A Report

On

Emotion Detection from Images Using Machine Learning

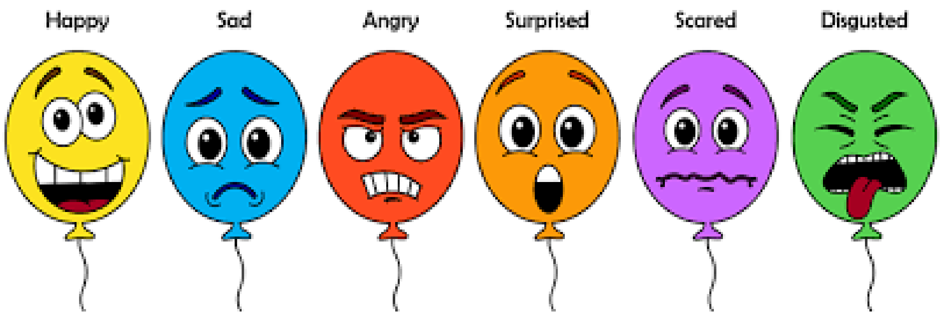
By

Saket Ingale

Project Repo: https://github.com/theingale/face_emotion_detection

CONTENTS

INTRODUCTION



In human communications, facial expressions contain critical nonverbal information that can provide additional clues and meanings to verbal communications. Some studies have suggested that 60–80% of communication is nonverbal. This nonverbal information includes facial expressions, eye contact, tones of voice, hand gestures and physical distancing. In particular, facial expression analysis has become a popular research topic.

Automatic emotion recognition based on facial expressions has presented and applied in several areas such as safety, health and in human machine interfaces. Researchers in this field are interested in developing techniques to interpret, code facial expressions and extract these features in order to have a better prediction by computer.

PROBLEM STATEMENT

Our aim is to develop a Machine Learning model that is capable of detecting the emotion from the face in the input image.

WHY MACHINE LEARNING IS USEFUL HERE?

Here, our input is Image data and the output is single label indicating the corresponding emotion. We can use the existing image data to make Machine Learning model learn the underlying relationship between the input images and output class label.

Once the model learns this relationship, we can use it to predict the corresponding emotion from the person's face in the input image.

LITERATURE SURVEY

To understand the Problem Statement thoroughly and to learn various possible approaches, following research papers and/or blogs were referred.

1. https://www.sciencedirect.com/science/article/pii/S1877050920318019
2. https://edps.europa.eu/system/files/2021-05/21-05-26_techdispatch-facial-emotion-recognition_ref_en.pdf
3. https://arxiv.org/pdf/1804.08348.pdf
4. https://www.robots.ox.ac.uk/~vgg/publications/2015/Parkhi15/parkhi15.pdf

LITERATURE REVIEW

Facial Expression Recognition is an exciting and challenging topic in the research field. It can be achieved using various approaches like Classical ML Model, Deep learning Model, CNN Networks for images, by using state of the art Image classification models as well.

Above reference literature helped us to finalize the workflow to achieve the Facial Emotion Recognition task given Images.

PROJECT WORKFLOW

Following steps are involved in this project workflow.

