Integration of Continuous Wavelet Transform and Convolutional Neural Network for multiclass EEG dataset classification

Author

SIDRA NASEEM

00000204793

Supervisor

DR KASHIF JAVED

Department of Robotics and Intelligent Machine Engineering

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

June 2021

Integration of Continuous Wavelet Transform and Convolutional Neural Network for EEG dataset classification

00000204793

A thesis submitted in partial fulfillment of the requirements for the degree of

MS ROBOTICS AND INTELLIGENT MACHINE ENGINEERING

Thesis Supervisor:

DR KASHIF JAVED

Thesis Supervisor's Signature: ____

Department of Robotics and Intelligent Machine Engineering

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY,

ISLAMABAD

June 2021

THESIS ACCEPTANCE CERTIFICATE

It is certified that final copy of MS/MPhil thesis written by Ms. Sidra Naseem (Registration No. 00000204793), of SMME (National University of science and Technology) has been vetted by undersigned, found complete in all respects as per NUST Statues/Regulations/MS Policy, is free of plagiarism, errors, and mistakes and is accepted as partial fulfillment for award of MS/MPhil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in this dissertation.

Signature of Supervisor:______________________

Name: Dr. Kashif Javed

Date: ____________________________________

Signature of HoD:____________________________

Date: ____________________________________

Signature (Dean/Principal): ___________________

Date: ____________________________________

Declaration

I certify that this research work titled "*Integration of Continuous Wavelet Transform and Convolutional Neural Network for multiclass EEG dataset classification"* is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

Signature of Student

SIDRA NASEEM

2017-NUST-MS-RIME-00000204793

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Signature of Student

SIDRA NASEEM

2017-NUST-MS-RIME-00000204793

Signature of Supervisor

DR KASHIF JAVED

Copyright Statement

Acknowledgements

Above all, I want to thank Allah Subhana-Wata'ala for the provision of knowledge, resources, health, and the strength to complete this work. I am grateful for all the support I received from other people by HIS will. Alhamdulillah!

I would like to express my gratitude to my primary supervisor, Dr Kashif Javed for his valuable guidance, constructive feedbacks and comments from the very beginning till the completion of this dissertation and research work. I am sincerely grateful for his patience and tremendous support during my thesis. I appreciate his thoughtfulness and encouragement throughout his supervision.

I would like to extend my gratitude to my Co-supervisor Dr Muhammad Jawad Khan for his consistent support and feedback to enhance the quality of this work. I appreciate his mentoring, cooperation, and comprehensive guidance throughout this research work. I am thankful for the insights he provided during my thesis.

I would like to express my appreciation to Dr Muhammad Hasan Sajid, Head of Department, and a great teacher, who has inspired me with his knowledge, strong research enthusiasm and hard-working attitude. His insights on Machine Learning will have a profound effect on my academic career forever.

Finally, I would like to express my deepest gratitude to my parents for their profound support and unconditional love throughout my studies regardless of the circumstances. My sincere thanks to my siblings for their support through all the good and bad times.

My deepest gratitude and appreciation go to everyone who helped me one way or another with my work.

JazakAllah!

Dedicated to my beloved brother Muhammad Ali(late). May Allah(SWT) forgive him and bless him with eternal peace.

Abstract

This thesis presents a novel method for EEG classification in time-frequency domain using deep learning architecture. Existing deep learning architectures suffer from poor performance when classifying EEG data in Time-frequency domain. The proposed method seeks to improve classification process and provide better accuracy and loss than previously has been achieved.

The Continuous Wavelet Transform is used to convert brainwaves into time-frequency domain and then Convolutional Neural Network is used for feature learning and classification of EEG data. The results have been cross-validated by Kfold cross validation and Leave-One-Out Cross Validation(LOOCV). The proposed method has also been compared with VGG16, Google Net, AlexNet models. This model produces results on publicly available dataset: Epilepsy dataset by UCI(Machine Learning Repository)

Key Words: *Deep Learning, 3D CNN, Electroencephalography, Epilepsy, Continuous Wavelet Transform,*

Table of Contents

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

Electroencephalography is an electrophysiological method to record neural activity generated by brain neurons(cf. ***Figure 1***). Electrodes are used to perform this method and record all brainwave patterns. These brain signals are helpful to analyze brain performance in different health conditions because EEGs are formed of brainwaves caused by emotional changes, motor movements and motor movement imagery, brain tumor, epileptic seizer, and many other systems of human body[70].

Electroencephalography has been an important and well-researched topic over past few years since it plays significant role in the diagnosis of neural abnormalities. Moreover, these brain signals have been of great interest for Brain computer Interfaces which are used to facilitate People who are suffering from Locked-In syndrome, paraplegia, and quadriplegia[44][47]. Because EEG has huge impact on human life, it is imperative to design reliable classification algorithms which are cost efficient and has better diagnostic accuracy.

Currently, most of EEG analysis is performed manually where the experts of neuroscience monitor and interpret brain signals to identify neural abnormalities. The neurologists perform visual inspection on EEG recordings based on their knowledge of normal and abnormal EEG patterns. For accurate clinical interpretation of EEGs through visual inspection, the neurologists should have complete understanding of EEG characteristics recorded from people of different age groups with diverse health conditions.

Since it is difficult and time consuming to analyze EEG signals manually and establish conclusions, various statistical and machine learning techniques has been proposed and widely used to analyze and classify EEG signals. There are many transformation techniques such as Fourier Transform, Hilbert Huang Transform and Continuous Wavelet Transform, which have been used for analysis and interpretation of brain signals and detect any abnormalities by comparing these signals with standard EEGs. Continuous wavelet transform provides enhanced and detailed analysis of signals in time-frequency domain.

Classification of EEG signals is greatly advantageous in neural disease predictions and brain-computer interfaces. Therefore, many machine learning techniques have been used for EEG classification e.g., SVM, KNN, and Neural networks etc. These classification techniques are useful only if accurate feature derivation and selection is performed beforehand. However, this limitation can be avoided with the help of Deep Learning Models which do not require prior feature engineering.

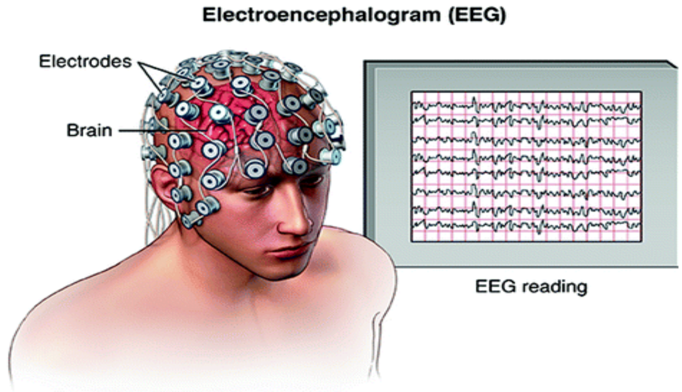


Figure 1. illustration of EEG recording Process (Siuly et al., 2017)

Epilepsy is a recurring neural disease which can cause brain dysfunction due to unexpected occurrences of seizures. These Seizures can result in loss of consciousness, limb Tremors, behavioral disorders, and transient sensory disorders etc., due to mental shocks caused by Seizures[71]. Brain tumors are also a major cause of Seizures, limb numbness and behavioral disorders[72]. EEG can be a reliable source for an early detection of Seizures due to Epilepsy or Brain Tumors but, it tends to be contaminated by various physiological activities. Therefore, it is imperative to design an efficient method which can separate useful information from noisy characteristics of EEG.

The application of Deep Learning in Neuroscience has been of great help in detection of neural abnormalities because DL models can recognize complex EEG patterns without predefined features. In this thesis, we combined two different techniques to classify Epilepsy dataset[3] which involves signal transformation and deep learning. The key idea is to use time-frequency coefficients of EEG signals to train the Convolutional Neural Network which classifies EEG signals into different classes based on their distinctive features. Our goal is to use generic feature learning property of deep learning to design a classification method that is fast, robust, and more accurate than the existing classification methods for EEG dataset.

1.1 Scope and Motivation

The proposed technique for EEG classification has a huge scope. It has the potential to revolutionize the EEG-based applications for the decade to come. Accurate EEG classification is a fundamental to facilitate people suffering from neural abnormalities and to avoid any accidents due to inaccurate classification in BCIs. Our motivation is to improve lifestyle and health standards by providing better EEG interpretation and reliable classification accuracy. Based on the results that we have achieved in this work, it safe to say that we have been successful in fulfilling our goal and we are hopeful that we will be able to take our work even further in the days to come.

1.2 Problem Statement

"To improve classification accuracy of Epilepsy dataset[] for better diagnosis of brain disfunctions and predict seizers."

1.3 Aims and Objectives.

The objective of this research is to classify EEG signals with maximum accuracy. The data obtained from brain activity monitoring devices can be contaminated by various factors which results in high dimensional data, and the data acquisition tends to be costly. This limitation implies a restriction on the number of data samples that can be collected and used for EEG analysis and classification. Since the data obtained from neuroimaging devices belongs to various classes therefore, the most challenging task in EEG Classification is to map a brain activity to a specific command. To design a method for differentiating between different EEG classes, a model needs to learn task specific features which requires signal preprocessing tools for feature extraction and features selection for classification. EEG classification problems introduce a unique set of challenges which need to be addressed in machine learning. The goal of this research to overcome these challenges and design a robust method improve classification accuracy.

1.4 Research Methodology

In our research work, we followed a simple yet effective methodology. We did an in-depth literature survey to identify possible areas of work where improvement was needed. Once identified we came with a hypothesis regarding our technique. In the next phase we implemented the technique and iteratively addressed the issues that arose. Over time we converged towards a solution that was both faster and better compared to the existing methods. The techniques used in our work are standard for EEG analysis and classification. Furthermore, the evaluation process is cross validated and compared against existing EEG classification methods.

1.5 Organization of Thesis Report

The thesis report is organized in 5 chapters, which are arranges as follows:

1. Introduction
2. Literature Review
3. Dataset
4. Methodology
5. Discussion and Experimental Results
6. Conclusion

1.6 Summary

To summarize, we proposed an EEG classification method which has better accuracy than existing methods. Instead of using traditional classification methods for EEG, we proposed deep learning integration with Time-frequency transformation for improved results and performance.

CHAPTER 2: REVIEW OF EEG CLASSIFICATION APPROACHES

This section consists of two parts, in part A, we did a brief literature aimed at EEG classification. In Part B, the previous works done using convolutional neural network was briefly reviewed to highlight the remarkable performance of these techniques.

2.1Traditional Approaches

The majority of published techniques used some form of feature extraction followed by a simple classifier model. These Existing techniques require a less implementation time than common deep learning models and, are less prone to overfitting. Traditional EEG classification methods consists of two main steps: Feature Extraction, and classification. Various statistical approaches have been used for EEG source localization and classification. The author of [9] used Blind Source Separation(BSS) to evaluate and classify real time EEG signals. Correlation Analysis have been implemented along with Gaussian Mixture clustering by [35] to enhance and denoise time-domain EEG signals. High density source imaging systems have also been used for source identification of different EEG characteristics[39]. Many source localization techniques are available for feature extraction and classification EEG signals as discussed in [4]. Empirical Mode Decomposition and its variants can also be used to localize EEG sources and concluded that these techniques can achieve good spatio-temporal brain source reconstruction of active brain resources[40].

The other feature extraction approaches are Fourier transforms(FT) and related methods such as STFT, Hilbert Huang[43,52], Wavelet Transforms[7,15,18], principal component analysis (PCA), independent component analysis (ICA), autoregressive methods[81], or combinations of those techniques. The most used technique for feature extraction and classification of EEG signals is Fourier transform[56]. The feature extracted from this approach involves power spectra, which has been a major interest of EEG literature and addressed neural activities occurring within different frequency bands. A limitation of Fourier Transform is that it loses temporal information whilst providing information on frequency spectrum of the data. To prevent this limitation, short time Fourier transforms are more common for EEG analysis. The author of [51] used a variant of STFT for Epileptic seizers detection and EEG classification. Another signal processing tool used in feature extraction process is wavelet transform which decomposes a signal into different information bands based on frequency components. In contrast to Fourier transform, there are built in time and frequency trade-offs into wavelet-based models[82]. Another, signal transformation technique is the Hilbert-Huang Transform (HHT), which is also a time frequency method. HHT works according to an adaptive basis rather than relying on an a priori basis therefore, the basis in HHT is empirically determined is not guaranteed to be complete[83].

Support Vector Machines have been widely used and preferred by researchers for EEG classification instead of traditional source localization techniques. For example, SVM and KNN has been used to evaluate the Hjorth parameters, a statistical method, for EEG classification[38]. In [50], The authors intended to detect n Creutzfeldt-Jakob disease (CJD) using SVM whereas PCA features served as inputs to SVM algorithm. SVM maps the training data to a very high dimensional space, and then tries to find a hyperplane in that high dimensional space which separates the classes of data accurately. Three feature extraction processes i.e. PCA, ICA and LDA have been useful to detect epileptic seizers using SVM classifier in [54,55]. The Hilbert-Huang features have also been used for epileptic data classification via RBF feature selection and SVM classifier[58]. In [76] the binary classification of Epilepsy dataset[3] have been performed using SVM classifier and Cross correlation analysis have been used for feature extraction. Empirical Mode Decomposition based features have also been classified using SVM classifier in [77]. The author of [27] used trials of 1000ms to identify temporal and spatial features through SVM classifier where Global Field Power (GFP) was used to get temporal features and spatially weighted SVM(sw-SVM) to extract spatial features. In [78], the wavelet transform is combined with SVM and ANN methods for EEG classification and early seizer detection.

