REPORT
[BASELINE]

Akshara Nair - MT22008 | Ayush Agarwal - MT22095 | Medha - MT22110
Mohit Gupta - MT22112 | Nidhi Verma - MT22044 | Pratik Chauhan - MT22118

Problem formulation:

With a massive amount of content being published on an individual's social media, there exists an emergent need to monitor if the posts being posted by an individual are indicative of the existence of early symptoms of depression. **An early and effective analysis and prediction through pre-learnt representations of depression by visual and textual data forms the basis of the problem of the proposed research based project.** The aim of the proposed research based project is development and deployment of Machine Learning and Deep Learning models to analyze multimodal data to predict depression. For the evaluation of baseline models, we have scraped the data as images and text from platforms where user based content is generated such as Reddit (Text) and Shutterstock (Images) and evaluate preliminary results from the best suited models for the purpose. We further look forward to finding the best suited models for classification of an image or text based input as depressed or not depressed and combine the predictions of the specific modalities to provide efficient results on Social Media Posts of an individual.

Literature review:

Physical activities and geographic movement patterns were classified using built-in phone sensors, such as the acceleration and Global Positioning System (GPS) sensors, respectively. A portion of the characteristics were chosen. With an accuracy of 87.2%, the SVM classifier was used to differentiate between the three different severity levels of depression (absence, moderate, and extreme) [1]. Information gathered by mobile phone sensors, including utilization of applications, brightness, acceleration, rotation, and orientation. They were combined by the creator to create higher-level feature vectors. The subjects' stress levels could be predicted using the fusions of these feature vectors [2]. Researchers looked at how the facial cues of depressed and normal individuals changed in the same circumstance. (while displaying positive, neutral, and negative pictures). To measure the facial cue changes on the face, they used a person-specific active appearance model to detect 68-point landmarks. Statistical features are extracted from distances between feature points of the eyes, eyebrows, and corners of the mouth to feed the SVM classifier. The classifier achieved 78% test accuracy [3]. They identified 68-point landmarks using a person-specific active appearance model to quantify the facial cue changes on the face. To provide the SVM classifier with statistical features, the distances between the feature points of the eyes, brows, and corners of the lips are measured. The classifier's test success score was 78% [3]. Due to the following factors, some scholars have recently attempted to combine various modalities: First, when the convergence of modalities is carried out, the input of each modality can be better understood. Second, each modality has benefits of its own. Consequently, a combination can produce a superior result. Third, the elements have compatibility traits. In order to identify psychomotor retardation, Williamson et al. [4] used feature sets drawn from facial movements and acoustic verbal signals. For dimensionality reduction, they used principal component analysis, and to categorize the combination of primary feature vectors, they used the Gaussian mixture model. The majority of the works covered in this section use ML-based techniques rather than deep learning techniques (Support Vector Machines, Gaussian Mixture Models, Random Forest, etc.). This could be due to a lack of training material availability. Deep learning

requires more training data than ML, whereas ML-based methods can be taught on less data. Another explanation might be that supervised machine learning is more effective when there is a known connection between the inputs and labels. To increase the accuracy of the model, many features can be extracted and then assessed. **Depression Detection by Analyzing Social Media Posts of Users :** This study suggests a data-analytic strategy for identifying depression in any human. The data used in this suggested model is gathered from user posts on Twitter and Facebook, two well-known social media websites. The entries are vectorized using the SVM, and their positivity, negativity, or neutrality is ascertained using Nave Bayes. It takes the user's username and examines their social media posts to gauge their degree of susceptibility to depression. The machine learning model is taught to categorize the six ranges of depression criteria. (Considered Normal, Mild, Moderate, Borderline, Severe, and Extreme). The model's assessed accuracy is 74.00% [5].

