Project Report: Sentimental Analysis using Text and Images

Mukand Krishna   **8/29/23**

Introduction

This comprehensive report encompasses two distinct projects: Text-Based Sentiment Analysis and Image-Based Sentiment Analysis. Each project focuses on employing machine learning techniques to analyze different types of data—text and images—ultimately categorizing sentiment to provide valuable insights into public opinions and emotional states.

Text-Based Sentiment Analysis

1. Introduction:

Sentiment analysis of text data derived from sources like Twitter and Reddit. The objective is to classify text into sentiment categories: Negative, Neutral, and Positive. It has practical applications in understanding public emotions and opinions.

2. Objectives:

The main objectives of this project are as follows:

I. Data Collection and Preprocessing: Importing the necessary libraries. Loading data from Twitter and Reddit sentiment datasets, preprocesses the data, and combines both datasets for training.

Null values


```
Null values in reddit:
 clean_text      4
category        7
dtype: int64


Null values in Twitter:
 clean_comment    100
category           0
dtype: int64
```

Data after Cleaning and merging both datasets.


```
                           clean_comment  category
0   when modi promised "minimum government maximum...     -1.0
1   talk all the nonsense and continue all the dra...      0.0
2   what did just say vote for modi  welcome bjp t...      1.0
3   asking his supporters prefix chowkidar their n...      1.0
4   answer who among these the most powerful world...      1.0
5           kiya tho refresh maarkefir comment karo       0.0
6   surat women perform yagna seeks divine grace f...      0.0
7   this comes from cabinet which has scholars lik...      0.0
8   with upcoming election india saga going import...      1.0
9                gandhi was gay does modi               1.0
clean_comment     0
category          0
comment_length    0
dtype: int64
```

ii. Exploratory Data Analysis (EDA): EDA is performed to understand the dataset's characteristics.

Histograms: **Distribution of comment lengths for different sentiment categories.**
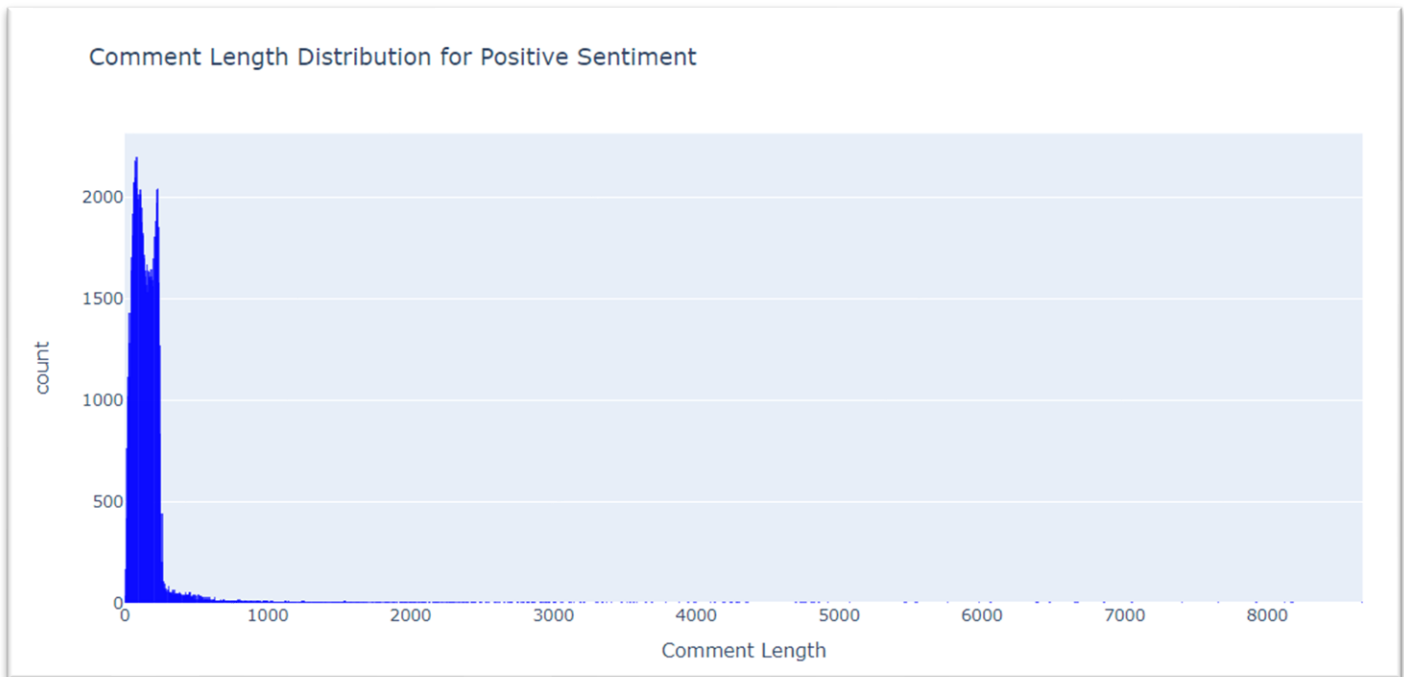
Bar Plots: **Distribution of sentiment categories.**