Additionally, there are other algorithms which uses similarity-based approaches to make decision about classification of data such as simple probabilistic classifier such as Naive Bayes and K Nearest Neighbors. The KNN have been used on Hjorth parameters i.e., activity, mobility, and complexity to classify time-domain EEG signals and to identify guilty and innocent in [5]. KNN has also been used for Schizophrenia detection[45] and mental stress recognition[46]. In [6], root mean square and polynomial fitting were implemented and Hjorth descriptors were classified using KNN algorithm. Artificial Neural Networks(ANN) have also been used for the purpose of EEG classification[30]. In [32], the author performed EEG classification using Continuous Wavelet transform and machine learning algorithms e.g., SVM and KNN. Traditional EEG classification methods requires in depth knowledge of EEGs to identify correct features[1], define appropriate thresholds, select right window size etc., for accurate feature extraction and classification.

2.2Deep Learning approaches

Neural networks provided great aid for EEG classification[13,23,33,41,42]. The accurate classification of EEG signals using machine learning algorithms is solely dependent on the correct detection and selection of distinctive EEG features. Lately with the success of Deep Learning(DL) models [11][36][48], this classification challenge has been resolved by deep learning models. Convolutional neural network(CNN), a subtype of Neural Networks, is considered best as compared to other machine learning techniques because it has end-to-end learning ability on raw data in terms of information extraction, online applications, usability, and classification accuracy[60][63].

The categorization of EEG signals in time domain have been performed using a variety of classification methods[2]. The author of [64] implemented EEG classification in Frequency domain using CNN. Image-based EEG classification have also been proposed by the author of [75]. CNN was initially designed to classify images, but it has been successfully used to classify EEG data in frequency domain. The binary classification of EEGs has been implemented in [67] using Radial Basis Function for feature extraction and a One Against One binary classifier.

In [64], the author proposed a 1D-CNN for epilepsy detection using Butterworth filter to denoise raw data and then created a spectrograms matrix which have been used to train CNN for classification. An Image based EEG classification method has been proposed in [65] which uses a specific window length to create a collection of sub-signals. The plot images of these sub-signal are created in temporal domain which are passed as input to the CNN model for classification. Similarly, a fixed size overlapping window is also used in [73] to generate a collection of sub-signals of EEGs in time domain whose plot images are used for binary ad ternary classification using CNN. A 3D image reconstruction and classification method has been proposed in[75] which used a sliding window to divide time series data into 2D segments and then, 3D image reconstruction is performed to create images suitable for 3D CNN. The author of [68] proposed binary classification of Epilepsy dataset[3] using KNN, Logistic regression, Decision Tree and Random Forest. Various machine learning methods are implemented and compared in [69]for binary classification of Epilepsy dataset[3]. [74] for epileptic classification. The author of [37] implemented a CWT to create images of 224x224 from Epilepsy dataset[3], which are used as input to a 3-layer CNN model. The maximum accuracy achieved for multiclass EEG classification using CWT and CNN is 72.49 % [37]. In [79], the author used CNN to detect epileptic seizer using time series EEG signals. The author of [80] used scalograms and CNN for epilepsy detection.

CHAPTER 3: Dataset

Electroencephalography records non-invasive brain signals generated by neurons due to some neural activity. These signals can be used to track brain functions, but EEGs tend to be noisy due to epilepsy, brain tumors, muscles movements, movement imagery, and Alzheimer's disease etc. Usually, these noisy characteristics in the EEG signals make it challenging to separate useful information from attributes of other classes with similar time-frequency patterns. For a precise identification and classification of multiclass EEG signals, machine learning models require a large amount of data since, the data will be divided, pooled, and normalized during feature learning process.

3.1 Data acquisition

The dataset[3] used in this research was composed of Five sets of EEG recordings denoted by A–E. All sets contain 100 single channel EEG segments recorded for 23.6-sec duration. For initial preprocessing, continuous multichannel EEG were selected and segmented after visual inspection. Various artifacts caused due to muscle activity or eye movements were discarded. The process of EEG recording for sets A and B was carried out extracranially using a standardized 10-20 electrode placement scheme(cf. ***Figure 2***) and five healthy volunteers contributed for this process. The volunteers were in a relaxing and awake state with their eyes open(A) and closed(B), respectively.

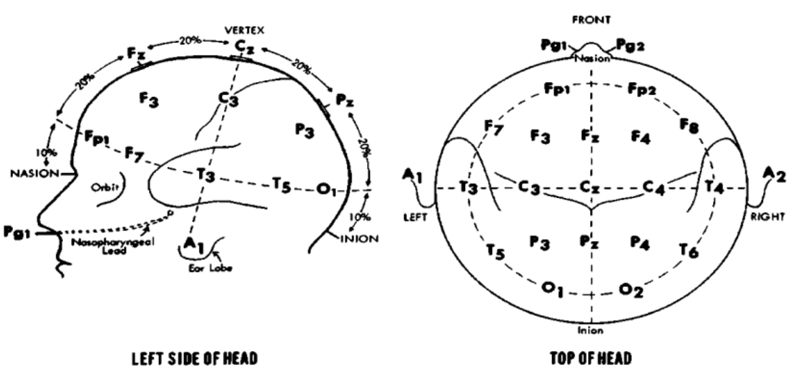


Figure 2. extracranially Electrode placement[86]

The remaining sets C, D, and E were created from the EEG archive at UCI machine learning repository for presurgical epilepsy diagnosis and recorded intracranially. For the present study, the EEG recordings from five patients were selected who had achieved complete seizure control after surgical resection of one of the hippocampal formations. Therefore, the epileptogenic zone in these patients was correctly diagnosed and set D was directly recorded within this epileptogenic zone. Set C was recorded from the hippocampal formation from the opposite hemisphere of the brain. Both sets C and D contained activities during seizure free intervals, where set E contained pure seizure activity.

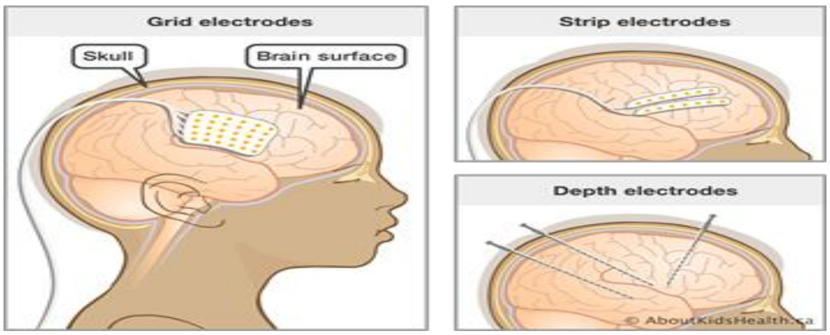


Figure 3. intracranial Electrode Placement[85]

The EEG segments in these sets were selected from all recording sites and exhibit ictal activity. All EEG recordings were carried out using the same 128- channel amplifier system based on an average common reference. The electrodes containing pathological activity(C, D, and E) or strong eye movement artifacts (A and B) were omitted. After performing 12-bit analog-to-digital conversion, the data was continuously written onto the disk of a data acquisition computer system with a sampling rate of 173.61 Hz. The signals were band-pass filtered using 0.53–40 Hz ~12 dB/oct. Exemplary EEGs are depicted in ***Figure 4***.

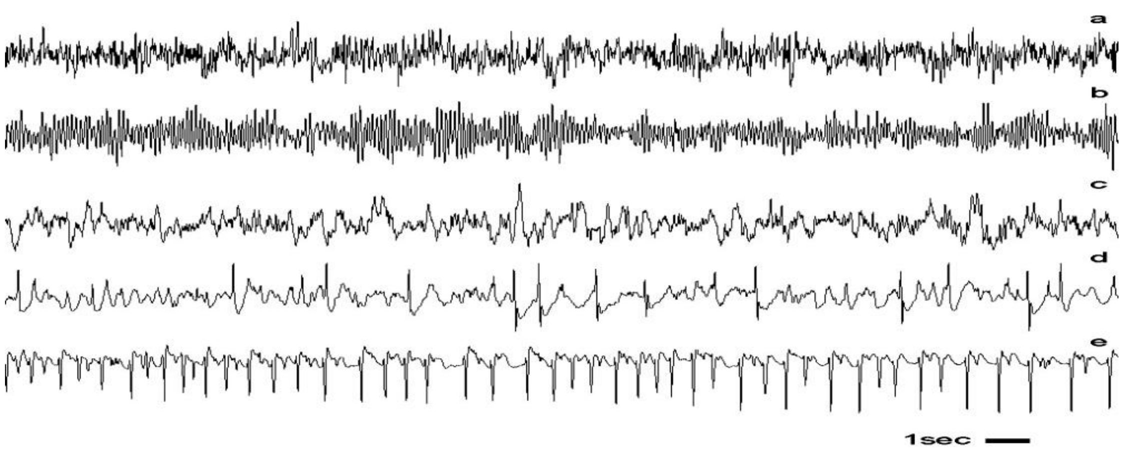


Figure 4. Types of EEGs in Epilepsy dataset

3.2 Dataset Preprocessing

The dataset used in proposed Integrated CWT and CNN method was rearranged by UCI Machine Learning Repository and it was published by[3]. Every 4097 data points are divided and shuffled into 23 chunks of 1 second. This process results in 23 x 500 = 11500 pieces of information(row), stored in 178 columns contains 178 data points per second(column). The 179th column "y" contains data labels representing classes for each information piece. Specifically, the classes {1,2,3,4,5} are:

1 - Recording of seizure activity.

2 - The recording of EEG from the area where the tumor was located during seizer free duration.

3 – Recording of the EEG activity from the healthy brain area in seizer free duration.

4 - eyes closed.

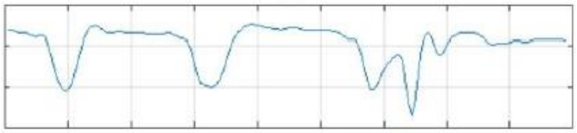
5 - eyes open.

All subjects from classes 2, 3, 4, and 5 are did not have epileptic seizure and only subjects from class 1 have epileptic seizure activities. This dataset contains collection attributes from five different classes, see **Table 1**.

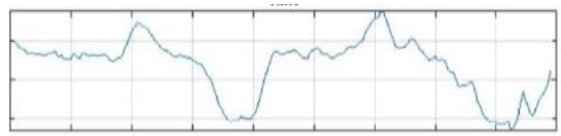
State Number	State Description	Prevalence
0	EEG Recording of seizure activity.	2300
1	EEG Recording from Tumor effected brain area	2300
2	EEG activity from the healthy brain area	2300
3	EEG recording with Eyes-Closed	2300
4	EEG recording with Eyes-Open	2300

Table 1. Attributes of EEG Dataset

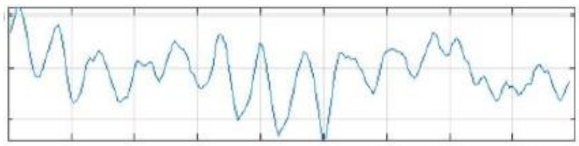
Additionally, the dataset is shuffled and reorganized to avoid biased classification of EEG data towards any class and, to make sure that the data points from each class get to be the part of CNN training. Essentially, samples from each class have different characteristics from each other which are illustrated in ***Figure 5***.



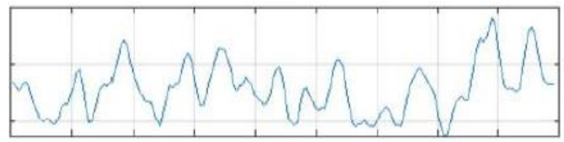
(a): Epileptic



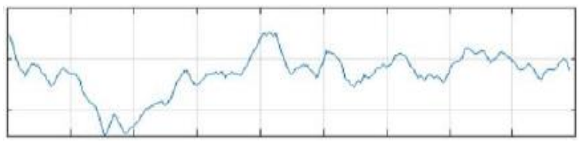
(b): Tumor



(c): Eyes closed



(d): Eyes open



(e): Healthy

Figure 5: Attributes of dataset

Chapter 4: Methodology

This chapter is divided into two sections: first we describe the theory of Continuous Wavelet Transform and Convolutional Neural Networks then, we discuss the flow of proposed integrated continuous wavelet transform and convolutional neural network method:

4.1 Continuous Wavelet Transform

In recent years, many signal transformation techniques have been useful to analyze signals such as Fast Fourier transforms(FFT) which provides frequency components of signals and Short-Time Fourier Transform(STFT), which uses a sliding window to extract time-frequency components of a signal. However, STFT has the limitation of window size and it is suitable for signals that do not change frequency over time. Empirical Mode Decomposition(EMD) also provides time-frequency analysis of signals, but it has complex exhibition modes of data which are difficult to interpret[24,28]. Wavelet transformation(WT) decomposes a signal into a set of frequency components[21,32] and present their distribution in temporal and spectral domain by compressing, scaling, and shifting signals(cf. ***Figure 6***). Since all frequencies become apparent in WT, it is advantageous over other transformation techniques[26,62].

