

## **Capstone Project Submission**

### **Instructions:**

- i) Please fill in all the required information.
- ii) Avoid grammatical errors.

### **Team Member's Name, Email and Contribution:**

#### **Name: Tanjul Gohar**

- Introduction
- Problem statement
- Data Summary
- Data Handling
- Build Xception model and CNN
- Accuracy and loss plots
- Confusion matrix
- Model analysis
- Ran the webcam on local host for face emotion detection
- Created streamlit webapp and deployment on Heroku
- Conclusion

#### **Name: Kanika Singh**

- Introduction
- Problem statement
- Data Summary
- Data Handling
- Build DeepFace model and ResNet50
- Accuracy and loss plots
- Confusion matrix
- Model analysis
- Ran the webcam on local host for face emotion detection
- Created streamlit webapp and deployment on Heroku
- Conclusion

### **Please paste the GitHub Repo link.**

#### **DRIVE LINK -**

<https://drive.google.com/drive/folders/1QeBiIuouzcVFViprHrc8fZtXy56YXAqj?usp=sharing>

**Github Link -** <https://github.com/Tanjul5/Face-Emotion-Recognition.git>

**Please write a short summary of your Capstone project and its components. Describe the problem statement, your approaches and your conclusions. (200-400 words)**

## Capstone Project 5: Live Class Monitoring System (Face Emotion Recognition)

**Problem Statement:** Digital E-learning platforms have limitation in terms of physical surveillance to learn how much their students understanding through videos to tackle this challenge we have to deep learning algorithm. To detect emotions of student while learning we have to use face emotion reorganization system.

**Dataset:** 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.

**Data Preprocessing and EDA:** In EDA we observed that neutral class have highest training images whereas disgust class have the least training as well as test images. In data preprocessing we converted pixels to the array then converted array and dimension for the transfer learning also we've used data augmentation technique to preprocess the data by horizontal mirroring,  $\pm 10$  degree rotations,  $\pm 10\%$  image zooms, and  $\pm 10\%$  horizontal/vertical shifting and randomly flip the image.

### **Model Creation:**

1) **DeepFace model-** DeepFace is a lightweight face recognition and facial attribute analysis (age, gender, emotion and race) framework for python. We imported above image which looks angry but our model gives us "27 years old white fear Man" this result. To get better results we decided to train our own model.

2) **Resnet50 model** - ResNet50 is a deep residual network with 50 layers. We used Adam as our optimizer after training for 50 epochs using Adam and a batch size of 785, we achieved 63.11% accuracy on the test set and 67% on train set there is much less over fitting.

3) **Xception model** - Xception architecture is a linear stack of depth wise separable convolution layers with residual connections. We used Adam as our optimizer after training for 50 epochs using Adam and a batch size of 785, we achieved 64 % accuracy on the test set.

4) **CNN model** - A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. We used RMSprop as our optimizer after training for 50 epochs using RMSprop with learning rate 0.001 and a batch size of 785, we achieved 65 % accuracy on the test set.

**Results: -** The below table shows results obtained after training the model.

Model	Train Score	Test Score
Deep Face	NA	NA
Resnet50	68%	62%
CNN	74%	69%
Xception	72%	64%

**Real time local video face emotion detection: -** We created patterns for detecting and predicting single faces as well as multiple faces using OpenCV video capture in local webcam. This gave pretty much accurate prediction on sample images of different emotions.

**Web APP: -** We deployed the app in Heroku if you saw in the starting section of GitHub repo you see the all the requirement files are there for creating an app on Heroku of name "**face-emotion-recognition4**". but due to high slug size the buffering takes time so we have run our app working on local and it ran properly and app is also fine also we've included video on GitHub repo.

**Heroku Link: -** <https://face-emotion-recognition4.herokuapp.com/>

**CONCLUSION: -** We explored several models including shallow CNNs, DeepFace, ResNet50, and Xception. To alleviate FER2013's inherent class imbalance, we employed class weights, data augmentation, and auxiliary datasets. By ensemble seven models we achieved 77% training accuracy and 69% accuracy for CNN model. Additionally, we demonstrated that FER models could be applied in the real world by developing a web application with real-time recognition speeds. It was interesting project and we learn lot from this project.

