Epilepsy detection from EEG data using 2D Convolutional Neural Network

1st Adnan Salman
Computer Science Department
An-najah National University
Nablus, Palestine
aalshaikh@najah.edu

***Abstract*—An epileptic seizure, a disorder in brain functionality, happens when electrical bursts spread across the brain, causing the person to lose control or consciousness. Predicting epileptic seizures before they occur is useful for seizure prevention with medicine or for neural pre-surgical planning. Machine learning and computational methods are used to predict epileptic seizures from electroencephalogram (EEG) recordings. However, noise removal and feature extraction in EEG data are two important challenges that have a negative impact on the effectiveness of both time and true positive prediction rate. We offer a model in this paper that provides a reliable strategy for both preprocessing and feature extraction. Our model is based on a two-dimensional Convolution Neural Network (CNN), with EEG input provided to the CNN in the form of two-dimensional images. On the University of Bonn data set's normal vs intericatl vs ictal instance, our proposed system achieved 97.8 percent accuracy and over 97 percent for the other parameters, which include precision, recall, F1-score, and ROC-AUC. Our findings are reproducible using the code available on github [1].**

***Index Terms*—epilepsy, convolution neural networks, machine learning, eeg, seizure**

I. Introduction

The human brain, the central part of the nervous system, is a complex network that contains billions of neurons and thousands of synapses per neuron that communicate with each other using electrochemical signals. Any condition that affects brain functions is considered a brain disorder. This includes conditions caused by illness, genetics, or traumatic injury. Excessive and abnormal firing of electrical signals in the brain causes a brain disorder called a seizure. The condition of frequent seizures is called epilepsy. Epilepsy is the fourth most common neurological disorder that affects about fifty million people worldwide at all ages [2]. Therefore, it is important to periodically monitor the brain to manage and prevent seizures.

An Electroencephalogram (EEG) is a noninvasive monitoring technology used widely to record electrical activities of the brain and detect potential problems associated with these activities, such as epilepsy, sleep disorders, and encephalitis. EEG is a low-cost, non-invasive medical tool that displays the status of the brain. It proved to be a powerful technique for diagnosing epilepsy. However, manual analysis of long recordings of EEG signals by neurologists is time-consuming and requires the availability of an experienced neurologist. In response, several automatic methods were proposed to help neurologists detect epilepsy in EEG signals [3]–[6]. Several of these methods have been used to distinguish between binary epileptic states, such as seizure vs. non-seizure or normal vs. ictal. However, the majority of them have difficulties distinguishing between the three states: ictal, normal, and inter-ictal. As a result, automatic diagnosis of brain disorders from EEG signals is a critical job in the field of neuroscience.

The majority of existing epilepsy detection algorithms involve signal processing to extract features from the EEG signal, which are then used to train a machine learning model to classify the EEG data. This strategy frequently produces excellent accuracy in the case of binary classification. However, it performs badly when comparing normal vs. ictal vs. inter-ictal [4]. As a result, machine learning algorithms must overcome a number of challenges in order to tackle this problem. Among these difficulties is a scarcity of EEG data pertaining to epilepsy. Furthermore, the presence of noise in the data creates an extra barrier for machine learning algorithms that aim to avoid learning from the noise. In this research, we present an EEG classification strategy based on grayscale images and a deep two-dimensional Convolution Neural Network (CNN). By transforming the EEG data to a 2D EEG image, the process of noise filtering, feature extraction, and feature reduction is simplified. CNN will handle these specifics. In this technique, CNN acts as a neurology expert examining EEG signals. When compared to earlier work, our results indicate an improvement in detection.

The proposed augmentation approach in [4], [7] is utilized to increase the quantity of the training data, which improves accuracy. By augmenting two-dimensional EEG images, CNN may learn with several views on a single EEG image. In addition, using EEG images for classification improves the model's robustness when compared to the one-dimension model, which is susceptible to noise. When the EEG image is fed into a CNN, the proposed model ignores the noise input while extracting essential characteristics throughout the convolution layers. Another advantage of this model is that it may be used with EEG inputs from multiple EEG devices with different sampling rates and amplitudes. While learning based on 1d signals may entail the creation of a new model for each device. The proposed model can be utilized for long-term

978-1-6654-3279-5/21/$31.00 ©2021 IEEE

EEG signal monitoring and can assist specialists in identifying brain illnesses.