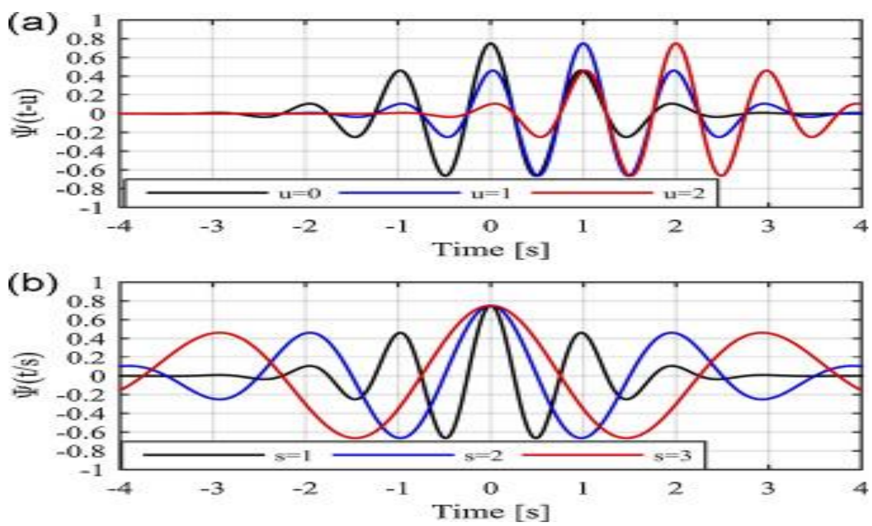


Figure 6. Wavelet transform.

A Wavelet($\psi \epsilon$ R) is function with zero average(i.e., $\int_R \psi$ =0), which is centered around t=0 and normalized(i.e., $\| \psi \|$=1) [12]. Eq. (4.1) represents mathematical form of a wavelet transform:

$$\psi_{u,v}(t)= \frac{1}{\sqrt{v}} \psi \left(\frac{t-u}{v}\right), u \epsilon R, v>0 \tag{4.1}$$

Where ψ is the basis function to compute frequency components, $\boldsymbol{u}$ is a shifting parameter along x-axis in the concerned region and $\boldsymbol{v}$ is a positive scaling parameter along y-axis because negative scaling is not established. The scaling and shifting parameters continuously vary to convolve mother wavelet over different portions of the signal and analyze it at different scales. Given for the original signal($f \epsilon$ R), the Continuous Wavelet Transform(CWT)[8,19] function of f using scaling and shifting parameters is presented in Eq.(4.2)

$$W f(u,v)=\int_{-\infty}^{\infty} f(t) \psi_{u,v}(t)dt \tag{4.2}$$

Through this transformation, a one-dimensional signal f(t) can be converted into two-dimensional form Wf(u,v) which are known as scalograms. These scalograms are used to detect and present most prominent frequencies(scales) of a signal in time-scale representation. From Eq. (4.3), the scalograms of a signal f(t) can be calculated as:

$$\Phi(s) =\| W f(u,v) \|=\left(\int_{-\infty}^{\infty} | Wf(u,v)|^2 du\right)^2 \tag{4.3}$$

Where, the function Φ denotes a scalogram which presents the energy of signal Wf at scale $\boldsymbol{u}$, and time-location $\boldsymbol{v}$. Thee exist many wavelet families which are different from each other based on their compactness and smoothness such as Gaussian, Mexican hat, Shannon and Morlet wavelet etc. In proposed Integrated CWT and CNN method, we used Morlet wavelet transform as the mother wavelet because it has the ability to extract features with equal variance in frequency and time[61]. Eq. (4) presents general mathematical form of Morlet wavelets.

4.1.1 Morlet Wavelet Transform

A Morlet wavelet is often used for time-frequency analysis in which a Gaussian function tapers a complex sine wave which convolves over the time series signal. Wavelet convolution can be considered as a process of “template-matching” in which the Gaussian-windowed sine wave acts as a template and each time point in the signal is compared against this template. This results in the “similarities” between the signal and the wavelet. The advantage of Morlet wavelet is that it causes the absence of sharp edges in the frequency domain. Since this transform uses Gaussianshaped function, it reduces the chances of signal misinterpretation by minimizing ripple effects. Second, wavelet convolution is computationally efficient than other transformation methods. Lastly, the results of Morlet wavelet transformation preserves the temporal resolution of the original signal.

The Morlet Mother wavelet can be defined as:

$$\psi(t) = e^{iw_0 t} e^{\frac{t^2}{2\sigma^2}} \tag{4.4}$$

Where, w_0 is the frequency and σ is the measure of spread of signal.

Translation and dilation of Morlet wavelet for scale factor u and dilation factor v can be defined as:

$$\psi_{(u,v)}(t) = \frac{1}{u} e^{\left[iw_0\left(\frac{t-v}{u}\right)\right]} e^{\frac{-\left(iw_0\left(\frac{t-v}{u}\right)\right)^2}{2\sigma^2}} \tag{4.5}$$

Morlet wavelet extract temporal features of a variety of signals and it can adapt to their time-frequency resolution. There is a criterion for selecting scale which is based on entropy of signals:

$$\mathrm{E}(f)=\sum_{k=1}^{n} pi \ \log pi \tag{4.6}$$

Where E is entropy of signal f and pi is the probability of k-th class in f.

4.2 Convolutional Neural Network

Deep learning has been of great interest for researchers in recent years and it has shown great advantage to every aspect of life where it has been used. Most prominent model of deep learning is Convolutional Neural Networks(CNN) which do not enforce prior selection of input features. CNNs were introduced in the late 1990s by Yann LeCun [84], Convolutional Neural Networks have gradually taken over the Computer Vision and classification community. To name a few, Denoising, Pattern recognition, fault and motion detection[22,59],Learning rotation-invariant features[10], Object Detection[20], segmentation[29][34], high-resolution reconstruction of images[14], and classification[16][66] are advance deep learning applications.

CNN takes raw data as input and learn certain patterns in time and scale dimensions (i.e., scalograms) without any handcrafted features and in-depth knowledge. CNN has the adaptability to any kind of transformation e.g., linear, and non-linear transformation[17].

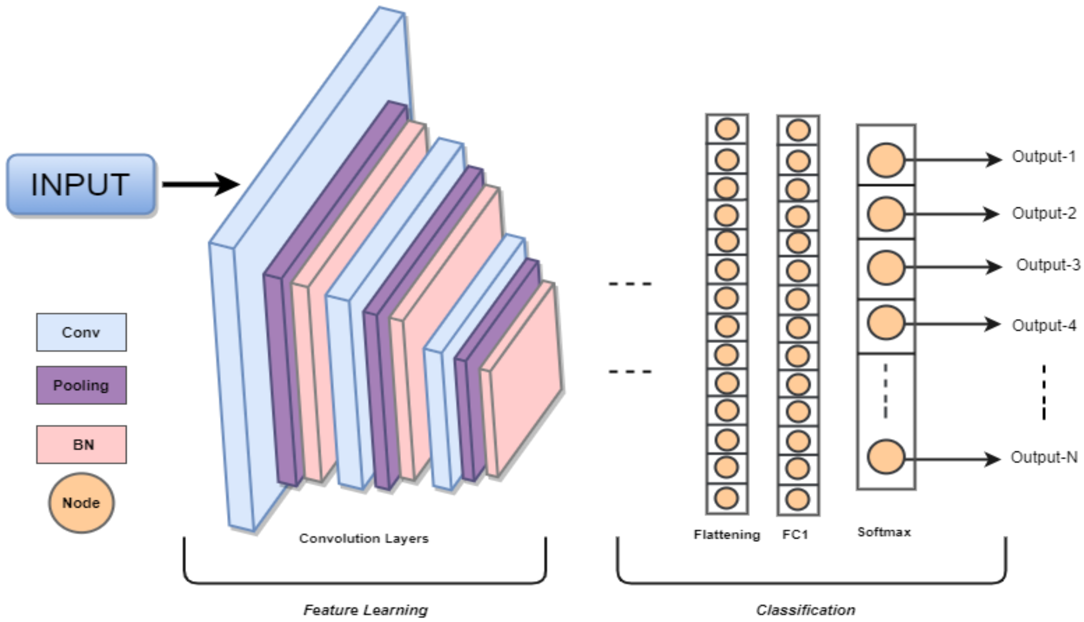


Figure 7: CNN architecture

As shown in ***Figure 7***, a CNN architecture has two main layers, an input and output layer, however, there can be variable number of hidden layers between these two layers. The hidden layers are comprised of combinations of Convolution layer, pooling layer, batch normalization layers, Rectified Linear Unit Layer, and one or more fully connected layers. CNN is just like feed-forward neural network which is composed of one or more layers with variable number of neurons. The input passes through the network as linear combinations of input culminated by each neuron from each layer so that the network can learn highly non-linear features. The neurons in each convolution layer learn distinctive features from the input and constructed feature maps. Hidden layers learn features from the selective activated neurons based on sparse interactions.

4.2.1Convolution

Convolutional Neural Networks are based on the principal of convolution, Convolution is a well-known operation in Image processing, which is given by:

$$I_{new} = \Sigma\, k.\, I \tag{4.7}$$

Where, I_{new} is the new feature map of image received after applying convolution, k is the kernel, a small image which acts as weights in Convolutional layer and I is the original image.

4.2.2 2D Convolution

Convolution layer is the essence of CNN architectures because this layer is composed of features maps which are generated by computing cross-correlation between previous layer's output and kernels in receptive neurons(cf. ***Figure 8***). Each neuron in current layer is associated to a different region of previous layer's input to extract distinct elements from input[21]. 2D convolution in a Neural network is performed to extract high-level features from local neighborhood on feature maps in the previous layer.

$$V_{ij}^{xy} = b_{ij} + \sum_m \sum_{p=0}^{P_i-1} \sum_{q=1}^{Q_1-1} w_{ijm}^{pq} \, v_{(i-1)m}^{(x+p)(y+q)} \tag{4.8}$$

Then,

$$activation = \max(V, 0)$$

Where the activation function can be replaced by other activation functions, like, tanh, sigmoid, etc., b_{ij} is the bias, m is a set of indexes of feature maps between two connected layers such as (i-1)th and (i) th layer, w_{ijm}^{pq} is the weight parameter at the position*(p, q)* of the kernel, and P and Q refer to the height and width of the kernel, respectively. The above equation is an equivalent to what we see in Fully connected layers.

$$y = Wx + b \tag{4.9}$$

Where, W is the matrix of calculated Weights, x is the input features and b is the bias.

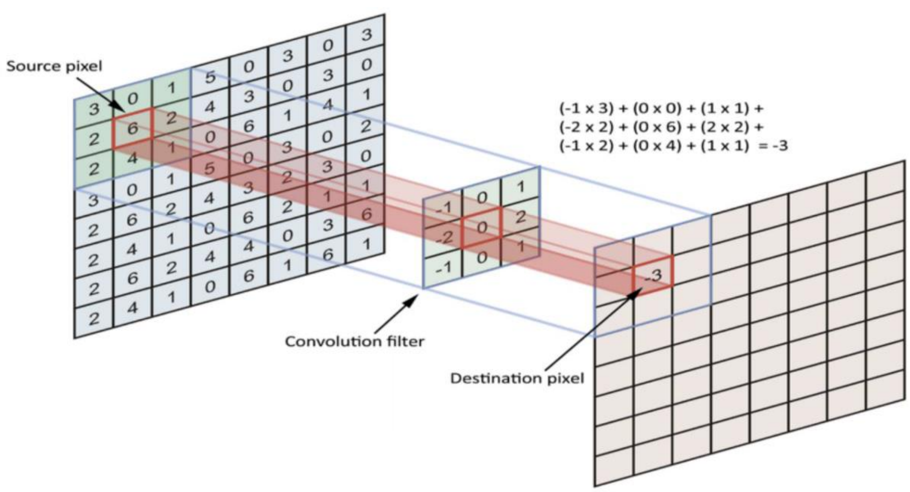


Figure 8. Illustration Of 2D convolution[87]

In case of convolutions in convolution layers, the total number of weights are a lot less than that of an equivalent fully connected layer because the same kernel is re-applied in case of convolutions all over the image, while in case of fully connected layers there is one weight for one corresponding pixel. This property of convolutions makes them more efficient in computations and they achieve better results. In practice multiple kernels are used to extract high-level features from the input image and then either the resultant image is passed to another convolution layer or to a pooling layer.

4.2.3 Pooling

Pooling layers help in reducing the size of the input image to make the operations less expensive, there two types of pooling layers used in Convolutional Neural Networks, Max Pooling and Average pooling, the concept of pooling is to select a value from a window of some size ($k \times k \times k$) from the input image of size ($N \times N$) and use that value as the representative of these value in the new image ($N / k \times N / k$). This method is also commonly known as sub-sampling, in max pooling the maximum value from the window is chosen while in average pooling, average of all the values in the window is taken(cf. ***Figure 9***).

