Project Report

On

Schizophrenia Detection & Image Classification using EEG Signals

Submitted

In partial fulfilment

For the award of the Degree of

Post Graduate Diploma in Artificial Intelligence

(C-DAC, ACTS (Pune))

Guided By: Smt. Dr. Priyanka Jain, Associate Director

Artificial Intelligence Group

C-DAC , Delhi

Submitted By:

SIbadatta Sasmal - 210540184033

Mittala Nandle - 210540184018

Apurv Kulkarni - 210540184017

Pranav Amlekar - 210540184007

Bhamidi Srujana - 210540184010

Centre of Development of Advanced Computing (C-DAC),ACTS

(Pune- 411008)

Acknowledgement

This is to acknowledge our indebtedness to our Project Guide, **Dr. Priyanka Jain.** C-DAC ACTS, Pune for her constant guidance and helpful suggestions for preparing this project **Schizophrenia Detection & Image Classification using EEG Signals.** We express our deep gratitude towards her for her inspiration, personal involvement, constructive criticism that she provided us along with technical guidance during the course of this project.

We take this opportunity to thank Head of the department **Mr. Gaur Sunder** for providing us such a great infrastructure and environment for our overall development.

We express sincere thanks to **Ms. Risha P.R.,** Program Head, and **Mr. Kaushal Sharma**, Process Owner, for their kind cooperation and extendible support towards the completion of our project.

It is our great pleasure in expressing sincere and deep gratitude towards **Mrs. Priyanka Ranade** (Course Coordinator, PG-DAI) for her valuable guidance and constant support throughout this work and help to pursue additional studies.

Also our warm thanks to **C-DAC ACTS Pune**, which provided us this opportunity to carry out this prestigious Project and enhance our learning in various technical fields.

Apurv Kulkarni (210540184017)

Mittala Nandle (210540184018)

Pranav Amlekar (210540184007)

Bhamidi Srujana (210540184010)

Sibadatta Sasmal (210540184033)

ABSTRACT

"Healthy body lives in a healthy mind."

Mental health in our Fastrack lives is neglected and not talked about. People tend to shy away from it and suffer. This becomes the root cause of many problems. This project is a step towards making a difference, to bring change and to cater a mental illness named Schizophrenia. Its early detection can save a person's life and can enjoy it.

This project is divided into two parts i.e., Classification of EEG signals for schizophrenia patients and healthy individuals and second part includes image label classification using EEG signals and classification of spectrograms of the above EEG signals for the above-mentioned labels.

First part contains 45 patients with schizophrenia syndromes and 39 healthy subjects are studied with electroencephalogram (EEG) signals. For each of the two groups, the signals were analyzed using 16 EEG channels. The data were compared for two groups of subjects. The results will be useful for further development of EEG signal classification algorithms for machine learning. This study shows that EEG signals can be an effective tool for classifying participants with symptoms of schizophrenia and control group. It is suggested that this analysis may be an additional tool to help psychiatrists diagnose patients with schizophrenia.

Second part of this project includes developing the visual object classifier driven by human brain signals. In particular, we employ EEG data from five different channels evoked by visual object stimuli combined with different Machine Learning (ML) models, Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) to learn a discriminative brain activity manifold of visual categories in an effort to read the mind.

Table of Contents

Chapter 1
Introduction

1.1 Introduction

For Schizophrenia is associated with disorders in the lobes and areas of the brain, which are responsible for information processing, temporary memory and executive functions. The diagnosis of schizophrenic spectrum disorders and other psychotic disorders is challenging. The scientific community is constantly working to integrate the latest clinical and scientific advances in the field of psychiatry into diagnostic and statistical manuals. However, quantifying and evaluating abnormalities in the cerebral cortex can help to understand the mechanisms of such psychotic disorders. Recent advances in the area of analysis of complexity of time series provide insights into nonlinear electroencephalogram (EEG) signals.

It is suggested that nonlinear EEG analysis can be a useful tool in the analysis of EEG data for studying the neurodynamic of the brain of patients with schizophrenia.

