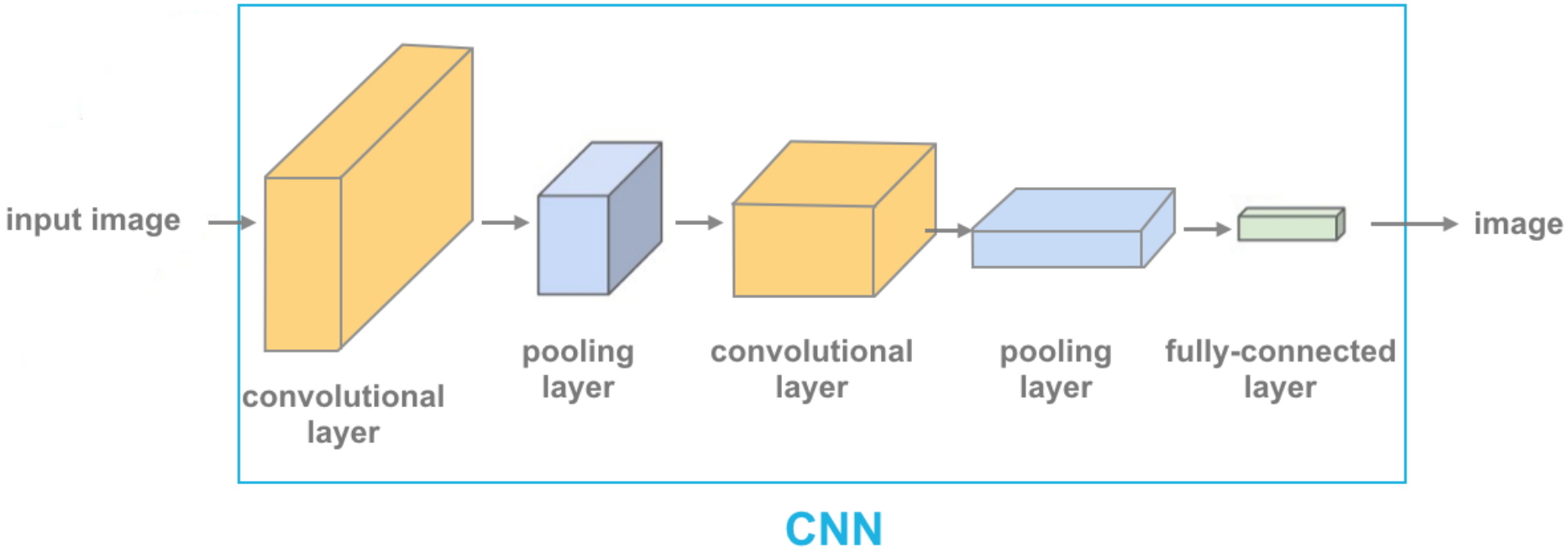
- Link of Dataset -**https://www.kaggle.com/msambare/fer2013**

| Label | Emotion | No. of images for Training | No. of images for Testing |
|---|---|---|---|
| 0 | Angry | 3995 | 958 |
| 1 | Disgust | 436 | 111 |
| 2 | Fear | 4097 | 1024 |
| 3 | Happy | 7215 | 1774 |
| 4 | Sad | 4830 | 1247 |
| 5 | Surprised | 3171 | 831 |
| 6 | Neutral | 4965 | 1233 |

# Pipeline


Data Exploration
• Understanding Data
• Types of images
• Properties
Modeling
• Modeling Structure
• Learning
• CNN Transfer
Model Evaluation and Deployment
• Loss and accuracy plots
• Heatmap
• Web app And Deployment

# Generic CNN Model



# Generic CNN Model

- **Input layer**-Input layer in CNN should contain image data
- **Convolution layer**-Convolution layer is sometimes called feature extractor layer because features of the image are get extracted within this layer.
- **Pooling layer** -Pooling is used to reduce the dimensionality of each features while retaining the most important information. It is used between two convolution layer.
- **FullyCL:**Fullyconnectedlayerinvolvesweights,biases,andneurons.Itconnectsneuronsinone layer to neurons in another layer. It is used to classify images between different category by training and placed before the output layer.