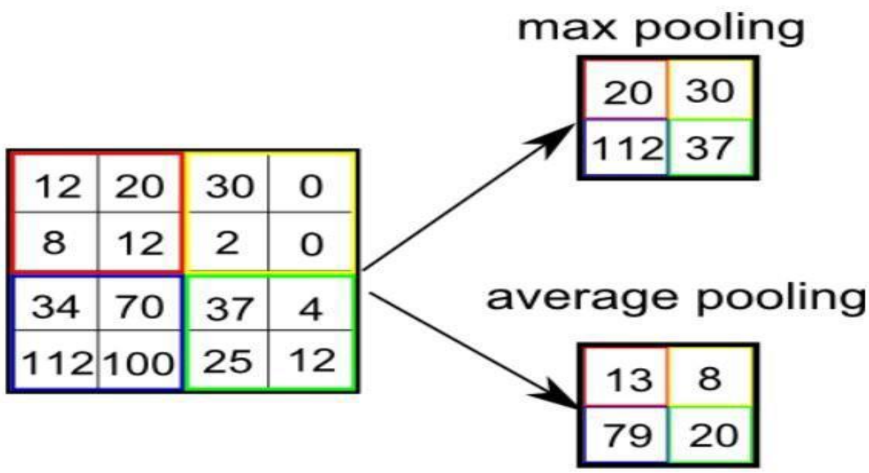


Figure 9. illustration of 2D Pooling[88]

This process of stacking convolution and pooling layers is applied a several times alternatively to extract features from the image then the resultant feature map is passed to the fully connected layers which learn the non-linear representation of the high-level feature maps extracted from the convolutional layer.

4.2.4 Batch Normalization

One of the most important steps while training a neural network is normalization because the input of each layer is affected by the output parameters of all previous layers. In deep networks, the distribution of the input data is not consistent in hidden layers therefore, even the smallest change of parameters will have a huge impact on output features distribution as the network becomes deeper. This is called to internal covariate shift problem which complicates the training process. Typically, due to internal covariation in training data, the distribution of feature maps changes due to the update of parameters. This phenomenon requires to select small learning rate and initialize parameters carefully. This problem seems to slow down the learning process and makes it harder to learn features with saturating nonlinearities. The basic idea of BatchNormalization [25] is to train a network at a higher learning rate by fixing the input distribution problem for each hidden layer. Batch normalization is performed such that the entire batch is normalized to have zero mean and unit variance(cf. ***Figure 10)***.

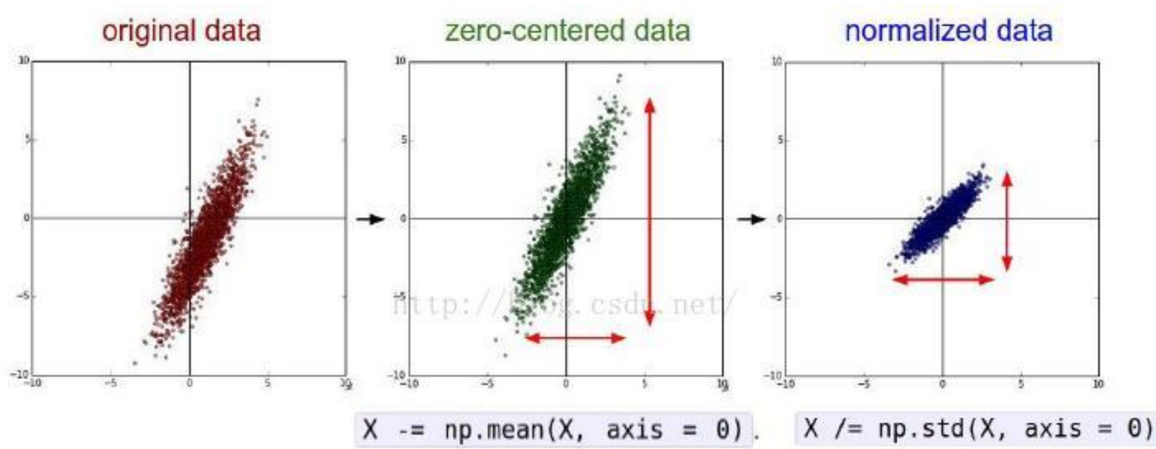


Figure 10. *Batch Normalization[89]*

4.2.5 Fully Connected Layer

In fully connected layers, each node of current layer should be connected to each node of next layer. This layer takes flattened output of previous Convolution layers and combines them to generate a vector of probability scores. The output layer of CNN architecture assigns data to the respective classes based on computed probability scores.

Suppose there are two consecutive layers, $l^{(k-1)} \in R^{m^{(k-1)}\times 1}$ and $l^{(k)} \in R^{m^{(k-1)}\times 1}$, the weight matrix to fully connect these two layers would be defined as $w^{(k)} \in R^{m^{(k-1)}\times m^{(k)}}$. This structure is represented in ***Figure 11***. A bias term ($b^{(k)} \in R^{m^{(k)}}$) is always added to this weight matrix to account for the existing constants in the system. Finally, the output of a fully connected layer, $o^{(k)}$, can be calculated using following layer function $\psi^{(FC)}$ as:

$$o^{(k)} = \psi_{(k)}^{(FC)}\left(o^{(k-1)}\right) = (o^{(k-1)})^T w^{(k)} + b^{(k)} \qquad (4.11)$$

The computational complexity for each fully connected layer $\psi_{(k)}^{(FC)}$ can be described as:

$$\mathcal{O}\left(\psi_{(k)}^{(FC)}\right) = \mathcal{O}(m^{(k-1)}m^{(k)}) \qquad (4.12)$$

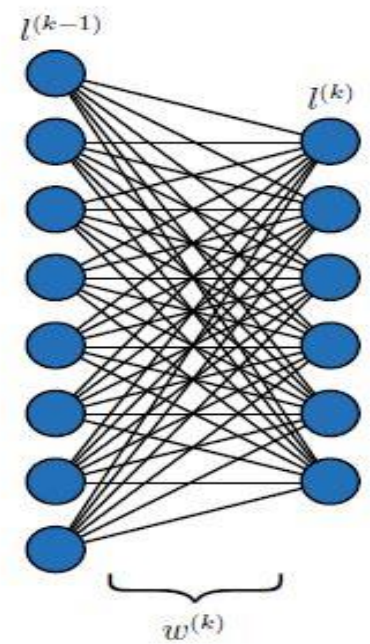


Figure 11. Graphical Representation of two FC layers

4.3 Loss

To determine the quality of an approximation by a classifier, a loss (or cost) function is used. A loss functions has two basic properties: loss is never negative and if we compare two approximations, the result with smaller loss tends to be better at approximating the data. A loss function compares the actual value with the predicted value. It will report a high number if the deviation is too much and a small number if the predicted output is near to the actual output value.

In Machine learning, the value(s) from a certain loss function helps to evaluate the performance of a specific architecture on given data. There exist various loss functions for regression and classification i.e., Mean Square error(L2 loss), Mean Absolute Error(L1 loss), Mean Bias Error, and cross entropy loss(CEL). Since our problem is based on classification well only discuss loss function for classification.

4.3.1 Cross-Entropy

Cross-entropy loss(CEL) function is a log loss, which estimates the performance of a classification model whose output is a probability which always lies between 0 and 1. Cross-entropy loss can be used for binary and multiclass classification. CEL for binary classification, can be defined as:

$$CE = -\sum_{m=1}^{M=2} y_{o,c} \log(P_{o,c}) = -y_1 \log(P_1) - (1 - y_1)\log(1 - P_1) \qquad (4.13)$$

Where M_1 and M_2 refer to two classes.

For multi-class data, CEL is best suited and can be calculated using Eq. (4.14):

$$\text{CE}= -\sum_{m=1}^{M} y_{o,c} \log(P_{o,c}) \qquad (4.14)$$

Where, M is number of classes in the data, y is true label, and P is predicted probability of a sample “o” in class C.

For Multi-Class classification the labels are one-hot encoded, so only the positive class keeps its term in the loss. There is only one element of the Target vector y which is not zero i.e., $y_m = y_p$ therefore, discarding the elements of the summation which are zero due to target labels, we can write the one-hot encoded labels for multi-class data using eq(4.15):

$$CE = -\log\left(\frac{e^{S_p}}{\sum_j^C e^{S_j}}\right) \qquad (4.15)$$

Where, S_p is the score for the positive class.

4.4 Optimizer

Optimizers are used while training the model to update the weight parameters. Loss functions act as input to the optimizer to indicate if the model training is moving towards local minima or not. There are many common optimizers available to optimize the weights during feature learning process, but we will only discuss the most used ones in classification problems.

4.4.1 Stochastic Gradient Descent

Stochastic Gradient Descent only uses the loss gradient of one training example at each iteration, it does not use the sum of loss gradient of all training examples. Data shuffling is required in its usage. Since it uses only one training example at a time, its path towards the minima is very noisy and very random which in results causes an unstable convergence during training process.

$$\boldsymbol{W}=\boldsymbol{W}-\boldsymbol{\alpha}.\boldsymbol{dw}$$

$$\boldsymbol{b}=\boldsymbol{b}-\boldsymbol{\alpha}.\boldsymbol{db} \qquad (4.16)$$

Where, ($\boldsymbol{\alpha}$) is the learning rate, $\boldsymbol{dw}$ and $\boldsymbol{db}$ are derivatives of weight $\boldsymbol{W}$ and bias $\boldsymbol{b}$.

4.4.2 RMSprop

RMSprop stands for root mean square propagation. It resolves adagrad's vanishing learning rate by using moving average of squared gradient. Learning rate gets adjusted on its own and its choses a different learning rate for each parameter.

$$\boldsymbol{v_{dw}}=\boldsymbol{\beta}.\boldsymbol{vdw}+(\boldsymbol{1}+\boldsymbol{\beta}).\boldsymbol{dw}$$

$$\boldsymbol{v_{db}}=\boldsymbol{\beta}.\boldsymbol{vdw}+(\boldsymbol{1}+\boldsymbol{\beta}).\boldsymbol{db} \qquad (4.17)$$

Where, ($\boldsymbol{v_{dw}}$) and ($\boldsymbol{v_{db}}$) are the gradient updates for the weights ($\boldsymbol{W}$) and bias ($\boldsymbol{b}$), ($\boldsymbol{\beta}$) is the momentum.

$$\boldsymbol{W}=\boldsymbol{W}-\boldsymbol{\alpha}.\frac{\mathrm{dw}}{\sqrt{v_{dw}}+\epsilon}$$

$$\boldsymbol{b}=\boldsymbol{b}-\boldsymbol{\alpha}.\frac{\mathrm{db}}{\sqrt{v_{db}}+\epsilon} \qquad (4.18)$$

Where, ($\boldsymbol{\alpha}$) is the learning rate, ($\boldsymbol{\epsilon}$) is the weight decay.

4.4.3. Adam

Adam is an moment estimation optimizer which estimates the individual moments adaptively. These learning rates are estimated from the moments of the gradients. It also helps overcome the problem of vanishing learning rates. It is computationally effective hence requiring very less memory.

$$\boldsymbol{m}=\boldsymbol{\beta 1}.\boldsymbol{m}+(\boldsymbol{1}-\boldsymbol{\beta 1}).\boldsymbol{g}$$

$$\boldsymbol{v}=\boldsymbol{\beta 2}.\boldsymbol{v}+(\boldsymbol{1}-\boldsymbol{\beta 2}).\boldsymbol{g2} \qquad (4.19)$$

Where, $\boldsymbol{m}$ and $\boldsymbol{v}$ are moving averages of gradient $\boldsymbol{g}$, $\boldsymbol{\beta 1}$ and $\boldsymbol{\beta 2}$ are hyper-parameters with default values of 0.9 and 0.999.

$$\hat{v} = \frac{v}{(1-\beta_2^t)} \quad (4.20)$$

Where, $\hat{m}$ and $\hat{v}$ are bias corrected 1st and 2nd momentum, $\boldsymbol{t}$ is the current iteration.

$$w = w - \eta \, \frac{\hat{m}}{(\sqrt{\hat{v}} + \epsilon)} \quad (4.21)$$

Where, $\boldsymbol{\eta}$ is the step size, $\boldsymbol{\epsilon}$ is the weight decay and $\boldsymbol{w}$ are the weights.

4.5 Weights Initialization

While training a model, it is important to randomly initialize the weights to ensure faster convergence during training and feature learning. Initializing weights the right way is very important because, if it is done wrong, it can cause problem of vanishing or exploding gradients. There are several ways to initialize the weights indifferent layers. We will only discuss only a few.

4.5.1 Zeros

In zero initialization, all the layers all initialized. This might be problematic as same all the errors propagated through the network will same hence affecting the learning process.

4.5.2 Random Normal

It initializes the tensors with a normal distribution.

4.5.3 Xavier Normal Initializer

It initializes the weights in network by drawing them from a distribution with zero variance and a specific variance of $\boldsymbol{f}(\boldsymbol{W}) = \frac{2}{n_i + n_0}$ where, $\boldsymbol{W}$ are the weights and $\boldsymbol{n_i}$ and $\boldsymbol{n_o}$ are the number of neurons in input and output layers.

4.5.4 He Normal Initializer

The He Normal Initializer takes samples from a truncated normal distribution which is centered on 0 and which has standard deviation = $\sqrt{2/fan\_in}$ where, fan_in corresponds to the number of input units in the weight tensor.

4.6 Activation Function

Activation functions are a significantly important choice to training neural networks efficiently[18]. They govern the flow of input features into the network and determine which features should be learned by the network. Usually, linear transformations of the inputs in each layer rules out the possibility of learning complex functional forms of non-linear data. Therefore, a non-linear activation function is favored for its improved expressibility and learnability of complex features.