The advancements in sensor technologies have facilitated the growth of wearable headband devices for development in Brain-Computer Interface (BCI) applications. In second part of this project, we are studying MindBigData "IMAGENET" of The Brain, open Data Base contains brain signals of 3 seconds each, captured using commercial EEG with emotive headset covering five brain location. In this work, we analyzed the relationship between EEG signals, channels and spectrograms with image labels. The objective was to extract the information (features) from EEG signals and spectrograms to classify or distinguish them based on the images that were used to stimulate brain activity.

1.2 Objective and specification:

The program works towards the objective of classification of schizophrenia detection using brain signals. Using the model we thus created to detect whether any person is having schizophrenia or not.

Second goal of this project is to explore different approaches to classify EEG signals from MindBigData "IMAGENET" dataset which is composed of records of brain signals of 3 seconds each, captured with the stimulus of seeing a random image (14012 so far) from the ImageNet ILSVRC2013 train dataset and thinking about it, and tried to focus on the different images for about three seconds. The goal is classifying EEG signals into 9 image categories.

The different processes which can be used in this project are:

- Studying EEG signals
- Different approaches for classification
- Schizophrenia Detection
- Feature extraction and classification based on spectrogram images

Chapter 2

Methodology and Techniques

2.1 Approach & Methodology/Techniques:

The approach for Schizophrenia detection and EEG classification, we needed raw data to train the model for classification and detection.

2.2 Dataset Preparation

2.2.1 Schizophrenia detection

Two EEG data archives were analyzed for two groups of subjects [http://brain.bio.msu.ru/eeg_schizophrenia.htm]. The subjects of the survey were adolescents who were tested by a psychiatrist and divided into two groups: healthy (n = 39) and with symptoms of schizophrenia (n = 45). Each file contains an EEG record for one subject. Each TXT file contains a column with EEG samples from 16 EEG channels, according to Fig.1. Signals were recorded by channels: 'F7', 'F3', 'F4', 'F8', 'T3', 'C3', 'Cz', 'C4', 'T4', 'T5', 'P3', ' Pz ',' P4 ',' T6 ',' O1 ',' O2 '. Each number in the column represented the EEG amplitude (μV) on a separate sample. The first 7680 samples represent 1 channel, then 7680 - channel 2, etc. The sampling rate is 128 Hz, so 7680 samples correspond to 1 minute of EEG recording.

```
inpDir = '/content/drive/MyDrive/Schizophrenia'
os.chdir(inpDir+"/sch")
lst = ['F7','F3','F4','F8','T3','C3','Cz','C4','T4','T5','P3','Pz','P4','T6','O1','O2']

iteration = 1
buffer = 7680

df_empty =  pd.DataFrame(columns=lst)
df_concatenated = pd.DataFrame()

all_files = glob.glob("*.eea")
```

Fig 2.1 Combining EEA files to dataset

2.2.2 EEG Classification

Dataset contains in total 70,060 brain signals captured in intervals of 3 seconds each, with the brain stimulus of seeing a random image (14,012 so far), saved in a csv file with a unique name representing image category and session id.

Each file contains signals from 5 channels (AF3, AF4, T7, T8, PZ). We combined all csv files and reduced the number of classes from 569 to 29 and then to 9 on the basis of the type of images and information provided in WordReport-v104.txt. We performed Exploratory Data Analysis (EDA) and experimented with different filters to reduce noise from signals.

	name	repetition	target	category
0	domestic cat, house cat, Felis domesticus, Fel...	24	n02121808	Mammals
1	rock python, rock snake, Python sebae	29	n01744401	Reptile

Fig.2.2 WordReport-v104.txt

EDA steps followed are: -

1. Checked the shape
2. Checked and fill null values
3. Retained columns with at least than 90% data
4. Checked data types of columns
5. dropped unnecessary columns
6. Class density reduction
7. Multivariate analysis (correlation matrix)
8. Univariate analysis (signals after and before filtering)