Word clouds for words associated with different sentiment categories.

Word Cloud for Neutral Sentiments

Preprocessing the text data to prepare it for training.

To more suitable format for analysis, like converting to lowercase letters, splitting text into word , remove words like "the", "is" which don't carry significant meaning.

```
Original tweet -> when modi promised "minimum government maximum governance" ex
rs get justice state should and not business and should exit psus and temples

Processed tweet -> ['modi', 'promis', 'minimum', 'govern', 'maximum', 'govern',
r', 'get', 'justic', 'state', 'busi', 'exit', 'psu', 'templ']
```

Preprocess and transform the text data into sequences of integers, making it suitable for input into machine learning models.

```
Before Tokenization & Padding
 when modi promised "minimum government maximum governance" expected him
ate should and not business and should exit psus and temples
After Tokenization & Padding
[  41    2  394   67 1956 1026   45 2254    1 1132  217    1  169   32
  143  100   57   60 1049  169   49    3    9  517    3   49 2525    3
 2462    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0]
```

Model Definition and Training

Combination of layers like Convolutional Neural Networks (Conv1D), Long Short-Term Memory (LSTM) units, and dense layers to analyze and classify text data.

```
Model: "sequential"

 Layer (type)                   Output Shape              Param #
=================================================================
 embedding (Embedding)          (None, 50, 32)            160000

 conv1d (Conv1D)                (None, 50, 32)            3104

 max_pooling1d (MaxPooling1D     (None, 25, 32)            0
 )

 bidirectional (Bidirectiona     (None, 64)                16640
 1)

 dropout (Dropout)              (None, 64)                0

 dense (Dense)                  (None, 3)                 195

=================================================================
Total params: 179,939
Trainable params: 179,939
Non-trainable params: 0
_____
None
```
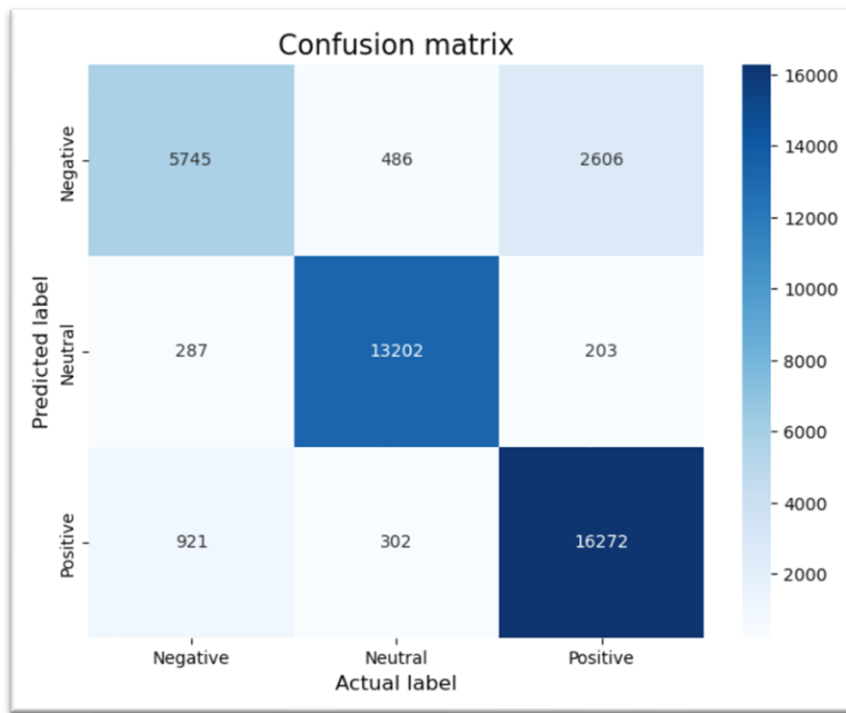
Train vs Test Loss
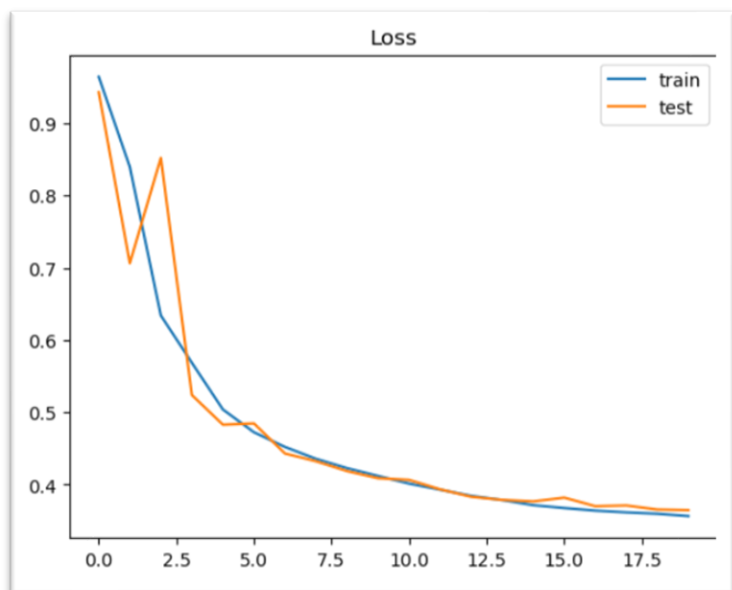
Train vs Test Accuracy

Confusion matrix
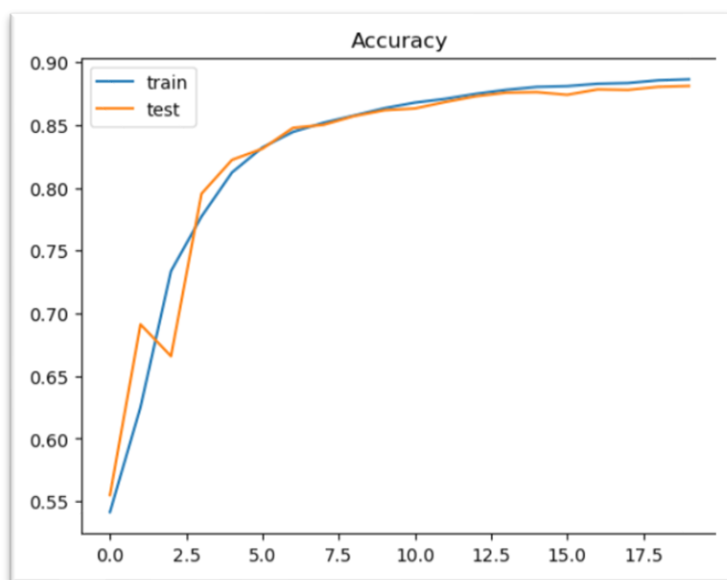
3. Results:

The project successfully performs sentiment analysis on text data from Twitter and Reddit. It preprocesses the data, trains a deep learning model, and evaluates its performance using various visualization techniques and metrics. Predicts the comment fine when it was tested.

```
predict_class(['"I hate when I have to call and wake people up'])

predict_class(['The food was meh'])

predict_class(['He is a best minister india ever had seen'])
```

```
Best model saved
1/1 [==============================] - 1s 792ms/step
The predicted sentiment is Negative
1/1 [==============================] - 0s 27ms/step
The predicted sentiment is Neutral
1/1 [==============================] - 0s 28ms/step
The predicted sentiment is Positive
```

Image-Based Sentiment Analysis

1. Introduction

Performing multi-class image classification to categorize facial images into 7 emotions - Angry, Disgust, Fear, Happy, Neutral, Sadness and Surprise. Using Transfer learning with DenseNet169 model pretrained on ImageNet dataset. Fine-tuning pretrained model on dataset enables better accuracy compared to training a model.

2. Objectives

The key objectives of this analysis are:

1. Loading and visualize sample images to understand the dataset.

2. Data Loading and Preprocessing

- The train and test datasets contain images of shape (48, 48, 3).
- Apply image augmentation like horizontal flips, width/height shifts to increase diversity of images
- Split training data into 80% training and 20% validation set.
- Apply DenseNet169's preprocessing function to images.

```
Model: "model"

Layer (type)                Output Shape           Param #
=================================================================
input_1 (InputLayer)        [(None, 48, 48, 3)]    0

densenet169 (Functional)    (None, 1, 1, 1664)     12642880

global_average_pooling2d (Gl (None, 1664)          0

dense (Dense)               (None, 256)            426240

dropout (Dropout)           (None, 256)            0

dense_1 (Dense)             (None, 1024)           263168

dropout_1 (Dropout)         (None, 1024)           0

dense_2 (Dense)             (None, 512)            524800

dropout_2 (Dropout)         (None, 512)            0

classification (Dense)      (None, 7)              3591
=================================================================
Total params: 13,860,679
Trainable params: 1,217,799
Non-trainable params: 12,642,880
```

- Use **early stopping callback** to terminate training when validation loss stops improving.
- The model is first trained for 30 epochs with frozen DenseNet169 base. Then, it is unfrozen and fine-tuned for 50 more epochs at lower learning rate for convergence.
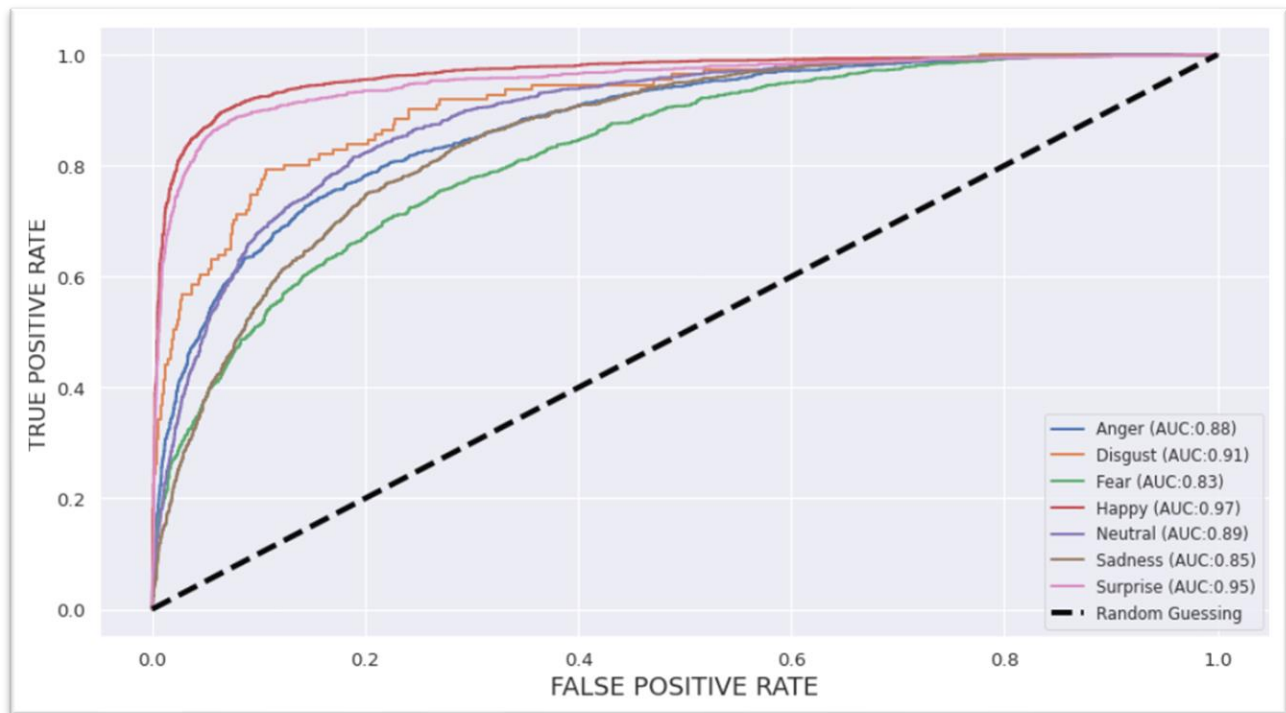

