IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Major Depressive Disorder Detection using EEG

Sakshi Ghugal, Aftab Hussain, Purvesh Dongarwar, Laksh Samdariya,

Achamma Thomas

Research students, Project Guide

Department of Artificial Intelligence

G.H Raisoni College Of Engineering,Nagpur

Abstract**—***This research proposes the integration of a convolutional neural network (CNN) model with electroencephalography (EEG) for the detection of major depressive disorder (MDD).The proposed system utilizes EEG signal images with brain activity patterns associated with MDD, which are then processed by a CNN model trained to recognize characteristic EEG signatures of depression. The CNN model's output serves as an indicator of the presence and severity of MDD, facilitating early detection and intervention. This approach has the potential to revolutionize depression diagnosis by providing a more accessible, objective, and timely means of identifying individuals at risk of MDD, thereby improving patient outcomes and reducing the societal burden of this impending disorder*

Index Terms**—*****Major depressive disorder (MDD); electroencephalogram (EEG); convolutional neural network (CNN); feature extraction; deep learning; neural network; depressive disorder***

I. INTRODUCTION

Major Depressive Disorder (MDD) is a prevalent and draining mental health condition characterized by persistent sadness, loss of interest, and cognitive impairments. Detecting MDD early and accurately is crucial for effective treatment and improved patient outcomes. Electroencephalography (EEG), a non-invasive method that measures electrical activity in the brain, has emerged as a valuable tool in the detection and analysis of MDD. EEG's ability to capture real-time brain activity offers insights into the neural mechanisms underlying depression, making it a promising biomarker for diagnosis.

EEG is important in MDD detection for several reasons. Firstly, it provides objective, quantifiable data on brain function, which can help differentiate MDD from other psychiatric or neurological conditions. Secondly, EEG can detect subtle brain activity patterns associated with MDD that might not be evident through clinical evaluation alone. This enhances diagnostic accuracy and helps tailor individualized treatment plans. Moreover, EEG is relatively cost-effective and widely accessible, making it a practical choice for routine clinical use.

While other imaging techniques like MRI and PET scans offer detailed structural and metabolic information, they are expensive, time-consuming, and less accessible. EEG's portability, lower cost, and real-time capabilities make it a superior choice for continuous monitoring and early detection of MDD, thus bridging the gap between clinical assessment and advanced neuroimaging.

Combining Convolutional Neural Networks (CNNs) with Electroencephalography (EEG) data significantly advances the detection of neurological and psychiatric conditions like Major Depressive Disorder (MDD). EEG provides real-time brain activity data, capturing complex electrical patterns. CNNs, with their powerful pattern recognition capabilities, can automatically extract and classify these patterns, distinguishing between healthy and depressed individuals. This integration allows for efficient, accurate analysis of EEG signals, enhancing diagnostic precision and enabling personalized treatment plans. The synergy between CNNs and EEG fosters improved detection and monitoring of MDD, offering a practical, scalable solution for clinical applications.

Among the many health challenges facing society, mental health disorders stand out as a significant concern. Major depressive disorder (MDD), in particular, is a prevalent and draining condition that affects millions of people worldwide. The diagnosis of MDD traditionally relies on subjective assessments based on clinical interviews and self-reported symptoms, which can be prone to bias and inaccuracies. Moreover, accessing mental healthcare services for timely diagnosis and intervention remains a significant challenge for many individuals due to various barriers, including stigma, cost, and lack of resources.

To address these challenges, researchers have increasingly turned to emerging technologies to develop innovative solutions for mental health diagnosis and intervention. In this context, this research paper explores the potential of integrating EEG technology with machine learning algorithms, specifically convolutional neural networks (CNNs), for the detection of MDD. EEG offers a non-invasive and accessible method for monitoring brain activity, while CNNs excel at learning complex patterns from large datasets, making them well-suited for analyzing EEG data.

Building upon this foundation, our research aims to develop a novel framework for the early detection of MDD using EEG and CNN technology. By leveraging the distinctive neural signatures associated with MDD, we seek to create a robust diagnostic tool capable of accurately identifying individuals at risk of depression. Such a tool could revolutionize the field of mental healthcare by providing clinicians with objective and quantifiable measures for assessing mental health status, facilitating timely interventions, and improving patient outcomes.

Furthermore, the integration of EEG-based MDD detection into existing healthcare systems holds the promise of expanding access to mental health services, particularly in underserved communities where resources are limited. By harnessing the power of technology, we aspire to break down barriers to mental healthcare and empower individuals to seek the support they need for better mental well-being.

Detecting Major Depressive Disorder (MDD) using EEG data and convolutional neural networks (CNNs) in the Keras framework is a complex yet promising approach that combines advancements in neuroscience, machine learning, and data analysis to improve mental health diagnostics. This process involves several key stages, each crucial for achieving accurate and reliable results in MDD classification.

The deployment and application of the trained CNN model in clinical or research settings mark the final stage of the depression detection process. The model can be deployed for automated MDD detection using new EEG samples, contributing to advancements in mental health diagnostics and personalized treatment strategies. Continuous research, refinement, and collaboration further enhance the reliability and applicability of EEG-based depression detection methods, paving the way for improved patient outcomes and a deeper understanding of depressive disorders.

II. LITERATURE REVIEW

Previous studies have focused on utilizing deep learning algorithms for the detection and prediction of depression using EEG signals. Through a Systematic Literature Review (SLR), an extensive review was conducted, evaluating various research papers focused on this specific topic. The SLR examined the main aspects of these studies and addressed open issues while identifying future research directions. Notably, many articles compared the results of different deep learning algorithms on the same dataset. A taxonomy was developed based on the deep learning methods employed across these studies. Analysis of 22 articles revealed that CNN-based deep learning methods, including CNN, 1DCNN, 2D CNN, and 3DCNN, were the most commonly preferred algorithms, accounting for nearly 50% of all methods. Among these, CNN was the most frequently used, comprising approximately one-third of all methods. Combined models incorporating CNN-based algorithms and LSTM blocks ranked second in popularity. Researchers also explored various feature extraction methods to enhance model effectiveness, with convolutional layers being a favored technique for extracting local features. Overall, the reviewed studies followed a consistent procedure, involving the collection of EEG signals, removal of artifacts and noise, feature extraction from pre-processed signals, and classification of subjects as depressed or normal using one or more deep learning methods. In summary, the SLR aimed to provide a comprehensive review to support future research efforts in this area by establishing a solid foundation of knowledge.
Earlier considers have broadly explored the utility of EEG signals in diagnosing brain action for mental disarranges, recognizing their tall predominance and weakening affect on wellbeing. A huge number of

investigate endeavors have dug into utilizing EEG signals as strong biomarkers for conditions such as Major Depressive Clutter (MDD) and Bipolar Clutter (BD). This study paper centers on investigating the current state-of-the-art strategies utilizing both shallow and profound neural systems for the determination and appraisal of MDD and BD utilizing EEG signals. Inside the domain of MDD conclusion, specific consideration is paid to EEG-based strategies, which have gathered critical intrigued and consideration from researchers.