Content of Data Frame:

	category	global_session_id	number_of_sessions	image_id	image_category	channels	t_1	t_2	t_3	t_4	t_5
0	Mammals	2602	1	4823	n02077923	AF3	4320.000000	4332.307692	4334.871795	4342.564103	4333.333333
1	Mammals	2602	1	4823	n02077923	AF4	4294.871795	4305.128205	4298.974359	4307.179487	4312.307692
2	Mammals	2602	1	4823	n02077923	T7	4187.692308	4200.000000	4208.717949	4199.487179	4199.487179
3	Mammals	2602	1	4823	n02077923	T8	4203.076923	4224.615385	4249.743590	4231.794872	4190.769231
4	Mammals	2602	1	4823	n02077923	Pz	4145.128205	4138.461538	4145.128205	4142.051282	4132.307692
...	...	...	...	...	...	...	...	...	...	...	...
70055	Mammals	144	1	4942	n02105162	AF3	4309.230769	4314.871795	4320.000000	4314.358974	4317.435897
70056	Mammals	144	1	4942	n02105162	AF4	4290.769231	4291.282051	4286.153846	4283.589744	4290.256410
70057	Mammals	144	1	4942	n02105162	T7	4322.051282	4313.333333	4299.487179	4297.948718	4303.589744
70058	Mammals	144	1	4942	n02105162	T8	4268.205128	4269.230769	4270.769231	4272.307692	4280.512821
70059	Mammals	144	1	4942	n02105162	Pz	4254.358974	4263.076923	4261.025641	4262.051282	4274.871795

70060 rows × 450 columns

Fig.2.3 EEG Signal Data Frame

	path	target	category	3D
3652	MindBigData_Imagenet_Insight_n03109150_3341_1_...	n03109150	Home_Appliances	[[[6.0, 9.0, 9.0], [50.0, 44.0, 14.0], [20.0, ...
971	MindBigData_Imagenet_Insight_n10148035_13340_1...	n10148035	Human	[[[10.0, 12.0, 7.0], [31.0, 14.0, 1.0], [2.0, ...
7366	MindBigData_Imagenet_Insight_n02088094_272_1_1...	n02088094	Animals	[[[3.0, 19.0, 11.0], [42.0, 0.0, 0.0], [13.0, ...
67	MindBigData_Imagenet_Insight_n04379243_21890_1...	n04379243	Home_Appliances	[[[5.0, 6.0, 29.0], [41.0, 48.0, 4.0], [31.0, ...
8805	MindBigData_Imagenet_Insight_n02007558_6509_1_...	n02007558	Animals	[[[1.0, 177.0, 12.0], [25.0, 175.0, 9.0], [8.0