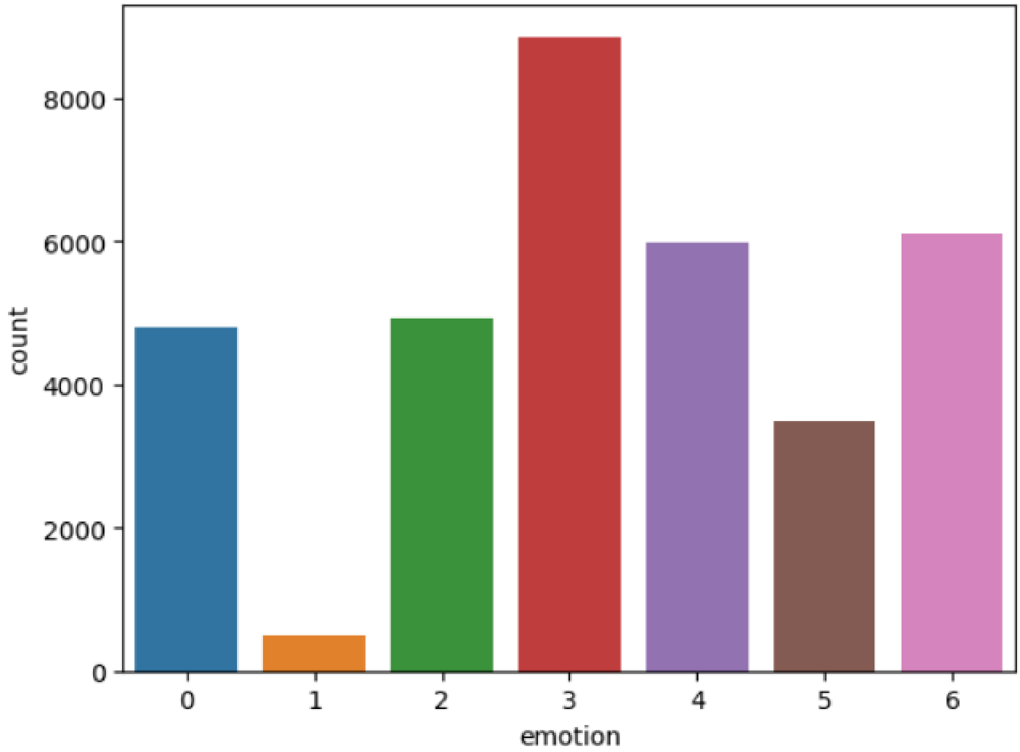
1. Exploratory Data Analysis
    1. Check Data Size
    2. Check Null values
    3. Check Duplicate Data points
    4. Check Class Imbalance
2. Data Pre-processing
    1. Extracting image pixel features for training ML Models
    2. Extracting Images features for Training DL (CNN) Models
3. Modelling Experiments
    a. Multi Class Classification using SVM
    b. Deep Multi-Layer Perceptron
    c. Custom CNN Model
    d. Transfer Learning using Pretrained CNN Model
    e. Training State of the art CNN Model from scratch.

EXPLORATORY DATA ANALYSIS

Data Source: https://www.kaggle.com/datasets/ashishpatel26/facial-expression-recognitionferchallenge/data

Data is zipped in single archive.zip file and has size 96.59 MB.

File fer2013.csv contains the image pixel values and the corresponding class labels.



fer2013.csv file contains 3 columns emotion, pixels, Usage

* emotion -> id of the emotion
* pixels -> flattened array of pixel values in the image
* Usage -> string indicating training or test image

Data contains 35887 images with corresponding class labels.

There are 1234 duplicate images which are dropped to get 34653 unique images

There are no null values in the dataset

There are 7 unique emotions

0:'anger', 1:'disgust', 2:'fear', 3:'happiness', 4: 'sadness', 5: 'surprise', 6: 'neutral'

Emotions distribution in imbalanced with only 491 unique images for 'disgust' emotion, all other emotions have at least 3400 images

PERFORMANCE METRICS

Since, this is a Multi-Class Classification Problem with Imbalanced Data, following Metrics can be used to monitor and compare performance of the ML Models.

1. Accuracy

   Accuracy score in machine learning is an evaluation metric that measures the number of correct predictions made by a model in relation to the total number of predictions made. We calculate it by dividing the number of correct predictions by the total number of predictions

   (https://scikitlearn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html#sklearn.metrics.accuracy_score)

2. Weighted_F1_score

   The F1 score can be interpreted as a harmonic mean of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is:

   F1_score = 2 * (precision * recall) / (precision + recall)

   In the multi-class and multi-label case, this is the average of the F1 score of each class with weighting depending on the average parameter. Weighted f1 Score Calculate metrics for each label, and find their average weighted by support (the number of true instances for each label)

   (https://scikitlearn.org/stable/modules/generated/sklearn.metrics.f1_score.html#sklearn.metrics.f1_score)

3. Confusion Matrix

   It helps in evaluating the accuracy of a classification in more depth. It provides the count of correctly and miss-classified points for all the class labels.