The remainder of this paper is structured as follows. Section 2 includes a summary of relevant work. Section 3 goes into detail on the approaches utilized for EEG classification, including a convolutional neural network classifier. Section 4 discusses our EEG classifier evaluation results. Finally, Section 6 summarizes the paper's discussion and conclusion.

II. Related Works

Classification of EEG signals into epileptic and non-epileptic involves extracting features from the EEG signals and using these features in training a machine learning model. In the literature, several feature extraction approaches and several classification methods are used. Several groups used the Discrete Wavelet Transform for feature extraction and then they applied different machine learning algorithm. Guo et al. [8] extracted features such as entropy, standard deviation, and energy from EEG signals using discrete wavelet transform (DWT). For classification, they used Artificial Neural Network (ANN). They reported 97.8% accuracy for the seizure vs non-seizure case (ABCD vs E) on the University of Bonn dataset. Swami et al. [9] took a similar method, employing the DWT to extract features including entropy, root-mean-square, and energy. They used the regression neural network classification approach on the University of Bonn dataset and reported an accuracy of 100 percent for seizure versus non-seizure (A-E) cases and 98.2 percent for normal vs seizure cases (AB-E).

Other studies utilized DWT to extract the attributes but classified the data using SVM and probabilistic neural networks (PNN) [10]. For normal vs. seizure classification, they reported a 95.44 percent accuracy rate (ABCD-E case). In a similar work, Shoeb [11] used energy as a feature and SVM as a machine learning classifier, as well as a patient-specific prediction approach. Permutation entropy was utilized as a feature, and the SVM was used as a classifier, by Nicolaou et al. [12]. On the University of Bonn dataset, they reported a 93.55 percent accuracy for the A-E example. The maximum accuracy was 86.1 percent in the remaining cases. The accuracy rate was claimed to be 96%. In addition, wavelet transform was employed for feature extraction in Khan et al's work [13].

The authors of [14] employed the wavelet transform to obtain the tensor representation of EEG signals and principal components analysis to extract features. The Bayesian Linear Discriminant Analysis (Bayesian LDA) was used to classify the tensor distance features, with a reported accuracy of 97.60 percent. Acharya et al. [15] employed four entropies features to detect seizures and compared the performance of seven different classifiers based on classification accuracy. The Fuzzy Sugeno classifier was shown to be the best model, while the Naive Bayes classifier was found to be the worst. The authors of [16] used spectral features derived from the Discrete Wavelet Transform (DWT) and the Discrete Fourier Transform in five frequency bands (DFT). For classification, a neural network (NN) classifier based on improved particle swarm optimization (IPSO) is used. They stated " their sensitivity had improved". Similar approch is done in [17] where they used Chebyschev filter. Fractal geometry is employed as a feature of EEG data in [3]. The neural network is trained using an extreme learning machine (ELM). The sensitivity was 91.72 percent and the specificity was 94.89 percent.

In [4], the features were extracted using variational mode decomposition and quadratic feature extraction, and the classification was done using Random Forest. For the case of normal vs interictal vs ictal, they reported a 97.4 percent accuracy. Ihasan et al. [7] employed an ensemble learning technique based on voting scheme and a 1D convolutional neural network model. They reported that the CNN findings were 96.1 percent accurate, and that the ensemble learning results were 99.1 percent accurate. In [18], multi-modal Machine Learning is used to integrate EEG features. EEG from patients were used to discriminate healthy subjects from patients. Continuous Wavelet Transform used to extract features from EEG sub-bands. They reported that a neural network classifier outperforms Logistic Regression and Support Vector Machine.

A convolutional network was used in another study to learn seizure characteristics from EEG data. Then, to understand the temporal dependencies in EEG data, these features are loaded into a Nested Long Short-Term Memory (NLSTM) model. They stated that this model performed well, with an accuracy range of 98.44-100 percent [5]. Convolutional neural networks are also used in [5]. They reported accuracy and specificity scores of 99.3 percent and 99.6 percent for the CHB-MIT dataset, respectively, as well as 98.0 percent and 98.3 percent for patients [6]. Patel et al [19] proposed seizure detection in ambulatory EEG. On 13 different subjects, they compared various classifiers. When trained and assessed on a single subject, linear discriminant analysis (LDA) performs best. It has an overall accuracy of 76.5 percent.