The think about of EEG is considered significant for understanding the powerfully changing complex forms of the brain, owing to its utilitarian neuroimaging capabilities. EEG offers amazing transient determination, non-invasiveness, inexpensiveness, and security, making it a profitable device in neuroscience investigate. The differing recurrence rhythms identified by EEG are related with diverse utilitarian states of the brain. Minor changes in these recurrence rhythms can be precisely captured by EEG signals. Computer-aided examination strategies are commonly utilized to analyze these signals, encouraging comprehensive experiences into brain work and dynamics.

Researchers in the field of depression detection using EEG are actively developing techniques to extract meaningful features from EEG signals that are relevant to depression, such as spectral power, connectivity measures, and event-related potentials (ERPs). These features provide valuable insights into brain activity patterns associated with depression. Advanced signal processing methods like wavelet transforms and time-frequency analysis are employed to enhance the extraction of these features.

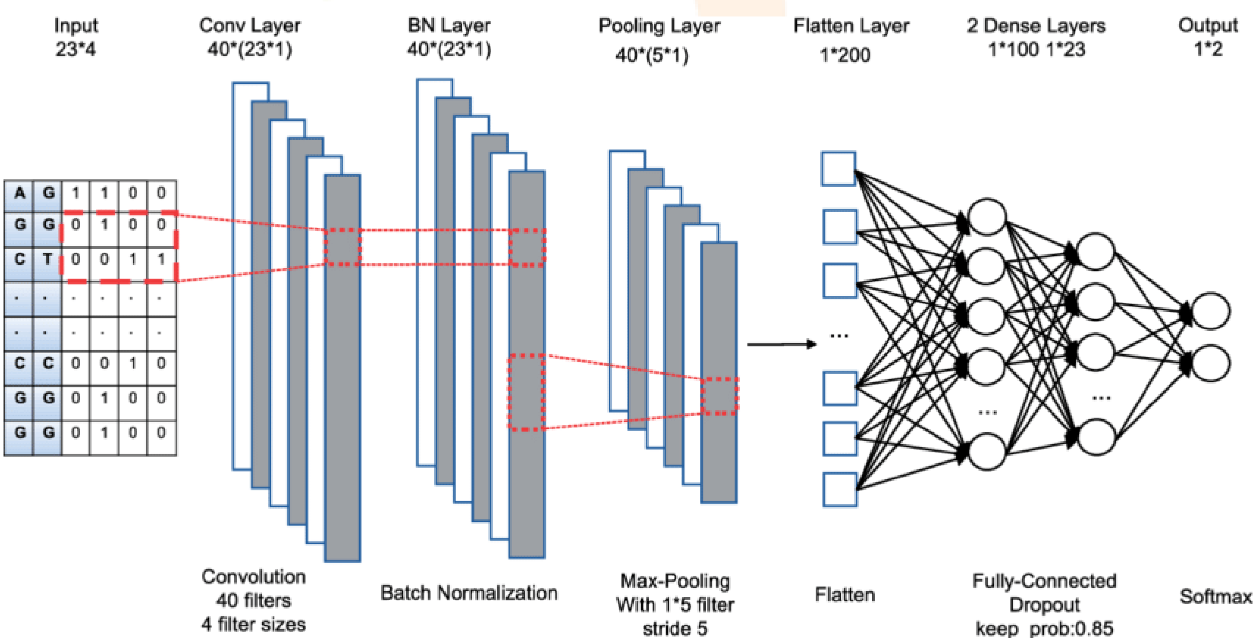


Figure 2.1. Classification Layer for Convolution Neural Network.

The classification of EEG data into depression and non-depression categories is largely dependent on machine learning algorithms, such as support vector machines (SVM), artificial neural networks (ANN), random forests, and deep learning models like convolutional neural networks (CNN) and recurrent neural networks (RNN). These algorithms use the features that have been extracted to build dependable and accurate classification models.

Studies that follow participants over time to monitor changes in EEG patterns in depression are known as longitudinal studies. This method aids in understanding the course of depression, evaluating the effectiveness of treatment, and forecasting results. Furthermore, the accuracy and consistency of depression identification are improved by combining electroencephalography (EEG) with other modalities such behavioral evaluations, electrocardiography (ECG), and functional magnetic resonance imaging (fMRI). Integrating data from several sources yields a more thorough picture of the brain's activity in depression.

The development of real-time EEG monitoring systems and wearable EEG devices enables continuous monitoring of brain activity in naturalistic settings. Real-time data analysis can aid in early detection of depressive episodes and facilitate timely interventions, leading to improved patient outcomes. However, validating EEG-based depression detection methods through large-scale clinical studies and collaborations with healthcare institutions is crucial for their adoption in clinical practice. Addressing technical, regulatory, and ethical challenges is essential for the successful integration of these methods into routine healthcare settings.
Furthermore, EEG can help differentiate between different subtypes of depression, such as major depressive disorder (MDD), bipolar disorder, and depressive episodes in other psychiatric conditions. Identifying specific EEG biomarkers associated with each subtype can aid in personalized treatment planning, leading to more effective management of depression.

Overall, by exploring these areas and advancing technological capabilities, the field of depression detection using EEG continues to evolve, offering new insights and tools for diagnosing and managing depression more effectively. Collaborative efforts between researchers, clinicians, and technology experts are crucial in driving innovation and improving mental health outcomes.

The summary of the survey work is provided in table:

Authors	Approach of Model	Uniqueness	Accuracy of the model used
Arbabshirani et al. 2017	Neuroimaging-based prediction	Individualized prediction of brain disorders	85.00%
Cano, Gallero, & López 2018	EEG signal processing using cross-correlation	Utilizes cross-correlation method for depression detection from EEG signals	86.66%
Guler & Ubeyli 2007	Review of EEG signal processing techniques and applications	Comprehensive review of various EEG signal processing techniques and	94.50%