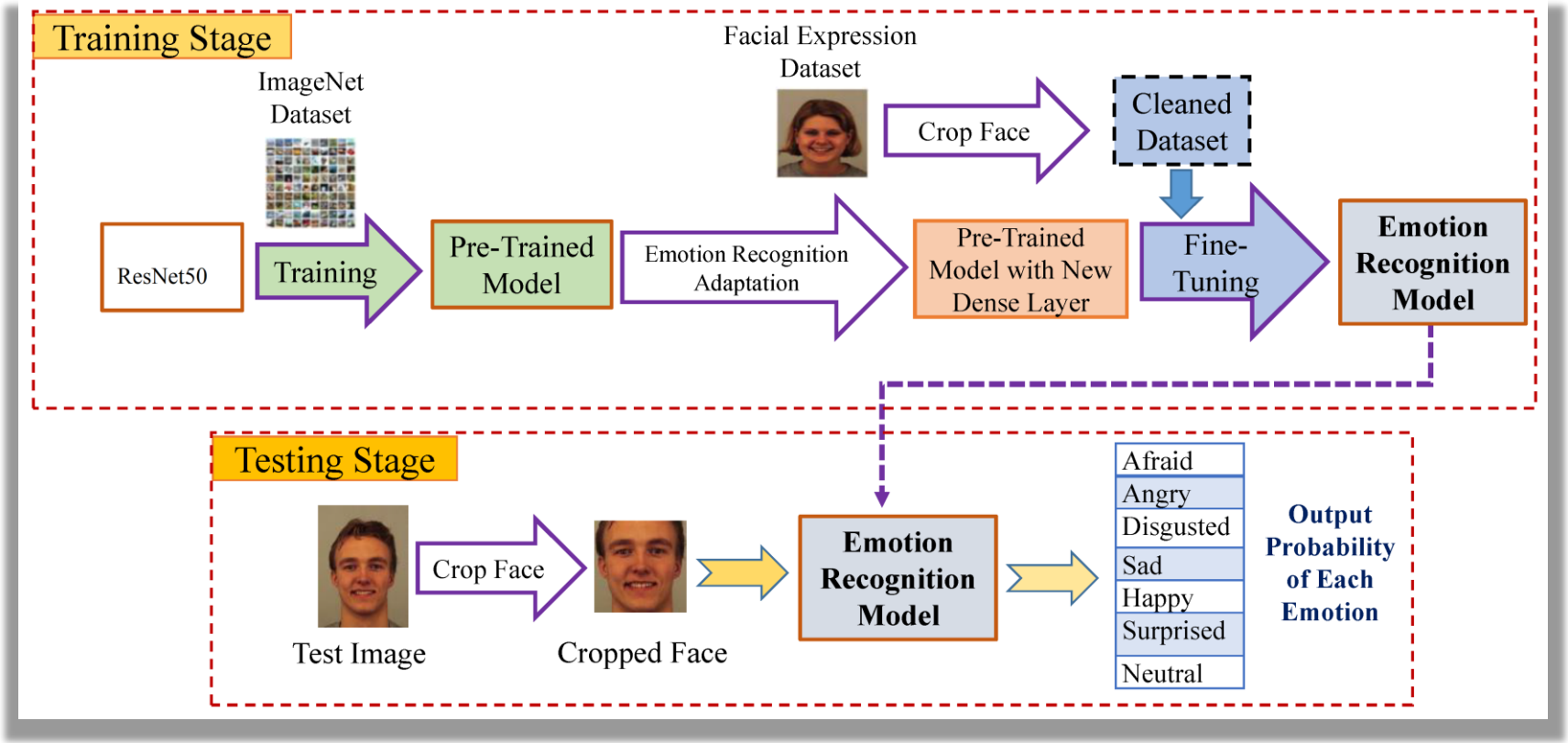
# Generic CNN Model

- **Output Layer** -Output layer contains the label which is in the form of one-hot encoded.