Fig.2.4 Spectrogram Data Frame

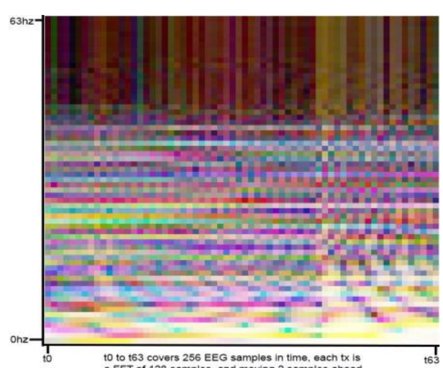

Fig.2.5 Spectrogram of signals

2.2.1 Data Visualization

Part I - Schizophrenia detection

<matplotlib.axes._subplots.AxesSubplot at 0x7f3e61691b10>

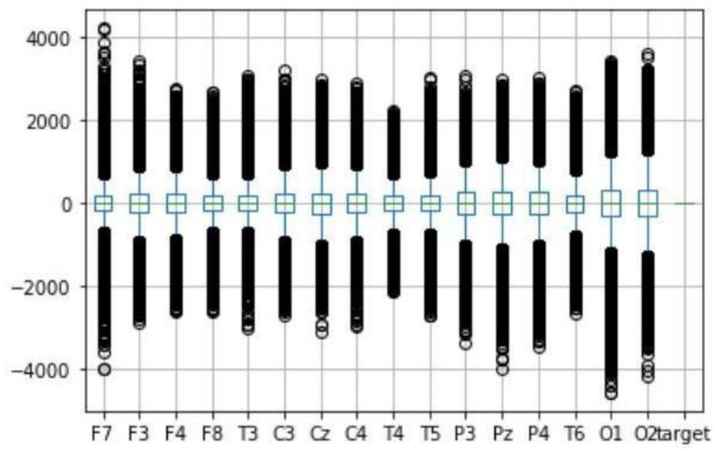

Fig 2.5 Dataset visualization Boxplot

<matplotlib.axes._subplots.AxesSubplot at 0x7f3e6b813a50>

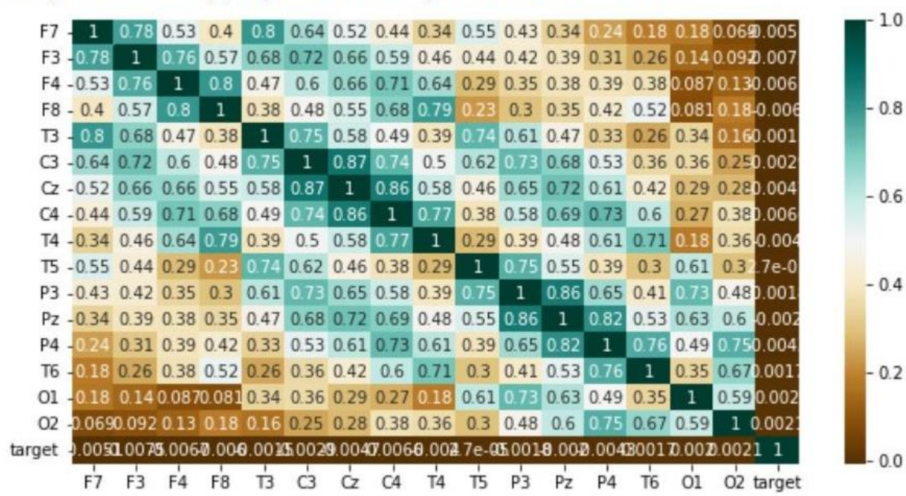

Fig 2.3 Dataset Correlation

Part-2 EEG Signals

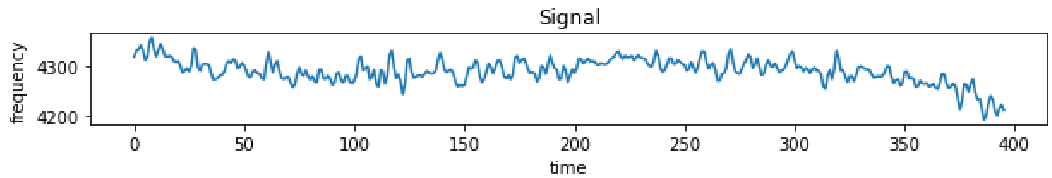

Fig.2.6 Signal before filtering

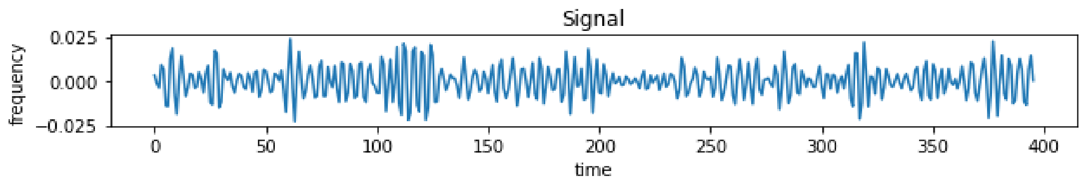

Fig.2.7 Signal after filtering

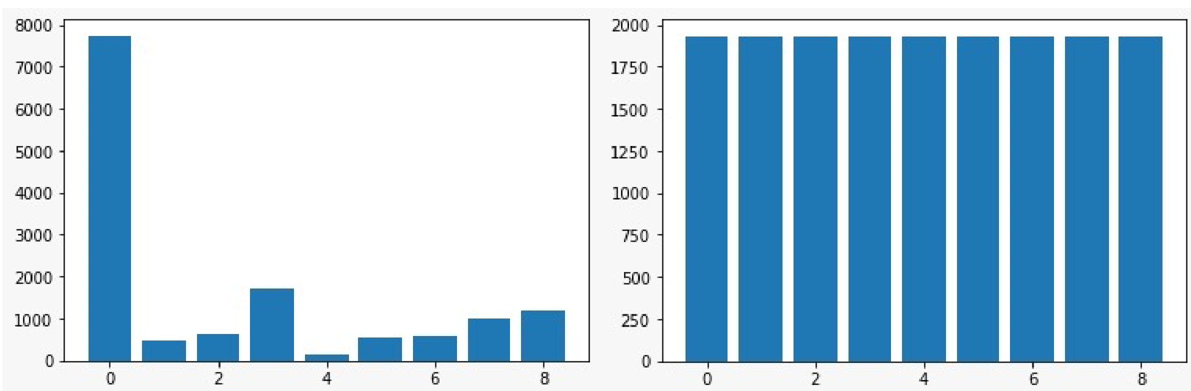

Fig.2.8 Class density before and after resampling