   (https://scikitlearn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html)

DATA PREPROCESSING

The given data is in a csv file format with column named 'pixels' containing flattened array of all the pixel values for each image. Machine Learning and Image based CNN networks require different kinds of preprocessing or feature extraction.

1. Extracting image pixel features for training ML Models

   'pixels' string for each image was split into each separate pixel value and was converted into float datatype

2. Extracting Images features for Training DL (CNN) Models

   'pixels' string for each image was split into each separate pixel value and was converted into float datatype. It was further converted from grayscale image format to RGB image format for more effective training of CNN based models.

MODELLING EXPERIMENTS

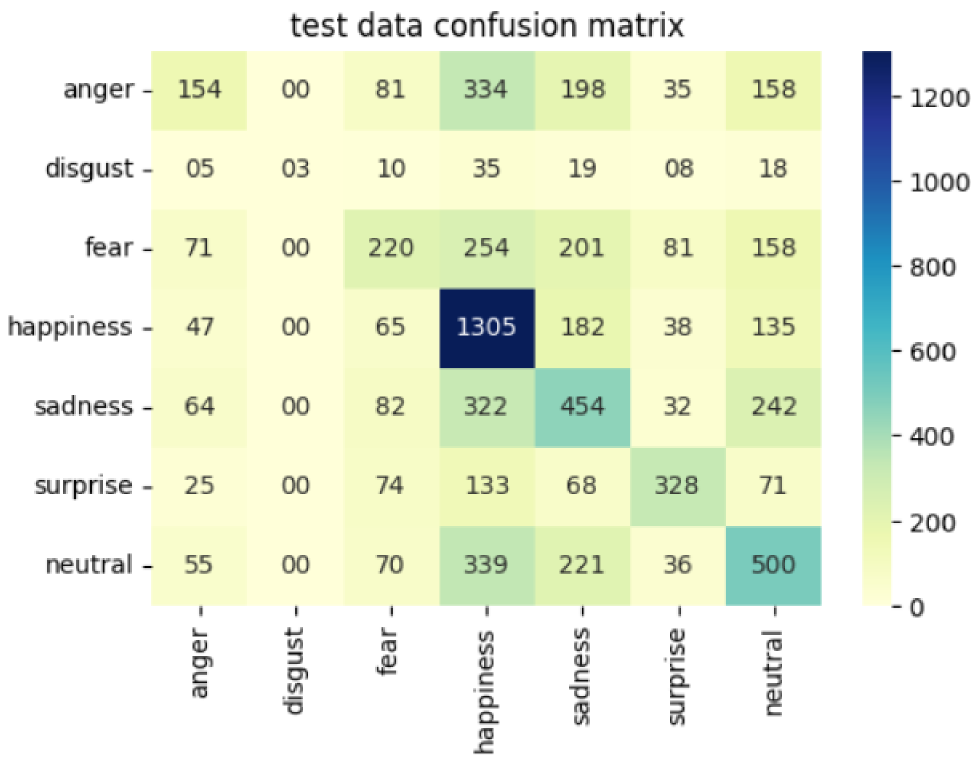
Many different modelling experiments can be carried out in order to achieve the best performance for our classification task. Here we will experiment with 5 such approaches.

a. Multi Class Classification using SVM

Pixel values can be used to train any ML algorithm that supports multiclass classification. We trained Support Vector Classifier model, since it support multiclass classification, is suitable for high dimensional data and also it can capture non-linear relationships to some extent due to kernel trick.