According to the preceding research, most feature extraction algorithms are theoretically designed using statistical or numerical methods and are not data-adaptive. To improve the accuracy and generalization of an epilepsy detection system, a deep learning technique is utilized instead of feature extractors and classifiers. With an accuracy of 99.1%, the deep learning algorithms proposed in [7] outperform all previous approaches. They employed a 1D convolutional network and ensemble learning to improve their method. As a result, we motivated to follow their example and use a deep learning method, but this time using a 2D convolutional network rather of a 1D convolutional network, and to apply ensemble learning scheme to improve the results.

III. Method

The suggested CNN classifier for detecting epilepsy from EEG signals is divided into two steps: preprocessing and classification. During the preprocessing step, the EEG signals are divided into sub-signals using an augmentation strategy. The sub-signals are then converted to 2D images. The 2D images are used as input to the CNN classifier in the classification step, which performs classification of the three epilepsy states.

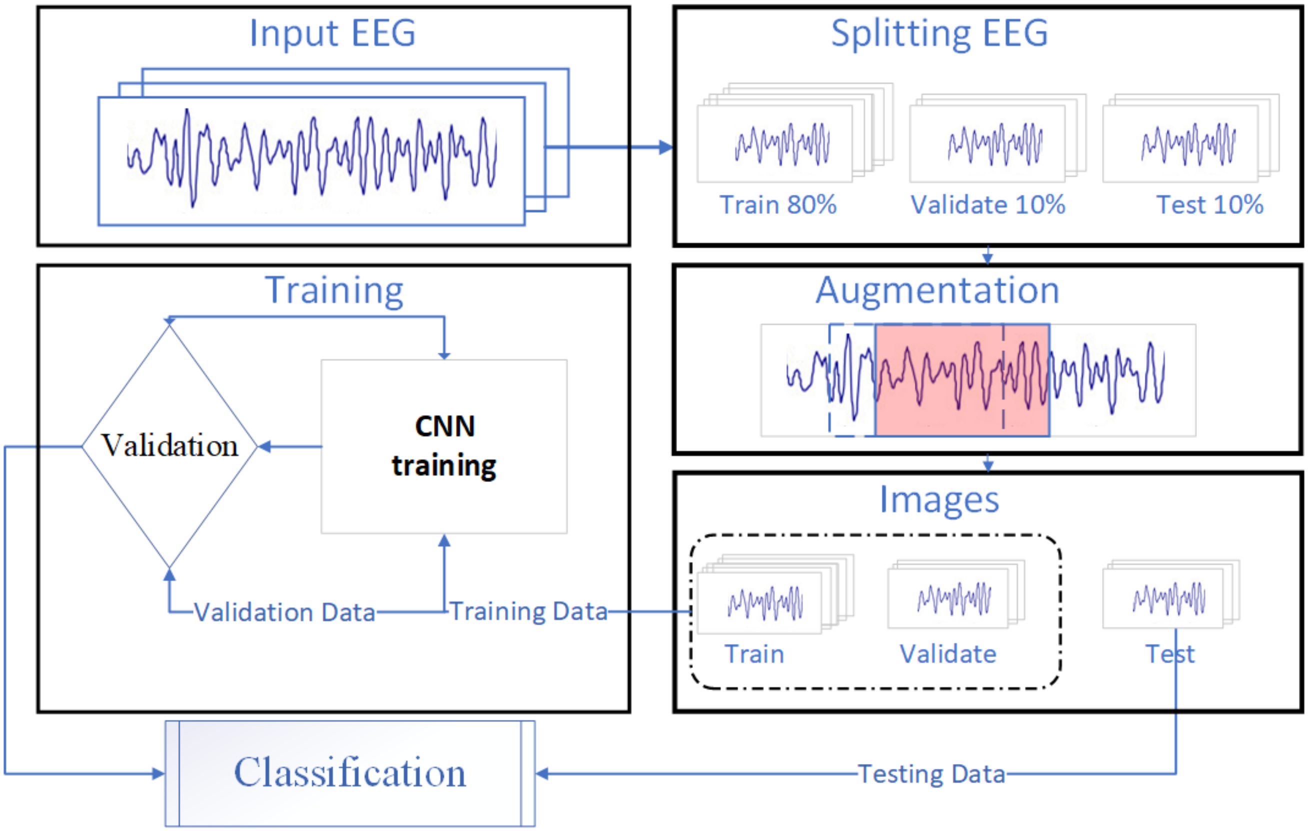

Fig. 1. System Architecture

The complete technique is depicted in Figure 1 and is detailed below.