4.6.1 Sigmoid and Tanh activation functions

The sigmoidal and hyperbolic tangent function are preferably used by researchers than other activation function because these functions provide the b benefit of non-linearity and differentiability, since they have easily calculable derivatives everywhere in the input data, therefore they are useful for efficient computation of gradients[20]. The mathematical form of these function is:

$$sigmoid = \sigma(z) = \frac{1}{1+e^{-z}} \tag{4.22}$$

$$hyperbolic\ tangent = \tanh(z) = \frac{e^{z}+e^{-z}}{e^{z}-e^{-z}} \tag{4.23}$$

4.6.2 ReLU and Softmax activation functions

Another most commonly used activation function is known as Rectified Linear Unit or ReLU[17]. The output of this function is either 0 or the input itself(z) therefore, it can be calculated faster than other activation functions. ReLU function helps networks to avoid vanishing gradients problem and imposes sparsity on the activations in each layer. The classification layer always uses softmax(·) function[19] and contains the final probabilities for all classes. The activation function for output layer is presented in Eq.(4.24).

$$P(y=j\ |z)=\text{Softmax}_j(z)= \frac{e^{z_j}}{\sum_{c=1}^{C} e^{z_c}} \tag{4.24}$$

Where, C is used for number of classes in an input vector z.

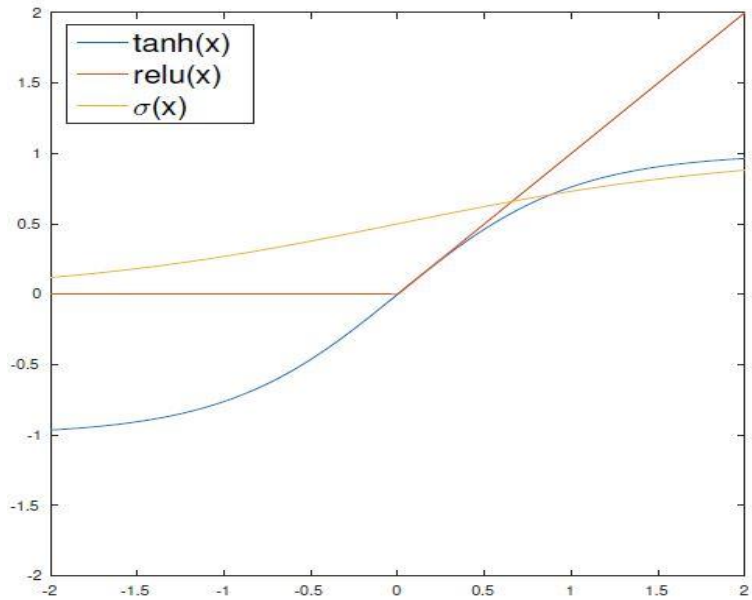


Figure 12. Plot of regularly used activation functions Tanh, ReLU, and sigmoid.

4.7 K-fold Cross validation

This technique is attractive because it has the ability to manage the computational cost whilst increasing the estimation bias. In K-fold cross validation, the original dataset(D) is divided into multiple(K) partitions of nearly equal size and each of these portions is called a "fold" of the actual dataset (thus there are K folds). This process can be considered as a leave-one-foldout technique. Typically, the classification model is trained on K −1 folds and the K-th fold is used for testing purposes. This process is repeated for K times such that each fold is part of the testing phase exactly once. Leaving out the K-th fold for testing purposes and using remaining k-1 folds for model training can generate estimated model performance for each iteration. The cross-validation statistic for prediction error is then computes as:

$$KCV = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}^{-k(i)})^2 \tag{4.25}$$

4.8 Proposed Integrated CWT and CNN method

We are working towards a solution of early seizure detection through EEG classification. To avoid adverse consequences of seizures, it is important to design a method which can achieve maximum classification accuracy. We want to classify EEG data using Convolutional Neural Network based on Continuous Wavelet Transformation. A flowchart of proposed Integrated CWT and CNN method is presented in ***Figure 13***

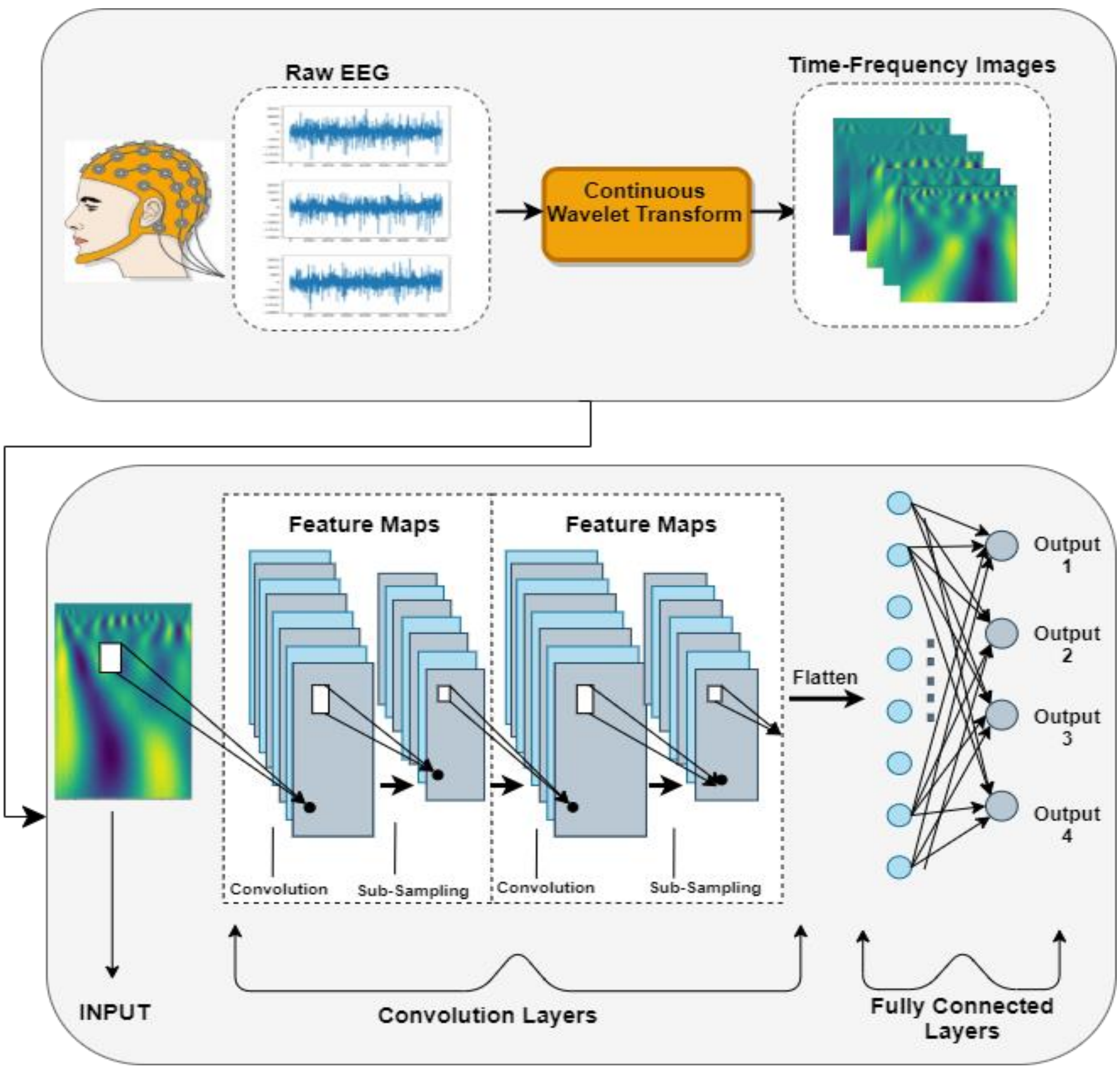


Figure 13: Proposed integrated CWT and CNN Method Flowchart

The CNN architecture is broken down into feature learning and classification. The feature learning part has three convolutional(Conv) layers present in proposed architecture followed by Batch normalization(BN) layer, three pooling layers to minimize the size of features to make computations faster and three fully connected layer in end to create a fully connected neural network to count the number of samples for each class. The convolution layers use ReLU function to activate neurons which contain linear combinations of data patterns learned by network.

For classification part, the network consists of a flatten layer which converts all features into one dimensional data so that, the data can be forwarded to tshe fully connected layer. There are three fully connected layers in this part where ReLU activation function is applied in the first two dense(FC) layers, and Softmax activation function is used in classification layer. Essentially, the output layer has five nodes as it must classify five-class data(cf. ***Figure 14*** **Figure *14***)

Initially, the dataset is shuffled randomly to make sure that samples from different classes are appearing in training, validation, and testing datasets equally. Furthermore, a standard scaler is used to normalize data with mean=0 and standard deviation=1. Furthermore, the collection of these images is then divided into Training, Validation and Testing datasets with a splitting factor 0.3. The training dataset contains 8050 samples where validation and testing datasets have 1725 samples each. The continuous wavelet transformation is used to create two dimensional images using scale values 0 to 128. These time-frequency coefficients are then rescaled by 128 to create 128x128 dimension images. These spatial domain images are reshaped and sent to network as input images. The proposed integrated CWT and CNN method takes these images as input to learn all possible features of Epilepsy Dataset[3].

We did brief experimentations with varying input sizes and with varying convolutional neural network architecture designs. The network with the best result is chosen for the paper submission and, for thesis. The architecture of proposed CWT and CNN method is presented in ***Figure 14***.

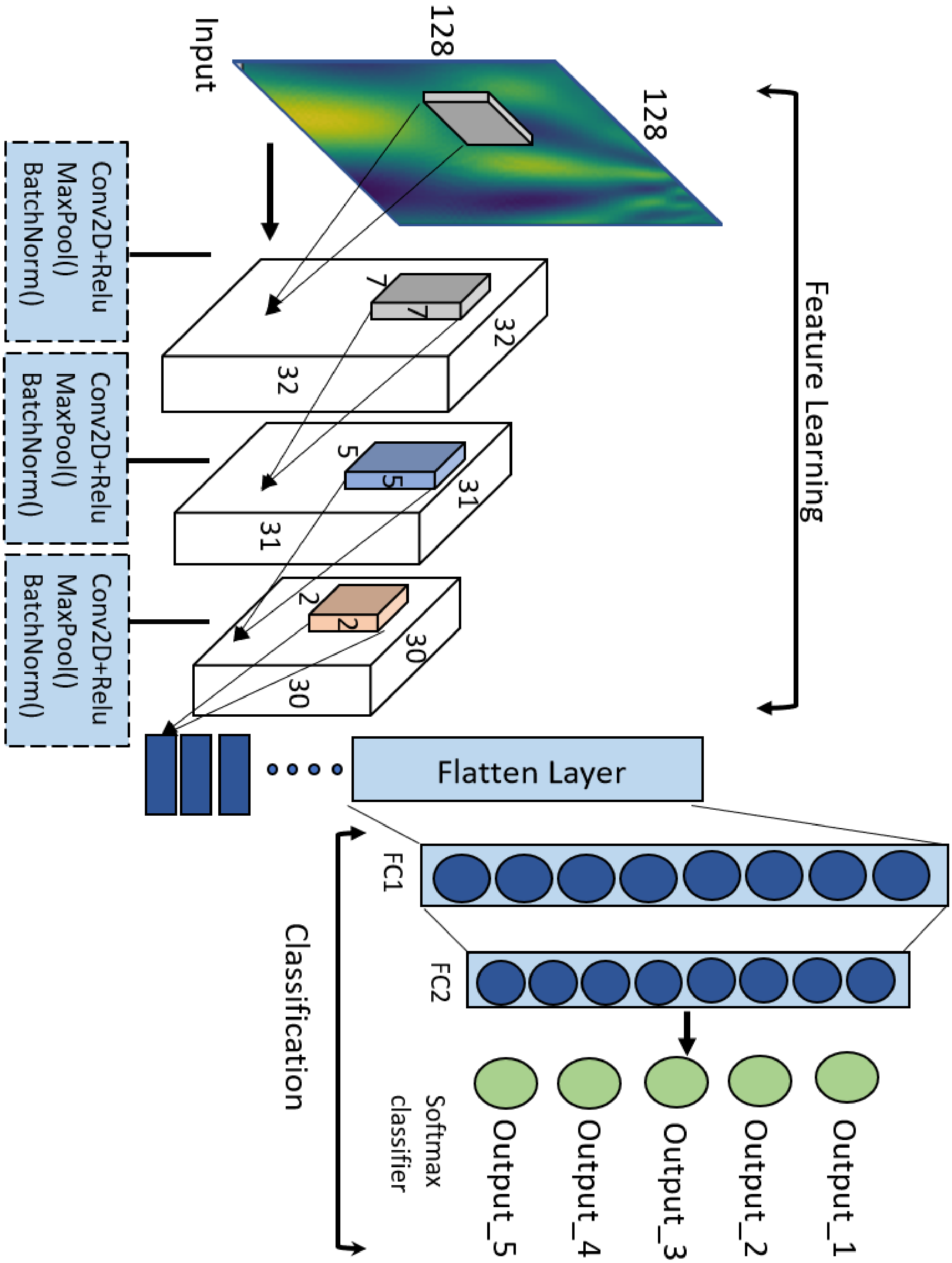


Figure 14:Architecture of Proposed Integrated CWT and CNN method

Layers	Output Shape	Description	Parameters
Input	(None,128,128,1)	-	0
Conv2d	(None,32,32,128)	Kernel=7, filters=128, ReLU, strides=4	6400
batch_normalization	(None,32,32,128)	-	512
MaxPooling2d	(None,31,31,128)	Kernel=2, strides=1	0
Conv2d_1	(None,31,31,128)	Kernel=5, filters=128, ReLU, strides=1	409728
batch_normalization	(None,31,31,128)	-	512
MaxPooling2d_1	(None,30,30,128)	Kernel=2, strides=1	0
Conv2d_2	(None,30,30,256)	Kernel=2, filters=256, ReLU, strides=1	131328
batch_normalization	(None,30,30,256)	-	1024
MaxPooling2d_2	(None,29,29,256)	Kernel=2, strides=1	0
Flatten	(None, 215296)	-	0
Dense	(None,128)	Nodes=128, ReLU	27558016
Dropout	(None,128)		0
Dense_1	(None,50)	Nodes=50, ReLU	6450
Dropout_1	(None,50)		0
Dense_2(Softmax)	(None,5)	Nodes=5, Softmax	255

Table 2. Description of Proposed integrated CWT and CNN Method

The input size of images for CNN is 128 x128 pixels. The first convolution layer outputs 128 feature and has 6400 parameters. The next layer is batch normalization to standardize inputs for each batch in the layer and it has 512 learnable parameters. After that, a max pooling is carried out spatially only on the output of previous convolution layer. The spatial features are not pooled as we want to preserve the temporal data at this point and not lose it too soon hence effecting the count result.