Model Performance on test data:

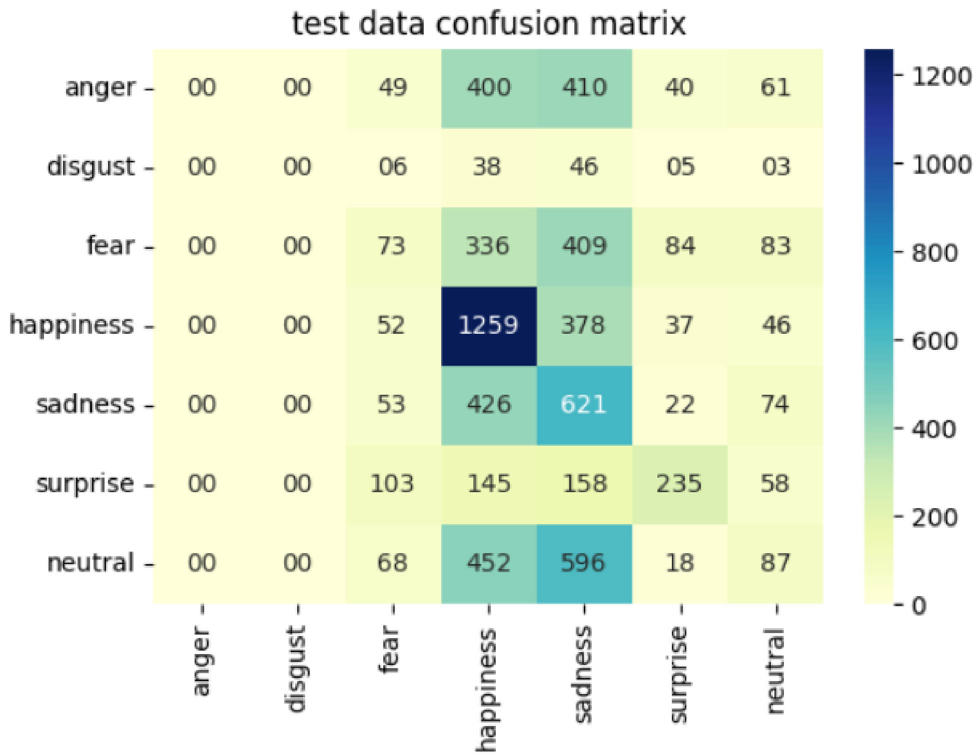
```
test data accuracy score: 0.42764391862646084
test data weighted f1 score: 0.40433925845207547
```



b. Deep Multi-Layer Perceptron

Pixel values can be directly used to train a Multi Layered Perceptron which is capable of learning non-linear relationships between input and target. We have trained an MLP to receive the following results on test data.

```
test data accuracy score: 0.32823546385802915
test data weighted f1 score: 0.264818713353675
```

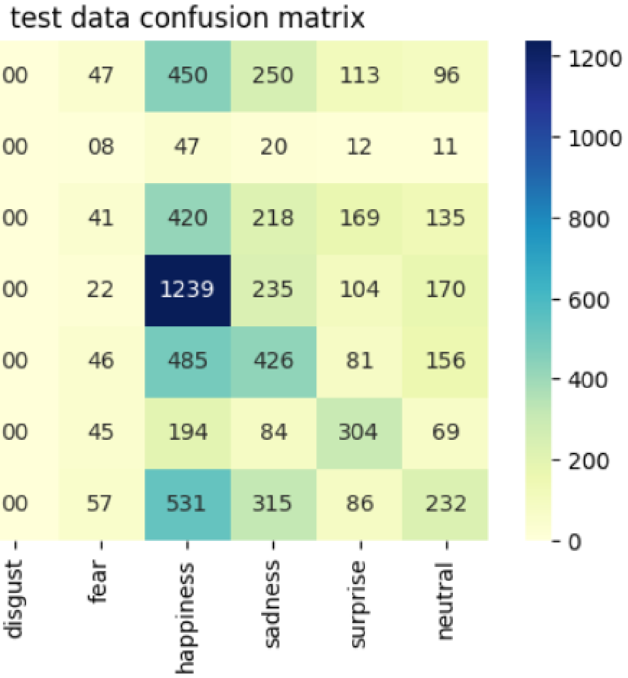


c. Custom CNN Model

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm that 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. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