A. Data set

In this research, we used the University of Bonn epilepsy dataset for CNN model training and testing [20]. The data set is divided into five sections labeled A, B, C, D, and E. Each set has 100 recordings of brain signals. Signals from the surface of the cerebral cortex were recorded from five healthy participants in groups A and B. Group A is recorded when the eye is open, and group B is recorded when the eye is closed. Groups C and D contain the EEG records of patient individuals between seizures. EEG signals from the epileptic region (epileptogenic zone) are found in Group D, but signals from the hippocampus on the opposite side of the epileptic area are found in Group C. Signals from patient individuals' seizure activities are contained in Group E. Figure 2 illustrates a representative sample from each group. Each record is 23.6 seconds long. Each set has 4097 samples. 173.61 hertz is the sampling rate. The data set is in text format, with 100 text files in each category.

Because the purpose of this study is to categorize EEG signals into three categories: normal, interictal, and ictal, the normal group is generated by joining groups A and B. The interictal group is formed by combining groups C and D, while group E makes up the ictal group. The available signals are separated into three categories: training, validation and testing sets. The test data set is created by randomly selecting 10% of each group's files. The remaining 90% of the data is used for training and validation.

B. Pre-processing

The number of examples in this dataset is insufficient to build a comprehensive learning model. As a result, the model is likely to overfit. Obtaining a larger number of EEG signals is usually not possible, and recognizing these cases by expert neurologists is a challenging task. As a result, in order to improve training and generalization accuracy, we adopted an augmentation technique to expand the size of the data set. In addition to increasing the amount of samples, the augmentation scheme can be used to keep the data balanced between classes. This is relevant in general, but particularly in medical imaging, because the vast majority of them are normal, with only a few abnormal instances. The augmentation method proposed in [4] and employed in [7] is also used in this work. The data set is augmented in this technique by using a sliding window of size w samples and a stride of s samples, as shown in Figure 1.

Because each EEG file provides a signal with a sample size of $n = 4097$. Each signal can be augmented with m sub-signals, where m equals to $floor((n-w)/s)+1$. Each sub-signal is handled as a seperate sample in training the model.

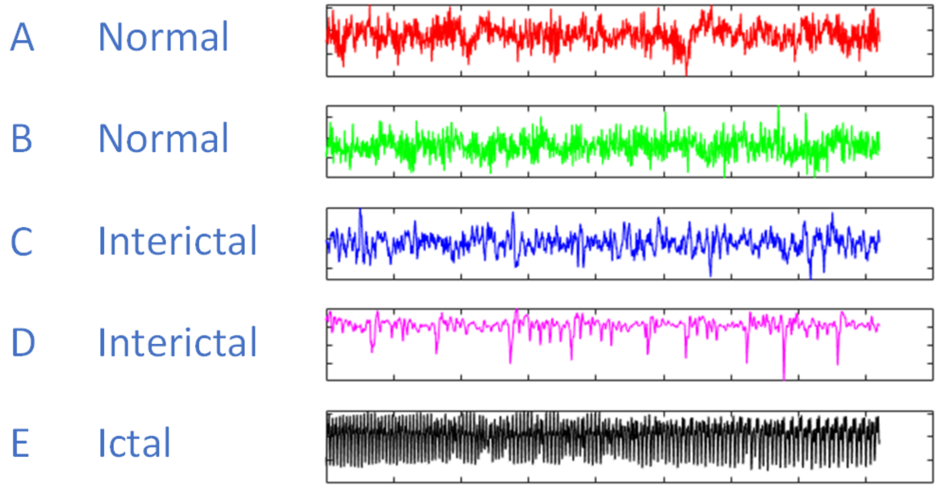

Fig. 2. University of Bonnepilepsy data set

In the case of unbalanced data sets, we used $w = 512$ and $s = 64$ in the training, validation and testing data sets for all groups. This would yield 57 sub-signals from each signal, for a total of 28500 sub-signals. We emphasize here that the data splitting technique is based on signals rather than sub-signals. This means that all sub-signals of a given signal will be in either the training, validation, or testing data sets, but not in more than one set in order to avoid data leaking. Using the augmentation strategy, we can balance the samples among the classes; for example, we can use $w = 512$ and $s = 32$ for group E to nearly quadruple the amount of sub-signals from group E. In this paper, we kept the data unbalanced, but we adopted a stratified sampling in all our experiments.