The second convolution layer in proposed architecture has 128 filters of size 5-by-5. It outputs 128 features maps and 409728 parameters. This layer is also followed by batch normalization and max pooling. The third convolution layer contains 256 kernels of size 2-by-2 resulting in 131328 parameters. The followed batch normalization layer results in 1024 learnable parameters. Then a max pooling is applied to the results of convolution layer.

The output dimension from the last layer of dimension 30x30x512 is flattened before passing it to the fully connected layers. The fully connected first layer eventually receives a feature map sized 27558016. The fully connected or dense layers are all interconnected with each other. This causes a huge increase in the number of parameters therefore, a Dropout layer is added in the network after Fully connected layers. For complete summary of our integrated CWT and CNN method, see ***Table 2***.

4.8 Training the Network

To train the network, we used Adam optimizer. Since it is adaptive to moment estimation, and it overcame the problem of vanishing learning rates/moments[49]. Also, it is computationally effective because it requires very less memory.

Cross-entropy(CE) is our choice for cost function. Our goal is to predict the number closer to the original number and CE tries to keep prediction closer to the output by keeping record of only positive class.We chose ReLU activation function since our data has complex features and we want to achieve fast convergence during training.

We used He initializer for tensors initialization because we want to bring the variance of outputs to approximately 1(He et al., 2015, Kumar, 2017).This weight initialization strategy initializes the weights randomly but with this variance:

$$v^2 = \frac{2}{N} \tag{4.26}$$

We trained the proposed Integrated CWT and CNN method using Google Colaboratory, which is a cloud-computing based service to train deep learning models in Python environment. We implemented this method in Keras using Tensorflow at the backend. Multiple python libraries such as Panadas, Numpy, and sklearn etc., have been used for data processing and simulations.

The classification accuracy and loss can be determined by using Eq. (4.27) and Eq. (4.28).

$$\text{Accuracy} = \frac{Validation\ Sample}{Number\ of\ samples\ in\ validation\ Data} \tag{4.27}$$

$$\text{Loss} = \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K}(Train\ samples - validation\ samples) \tag{4.28}$$

CHAPTER 5: Experiments and Results

The dataset mentioned in Chapter 3 was used for training and testing. In training and experiment phases, the biggest challenge was finding the right architecture and right best performing settings. Training the network also brought its own issues as initially we faced issues regarding vanishing gradients. After training, the performance of trained model is evaluated on test data. Also, a comparison of the results produced by proposed method and existing DL models is carried out to evaluate performance of our model. The performance of proposed Integrated CWT and CNN method is validated using Kfold cross validation. We compared our results with a baseline which uses the technique of CWT and CNN for EEG classification.

Famous deep learning(DL) models such as GoogleNet, VGG16, and AlexNet are also trained and tested on dataset described in chapter 3 to perform the comparison between them and our work. AlexNet architecture won ILSVRC-2012[31]. AlexNet is trained for 150 epochs and the resulting accuracy and loss of AlexNet for EEG dataset classification can be seen in ***Figure 15***. GoogleNet won ILSVRC—2014[57]. This model is trained for 2000 epochs and the resulting accuracy and loss graph of GoogleNet performance on EEG dataset are shown in ***Figure 16***. VGG16 is also a famous deep learning architecture which is named after Visual Geometry Group at Oxford. This model outperformed many previous generation models in ILSVRC-2012 and ILSVRC-2013 competitions[53]. Due to long training duration and limited resources, VGG16 is only trained for 500 epochs. ***Figure 17*** presents the resulting accuracy and loss of VGG16 for 150 epochs.

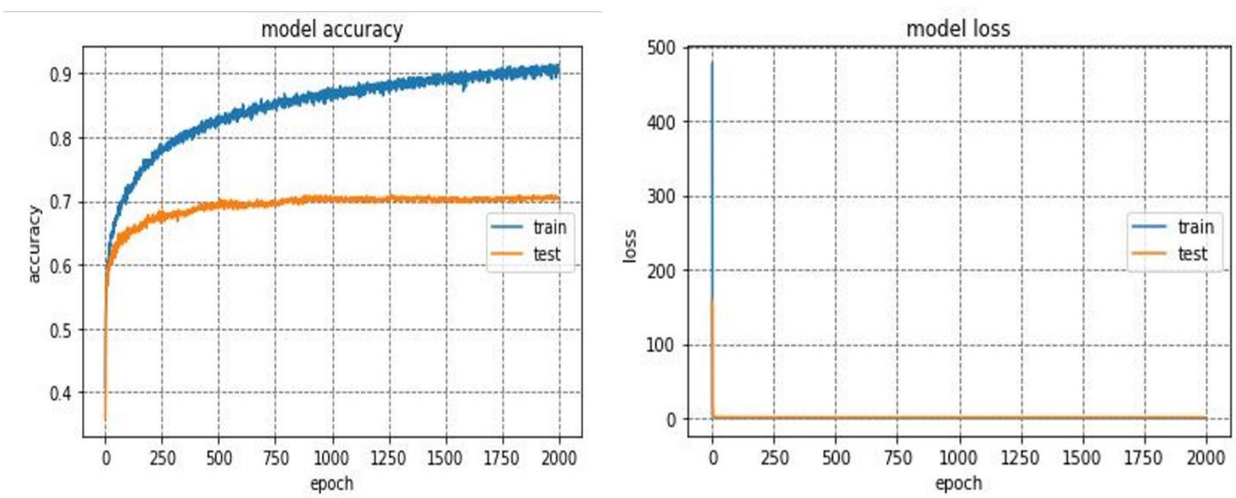


Figure 15: AlexNet Performance

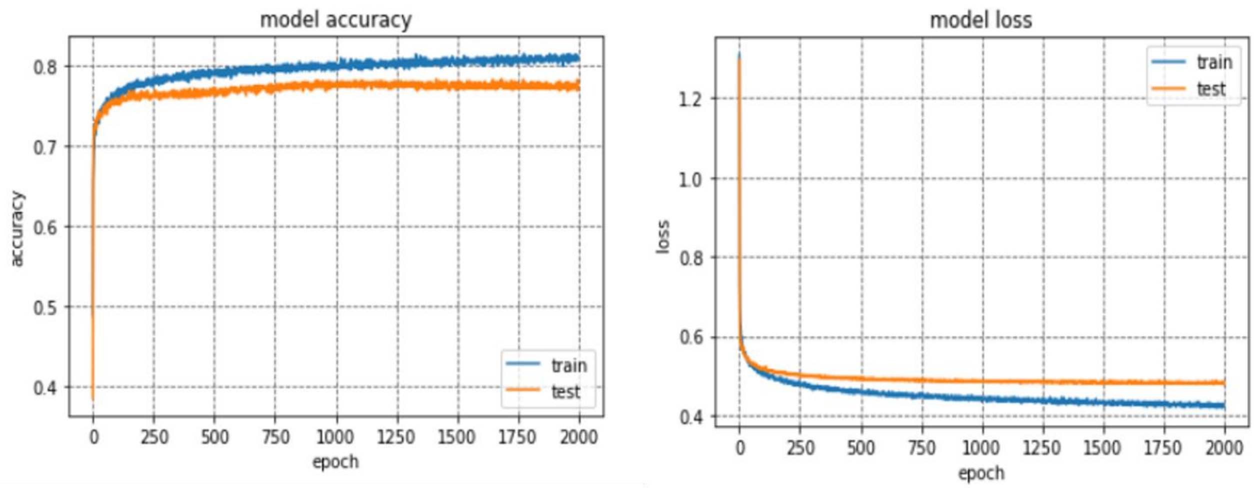


Figure 16: GoogleNet Performance

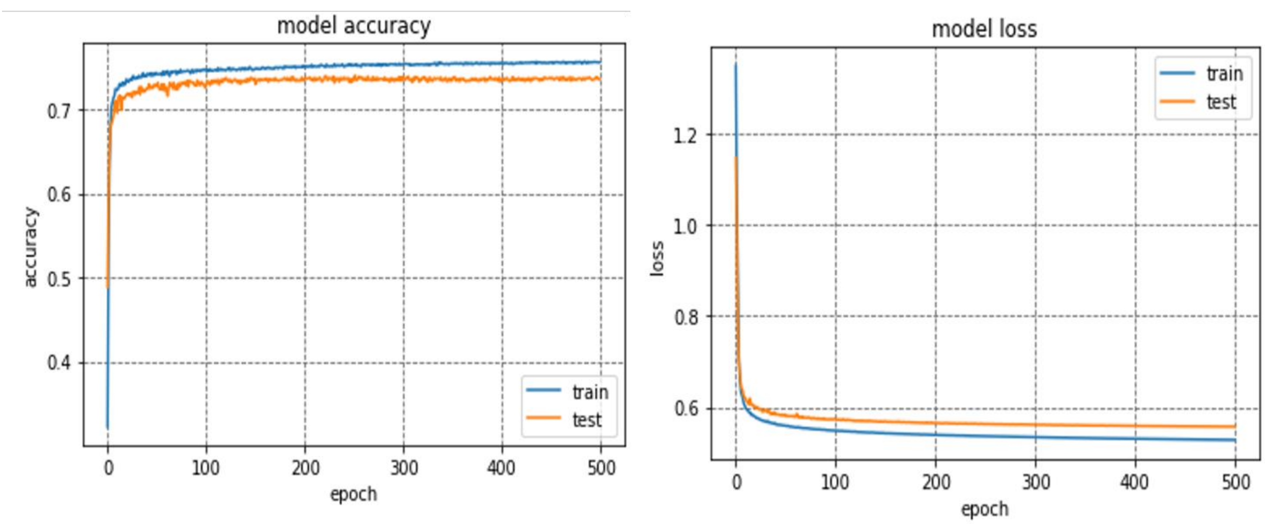


Figure 17: VGG16 Performance

GoogleNet, VGG16 and AlexNet architectures seem to perform well on Epilepsy dataset[3] during training and validation steps whereas, these architectures suffer from overfitting while performing classification on test data which results in bad accuracy scores. The proposed Integrated CWT and CNN method seems to perform well on the same Epilepsy dataset[3]. and the results of proposed Integrated CWT and CNN method does not show huge discrepancy in accuracy scores of training, validation and testing phases.

The results generated by all DL models and proposed Integrated CWT and CNN method for their performance assessment is shown in ***Table 3***.

Model	Accuracy(%)		Loss(%)		Test Accuracy(%)	Time (m)
	Train	Validation	Train	Validation		
GoogleNet	81.29	77.97	42.34	47.72	77.97	448
AlexNet	88.83	72.35	29.45	73.90	71.25	219
VGG16	75.86	74.38	52.91	53.59	74.32	305
Mao et al.(2019)[]	72.49		59.60		-	820
Our Model	90.46	78.84	27.55	53.04	78.84	36

Table 3. Performance of proposed method Vs Other DL architectures

The results of proposed Integrated CWT and CNN method are also cross validated using Kfold cross validation using 10 folds of Epilepsy dataset. Kfold cross-validation method is useful to evaluate the performance of a trained model on unseen data from the original dataset. This cross-validation process makes sure that the model is not performing in a bias manner towards a specific class. In this process, each data point gets to be a part of the testing process at least once and the model gets to trains on this data on multiple times depending on the number of folds i.e., if we use k=10 folds then each data point is used as a part of training for k-1 times(cf.)