We trained a Custom CNN to get following performance on test data

```
test data accuracy score: 0.3240513634396191
test data weighted f1 score: 0.2655519458422188
```



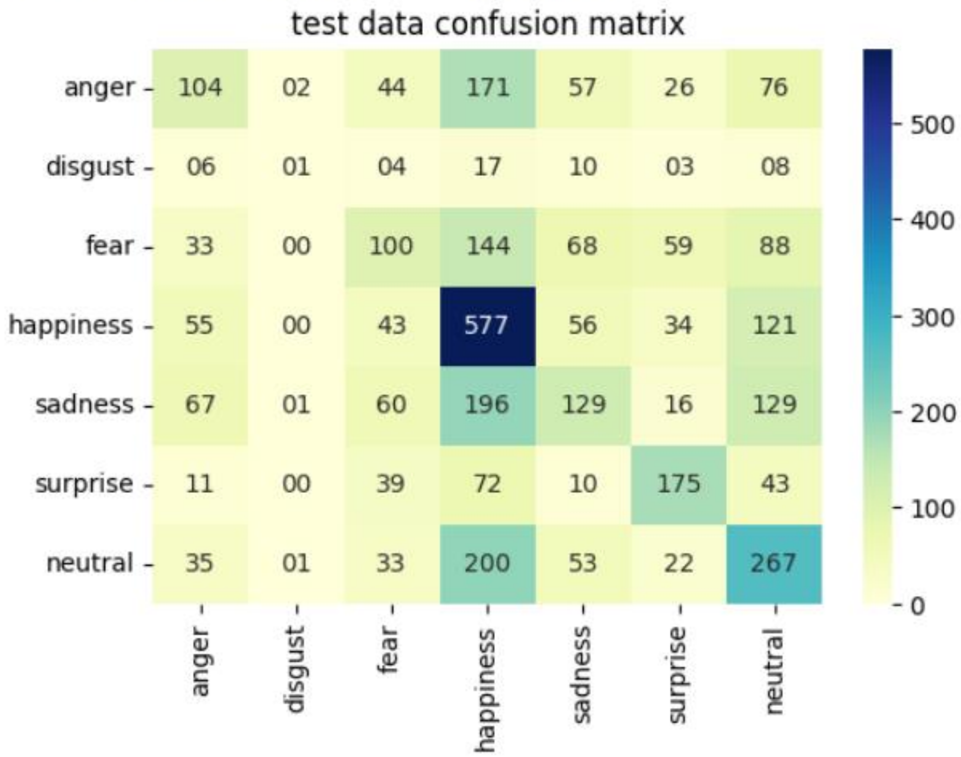
d. Transfer Learning using Pretrained CNN Model

We can use state of the art CNN Models that are trained on industry standard datasets and their learning to solve our task by doing some modifications. This allows faster training of models with better performance.

We used a pre-trained VGG19 model from tensor flow applications with ImageNet weights. We added a simple dense network on top of output of last layer of VGG19 that helped us to perform effective emotion classification.

VGG19 Transfer Learning results on test data:

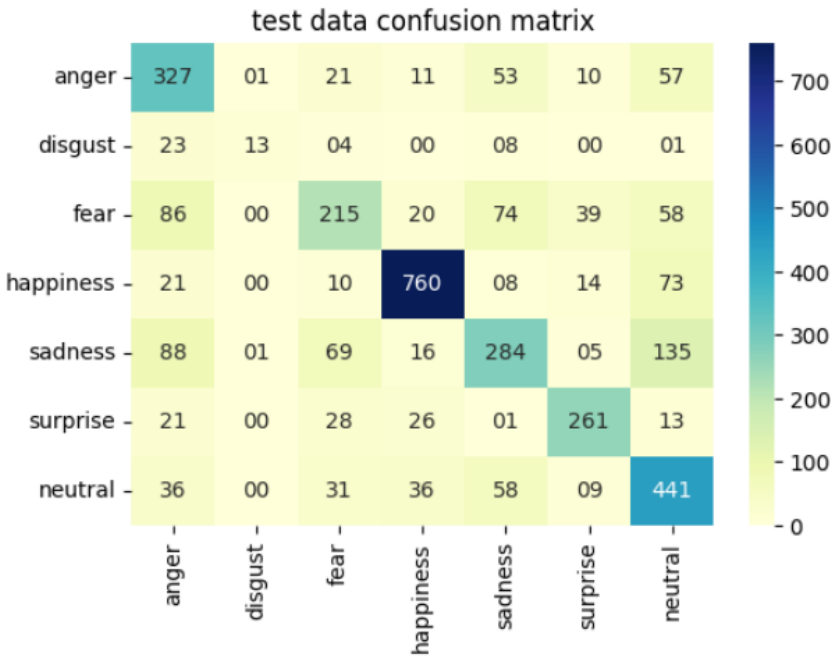
```
test data accuracy score: 0.39036353144835545
test data weighted f1 score: 0.3691946138861337
```



e. Training State of the art CNN Model from scratch.

Transfer Learning with VGG19 showed poor results and we have just enough data to train a large CNN network like VGG19 from scratch. So, we tried training entire VGG19 network with classification layers on top using our train data. It showed the best performance on test data:

```
test data accuracy score: 0.6638776687824581
test data weighted f1 score: 0.6599664679499141
```