References

[1]: Masud MT et al (2020) Unobtrusive monitoring of behavior and movement patterns to detect clinical depression severity level via smartphone. J Biomed Inform 103:103371
[2]: Fukazawa Y et al (2019) Predicting anxiety state using smartphone-based passive sensing. J Biomed Inform 93:103151
[3]: Wang Q, Yang H, Yu Y (2018) Facial expression video analysis for depression detection in Chinese patients. J Vis Commun Image Represent 57:228–233
[4]: Williamson JR, Quatieri TF, Helfer BS, Ciccarelli G, Mehta DD (2014) Vocal and facial biomarkers of depression based on motor incoordination and timing, 65–72
[5]: N. A. Asad, M. A. Mahmud Pranto, S. Afreen and M. M. Islam, "Depression Detection by Analyzing Social Media Posts of User," 2019 IEEE International Conference on Signal Processing, Information, Communication & Systems (SPICSCON), Dhaka, Bangladesh, 2019, pp. 13-17, doi: 10.1109/SPICSCON48833.2019.9065101.

METHODOLOGY AND RESULTS : -

1. **Scraping Process**

- **Image Scraping:**

We are scraping images from **shutterstock**. For that we are using selenium, and beautifulSoup. We have created a method which takes 4 parameters, searchTerm, searchPage, image_type, directory_path to store the images.

When we get the 'lxml' object of the web page, then we find all the **<img>** tags, having class name **'mui-1l7n00y-thumbnail'.** Then for each image, we are taking the src value and using the urllib library to retrieve the image and storing it in the given directory. Using this, we are scraping almost 2500 images containing both depression and not_depression images.Then after scraping images from the web, we need to preprocess all images. For preprocessing all the images, we apply:

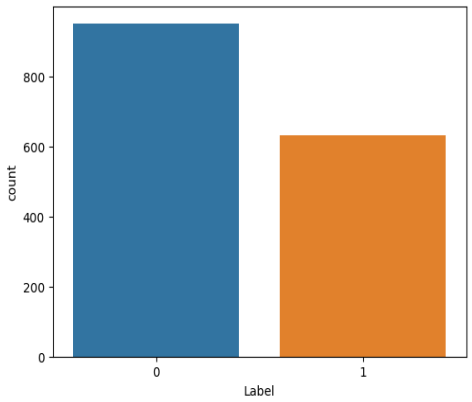
1. **Reduce the dimension of each image to (128, 128)**
2. **Convert the image into grayscale.**
3. **Remove shutterstock watermark.**

Then we split the images into TRAIN and TEST in the ratio of 80:20. The images data folder contains two folders named test and train, which further contains two folders one for each class i.e. depression and not_depression. The specific format is used to load the data efficiently from Image Data Generators for training the Deep Learning Models through Libraries.

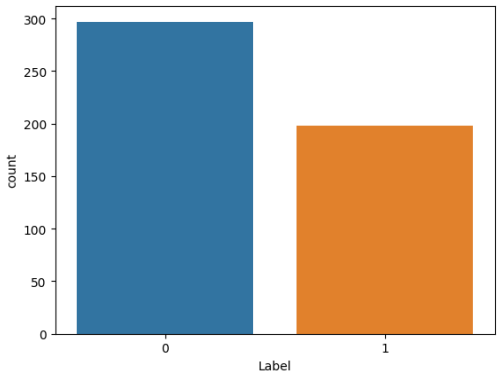
Image Based Classification

A. Data Distribution : - The data distribution is balanced across the categories. The relative ratio of the difference amongst the number of samples in the dataset is followed across the data split in all of the three sets namely, the data being used for Training, Validation and Testing and hence ensures that the results are not biased due to imbalanced data distribution. The Training and Validation split is 80:20 from all the images currently in the Training Dataset.