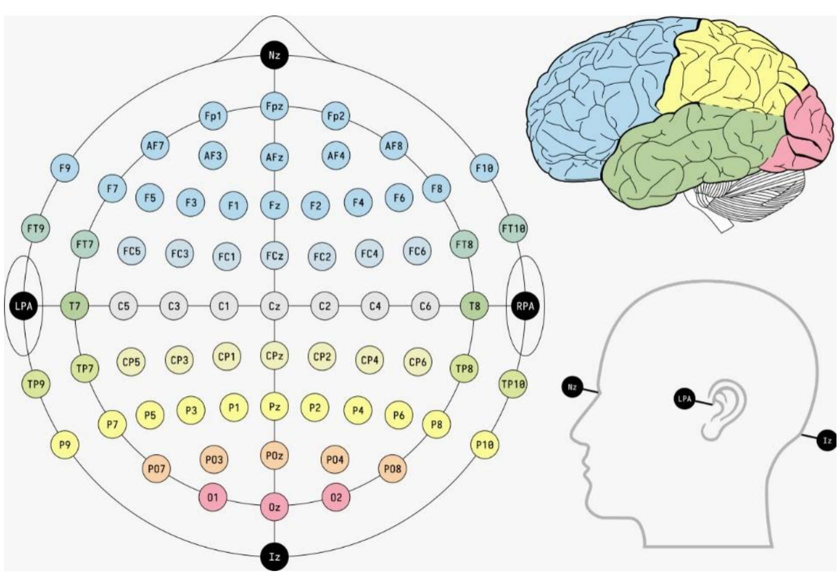

Fig.2.9 EEG signal location

2.3 Model Description

Part-I - Schizophrenia detection

```
parameters = {'max_features': np.arange(1,7)}

model_rf = RandomForestClassifier(random_state=2021)
cv = GridSearchCV(model_rf, param_grid=parameters,
                  cv=kfold,scoring='roc_auc')

cv.fit( X , y )

results_df = pd.DataFrame(cv.cv_results_  )

print(cv.best_params_)

print(cv.best_score_)

print(cv.best_estimator_)
```

```
{'max_features': 6}
0.8408496060421211
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features=6,
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=None, oob_score=False, random_state=2021,
                       verbose=0, warm_start=False)
```