The average accuracy achieved by Kfold cross validation for proposed Integrated CWT and CNN method is 76.02%. The overall Accuracy lies in the range of 74% to 79% and loss lies in the range of 50% to 57%. The proposed Integrated CWT and CNN method has improved classification accuracy by 6.35 %, loss is reduced by 6.02% and the performance time of proposed

Integrated CWT and CNN method is also efficient. Lastly, the performance of proposed Integrated CWT and CNN method can be observed in ***Figure 18***

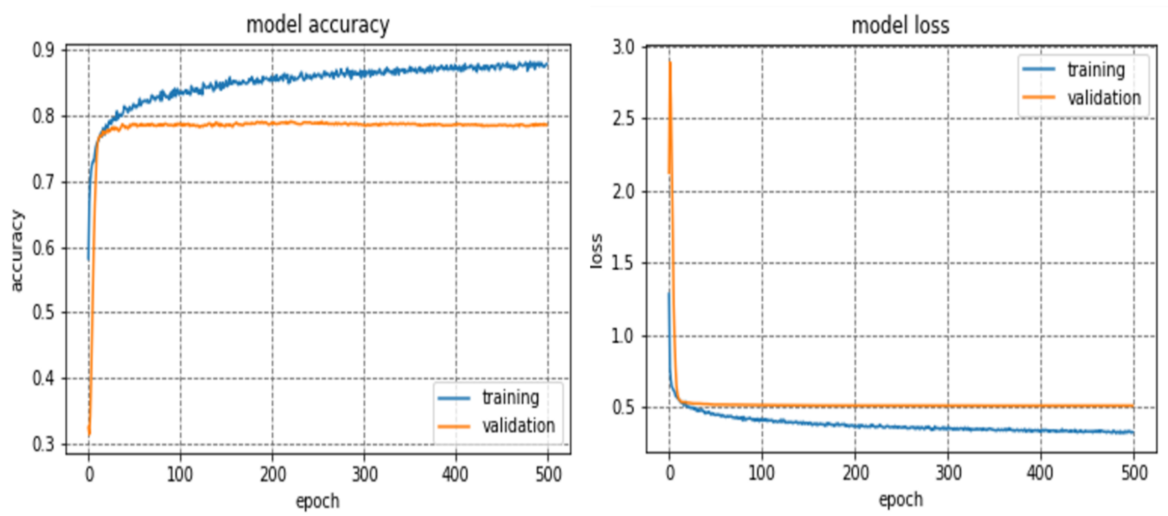


Figure 18: Performance of proposed integrated CWT and CNN method

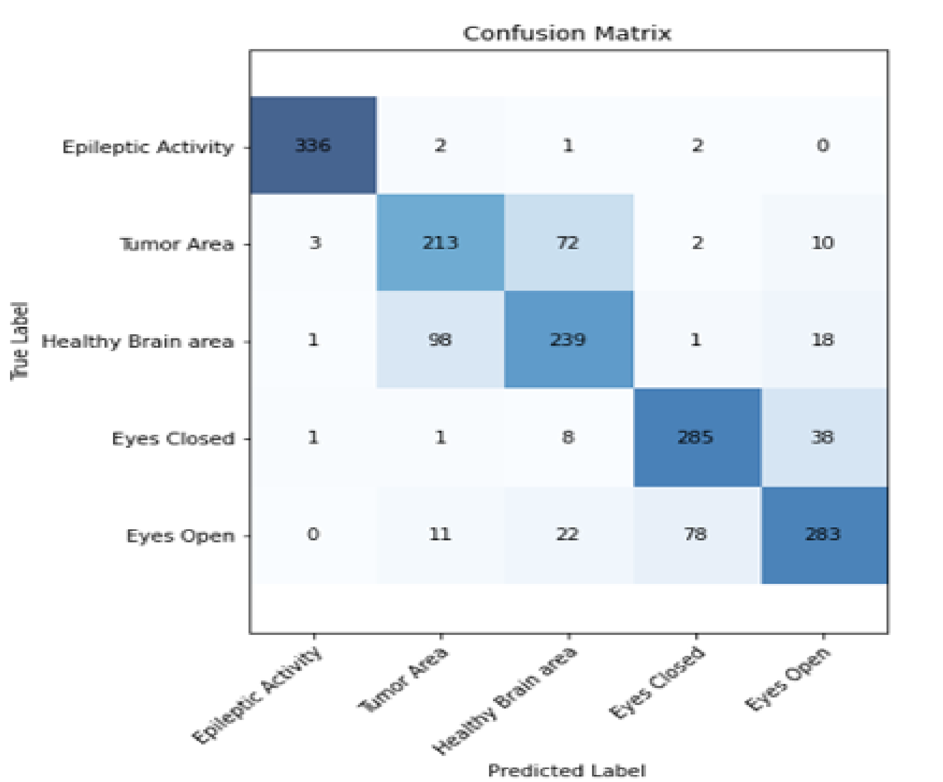


Figure 19: *Confusion Matrix*

Figure 19 shows the confusion Matrix for proposed integrated CWT and CNN method presents the predictions for samples from each class. The number and color of each cell in confusion matrix refer to correctly and incorrectly identified samples. Darker cell color means large number of correctly identified samples from each class whereas, the lighter shade cells refer to incorrectly classified samples. It can be observed that the Epilepsy data has the highest precision than other classes and Healthy Brain Area has the lowest precision of all classes.

CHAPTER 6: Conclusion

In present work, we demonstrated that Deep learning models are advantageous for EEG classification and timely prediction of epileptic seizures to avoid damage caused by recurrent seizure occurrences. We proposed an Integrated CWT and CNN method to classify EEG data and detect seizures caused by Epilepsy and Brain Tumors. The configurations of three existing Deep Learning models are experimented on Epilepsy Dataset[3] and their results are compared to the proposed Integrated CWT and CNN method.

The proposed CWT and CNN method generated better accuracy and loss results in timely manner. As shown in Table 3, our program generated better loss and accuracy results than Baseline[37]. Specifically, the proposed Integrated CWT and CNN achieved better test accuracy than GoogleNet, VGG16, AlexNet, and Baseline [37]. Consequently , the proposed Integrated CWT and CNN method has better loss results than VGG16, AlexNet and Baseline [37]. Moreover, the proposed Integrated CWT and CNN method has better learning time against GoogleNet, VGG16, AlexNet and Baseline [37].

For future work, we hope to improve performance time, and Accuracy of proposed Integrated CWT and CNN method. Additionally, we aim to reduce the number of false positives (cf. **Figure 19**.) while performing EEG classification on Epilepsy Dataset[3]. The intent of this research is to utilize the proposed method in medical applications for early seizure and brain tumor detection in future. Further study is required to refine the performance of proposed Integrated CWT and CNN method and achieve maximum classification accuracy with minimum loss score.

Appendix-A

Reading data

```
func read_csv:
        data=[]
        lables=[]
        dataframe=read_csv()
        labels= label values for each sample in dataframe
        data= append data points rows  from dataframe into array
return data and labels
```

Data Normalization

```
func Normalization():
        data_norm
        scalar= standard zero mean and unit variance scalar()
        data_norm=Normalize data using scalar
return data_norm
```

Data Division

```
Func data_divide:
        TrainingData=[]
        ValidationData=[]
        TestingData=[]
        TrainingData ValidationData = train_test_split(fraction=0.3)
        TestingData = train_test_split(frac=0.5)
        Save TrainingData _csv, ValidationData _csv, TestingData _csv
Return TrainingData _csv, ValidationData _csv, TestingData _csv
```

Scalograms

```
Func CWT():
        Scale=S
        Rescale=R
        Coefficients=[]
        Rescale_ Coefficients=[]
        N_sample=number of samples
        For sample in range(N_sample):
                Coefficients =continuous wavelet transform(sample)
                Rescale_ Coefficients= Resize coefficients SxR
Return Rescale_ Coefficients
```

Function Main

```
Func main():
        DATA=read_csv()
        DATA_NORM=Normalization()
        DATA_DIVISION= data_divide()
        IMAGES=CWT()
        reshaped_images =Reshape images to RxSx1
        MODEL=train(reshaped_images, train_labels)
        Save MODEL
        Save Weights
```

Testing Phase

```
load model
read testing_data_csv
for each sample in  testing_data_csv:
        TESTING_IMAGES = CWT()
        Reshape TESTING_IMAGES to SxRx1
        predict Using trained model
        True_Output = write original label in a file
        Predicted_Output = write predicted label in a file
        accuracy= evaluate model on  True_Output and Predicted_Output
return Accuracy
```

REFERENCES

[1] A. Adam, M. I. Shapiai, M. Zaidi, M. Tumari, M. S. Mohamad and M. Mubin, "Feature Selection and Classifier Parameters Estimation for EEG Signals Peak Detection Using Particle Swarm Optimization", *The Scientific World Journal,* Vol. 2014, Article No. 973063, 2014

[2] H. U. Amin, W. Mumtaz, A. R. Subhani, M. Naufal, M. Saad and A.S. Malik, "Classification of EEG Signals Based on Pattern Recognition Approach", *Frontiers in Computational Neuroscience,* Vol. 11, p. 103, 2017.

[3] R. G. Andrzejak, K. Lehnertz, C. Rieke, F. Mormann, P. David and C. E. Elger, " Indications of nonlinear deterministic and finite dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state", *Physical Review E Stat Nonlin Soft Matter Phys.,* Vol.64, p. 061907, 2001.

[4] S. Asadzadeh, T. Y. Rezaii, S. Beheshti, A. Delpak and S. Meshgini, "A systematic review of EEG source localization techniques and their applications on diagnosis of brain abnormalities", *Journal of Neuroscience Methods,* Vol 339, 2020.

[5] U. I. Awan, U. H. Rajput, G. Syed, R. Iqbal, I. Sabat and M. Mansoor, "Effective Classification of EEG Signals Using K-Nearest Neighbor Algorithm", *2016 International Conference on Frontiers of Information Technology (FIT),* pp. 120-124, 2016.

[6] O. Ayedemir, S. Pourzare and T. Kayikcioglu, "Classifying Various EMG and EOG Artifacts in EEG Signals", *PRZEGLĄD ELEKTROTECHNICZNY*, Vol.88, pp. 218-222, 2012.

[7] B. Boashash, N. A. Khan and T. B. Jabeur, "Time-frequency features for pattern recognition using high-resolution TFDs: A tutorial review", *Digit. Signal Process.* , Vol. 40, pp.1–30, 2015.

[8] Briassouli; D. Matsiki and I. Kompatsiaris," Continuous wavelet transform for time-varying motion extraction", *IET Image Processing,* Vol. 4, pp. 271 – 282, 2010.

[9] D. A. Bridwell, S. Rachakonda, R. F. Silva, G, D. Pearlson and V. D. Calhoun, "Spatio-spectral Decomposition of Multi-subject EEG: Evaluating Blind Source Separation Algorithms on Real and Realistic Simulated Data", *J. Springer Science+Business Media,*Vol.31, pp. 47-6, 2018.

[10] G. Cheng, P. C. Zhou and J. W. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images*", IEEE Transactions on Geoscience and Remote Sensing*", Vol. 54, 7405–7415, 2016.

[11] A. Craik, Y. He and J. L. C. Vidal, "Deep learning for electroencephalogram (EEG)classification tasks: a review", *Journal of Neural Engineering,* Vol. 16, 2019.

[12] I. Daubechies " Ten lectures on wavelets. *Computational Physics,* 1992.

[13] R. Dhanapal and D. Bhanu, "ELECTROENCEPHALOGRAM CLASSIFICATION USING VARIOUS ARTIFICIAL NEURAL NETWORKS", *Journal of critical Reviews,* Vol. 7, no. 4, pp. 891-894, 2020.

[14] C. Dong, C. C. Loy, K. M. He and X. O. Tang, "Image super-resolution using deep convolutional networks", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.38, pp.295–307, 2016.

[15] O. Faust, U. R. Acharya, H. Adeli and A. Adeli, "Wavelet-based EEG processing for computer-aided seizure detection and epilepsy diagnosis", *Seizure,* Vol. 26, pp. 56 - 64, 2015.

[16] G. Fu, C. J. Liu, R. Zhou, T. Sun and Q. J. Zhang, "Classification for high resolution remote sensing imagery using a fully convolutional network", *Remote Sensing,* Vol. 9, no. 5, Article no. 498, 2017.