Because the CNN model's inputs are 2D EEG images, employing different overlapping parts of the EEG signals expands the training set while maintaining performance. In the next step, the EEG data was converted to a 2D image by plotting each EEG sub-signal on fixed-sized axes, and the figure frame was captured as a 2D gray-scale image. For this task, we used the Matlab environment. These images were then used to train and evaluate the performance of the convolution neural network. In this study, we investigated image resolutions of 64×64 and 128×128.

C. EEG classifier

In this study, we used CNN as an epilepsy classifier. CNN was created in response to the Artificial Neural Network's (ANN) limitations in image classification. The exponential increase in the number of free parameters made ANN ineffective for image classification. The reason is that ANN does not take the image's topology into account while CNN can extract local visual features automatically utilizing convolution and pooling layers which reduces the need for signal processing and the challenge of feature engineering. As a result, using 2D CNN is similar to how a neurologist analyzes EEG information

There are various CNN architectural models, including AlexNet, GoogleNet, VGGNet, ResNet, and DenseNet. GoogleNet and VGGNet won first and second in the ILSVRC-2014 classification challenge, respectively. However, VGGNet is more popular due to its simpler structure. ResNet and DenseNet are two more recent CNN models. They developed a deep neural network to overcome the vanishing gradient problem, which happens when early features have little influence on the output. In this work, we looked at VGGNet's design, then modified its structure and evaluated its performance in identifying EEG signals. The architecture of the CNN that we utilized is shown in Figure 3.

As shown in Figure 3, the proposed CNN architecture contains six convolutional layers interspersed with three max pooling layers for feature extraction and two fully-connected layers. The convolutional layer kernel size is 3x3 and has a stride of 1 and a padding of 1. Each convolutional layer is made up of several channels. The architecture begins with a small number of kernels equal to 64 in the first two layers, then climbs to 128 in the following two layers, and finally 256 in the last two layers. The convolutional layer's basic principle is that it learns the hierarchy of low to high-level features from the provided input image. These features are obtained via convolution of the input with a number of filters learned during training.

A max pooling layer is inserted after each block of convolution layers to reduce the spatial dimension. The max pooling layers use 2×2 kernels with a stride of 2, which reduces the size of the previous layer by half. The function of this layer is to summarize the features activated by the convolutional layer. To accelerate convergence, we used batch normalization after each convolutional layer, followed by an activation function. The ReLu activation function is employed in this work. A dropout layer is used after each convolutional layer block and after the fully connected layer to decrease or eliminate overfitting. The Fully connected layer is the most effective way for learning non-linear features generated by the convolutional layer. The semantic representation of the high-level features is provided to the softMax classifier in the final layer, which