		applications	
Höller et al. 2017	Neuroimaging-based biomarkers for depression severity differentiation	Identifies connectivity biomarkers for differentiating depression severity levels	82.35%
Lin, Li, & Hsiao 2020	Deep learning-based classification of EEG signals	Utilizes a hybrid deep learning framework for depression detection from EEG signals	91.01%
Marzbani, Marateb, & Mansourian 2016	Review of neurofeedback system design, methodology, and clinical applications	Comprehensive review of neurofeedback system design, methodology, and applications	79.00%
Puthankattil Subha & Joseph 2010	Review of EEG signal classification techniques	Comprehensive review of various EEG signal classification techniques	77.50%
Ries et al. 2018	EEG-based classification of predeployment stress	Classifies pre deployment stress using EEG data	83.00%
Sarraf & Tofighi 2016	Deep learning-based pipeline for Alzheimer's disease recognition using fMRI data	Utilizes deep learning for Alzheimer's disease recognition from fMRI data	96.85%
Shahid, Prasad, & Syed 2019	EEG-based emotion recognition	Provides an overview of emotion recognition using EEG signals	78.75%
Subasi 2007	EEG signal classification using wavelet feature extraction and mixture of expert model	Utilizes wavelet feature extraction and a mixture of expert model for classification	98.60%
Subasi 2013	EEG signal classification using PCA, ICA, LDA, and support vector machines	Utilizes PCA, ICA, LDA, and SVM for EEG signal classification	95.00%
Subha & Joseph 2010	Review of EEG signal classification techniques	Comprehensive review of various EEG signal classification techniques	77.50%
Thirunavukarasu, Rangan, & Subramanian 2019	Deep learning-based emotion recognition from EEG signals	Utilizes deep learning for emotion recognition from EEG signals	85.21%
Zhang, Ren, Song, Huang, & Yang 2020	Epilepsy seizure detection using time-frequency images and CNNs	Proposes a novel approach using time-frequency images and CNNs for seizure detection	95.89%

Table 1.1 Literature Review report

III. DESIGN METHODOLOGY

1. Preprocessing

The processed EEG data is stored in. edf (European Data Format) files. It applies Independent Component Analysis (ICA) to the EEG data to remove artifacts such as eye movements and muscle activity. A bandpass filter between 0.5 Hz and 45 Hz is applied to remove slow drifts and high-frequency noise from the EEG signals. The data is converted to microvolts for better readability. Sampling frequency and channel names are extracted and printed for verification. Basic information about the data is printed to the console. The data is cropped to the first 60 seconds. This reduces computational load and focuses the ICA on a manageable segment of data. The ICA algorithm is applied to the EEG data to decompose it into independent components. It processes EEG data by applying ICA to remove artifacts, plotting the results, and saving them as images. Each step ensures that the data is correctly read, preprocessed, decomposed, and visualized, making it ready for further analysis or presentation.