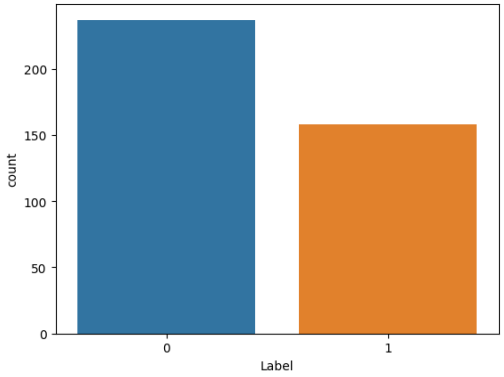
1. **For Training Data**



2. **For Test Data**



3. For Validation Data

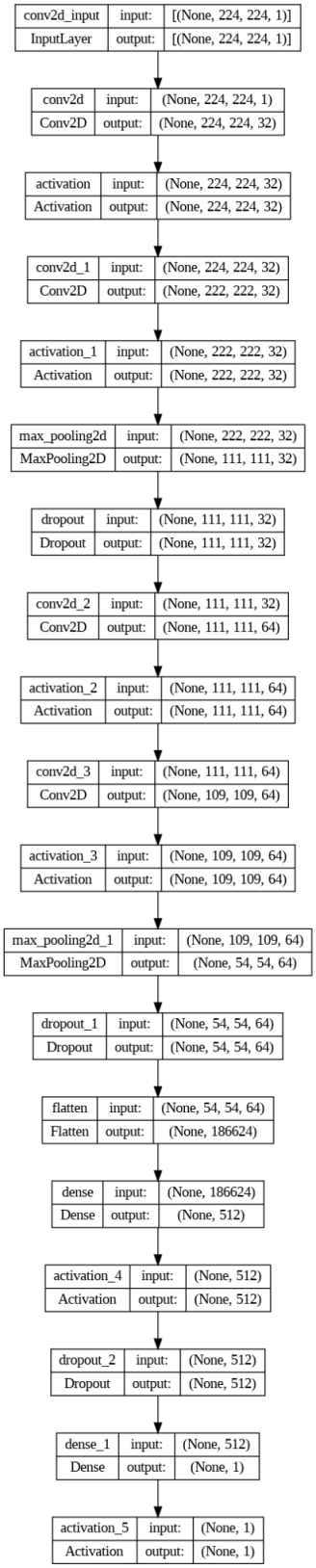


B. Data Visualization : -

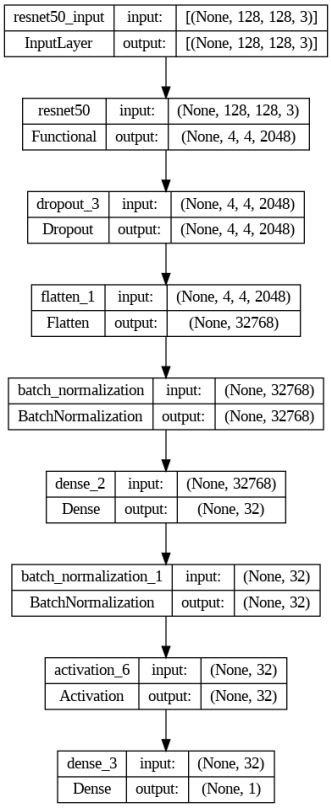
Visualizing the first five images of each batch (BATCH - SIZE = 64) - The first five images of each batch with the corresponding labels are as depicted below.