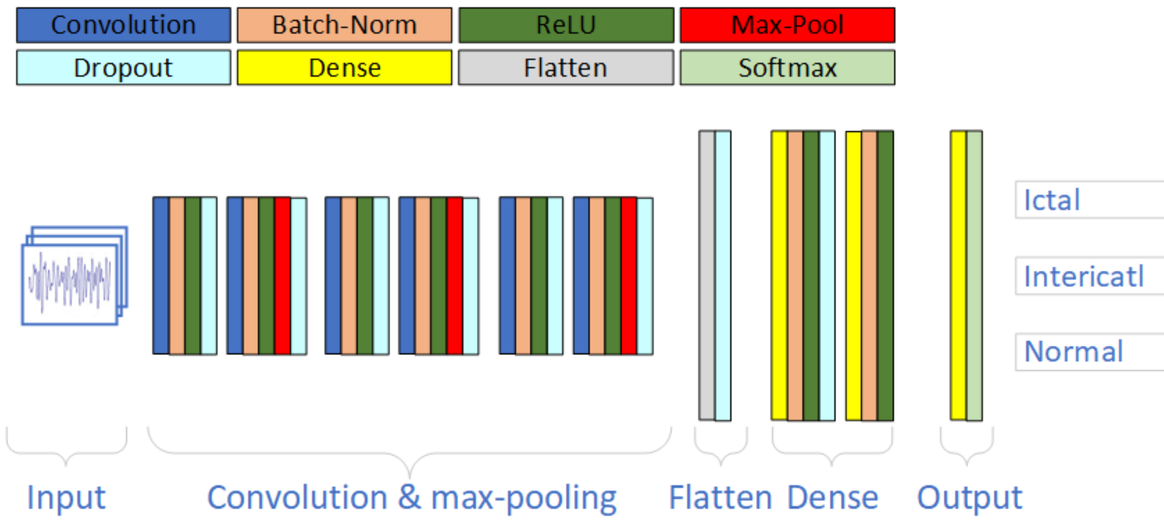

Fig. 3. The convolution neural network structure

predicts the class of the input EEG image. Figure 3 depicts an overview of our model at a high level. Some of the CNN parameters are covered in further detail below.

a) **Kernel initialization::** In deep learning, the initialization of the kernel weights is important for optimal convergence of the gradient descent to avoid being trapped in local minima. Keras allows a variety of kernel and bias initialization options per layer. After experimenting with different initializers, we used the Glorot normal initializer, which is also called the Xavier normal initializer.

b) **Activation function::** The primary purpose of the activation function is to introduce non-linearity to the system which allows it to learn complex pattern from the data. If no activation function is applied, the neural network would be just a linear regression model. Several non-linear function are used in the literature including rectified linear units (ReLU), Leakage rectified linear unit (LReLU), and exponential linear units (ELU). In this paper we experimented with ReLU and LReLU and ELU activation functions.

c) **Batch normalization::** The goal of batch normalization is to speed up convergence and increase the stability of a network. It works by normalizing the output of a layer by subtracting the batch mean and dividing by the standard deviation. The reason for this is that in a deep network, a small change in a parameter can cause a large influence on the input distribution of the next layer which is known as internal covariate shift problem. The goal of batch normalization is to reduce this effect. Batch normalization is usually applied just before the activation function and after the convolution layer and it can be applied after the activation of a layer including the fully connected block.

d) **Regularization::** Regularization is used to minimize overfitting when training the network. L1 and L2 norm are used in most recent networks and they require a hyperparameter to be tuned. Keras allow adding a penalty for weight size to the loss function in what is called weight regularization. By default no regularization is used in any layer. Another approach for regularization is dropout. It is a form of regularization to reduce overfitting by randomly dropping out nodes during training. This technique is computationally cheap and provides an effective regularization method in deep neural networks and it improves the generalization of deep neural networks. Dropout reduces dependency between layers. The net effect is a voting effect by the model combination. In this paper, we applied a dropout with a probability of 0.5. The dropout layer is placed after the batch-normalization layer.

e) **Cost and optimizer function::** The cost or fitting function indicates how well the network has been trained. It computes the difference between the expected values and the desired output, with the objective of minimizing this difference. Cost functions of various forms have been employed. We used cross-entropy, which is commonly used in deep learning problem classification.

D. Performance Measures

The evaluation metrics typically used in the binary classification include: Accuracy, Precision, Recall, f1-score, and AUC-ROC. The calculation of these measures is based on the outcome of the classification model which can be written in a confusion matrix form. In case of binary classification, the confusion matrix is a table with 4 different combinations of predicted and actual values. The elements of the matrix TP, TN, FP, and FN correspond to the True Positive, True Negative, False Positive, and False Negative counts, respectively. These measures are defined as follows:

- Accuracy: The fraction of correct predictions: $accuracy = TP + TN/(TP + TN + FP + FN)$
- Precision: The fraction of the actually positive samples out of the predicted positive samples. It provides a measure of the accuracy of positive predictions: $Precision = TP/(TP + FP)$