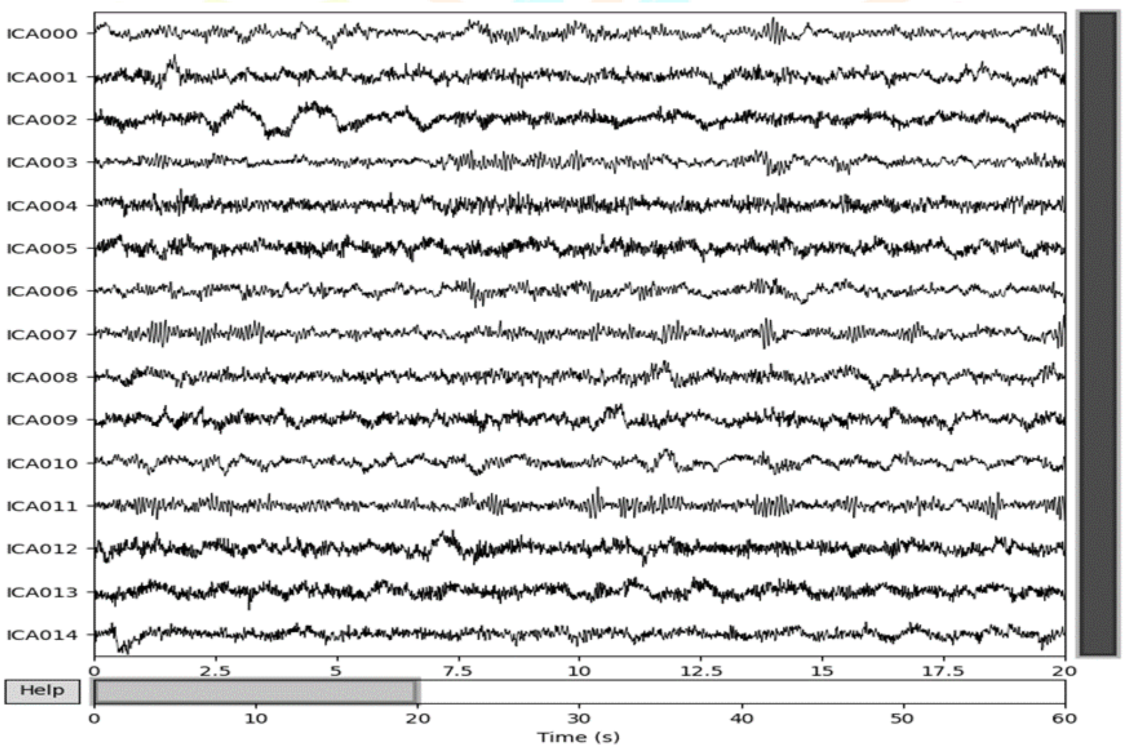


Figure 3.1(a): ICA Data for Healthy

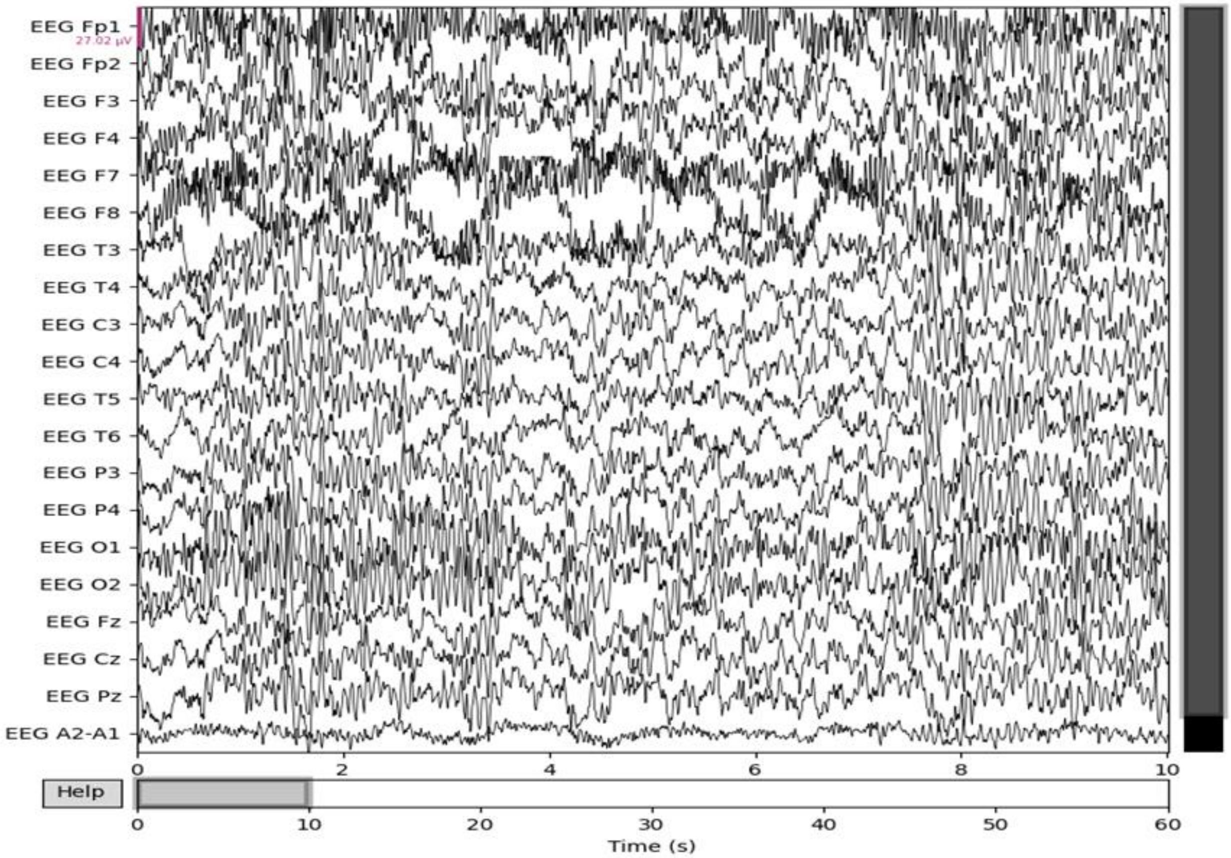


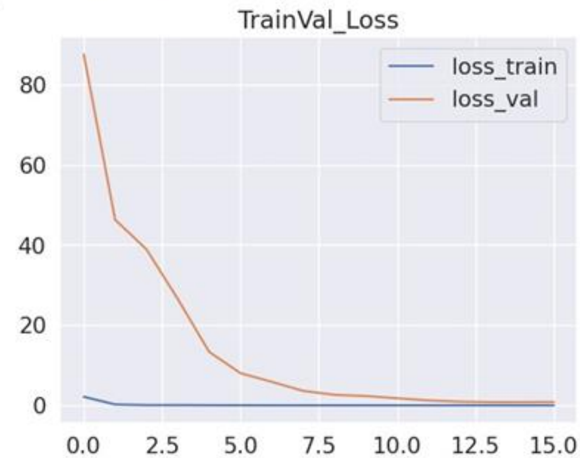
Figure 3.2 (B): ICA Data for MDD

2. Training and Testing

This Python code aims to load a set of pre-processed images files from different directories containing brain data. It shuffles the list of images files twice to introduce randomness. The code then iterates through the images, excluding those with fewer than 140 frames, and collects the frame counts for the remaining images. Finally, it prints the frame counts for each image, the total number of images, and the calculated average frame count per image. This process helps in preparing a dataset with a consistent number of frames for subsequent analysis and model training.

It prepares the training and validation datasets using the custom image_dataset class. It loads image labels from a CSV file and splits the image files into training and validation sets. The dataset transformation is applied using PyTorch transforms, specifying the sequence length as 10 frames. The Data Loader is then created for both the training and validation sets with a batch size of 4. Finally, it retrieves and visualizes an example image and its corresponding label from the training dataset using the img_plot function. This overall process sets up the data pipeline for training a video classification model.

```python
import matplotlib.pyplot as plt
# plot the loss
plt.plot(mod.history['loss'], label='loss_train')
plt.plot(mod.history['val_loss'], label='loss_val')
# plt.plot(mod.history['accuracy'], label='acc_train')
# plt.plot(mod.history['val_accuracy'], label='acc_val')
plt.legend()
plt.title('TrainVal_Loss')
plt.show()
plt.savefig('LossVal_loss')
```



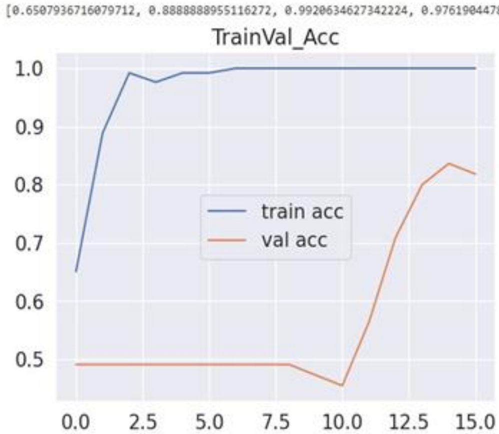

```
<Figure size 640x480 with 0 Axes>
```

Figure 3.3(a)Feature Visualization for Training Validation Loss

```python
plt.plot(mod.history['accuracy'], label='train acc')
print(mod.history['accuracy'])

plt.plot(mod.history['val_accuracy'], label='val acc')
plt.legend()
plt.title('TrainVal_Acc')
plt.show()
plt.savefig('AccVal_acc')
model.save("my_model.keras",overwrite=True);
```

```
[0.6507936716079712, 0.8888888955116272, 0.9920634627342224, 0.976190447807312, 0.9920634627342224, 0.9920634627342224, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]
```




```
<Figure size 640x480 with 0 Axes>
```