- **Batch normalization:** It improves the performance and stability of NNs by providing inputs with zero mean and unit variance.

- **Dropout:** It reduces overfitting by randomly not updating the weights of some nodes. This helps prevent the NN from relying on one node in the layer too much.

# Overview of Model

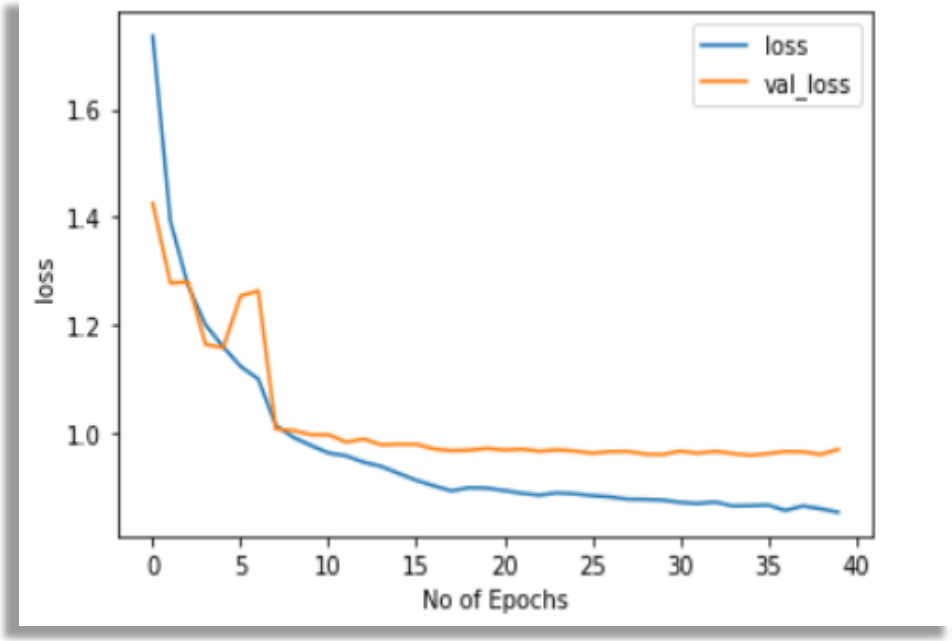


# Overview of Model

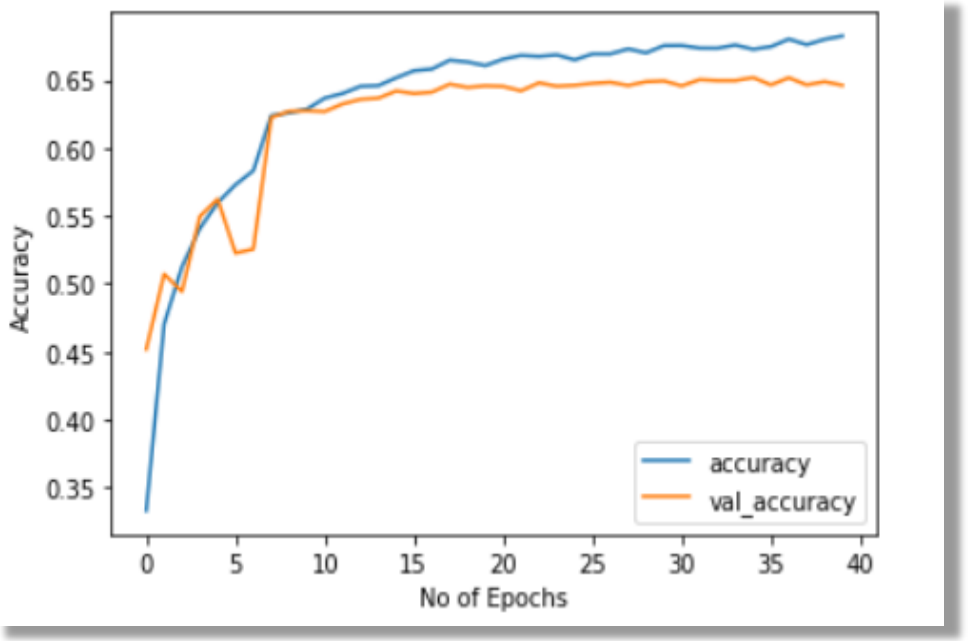
- Created a custom CNN model using Conv2D, MaxPooling, BatchNormalization, Dropout and Dense layers.
- Activation function used -**"ReLU**“ and **“Softmax”**
- Output layer has **7 nodes**
- Optimizer used – **Adam**
- Epochs -**50**
- Total params: 4,496,903
- The training accuracy obtained from this model is **68%** and validation accuracy is **64%** after **50 epochs**. Model is performing good in live video feed. Disgust images are not getting predicted correctly as there were very less images present for that in train dataset.