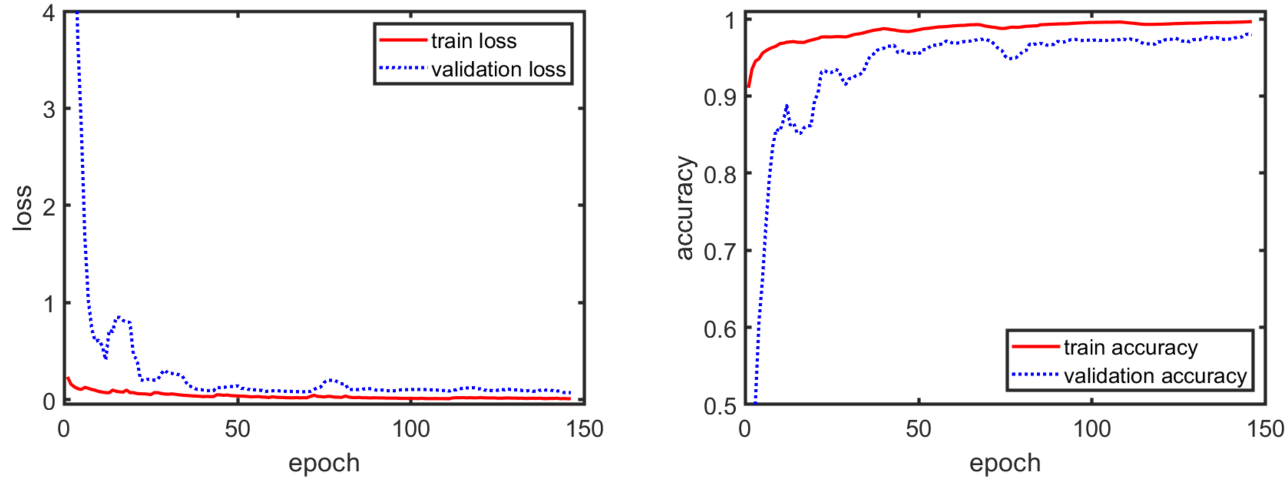

Fig. 4. Learning curve, accuracy (left) and loss (right)

- Sensitivity or Recall or True Positive Rate (TPR): The fraction of the correctly predicted positive values out of the actually positive samples. It measures the ability of the classifier to find all the positive samples: $Recall = TP/(TP + FN)$
- Specificity or True Negative Rate (TNR): The fraction of negative test results that are correctly identified as negative (true negative). It measures the effectiveness of a classifier in the identification of negative labels: $Specificity = TN/(TN + FP)$
- F1–score: Combines the precision and recall into a single metric that provide a simple way to compare two classifiers. F1-score is defined as the harmonic mean of Precision and Recall. As a result, the classifier will only get a high F1-score if both recall and precision are high: $F1 = 2 * ((precision * recall)/(precision + recall))$
- AUC-ROC: The classification model predicts the class of an input sample by predicting the probability of the sample belonging to the class. Based on the threshold value the class of the sample is determined. Therefore, by varying the threshold value for the classification model, the sensitivity and the specificity will be changed. Then we can select the threshold value depending on whether we want to lower the False Negatives or False Positives. By changing the threshold the confusion matrix result will change. And so by plotting the sensitivity vs the specificity at different threshold values, we get the ROC curve which can be used to chose the desired threshold value. Both sensitivity and specificity are used to plot the Receiver Operating Characteristics Curve (ROC). The Area Under the ROC curve gives another useful measure. The calculation of the AUC curve is done by using a set of thresholds to compute pairs of True Positive Rate (TPR) and False Positive Rate (FPR). The well accepted interpretation of the AUC values follows the academic point system: 0.9-1.0 (Excellent), 0.8-0.9 (Good), 0.7-0.6 (Acceptable), 0.6-0.7 (Poor discrimination), 0.5-0.6 (Fail – no discrimination).

The generalization of binary classification measures to multi-class classification can be achieved by applying them to each class independently (one–versus–rest) and then averaging the results [21]. For each class C_i, the binary confusion-matrix has the elements tp_i, fn_i, tn_i, fp_i. The overall measure is calculated by two ways: 1) macro-averaging, the overall measure is just the average value of the measures calculated for each class $C_1; ...; C_l$, and 2) micro-average, the sum of counts (tp_i, tn_i, fp_i, fn_i) is calculated first and then the overall measures are computed as in the binary classification using the accumulative values. Macro-averaging gives equal weight to each class and suitable when classes are equal. Micro-averaging favors bigger classes and more suitable when the classes are not equal.