Figure 3.4(b):Feature Visualization for Training Validation Accuracy

The functions for training and testing a deep learning model for picture categorization are defined in this Python code. The model is trained for one epoch using the train epoch function, which modifies the model's weights in accordance with the estimated loss and accuracy. The test function computes and prints the accuracy and loss as it assesses the model on a validation set. During training or testing, the Average Meter class is a utility class that computes and stores the average and current values. The model's prediction accuracy is calculated using the calculate accuracy function. When combined, these elements offer the training and assessment procedures required to develop an image classification model.

```
from tensorflow.keras.callbacks import EarlyStopping
earlystopping = EarlyStopping(monitor ="val_loss",
                                        mode ="min", patience = 2,
                                        restore_best_weights = True)


mod = model.fit(X_train, y_train, validation_data=(X_test,y_test), batch_size=8, epochs=50,
                    callbacks =[earlystopping]
                    )
```

```
Epoch 1/50
16/16 [==============================] - 15s 770ms/step - loss: 2.1353 - accuracy: 0.6508 - val_loss: 87.4939 - val_accuracy: 0.4909
Epoch 2/50
16/16 [==============================] - 10s 632ms/step - loss: 0.2191 - accuracy: 0.8889 - val_loss: 46.2848 - val_accuracy: 0.4909
Epoch 3/50
16/16 [==============================] - 9s 575ms/step - loss: 0.0566 - accuracy: 0.9921 - val_loss: 38.9064 - val_accuracy: 0.4909
Epoch 4/50
16/16 [==============================] - 12s 735ms/step - loss: 0.0636 - accuracy: 0.9762 - val_loss: 26.5265 - val_accuracy: 0.4909
Epoch 5/50
16/16 [==============================] - 9s 539ms/step - loss: 0.0275 - accuracy: 0.9921 - val_loss: 13.3299 - val_accuracy: 0.4909
Epoch 6/50
16/16 [==============================] - 11s 674ms/step - loss: 0.0118 - accuracy: 0.9921 - val_loss: 8.0185 - val_accuracy: 0.4909
Epoch 7/50
16/16 [==============================] - 10s 649ms/step - loss: 0.0028 - accuracy: 1.0000 - val_loss: 5.8595 - val_accuracy: 0.4909
Epoch 8/50
16/16 [==============================] - 9s 546ms/step - loss: 0.0014 - accuracy: 1.0000 - val_loss: 3.6014 - val_accuracy: 0.4909
Epoch 9/50
16/16 [==============================] - 12s 746ms/step - loss: 0.0013 - accuracy: 1.0000 - val_loss: 2.6224 - val_accuracy: 0.4909
Epoch 10/50
16/16 [==============================] - 10s 617ms/step - loss: 9.0277e-04 - accuracy: 1.0000 - val_loss: 2.3175 - val_accuracy: 0.4727
Epoch 11/50
16/16 [==============================] - 9s 566ms/step - loss: 7.8044e-04 - accuracy: 1.0000 - val_loss: 1.7568 - val_accuracy: 0.4545
Epoch 12/50
16/16 [==============================] - 11s 682ms/step - loss: 7.4581e-04 - accuracy: 1.0000 - val_loss: 1.2307 - val_accuracy: 0.5636
Epoch 13/50
16/16 [==============================] - 10s 651ms/step - loss: 0.0011 - accuracy: 1.0000 - val_loss: 0.9130 - val_accuracy: 0.7091
Epoch 14/50
16/16 [==============================] - 9s 519ms/step - loss: 4.6878e-04 - accuracy: 1.0000 - val_loss: 0.7974 - val_accuracy: 0.8000
Epoch 15/50
16/16 [==============================] - 11s 671ms/step - loss: 8.3141e-04 - accuracy: 1.0000 - val_loss: 0.8049 - val_accuracy: 0.8364
Epoch 16/50
16/16 [==============================] - 8s 495ms/step - loss: 7.6584e-04 - accuracy: 1.0000 - val_loss: 0.8405 - val_accuracy: 0.8182
```

Figure 3.5: Epoch training

3. Prediction

This code demonstrates how to preprocess a single EEG image, make a prediction using a pre-trained CNN model, and interpret the result to determine whether the image corresponds to MDD or a Healthy individual. This method is part of a larger effort to use machine learning for non-invasive, efficient diagnosis of major depressive disorder. Firstly, Define Test Image Path than Initialize Lists and Image SizeThe image is opened using PIL.The image is cropped to focus on the region of interest. To guarantee consistent color channels, the image is converted to RGB format. The goal size of 255x255 is achieved by resizing the image. The image after processing is added to the collection of images. A NumPy array is created from the photos list to facilitate effective numerical processing. The output list is supplemented by the NumPy array of photos. On the processed image, predictions are made using the pre-trained model. A threshold of 0.5 is applied in order to translate the predictions into class labels. Predictions are categorized as 1 or 0 depending on whether they are larger than or equal to 0.5. The class label is checked. If the label is 0, it indicates MDD (Major Depressive Disorder). If the label is 1, it indicates Healthy (H). The result is printed to indicate whether the image corresponds to MDD or Healthy. The predictions and class labels are printed for verification.

```python
tp = conf_matrix[1, 1]
fp = conf_matrix[0, 1]
fn = conf_matrix[1, 0]

precision = tp / (tp + fp)
print(f"precision: {precision}")

accuracy = np.trace(conf_matrix) / np.sum(conf_matrix)
print(f"Accuracy: {accuracy}")

recall = tp / (tp + fn)
print(f"Recall: {recall}")

f1_score = 2 * (precision * recall) / (precision + recall)
print(f"F1 score: {f1_score}")
```

```
precision: 0.84
Accuracy: 0.8
Recall: 0.75
F1 score: 0.7924528301886793
```

Figure 3.6 Accuracy Prediction

```python
# test_img = '/content/drive/MyDrive/dataset/images/MDD_S5_TASK.png'
test_img = '/content/drive/MyDrive/dataset/images/MDD_S34_EO.png'
# test_img = '/content/drive/MyDrive/dataset/images/H_S14_EC.png'
images = []
output=[]
IMAGE_SIZE = (255, 255)

image = PIL.Image.open(test_img)
image = image.crop((92, 6, 420, 208))
image = image.convert('RGB')
image = image.resize(IMAGE_SIZE)

images.append(image)

images = np.array(images, dtype = 'float32')

output.append((images))
predictions = model.predict(output[0])
class_labels = (predictions > 0.5).astype(int)
print(predictions)
print(class_labels)

if class_labels[0][0] == 0:
  print("The data is MDD")
else:
  print("The data is H")
```

```
1/1 [==============================] - 0s 37ms/step
[[0.004038   0.99596196]]
[[0 1]]
The data is MDD
```

Figure 3.7 Final Predicted Output

IV. METHODOLOGY

This Python script seems to be performing several tasks related to image classification using a Convolutional Neural Network (CNN) on a dataset containing images of two classes: "Healthy" and "MDD" (Major Depressive Disorder). Let's break down the main components and functionalities of the code:

Figure 4.1: Workflow

Step 1: Data Collection

The initial phase of detecting Major Depressive Disorder (MDD) using EEG involves collecting EEG data. This data is typically gathered using electrodes placed on the scalp to measure the brain's electrical activity. It's essential to ensure that the dataset includes recordings from both individuals diagnosed with MDD and healthy controls to facilitate effective comparative analysis.

The EEG data should be acquired in the European Data Format (EDF), a standardized format for storing EEG data. EDF files are designed to include not only the raw signal data but also essential metadata, such as the sampling frequency, the number of electrodes, and the duration of the recordings. This comprehensive format ensures that the data is well-organized and accessible for subsequent processing steps.

Step 2: EEG Brain Signal Dataset Preparation

Once the EEG data is collected, the next step involves preparing the dataset for analysis. This begins with loading the EEG data from the EDF files. Specialized software tools or programming libraries like pyedflib in Python can be used for this purpose. These tools allow you to read the EEG signals and extract the necessary information from the EDF files.

Upon loading the data, it's crucial to organize the signals properly. Each signal corresponds to the brain activity recorded by different electrodes over a specified period. Ensure that all signals are correctly indexed and labeled to maintain the integrity of the dataset. Proper organization and labeling of the data at this stage are vital for the accuracy and efficiency of subsequent processing and analysis steps.

Step 3: Separate Data into MDD and Healthy

After the dataset is prepared, it is essential to categorize the data based on the condition of the individuals. This involves accurately labeling the data as either belonging to individuals with MDD or healthy controls. Metadata files accompanying the EEG recordings often provide this information, detailing the clinical diagnosis associated with each recording.