3. Results

Some key results from model evaluation:

Training accuracy reaches 70% while validation accuracy reaches 65% after fine-tuning.
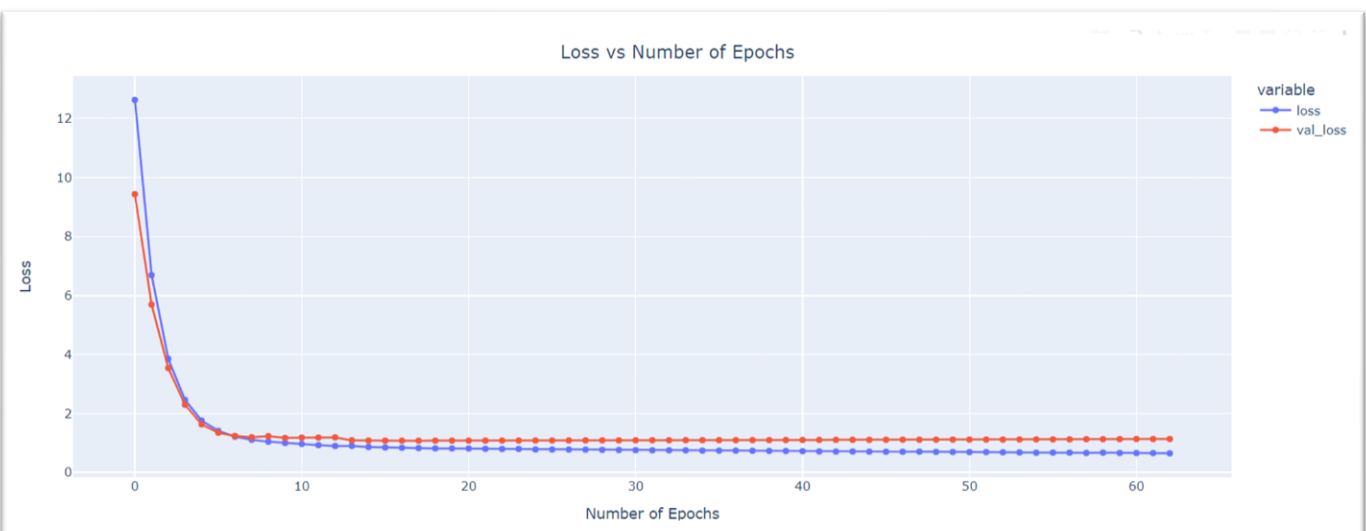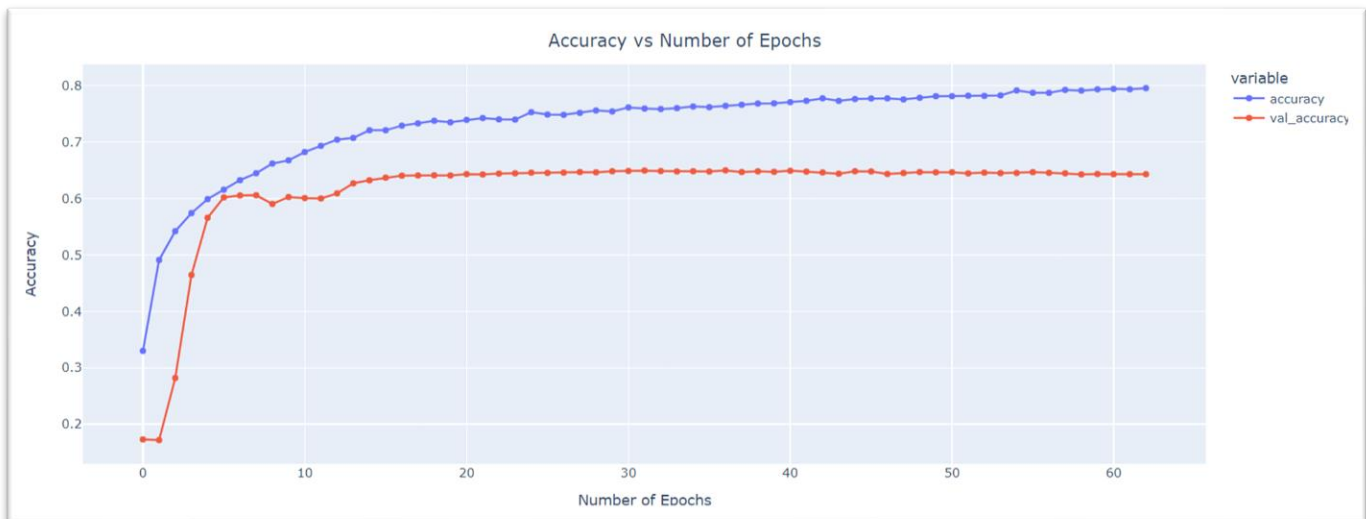
Model achieves ROC-AUC score of 0.9 showing reliable skill in multi-class classification.