Fig 2.10 RandomForestClaassifier on Schizophrenia Dataset

Part 2 – EEG Classification

The model we used is built with Keras using **Convolutional Neural Networks (CNN)**. A convolutional neural network is a special type of deep neural network which performs well for EEG classification purposes. A CNN basically consists of an input layer, an output layer and multiple hidden layers. In our CNN model there many convolution operations on 1D matrix on different layers.

```
def make_model(input_shape):

    input_layer = Input(input_shape)

    conv1 = Conv1D(filters=64, kernel_size=3, padding="same")(input_layer)
    conv1 =BatchNormalization()(conv1)
    conv1 =Dropout(0.3)(conv1)  #extra
    conv1 =ReLU()(conv1)

    conv2 =Conv1D(filters=64, kernel_size=3, padding="same")(conv1)
    conv2 = BatchNormalization()(conv2)
    conv2 =Dropout(0.3)(conv2)  #extra
    conv2 = ReLU()(conv2)

    conv3 = Conv1D(filters=64, kernel_size=3, padding="same")(conv2)
    conv3 = BatchNormalization()(conv3)
    conv3 = Dropout(0.3)(conv3)  #extra
    conv3 =ReLU()(conv3)

    gap = GlobalAveragePooling1D()(conv3)

    output_layer = Dense(num_labels, activation="softmax")(gap)

    model = keras.Model(inputs=input_layer, outputs=output_layer)

    return model
```

Fig.2.11 CNN model on EEG signals

```
model2 = Sequential()
model2.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(64,64,3)))
model2.add(BatchNormalization())

model2.add(Conv2D(32, kernel_size=(3, 3), activation='relu'))
model2.add(BatchNormalization())
model2.add(MaxPooling2D(pool_size=(2, 2)))
model2.add(Dropout(0.25))

model2.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model2.add(BatchNormalization())
model2.add(Dropout(0.25))

model2.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model2.add(BatchNormalization())
model2.add(MaxPooling2D(pool_size=(2, 2)))
model2.add(Dropout(0.25))

model2.add(Flatten())

model2.add(Dense(512, activation='relu'))
model2.add(BatchNormalization())
model2.add(Dropout(0.5))

model2.add(Dense(128, activation='relu'))
model2.add(BatchNormalization())
model2.add(Dropout(0.5))

model2.add(Dense(9, activation='softmax'))
```

Fig.2.12 CNN model on EEG Spectrograms

Chapter 3

Implementation

3.1 Implementation

1. Use of Python Platform for writing the code with Keras, TensorFlow, SciPy, Sklearn.

2. Hardware and Software Configuration:

Hardware Configuration:

- CPU: 8 GB RAM, Quad core processor
- GPU: 12GB NVIDIA TeslaK80 GPU (Google Colab)
- Software Required:

- **Google Colab**: Colab is a free Jupyter notebook environment that runs entirely in the cloud. It does not require a setup and the notebooks that you create can be simultaneously edited by multiple users. Colab supports many popular machine learning libraries which can be easily loaded in your notebook.

3.2 Algorithm flow

Part 1: EEG of healthy adolescents and adolescents with symptoms of schizophrenia

Step 1: There are two EEG data archives for two groups of subjects. The subjects were adolescents who had been screened by a psychiatrist and divided into two groups: healthy (n = 39) and with symptoms of schizophrenia (n = 45).

Step 2: Each file contains an EEG record for one subject. First 7680 samples represent 1st channel, then 7680 - 2nd channel, etc. The sampling rate is 128 Hz, thus 7680 samples refer to 1 minute of EEG record.

Step 3: Made 2 csv’s from two EEG data archives and combined them by appending the target column to them in a single csv file.

Step 4: Perform Exploratory Data Analysis (EDA) on the whole dataset. Use matplotlib library for visualization of data features, Plotted histogram and boxplot to analyze properties of features.

Step 5: Applied appropriate filters. Used SciPy.signal.Butter() function for noise reduction in signals, using high pass and low pass filtering methods
.

Step 6: First we divide the data into features data frame and target data frame and then split data using StratifiedKFold technique into 5 splits by shuffling it
.
Step 7: Developed different Machine learning and Deep Learning models. Among these all models including ML and DNN the good performance comes with Random Forest (ML). The Random Forest model is trained for hyperparameter max_features 1 to 7 and gives best accuracy of 0.84 for max_features = 6

Part 2: MindBigData ImageNet

Part 2.1: CNN for EEG Signals.

Step 1: Merged all csv data files. MindBigData dataset contain 140072 files which have five signals from five channels we merged into one csv file.
named merged.csv It contains 70060 rows and 450 columns.

Step 2: Merged categories with merged.csv file on the basis of data from WordReport-v1.04.