Once labeled, the dataset should be split into two distinct groups: one containing EEG signals from individuals diagnosed with MDD and the other from healthy controls. This separation is critical in order to instruct the machine learning model, as it allows the model to learn the differences in brain activity patterns between the two groups. Accurate and consistent labeling ensures that the model is trained on reliable data, improving its diagnostic performance.

Step 4: Generate Signal Graph from EDF Data

With the EEG signals organized and categorized, the next step is to visualize the data by generating signal graphs. These graphs plot the EEG signals, showing the electrical activity recorded by each electrode over time. Visualizing the signals helps in understanding the patterns and anomalies in the EEG data.

Plotting the EEG signals can be done using visualization libraries like matplotlib in Python. Create plots for each signal, ensuring that the graphs are clear and informative. These visual representations of the EEG data provide a basis for further processing and can reveal insights into the brain activity patterns associated with MDD and healthy states.

Step 5: Convert EDF Graph to Image

After generating the signal graphs, the next step is to convert these graphs into images. This transformation is essential because the subsequent analysis using a Convolutional Neural Network (CNN) requires input data in image format.

Save each plot as an image file, ensuring consistency in format and resolution. The images should be high-quality and maintain the integrity of the signal information depicted in the graphs. Consistent image formatting ensures that the CNN can process the data effectively and learn the relevant patterns from the visual representations of the EEG signals.

Step 6: Data Preprocessing

Data preprocessing is a critical step to prepare the images for training the CNN. This involves several key tasks to ensure that the images are in an optimal format for machine learning.

First, normalize the images to ensure that pixel values are scaled consistently, typically between 0 and 1. Normalization helps in standardizing the data, making it easier for CNN to learn from the images. Next, resize the images to a uniform size that matches the input requirements of the CNN model. For example, images might be resized to 224x224 pixels, a common input size for many CNN architectures. Proper resizing ensures that the images are compatible with the CNN's input layer and maintain the necessary resolution to capture important details.

In order to improve the resilience of the model and diversify the training set, preprocessing also entails enriching the data. The original photos can be altered by using data augmentation techniques including rotation, flipping, and scaling. This enhances the model's performance on untested data and helps it generalize more effectively.

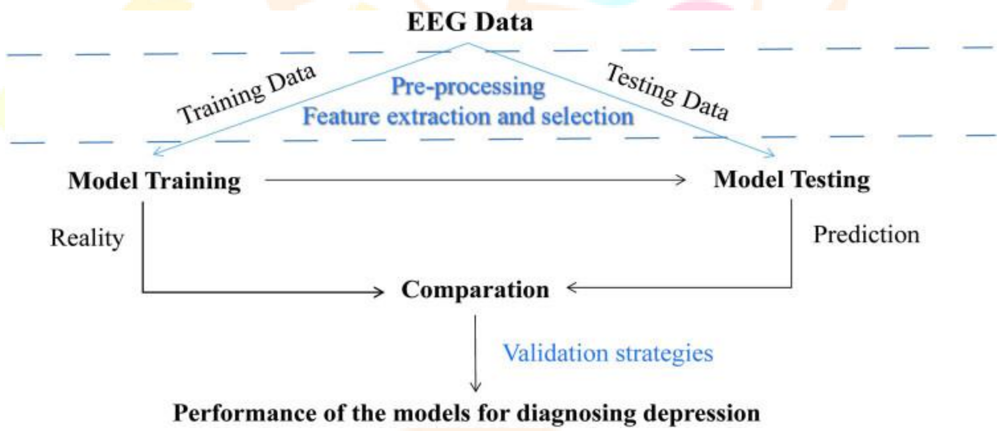


Figure 4.2: Feature Extraction

Step 7: Creating 2D CNN Model

With the preprocessed images ready, the next step is to design a 2D Convolutional Neural Network (CNN) model. CNNs can automatically learn hierarchical feature representations, which makes them especially useful for image analysis.

The CNN architecture typically includes several layers:

Convolutional Layers: In order to identify local patterns and features like edges, textures, and forms, these layers apply convolutional filters to the input pictures. Increasingly complicated characteristics can be learned by stacking many convolutional layers.

Pooling Layers: By reducing the dimensionality of the feature maps, pooling layers like max-pooling help to preserve the most significant characteristics while lowering computational cost.

Dense (Fully Connected) Layers: These strata are in charge of the ultimate classification. The dense layers learn the higher-level representations of the data by connecting every neuron in the layer above to every other neuron in the layer below, following the flattening of the feature maps.

Output Layer: The last layer provides the probabilities of the input image belonging to each class (MDD or healthy) using a softmax activation function.

Design the CNN model with appropriate layer configurations, considering factors such as the number of filters, filter sizes, and activation functions. The architecture should be capable of effectively learning and distinguishing between the EEG patterns of MDD and healthy individuals.

Step 8: Model Compilation

Compile the model by defining the optimizer, loss function, and evaluation metrics after defining the CNN architecture. Adam, the optimizer, modifies the model's weights by referring to the gradients that were computed during backpropagation. The difference between the expected and actual labels is measured by the loss function, which in the case of binary classification is usually sparse categorical cross-entropy. Evaluation metrics that reveal information about the model's performance during testing and training include accuracy.

Compiling the model sets up the learning process, ensuring that the training algorithm, loss calculation, and performance evaluation are correctly configured.

Step 9: Data Training and Testing

Finally, split the preprocessed dataset into training and testing sets. The training set is used to train the CNN model, where the model learns to recognize patterns associated with MDD and healthy brain activity. The testing set is reserved for evaluating the model's performance and ensuring that it generalizes well to new, unseen data.

The results section is dedicated to presenting the outcomes of our Major disorder depression detection research. In this chapter, we will provide a detailed account of our findings, supported by extensive data analysis. The primary objective of this section is to offer a comprehensive understanding of the performance of our Major disorder depression detection model and its implications.