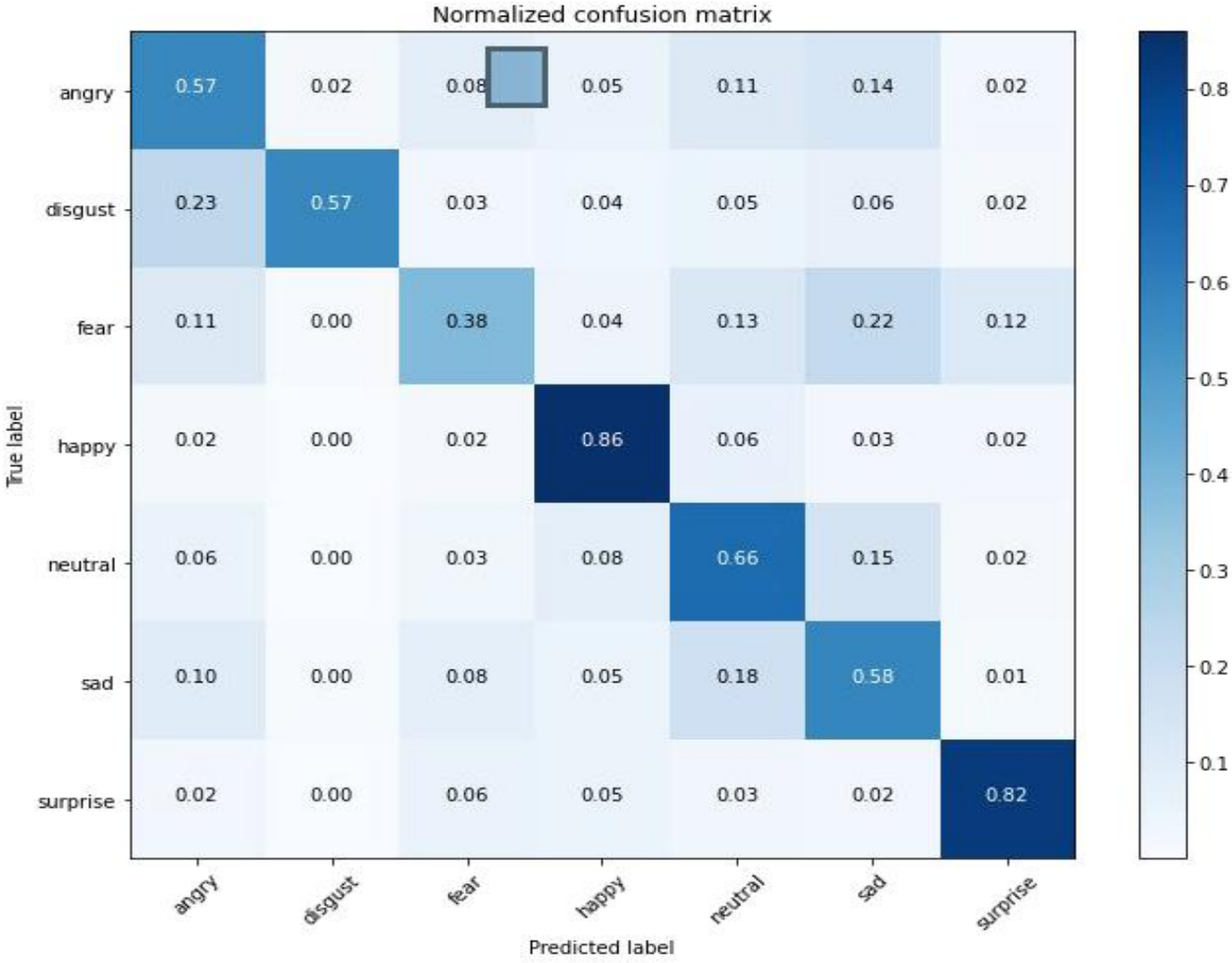
# Model Evaluation

## Loss




## Accuracy




Normalized confusion matrix
angry
disgust
fear
happy
neutral
sad
surprise
True label
Predicted label
0.57 0.02 0.08 0.05 0.11 0.14 0.02
0.23 0.57 0.03 0.04 0.05 0.06 0.02
0.11 0.00 0.38 0.04 0.13 0.22 0.12
0.02 0.00 0.02 0.86 0.06 0.03 0.02
0.06 0.00 0.03 0.08 0.66 0.15 0.02
0.10 0.00 0.08 0.05 0.18 0.58 0.01
0.02 0.00 0.06 0.05 0.03 0.02 0.82
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

# Real Time Face Emotion Detection

# Real Time Face Emotion Detection

# Deployment

## Creating Web App Using Streamlit

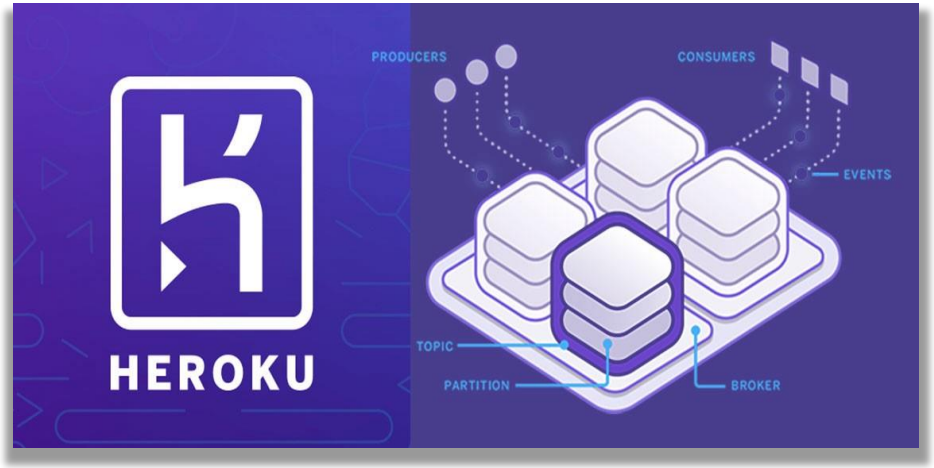
- Streamlit is an open source app framework in Python language.
- It helps us create web apps for data science and machine learning in a short time. It is compatible with major Python libraries such as scikit-learn, Keras, PyTorch, SymPy (latex), NumPy, pandas, Matplotlib etc.


https://share.streamlit.io/akashsalmuthe/face-emotion-recognition/main/app.py

# Deployment

**Heroku**

- Heroku is a platform as a service (PaaS) that enables developers to build, run, and operate applications entirely in the cloud.
- One of the first cloud platforms, since June 2007, when it supported only the Ruby programming language, but now supports Java, Node.js, Scala, Clojure, Python, PHP, and Go.




https://face-akash.herokuapp.com/

# Challenges

- Large image dataset with mislabeled data
- Model training take lots of time and system resources
- To access webcam in streamlit app
- Deployment on Heroku – issues with slug size


- Build a **FER** webapp using streamlit and deployed on Heroku, With live webcam detection.
- The model created with CNN layers gave training accuracy of **68%** and validation accuracy of **64%** after 50 epochs.
- Difficult to classify disgust images
- Model also work for multiple face detection.


THANK YOU