Step 3: Perform Exploratory Data Analysis (EDA) on the whole dataset. Use matplotlib library for visualization of data features, Plotted histogram and pie charts to analyze properties of categorical features and signals.

Step 4: Applied appropriate filters. Used scipy.signal.butter() function for noise reduction in signals, using high pass and low pass filtering methods.

Step 5: Splitting data and preparing data for the final model. First, we shuffled the data frame to remove bias in classification and Splitted data using from sklearn.model_selection import train_test_split where test size is 30% of the data frame and remaining for the train dataset.

Step 6: Developed different Machine learning (ML) and Deep Learning models (DNN). Among these all models including ML and DNN the good performance comes with Convolutional Neural Network (CNN).

The CNN model contains five layers which accept multivariate time series signals in nd array format.

1. input layer which accepts data - input_layer = **Input(input_shape)**

2. Three Convolutional 1D layers with batch normalization and dropout layer. - **conv1 = Conv1D (filters=64, kernel_size=3, padding="same")(input_layer)**

3. GlobalAverage Pooling layer 1D layer- gap = **GlobalAveragePooling1D()(conv3)**

4. Output layer with softmax activation function - **output_layer = Dense(num_labels, activation="softmax")(gap)**

Step 7: Compiled with Adam optimizer and fit the model on the test dataset.

Step 8: Evaluated performance of model using train, test accuracies and loss.

Part 2.2: CNN for Spectrogram Images

Step 1: Read all spectrogram image files present in the folder, and stored them in a list

Step 2: Passed it through a custom function to extract the target.

Step 3: Inner joined with WordReport-v104.txt to get the actual target.

Step 4: Reduced the labels from 29 to 9 via custom function.

Step 5: Shuffled the data to reduce bias.

Step 6: Convert images to ndarray(N-dimensional array) and stored in the data frame

Step 7: Divided the independent column and dependent column into X & Y respectively. Categorically label encoded the dependent column. And resampled

Step 8: Split the data into train and test with 80% in train and remaining in test.

Step 9: created a data augmentation pipeline and generated an augmented dataset.

Step 10: Build two CNN models with early stopping. First passed the normal data and then passed the augmented data. Loss and accuracy were noted down.

Chapter 4

Results

4.1.1 Schizophrenia Dataset Accuracies

Model	Accuracy
Logistic Regression	51
Gaussian Naive Bayes	57
Random Forest	84
Logistic Regression with Low Pass Filter	51
Gaussian Naive Bayes with Low Pass Filter	50
Random Forest with Low Pass Filter	83
Logistic Regression with High Pass Filter	51
Gaussian Naive Bayes with High Pass Filter	50.7
Random Forest with High Pass Filter	83
Logistic Regression with PCA	53
Gaussian Naive Bayes with PCA	52
Random Forest with PCA	75
DNN	53.57
DNN with Low Pass Filter	53.57
DNN with High Pass Filter	53.57

4.1.2 CNN of EEG signals

DNN	**Train accuracy** 92.60 **Test accuracy** 70.90
LSTM	**Train accuracy** 98.27 **Test accuracy** 71.57
CNN	**Train accuracy** 90.72 **Test accuracy** 69.90
Logistic	15.57
Bagging default	73.24
Bagging with logistic	15.98
Decision tree	12.63
Decision tree tuning	13.65
XGBoost	32.76

4.1.3 CNN for Spectrogram image results

Model	accuracy	val_accuracy
Model1-w/o augmentation	**11.5**	**10.23**
Model1-with augmentation	**11.75**	**11.39**
Model2-w/o augmentation	**97**	**74**
Model2-with augmentation	**14.22**	**18.62**