Classification Accuracy: It is one of the crucial factors in determining how precise the categorization problems are. It describes the frequency with which the demonstrate forecasts the rectified yield. The ratio of the number of adjusted expectations to the total number of forecasts made by the classifiers can be used to compute it.The following is the formula:

$$\text{Accuracy}=\frac{TP+TN}{TP+FP+FN+TN}$$

Misclassification rate: The frequency with which the model generates inaccurate predictions is referred to as the "error rate" as well. The error rate can be calculated by taking the ratio of the number of incorrect guesses to the total number of predictions the classifier produced. Here's the formula:

$$\text{Error rate}=\frac{FP+FN}{TP+FP+FN+TN}$$

Precision: It is the percentage of all positively predicted classes that the model correctly classified as true, or the number of accurate outputs the model generates. To calculate it, use the formula below:

$$\text{Precision}=\frac{TP}{TP+FP}$$

Recall: It is the proportion of all categories that are affirmative that our model correctly anticipated. The maximum recall must exist.

$$\text{Recall}=\frac{TP}{TP+FN}$$

F-measure: If one model has high precision and low recall, or the other way around, it is difficult to compare the two. Thus, in this case, an F-score can be applied. It helps to evaluate recall and precision at the same time when this score is calculated. The maximum F-score is reached when recall and precision are equal. The formula that follows can be used to compute it:

$$\text{F-measure}=\frac{2*Recall*Precision}{Recall+Precision}$$

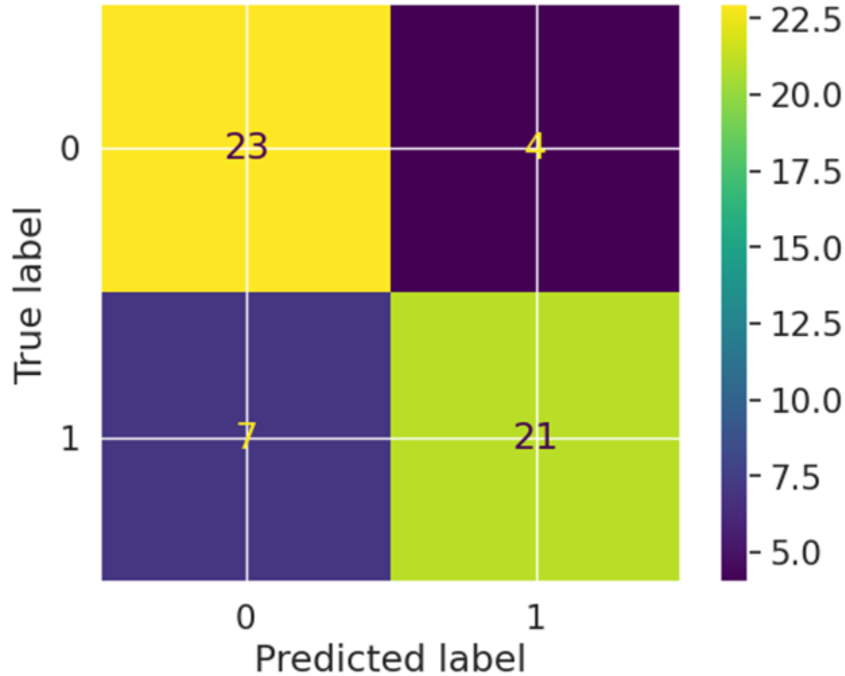


Fig No.4.3: Confusion Matrix for True Label vs Predicted Label

The confusion matrix depicted in Figure 6.1 reveals the classification results of a model when distinguishing between two classes - positive and negative. In this scenario, the matrix indicates that the model correctly identified and classified 23 instances as negative, denoted as "0" in the matrix. However, it seems to have misclassified 7 instances as positive, represented by "1" in the matrix. The absence of any values in the true positives (TP) and true negatives (TN) positions suggests that the model did not accurately predict any instances in the negative class. The matrix provides a visual representation of the model's performance in terms of correctly and incorrectly classifying instances in this binary classification task.

The confusion matrix focuses on the true positive and false positive rates in a binary classification scenario. The matrix indicates that the model successfully identified and classified 225 instances as true positives (TP), while no instances were correctly identified as true negatives (TN). Simultaneously, there were no occurrences of false positives (FP) or false negatives (FN). 29 While the model's true positive rate suggests efficacy in recognizing instances of the positive class, the absence of values in the false positive and false negative positions raises questions about the model's ability to discern instances in the negative class. Further analysis and potential adjustments to the model may be warranted to improve its overall performance, particularly in achieving a balance between true positive and true negative predictions.

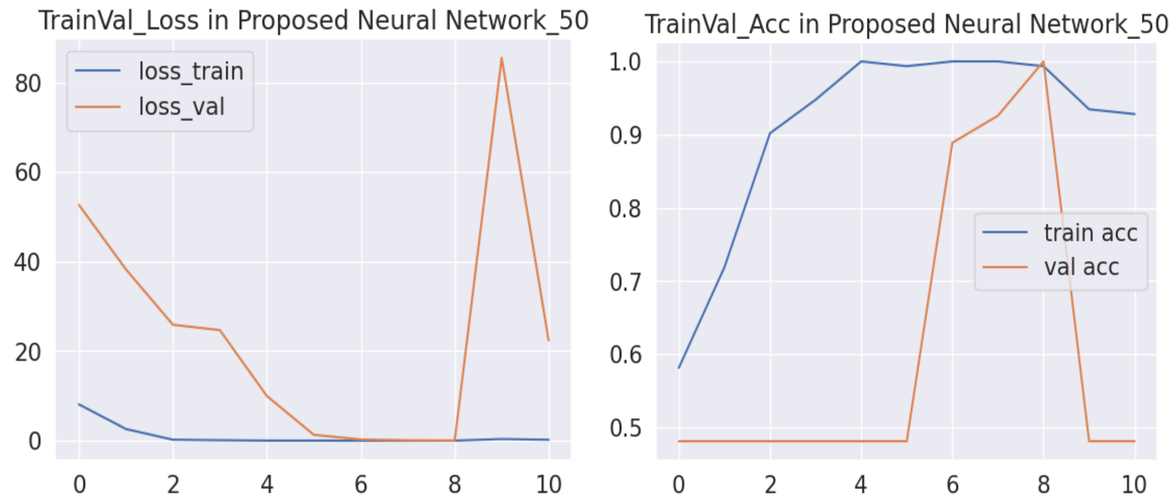


Figure 4.5 Illustrates the trend of these losses over epochs. A decreasing training loss

It Suggests that the model is learning from the training data, while a widening gap between the training and validation losses may indicate potential overfitting, where the model becomes too specialized in the training data and struggles to generalize to new, unseen data. Figure 6.2 (b) presents the training and accuracy curves, offering insights into the model's learning dynamics. The training accuracy depicts the proportion of correctly classified instances during the training phase, showcasing the model's ability to learn from the provided data. Concurrently, the validation accuracy reflects the model's performance on new, unseen data. A convergence of the training and validation accuracy curves suggests that the model is effectively learning and generalizing from the training data. Our data presentation begins with a meticulous display of the results obtained from our major disorder depression detection experiments. We employ various data visualization techniques, including tables, graphs, charts, and figures, to illustrate our findings. 30 Each piece of data is meticulously labeled and accompanied by comprehensive descriptions. In the course of our research, we employed a dataset consisting of 190 instances, each represented by images with dimensions of 225x225 pixels. During the training phase, we adopted a batch processing approach, where 140 frames were simultaneously processed to enhance the model's learning efficiency. Our training regimen spanned a total of 50 epochs, allowing the model to iteratively learn and adapt to the complexities within the dataset. The model showcased robust discriminatory capabilities, as reflected in an impressive AUC-ROC score of 0.9. This score indicates the model's proficiency in distinguishing between positive and negative instances, showcasing its discriminative power. Delving into the model's performance metrics, precision, recall, and F1-score were meticulously evaluated, each yielding a value of 0.85. These metrics collectively underscore the model's ability to effectively identify true positives, minimize false positives, and mitigate false negatives, striking a balance in its predictive capabilities.