[17] F. Gama, A. G. Marques, G. Leus and A. Ribeiro, "Convolutional Neural Network Architectures for Signals Supported on Graphs," *IEEE Transactions on Signal Processing*, Vol. 67, pp. 1034-1049, 2019.
[18] S. Garg and R. Narvey, "DENOISING & FEATURE EXTRACTION OF EEG SIGNAL USING WAVELET TRANSFORM", *IEEE- presented in International Journal of Engineering Science and Technology (IJEST),* Vol. 5 No.06 pp. 1249-1253, 2013.
[19] P. Ghorbanian, D. M. Devilbiss, A. J. Simon, A. Bern-stein, T. Hess et al., "Continuous Wavelet Transform EEG Features of Alzheimer's Disease*", 5th Annual Dynamic systems and control conference*, paper no. DSCC2012-MOVIC2012-8684, pp. 567-572, 2013.
[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation*", IEEE Transactions on Pattern Analysis ans Machine Intelligence*, Vol.38, pp. 142–158, 2016.
[21] E. Gomez-Luna, D. Silva, G. Aponte, J. G. Pleite and D. Hinestroza, "Obtaining the Electrical Impedance Using Wavelet Transform from the Time Response", *IEEE Transactions on Power Delivery*, Vol. 28, pp. 1242-1244, 2014.
[22] M. G. Gong, H. L.Yang and P. Z. Zhang, "Feature learning and change feature classification based on deep learning for ternary change detection in SAR images", *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 129, pp. 212–225, 2017.
[23] M. P. Hosseini, A. Hosseini and K. Ahi, "A Review on Machine Learning for EEG Signal Processing in Bioengineering", *IEEE Reviews in Biomedical Engineering,* Vol. PP, 2020.
[24] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, et al., "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis", *Proceedings of the royal society. Math. Phys. Eng. Sci.*, Vol. 454, pp. 903–995, 1998.
[25] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", *Proceedings of the 32nd International Conference on Machine Learning,* Vol. 37, pp. 448-456, 2015.
[26] J. Gilles, " Empirical Wavelet Transform", IEEE Transactions on Signal Procedings, Vol. 61, pp.3999–4010, 2013.
[27] N. Jrad and M. Congedo, "Identification of spatial and temporal features of EEG", *Neurocomputing,* Vol.90, pp. 66-71, 2012.
[28] G. Kaiser, "A friendly guide to wavelets", Birkhauser, Boston,1994).
[29] K. Kamnitsas, C. Ledig, V. F. J. Newcombe, J. P. Sirnpson, A. D. Kane et al., "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation", *Medical Image Anal*ysis, Vol. 36, pp. 61–78, 2017.
[30] M. R. N. Kousarrizi, A. A. Ghanbari, M. Teshnehlab, M. Aliyari and A. Gharaviri, "Feature Extraction and Classification of EEG Signals using Wavelet Transform, SVM and Artificial Neural Networks for Brain Computer Interfaces", *International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing,* Vol. 1, pp. 352-355, 2009.
[31] A. Krizhevsky, I. Sutskever and G. E. Hinton, " ImageNet Classification with Deep Convolutional Neural Networks", *Proceedings of the 25th International Conference on Neural Information Processing Systems(NIPS'12),* Vol. 1, pp. 1097–1105, 2012.
[32] N. Kumar, K. Alam and A. H. Siddiqi, "Wavelet Transform for Classification of EEG Signal using SVM and ANN", *Biomedical & Pharmacology Journal*, Vol.10, pp.2061-2069, 2017.
[33] M. Kwon, S. Han, K . Kim and S. C. Jun, " Super-Resolution for Improving EEG Spatial Resolution using Deep Convolutional Neural Network—Feasibility Study", *Sensors,* Vol.19, no.23, 2019.

[34] M. Langkvist, A. Kiselev, M. Alirezaie and A. Loutfi, “Classification and segmentation of satellite ortho-imagery using convolutional neural networks” *Remote Sensing*, Vol. 8, no. 4, Article no. 329, 2018.
[35] C. T. Lin, C. S. Huang, W. Y. Yang, A. K. Singh, C. H. Chuang et al., “Real-Time EEG Signal Enhancement Using Canonical Correlation Analysis and Gaussian Mixture Clustering”, *Journal of Healthcare Engineering*,Vol.2018, pages.11, 2018.
[36] Q. Liu, J. Cai, S. Z. Fan, M. F. Abbod, J. S. Shieh, Y. Kung and L. Lin, “Spectrum Analysis of EEG Signals Using CNN to Model Patient's Consciousness Level Based on Anesthesiologists' Experience, *IEEE*, Vol. 7, pp. 53731-53742, 2019.
[37] W. L. Mao, H. I. K. Fathurrahman, Y. Lee and T. W. Chang, “ EEG dataset classification using CNN method”, *Phys.: Conf. Ser.* , Vol.1456 012017, 2020.
[38] R. M. Mehmood and H and J. Lee, “EMOTION CLASSIFICATION OF EEG BRAIN SIGNAL USING SVM AND KNN”, *2015 IEEE International Conference on Multimedia and Expo Workshops*, Vol. 1, pp. 1-5, 2015.
[39] C. M. Michel and D. Brunet, “EEG Source Imaging: A Practical Review of the Analysis Steps”, *Frontiers in Neurology*, Vol. 10, 2019.
[40] P. A. Muñoz-Gutiérrez, E. Giraldo, M. Bueno-López and M. Molinas, “Localization of Active Brain Sources From EEG Signals Using Empirical Mode Decomposition: A Comparative Study”, *Frontiers in Integrative Neuroscience*, Vol. 12, 2018
[41] M. Nandish, S. Michahial, H. Kumar and F. Ahmed, “Feature Extraction and Classification of EEG Signal Using Neural Network Based Techniques”, *International Journal of Engineering and Innovative Technology (IJEIT)*,Vol.2, no. 4, 2012.
[42] V. T. N. Nguyen, V. T. Huynh and T. H. H. Nguyen, ” Electroencephalography Analysis Using Neural Network*”, 5th NAFOSTED Conference on Information and Computer Science (NICS)*, pp. 144-147, 2018.
[43] R.J. Oweis and W. E. Abdulhay, “Seizure classification in EEG signals utilizing Hilbert-Huang transform”, *Springer- BioMedical Engineering Online*, Vol. 10, p. 38, 2011.
[44] N. Padfield, J. Zabalza, H. Zhao, V. Masero and J. Ren, “EEG-Based Brain-Computer Interfaces Using Motor-Imagery: Techniques and Challenges”, *sensors*, Vol. 19, 2019.
[45] E. Parvinnia, M. Sabeti, M. Z. Jahromi and R. Boostani, “ Classification of EEG Signals using adaptive weighted distance nearest neighbor algorithm”, Journal of King Saud University – Computer and Information Sciences, Vol. 26, pp. 1-6, 2014.
[46] T. Rahman, A. K. Ghosh and M. M. H. Shuvo, “Mental Stress Recognition using K-Nearest Neighbor (KNN) Classifier on EEG Signals”, *International Conference on Materials, Electronics & Information Engineering (ICMEIE-2015)*, 2015.
[47] M. B. I. Reaz, M. S. Hussain, M. Ibrahimy and M. Mohd-Yasin, “EEG signal analysis and characterization for the aid of disabled people”, *Modelling in Medicine and Biology*, vol. 287, 2007.
[48] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T. H. Falk et al., “Deep learning-based electroencephalography analysis: a systematic review, *Journal of Neural Engineering*, Vol. 16, 2019.
[49] S. Ruder, "An Overview of Gradient Descent Optimization Algorithms," *CoRR. In arXiv*, Vol. abs/1609.04747, 2016.
[50] V. Saccà, M. Campolo, C. Mirarchi and F. C. Morabito, “On the Classification of EEG Signal by Using an SVM Based Algorithm, *Smart innovation, systems and technologies,Springer*, Vol. , p. 69, 2018.

[51] K. Samiee, P. Kovács and M. Gabbouj, "Epileptic Seizure Classification of EEG Time-Series Using Rational Discrete Short-Time Fourier Transform*", IEEE Transactions on Biomedical Engineering*, Vol. 62, pp. 541-552, 2015.

[52] M. Şeker and M. S. Özerdem, "Classification of EEG data sets with Hilbert transform", *24th Signal Processing and Communication Application Conference (SIU)*, pp. 1949-1952, 2016.

[53] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", *CoRR*, Vol. abs/1409.1556, 2014.

[54] S. and Y. Lib, "Designing a robust feature extraction method based on optimum allocation and principal component analysis for epileptic EEG signal classification", *Computer Methods and Programs in Biomedicine,* Vol. 119, pp. 29-42, 2015.

[55] A. Subasi and M. I. Gursoy, "EEG signal classification using PCA, ICA, LDA and support vector machines", *Expert Systems with Applications,* Vol. 37, pp. 8659–8666, 2010.

[56] S. K. Swee and L. Z. You, "Fast Fourier analysis and EEG classification brainwave-controlled wheelchair", *2016 2nd International Conference on Control Science and Systems Engineering (ICCSSE),* pp. 20-23, 2016.

[57] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed et al.," Going Deeper with Convolutions", *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 1-9, 2015.

[58] A. Tripathi, G. Srivastava, K. K. Singh and P. K. Maurya, "Epileptic Seizure Data Classification Using RBAs and Linear SVM", *Biomedical & Pharmacology Journal,* 2019.

[59] Q. Wang, X. D. Zhang, G. Z. Chen, F. Dai, Y. F. Gong et al., "Change detection based on faster r-cnn for high-resolution remote sensing images", *Remote Sensing Letters,* Vol. 9, pp. 923–932, 2018.

[60] T. Yang, K. S. Yu, J. Phua, T. Selvaratnam, V. Toh et al., "Image-based Motor Imagery EEG Classification using Convolutional Neural Network", *IEEE EMBS International Conference on Biomedical Health Informatics (BHI),* pp. 1-4, 2019.

[61] H. Yi, S. Y. Xin and J.F. Yin, "A Class of Algorithms for Continuous Wavelet Transform Based on the Circulant Matrix", *Algorithms,* Vol. 11, 2018.

[62] L. Zhao, Q. Zhang, Y. Li, H. Wang and X. Du, "Integrating the Continuous Wavelet Transform and a Convolutional Neural Network to Identify Vineyard Using Time Series Satellite Images, *Remote sensing,* Vol. 11, 2019.

[63] M. A. Zulkifley and S. R. Abdani, " EEG Signals Classification by using Convolutional Neural Networks*", SASSP(IEEE SYMPOSIUM ON ACOUSTIC, SPEECH AND SIGNAL PROCESSING),* 2019.

[64] G. C. Jana, R. Sharmab and A. Agrawal, "A 1D-CNN-Spectrogram Based Approach for Seizure Detection from EEG Signal", *International Conference on Computational Intelligence and Data Science (ICCIDS 2019),* Vol. 167, pp. 403-412, 2019.

[65] A. Emami, N. Kunii, T. Matsuo, T. Shinozaki, K. Kawai et al., "Seizure detection by convolutional neural network-based analysis of scalp electroencephalography plot images", *NeuroImage: Clinical*, Vol. 22, 2019.

[66] F. Li, F. He, F. Wang, D. Zhang, Y. Xia, and X. Li, "A Novel Simplified Convolutional Neural Network Classification Algorithm of Motor Imagery EEG Signals Based on Deep Learning", *Applied sciences,* Vol. 22, pp. 101684, 2020.

[67] D. Zhou and X. Li, "Epilepsy EEG Signal Classification Algorithm Based on Improved RBF", *Frontiers in Neuroscience,* Vol. 14, pp. 606, 2020.

[68] K. M. Almustafa, "Classification of epileptic seizure dataset using different machine learning algorithms", *Informatics in Medicine Unlocked,* Volume 21, Article no. 100444, 2020.

[69] H. Liu , L. Xi , Z. Li and Y. Zhao, “Using Deep Learning and Machine Learning to Detect Epileptic Seizure with Electroencephalography (EEG) Data, *Machine Learning Research,* Volume 4, Issue 3, Pages: 39-44, 2019.

[70] M. Soufineyestani , D. Dowling and A. Khan, “Electroencephalography (EEG) Technology Applications and Available Devices”, *applied Sciences,* Vol 10, no. 21, Article no.7453,2020.

[71] Y. Gao, B. Gao, Q. Chen, J. Liu and Y. Zhang, “Deep Convolutional Neural Network-Based Epileptic Electroencephalogram (EEG) Signal Classification”, *Frontiers in Neurology,* Vol. 11, pages 375, 2020

[72] J. W. Lee, P. Y. Wen and S. Hurwitz et al., “Morphological Characteristics of Brain Tumors Causing Seizures” , *Archives of Neurology,* Vol. 67, Article no. 3, pp. 336-342, 2010

[73] I. Ullah, M. Hussain, E. H. Qazi, and H. Aboalsamh, “An Automated System for Epilepsy Detection using EEG Brain Signals based on Deep Learning Approach”, *Expert systems with applications,* Vol. 107, pp. 61-7, 2018

[74] A. R. Johansen, J. Jin, T. Maszczyk, J. Dauwels, S. S. Cashs et al. “EPILEPTIFORM SPIKE DETECTION VIA CONVOLUTIONAL NEURAL NETWORKS” Proc IEEE Int Conf Acoust Speech Signal Process 2016 , 2016, pp.754-758, 2016.

[75] X . Wei, L. Zhou, Z. Chen, L. Zhang and Y. Zhou, “Automatic seizure detection using three-dimensional CNN based on multi-channel EEG”, *BMC Medical Informatics and Decision Making,* Vol.18(Suppl 5):111, 2018

[76] Chandaka et al. “Cross-correlation aided support vector machine classifier for classification of EEG signals”, 2009

[77] Li et al. “Feature extraction and recognition of ictal EEG using EMD and SVM”, 2013

[78] Kumar et al. “Wavelet Transform for Classification of EEG Signal using SVM and ANN”, 2017

[79] Zhou et al, “Epileptic Seizure Detection Based on EEG Signals and CNN” 2018

[80] Turk et al. “Epilepsy Detection by Using Scalogram Based Convolutional Neural Network from EEG Signals”, 2019

[81] Chen-SenOuyang, Rei-ChengYang, Ching-TaiChiang, Rong-ChingWu Lung-ChangLin, “EEG autoregressive modeling analysis: A diagnostic tool for patients with epilepsy without epileptiform discharges”, Volume 131, Issue 8, August 2020, Pages 1902-1908.

[82] Michael Unser,Akram Aldroubi “A Review of Wavelets in Biomedical Applications” , Proceedings of the IEEE, vol. 84, no. 4, pp. 626-638, April 1996.

[83] Norden E. Huang Zhaohua Wu , “A review on Hilbert-Huang transform: Method and its applications to geophysical studies”, Reviews of Geophysics, Volume 46, Issue 2, 06 June 2008.

[84] Yann Lecun, Koray Kavukcuoglu, Clement Farabet “Convolutional Networks and Applications in Vision”, Conference: International Symposium on Circuits and Systems (ISCAS 2010), 2010.