4.1.2 Output Graphs of accuracy and loss for training and validation

- CNN for classifying EEG signals

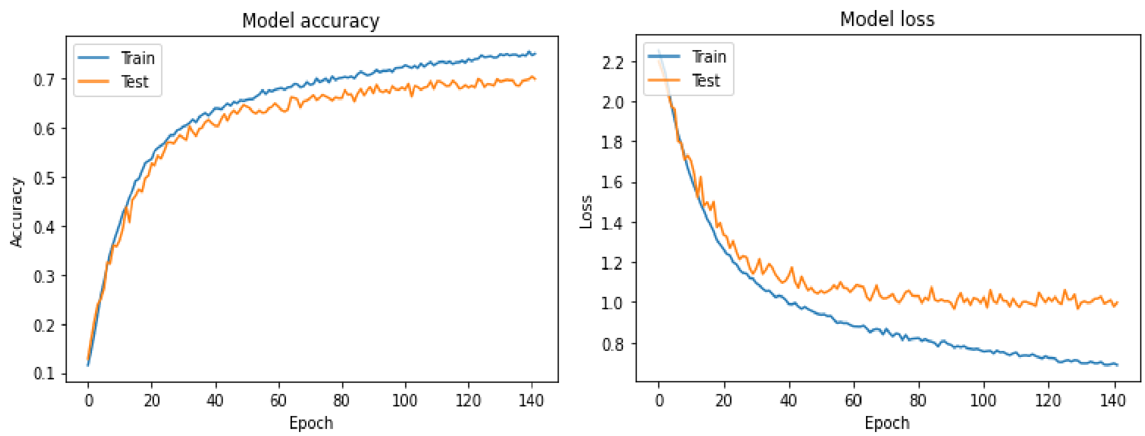

Fig. 2.12 Accuracy and loss graph for CNN

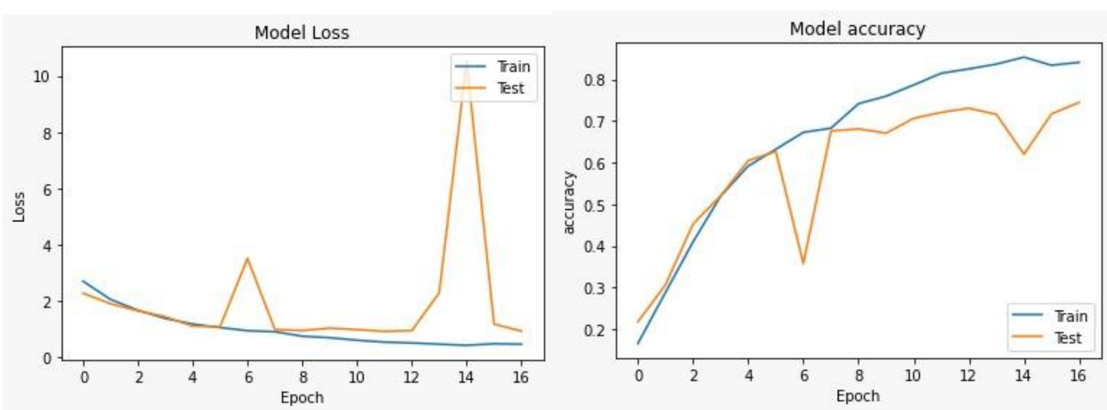

Fig. 2.13 Accuracy and loss for CNN Spectrogram dataset

The Epochs run were 500 with early stopping.

We were able to achieve the accuracy of 90% on the Train dataset and 69% on validation dataset.

The training and Validation Loss were comparable.

Chapter 5

Conclusion

5.1 Conclusion.

- We could detect whether a patient is having schizophrenia using the brain EEG signals. The Machine Learning model of Random Forest is giving the best accuracy for the prediction.
- We can use CNN effectively for classification of EEG signals generated by seeing the random images. We can further improve performance of the model by tweaking number of epochs, layers, resampling of the data, noise reduction of the signals.
- Need more Spectrogram images to achieve better result. Hyperparameter tuning and tweaks in the CNN model can help us read spectrogram with more accuracy and take us a step forward in predicting the correct label.

Chapter 6

References

References:

Dataset 1: Schizophrenia Database
Dataset 2: MindBigData "IMAGENET" of The Brain
Refrence_1:Developing a Data Visualization Tool for the Evaluation Process of a Graphical User Authentication System
Refrence_2: Object classification from randomized EEG trials