C. Model Architectures for Image Classification : -

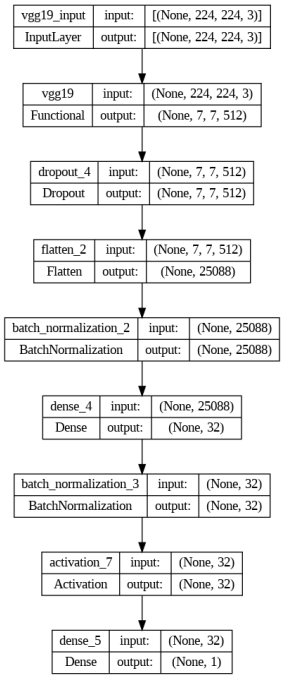
1. Deep CNN



2. ResNet50(Pre-Trained and Fine Tuned)



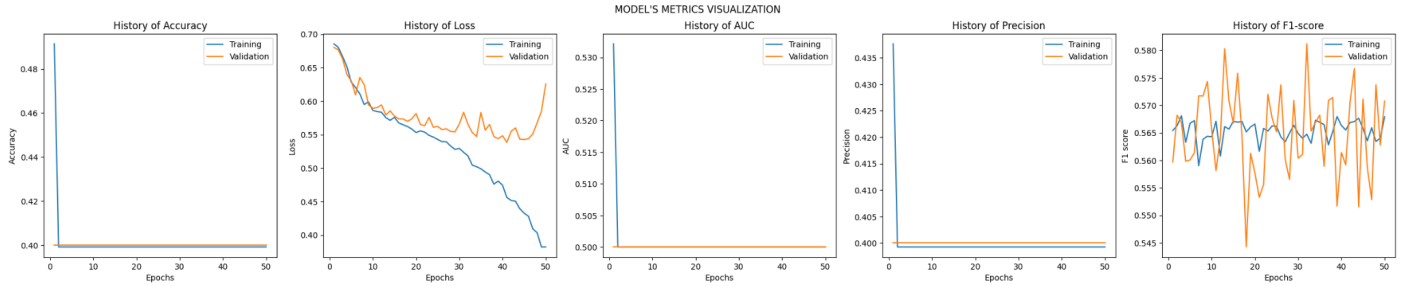
3. VGG19(Pre-Trained and Fine Tuned)



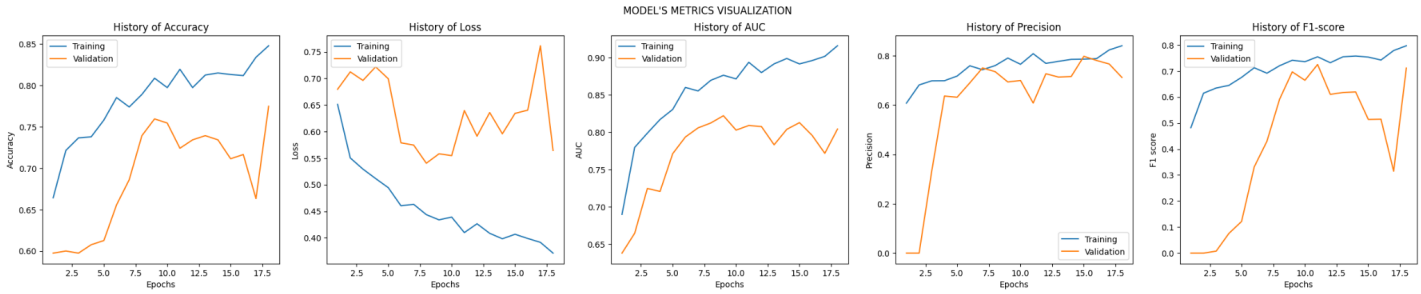
D. Results

1. Deep CNN

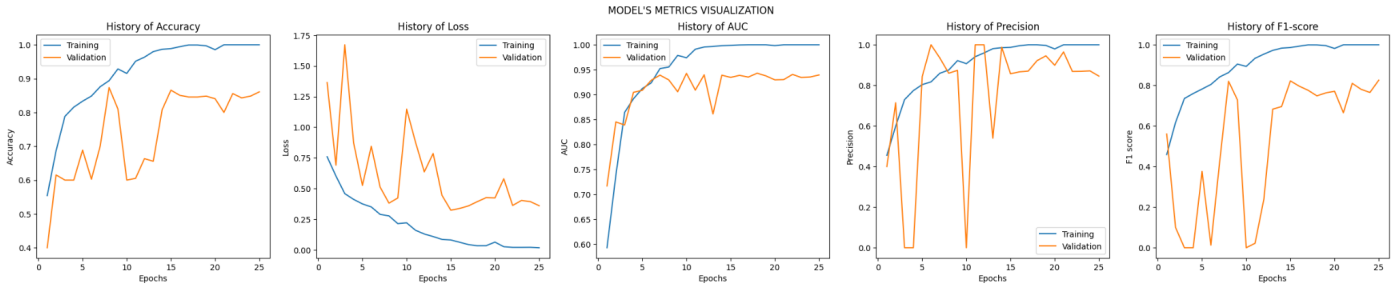
Results :-



2. **Model Architecture - ResNet50(Pre-Trained on ImageNet)**
   Results : -



3. **Model Architecture - VGG19**
   Results : -



The Training and Validation Accuracies for the specified models, being implemented are as below : -