MODEL EXPERIMENTS SUMMARY

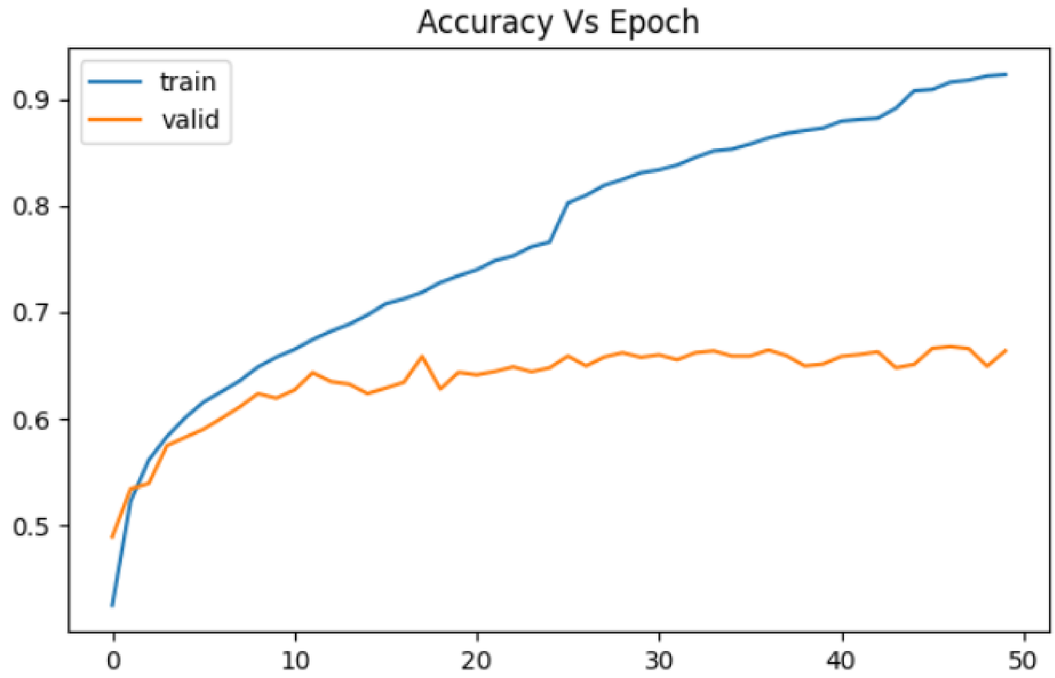
Summary Table for Model Experiments

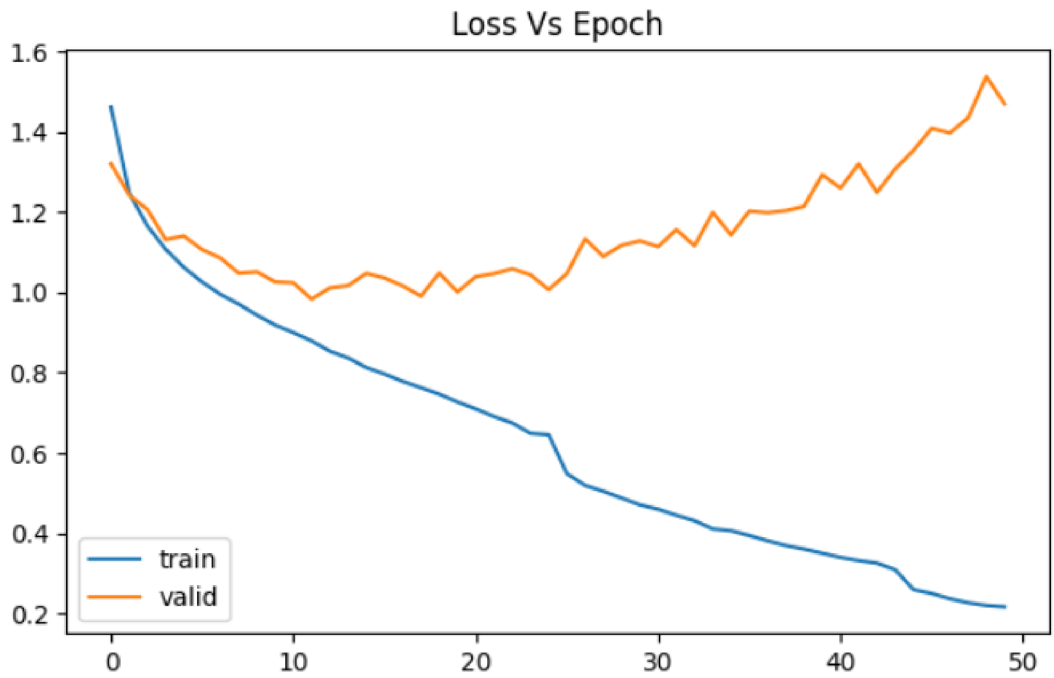
Model Info	Test Accuracy	Test Weighted F1
SVM Classifier	0.43	0.40
MLP Model	0.33	0.26
Custom CNN	0.32	0.27
Transfer Learning (VGG19)	0.39	0.37
Fully Trained VGG19	0.66	0.66

As we can observe in table above, fully trained VGG19 From scratch showed best performance.

BEST MODEL PERFORMANCE ANALYSIS

VGG19 Training Log Plots:



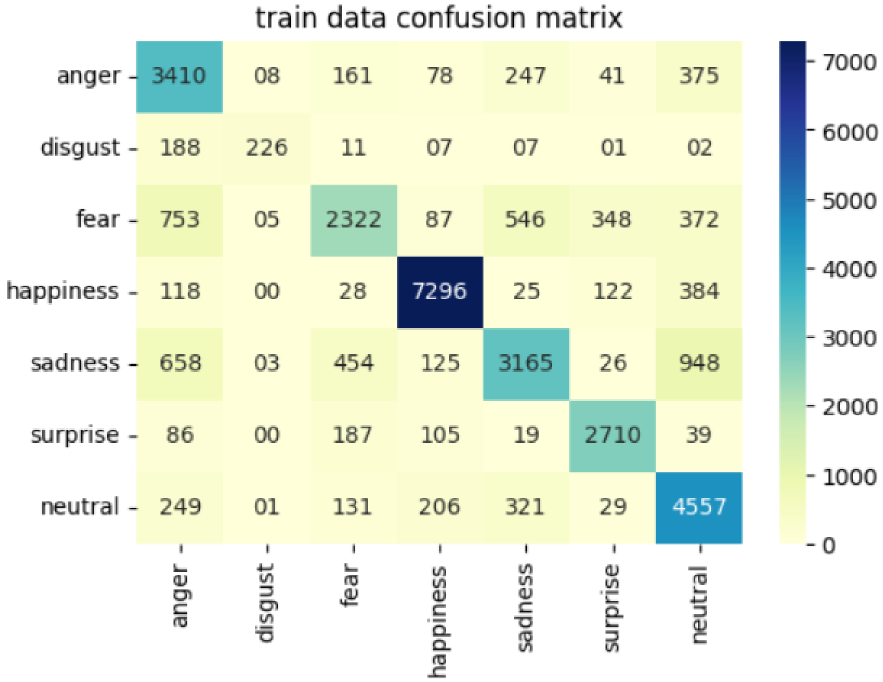


As we can observe in the plots above, VGG19 showed huge overfitting after 18th epoch.

Hence model state at 18th epoch was chosen as Best Model.

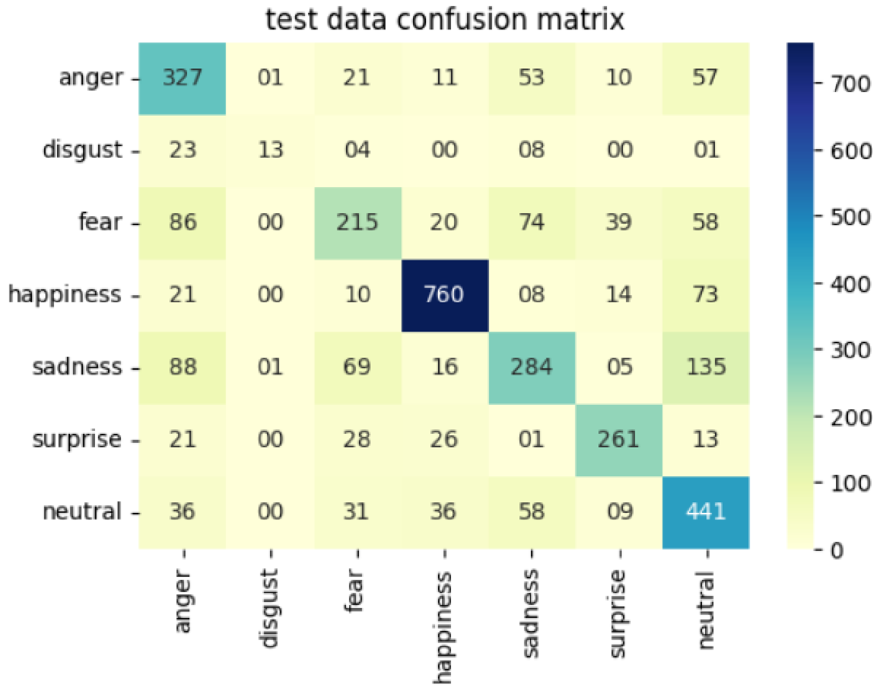
Best Model **Performance on Train Data**:

```
train data accuracy score: 0.7594831179658191
train data weighted f1 score: 0.755805075671032
```



Best Model **Performance on Test Data**:

```
test data accuracy score: 0.6638776687824581
test data weighted f1 score: 0.6599664679499141
```



BEST MODEL REMARKS:

1. Our model is best able to detect happiness with 87% f1 score.

2. It can also detect surprise, neutral with moderate performance at 76%, 63% and 60% f1 score respectively.

3. Model is confused between sadness-neutral, anger-sadness, anger-fear.

Adding more data and/or further tuning/experimenting can improve model performance.

FUTURE SCOPE

1. Different state of the art CNN Models like VGGFace, InceptionV3, ResNet, EfficientNet etc. can be experimented.

2. Transfer learning techniques can be used with more prominent dense layers and enabling training for bottom 10-15% layers.

3. Different more deeper architectures can be tried for MLP and Custom CNN networks.

4. Other ML algorithms with different hyperparameters can be experimented with for MultiClass Classification.

5. Some emotions like sadness and neutral look similar for some people, combining these to labels could improve overall model performance.