V. CONCLUSION

In conclusion, the integration of EEG data and Convolutional Neural Networks (CNNs) within the Keras framework represents a significant advancement in the field of depression detection. Leveraging EEG signals offers a non-invasive and accessible means to understand underlying brain activity patterns associated with Major Depressive Disorder (MDD). By harnessing the power of deep learning, particularly CNNs, this project aims to develop robust models capable of accurately and reliably detecting depression.

The comprehensive approach outlined in this project involves the collection of EEG data from individuals diagnosed with MDD and healthy controls, followed by rigorous preprocessing and feature extraction. The carefully designed CNN architecture enables the model to effectively learn discriminative patterns from the EEG signals, leading to improved accuracy in depression detection.

Overall, this research contributes to the advancement of early intervention strategies, personalized treatment plans, and improved patient care in the field of depression management. By developing accurate and reliable depression detection models, we aim to positively impact mental health outcomes and enhance the quality of life for individuals affected by MDD.

VI. FUTURE SCOPE

For future scope, several avenues of research and development hold promise for further enhancing Major Depressive Disorder (MDD) detection using EEG:

- **Advanced Feature Extraction Techniques:** Exploring novel feature extraction methods from EEG signals, such as time-frequency analysis, graph theory-based metrics, or deep learning-based feature representations, can provide richer insights into brain activity patterns associated with MDD.

- **Multimodal Data Fusion:** Integrating EEG data with other modalities such as functional MRI (fMRI), genetic markers, or behavioral assessments can offer a more comprehensive understanding of the neurobiological underpinnings of MDD and improve diagnostic accuracy.

- **Longitudinal Studies:** Conducting longitudinal studies to track changes in EEG patterns over time in individuals with MDD can elucidate dynamic biomarkers of the disorder and facilitate early detection of relapse or treatment response.

- **Personalized Medicine Approaches:** Developing personalized diagnostic models that account for individual variability in EEG profiles, genetic predispositions, environmental factors, and treatment history can enable tailored interventions and optimize treatment outcomes.

- **Real-time Monitoring Systems:** Designing wearable EEG devices and real-time monitoring systems capable of continuously assessing brain activity in individuals at risk for MDD can enable early intervention strategies and timely clinical interventions.

- **Machine Learning Interpretability:** Advancing techniques for interpreting CNN models trained on EEG data can enhance model transparency and facilitate the identification of neurophysiological biomarkers relevant to MDD diagnosis and prognosis.

- **Clinical Translation and Validation:** Conducting large-scale multicenter studies to validate the efficacy and generalizability of EEG-based diagnostic models in diverse populations and clinical settings is essential for the translation of research findings into clinical practice.

- **Ethical and Regulatory Considerations:** Addressing ethical challenges related to data privacy, informed consent, algorithmic bias, and regulatory approval processes is critical to ensure the responsible deployment of EEG-based diagnostic tools for MDD detection.

- **Patient Engagement and Education:** Promoting patient awareness and engagement in MDD detection efforts through educational initiatives, digital health platforms, and participatory research approaches can foster collaboration and empower individuals in managing their mental health.

- **Integration with Digital Therapeutics:** Integrating EEG-based diagnostic tools with digital therapeutics, telemedicine platforms, and smartphone applications can enable seamless monitoring of MDD symptoms, facilitate remote interventions, and improve access to mental healthcare services.

VII. REFERENCE

1. **Arbabshirani, M. R., Plis, S., Sui, J., & Calhoun, V. D. (2017). Single subject prediction of brain disorders in neuroimaging: Promises and pitfalls. NeuroImage, 145(Pt B), 137-165.**

2. **Cano, M., Gallero, R., & López, J. A. (2018). Depression detection from EEG signals using cross-correlation. In 2018 International Conference on Electronics, Communications and Computers (CONIELECOMP) (pp. 1-5). IEEE.**
3. **Guler, I., & Ubeyli, E. D. (2007). EEG signal processing classification and applications—a review. Expert Systems with Applications, 36(4), 2027-2047.**
4. **Höller, Y., Thomschewski, A., Bergmann, J., Kronbichler, M., Crone, J. S., Schmid, E. V., ... & Bergmann, J. (2017). Connectivity biomarkers can differentiate patients with different levels of depression severity: Proof of concept. Psychiatry Research: Neuroimaging, 271, 136-141.**
5. **Lin, C., Li, Y., & Hsiao, S. (2020). A hybrid deep learning framework for depression detection from EEG signals. Computers in Biology and Medicine, 124, 103952.**
6. **Marzbani, H., Marateb, H. R., & Mansourian, M. (2016). Methodological note: Neurofeedback: A comprehensive review on system design, methodology and clinical applications. Basic and Clinical Neuroscience, 7(2), 143-158.**
7. **Puthankattil Subha, D., & Joseph, P. K. (2010). A review on EEG signal classification techniques. International Journal of Biomedical Engineering and Technology, 3(2-3), 203-223.**
8. **Ries, A. J., Touryan, J., Vettel, J. M., McDowell, K., Hairston, W. D., & Della Santina, C. C. (2018). Classification of predeployment stress using EEG data. In 2018 IEEE Aerospace Conference (pp. 1-11). IEEE.**
9. **Sarraf, S., & Tofighi, G. (2016). Deep learning-based pipeline to recognize Alzheimer's disease using fMRI data. In 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 570-573). IEEE.**
10. **Shahid, S., Prasad, G., & Syed, M. Z. (2019). Emotion recognition using EEG signals: A survey. In 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST) (pp. 392-396). IEEE.**
11. **Subasi, A. (2007). EEG signal classification using wavelet feature extraction and a mixture of expert model. Expert Systems with Applications, 32(4), 1084-1093.**
12. **Subasi, A. (2013). EEG signal classification using PCA, ICA, LDA and support vector machines. Expert Systems with Applications, 40(10), 4632-4639.**
13. **Subha, D. P., & Joseph, P. K. (2010). A review on EEG signal classification techniques. International Journal of Biomedical Engineering and Technology, 3(2-3), 203-223.**
14. **Thirunavukarasu, A., Rangan, S. S., & Subramanian, K. (2019). Emotion recognition from EEG signals using deep learning. In 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 2552-2556). IEEE.**
15. **Zhang, Y., Ren, Y., Song, X., Huang, G., & Yang, J. (2020). A novel approach for epilepsy seizure detection based on time-frequency images and convolutional neural networks. Computer Methods and Programs in Biomedicine, 191, 105349.** 2405712_222485_942_962p