1. Deep CNN - Training Accuracy - 39% and Validation Accuracy - 40%.
2. ResNet50 - Training Accuracy - 84.78% and Validation Accuracy - 77.47%.
3. VGG19 - Training Accuracy - 100% and Validation Accuracy - 86.08%.

Text Based Classification

A. **Text Scraping: -** For scraping text data, we are using the reddit api **'praw'.** What we are doing is, we first find all the top posts having the subreddit 'depression anxiety', then fetching the title, and the body of the post and storing it in a pandas dataframe. Then after we are saving the dataframe

in the .csv file. We have extracted around 12000 text posts containing both depressed and not_depressed posts.

B. **Text Preprocessing-**

These preprocessing techniques are:

1. **Whitespace removal**: The re.sub() method is used to remove multiple whitespace characters with a single space character.
2. **URL removal:** The re.sub() method is used to remove URLs from the input text data.
3. **User mention removal:** The re.sub() method is used to remove user mentions, which start with the "@" character.
4. **Number removal**: The re.sub() method is used to remove all digits from the input text data.
5. **Emoji removal:** The re.sub() method is used to remove all emojis from the input text data. **Lowercasing and tokenization**: The input text data is converted to lowercase and tokenized using the word_tokenize() method from the nltk library.
6. **Stopword removal:** Stop words, which are common words such as "the" and "and" that do not provide much meaning, are removed using list comprehension and the stopwords corpus from the nltk library.
7. **Punctuation removal:** Punctuation marks are removed using list comprehension and the isalnum() method.
8. **Spelling checking:** The Speller() method from the enchant library is used to check the spelling of each word in the input text data.
9. **Lemmatization:** The WordNetLemmatizer() method from the nltk library is used to lemmatize each word in the input text data, reducing them to their base form.

Overall, these preprocessing techniques help to clean and normalize the input text data, making it easier to analyze and extract meaning from it.

C. **Text Models: -** Firstly, we created the tokenizer which uniquely labeled our top 1000 words and rest as OOV words.

Now, we pass this to our model, which consists of three layers:

1.Embeddings Layer: Embedding class used to generate 32-dimensional embeddings of every input sample sentence present in the dataset.

2. Bi-LSTM Layer : An RNN type, modeling the sequential dependencies between words and phrases in both directions of the sequence.

3. Dense Layer :

```
0s
 2 model.summary()

Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_4 (Embedding)     (None, 764, 32)           724896

 bidirectional_4 (Bidirectio  (None, 64)               16640
 nal)

 dense_4 (Dense)             (None, 1)                 65

=================================================================
Total params: 741,601
Trainable params: 741,601
Non-trainable params: 0
_________________________________________________________________
```

D. Results on few samples :

```
 9 #bracket represent probability of class 1(depressed). if prob > 0.5, its classified as depressed, else not.

1/1 [==============================] - 1s 665ms/step
thing puts smile face brother hes best -> not depressed (0.002152541186660528)
anxiety literally ga kill someone anyone help pls mens sleep drinking water help -> depressed (0.9907008409500122)
I am so depressed kill  -> depressed (0.9696369767189026)
life is beautiful -> not depressed (0.01041035819798708)
lets play in the sun -> not depressed (0.033363226801157)
i am very much scared of medha -> depressed (0.7482262849807739)
```

The Training and Test Accuracies for the specified model being implemented are as below : - Training Accuracy - 0.9914 and Validation Accuracy - 0.9291