Confusion matrix shows most errors come from misclassifying Neutral as Happy and Fear as Surprise.

Confusion matrix is a tool used to <u>evaluate the performance of predictive models</u>. This helps to understand how well the model's predictions align with the actual outcomes.

Consists of <u>four values</u>: **True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).**
TP: instances where the <u>model correctly predicts people's sentiments.</u>

TN: represents correct predictions of <u>people's sentiments</u>.

FP: instances where the model incorrectly predicts sentiments when it shouldn't.

FN: represents where the model fails to predict sentiments when it should.

Classification Report

```
print(classification_report(y_test, y_preds))

              precision    recall  f1-score   support

           0       0.54      0.60      0.57       958
           1       0.00      0.00      0.00       111
           2       0.50      0.45      0.47      1024
           3       0.87      0.85      0.86      1774
           4       0.59      0.66      0.63      1233
           5       0.53      0.54      0.54      1247
           6       0.79      0.76      0.77       831

    accuracy                           0.65      7178
   macro avg       0.55      0.55      0.55      7178
weighted avg       0.64      0.65      0.65      7178
```

4. Conclusion

Sentiment analysis is a valuable application of natural language processing, enabling the understanding of public opinions and sentiments. The project demonstrates the entire process, from data collection and preprocessing to model training and evaluation. By following this approach, the code achieves accurate sentiment predictions, which can be useful for understanding user sentiments on social media platforms and other text-based sources.

Then successfully trained a Deep CNN model using transfer learning to classify facial emotion images into 7 categories. The model shows good performance, but can further be improved by techniques like using more training data, better regularization, data balancing, and predictive image augmentation.