ROC-AUC score extends to problems with three or more classes by using two averaging strategies, one-vs-rest (ovr) and one-vs-one (ovo) approach. In the one-vs-rest approach, the average of the ROC-AUC scores is computed by calculating the average of ROC-AUC against all other classes. In the one-vs-one approach, the average is computed pair wise for all possible combinations of the classes.

IV. Results

The proposed classifier is implemented in Python language using the Keras-Tensorflow Implementation. Keras and Tensorflow are open source software libraries for deep learning. In training the CNN, we used the RTX 2060 super GPGPU graphics card to speed up the training process. Our computational system consists of a workstation running Intel 6 core I7 and 16 GB of RAM. TensorFlow is accelerated by using CUDNN[4] on the RTX graphics card. The complete code and the version of the software is available on Githup [22].

TABLE I
CROSS VALIDATION USING IMAGES OF SIZE 64 X 64

$Fold$	$Loss$	$Accuracy$	$Precision_M$	$Recall_M$	$F1-score_M$	$auc-ovo_M$
1	0.06567	98.74%	98.93%	98.39%	98.65%	99.90%
2	0.11017	96.88%	97.22%	97.22%	97.22%	99.75%
3	0.07448	98.25%	98.54%	98.54%	98.54%	99.89%
4	0.04350	98.21%	98.42%	98.01%	98.21%	99.94%
5	0.10646	97.37%	97.90%	97.22%	97.52%	99.85%
6	0.05019	98.42%	98.69%	98.27%	98.48%	99.92%
7	0.10238	96.88%	97.41%	97.19%	97.30%	99.78%
8	0.32400	95.58%	96.58%	93.68%	94.85%	97.67%
9	0.04585	98.53%	98.78%	98.60%	98.69%	99.93%
10	0.02431	99.12%	99.07%	99.18%	99.12%	99.98%
Average	0.09470	97.80%	98.15%	97.63%	97.86%	99.66%
stdev	0.08577	0.01092	0.00837	0.01542	0.01237	0.00702

A. Training and model selection

Training a CNN is the process of determining the optimal values of the weight parameters (kernels) by minimizing an objective function using the training set. The cross-entropy objective function, which is commonly used for classification, was applied. For back propagation, we used the Adam mini-batch gradient descent optimizer. We used all of the Adam algorithm's default hyperparameters except the learning rate, which we approached using the "Cosine annealing learning rate scheduler with periodic restarts" method described in [22] and implemented in [23]. After experimenting with various batch sizes, we discovered that as the batch size increases, performance measures improve up to a batch size of 128 for the 64×64 images. Due to the memory restriction in our training system, we did not test for more than 128 batch size for 64×64 images and 48 for 128×128 image sizes. This can be explained due to our augmentation technique, which replaces each signal with 57 overlapping signals, it's possible that the majority of data in small batches will come from one or two classes. As a result, the gradient descent updated step will be dependent solely on these classes. As a result, increasing batch size improves the possibility that multiple classes will be included in the batch, thus reducing this effect.

The number of epochs varies between executions due to the use of early stopping. We used a rate of 0.5 for dropout. Tensorflow version 2.4.1 and Sklearn version 1.0 were used to build the model. To address the problem of a limited amount of data in the detection of epilepsy, we applied the data augmentation technique described in §III-B. Each EEG sub-signal is treated as a separate instance in this training method. Each sub-signal image is identified as normal (0), interictal (1), or ictal (2) . The learning curve for a given training instance is depicted in Figure 4. As illustrated Figure 4(left), the figure reveals that the training and validation accuracies are near to one another, indicating that the model is not overfit. Figure 4 (right) depicts the training and validation losses at the end of each training epoch, with both the validation and training losses decreasing as the training progresses.

B. Model Selection

To select the best model that can classify the EEG signals into three classes with highest accuracy, we considered two different image resolution, 128×128 pixel resolution and 64×64 pixels. To properly evaluate and tune the parameters of the proposed CNN classifier, 10-fold stratified group cross-validation was used during the evaluation

All 10-fold cross validations are generated at random, with 10% of each group included in a test set and the remaining 90% included in a training set. The signals were then augmented with the previously established augmentation scheme. The average performance for each measure is then calculated using the ten distinct testing sets. Table I displays the results of a 10-fold cross validation experiment with an image resolution of 64×64. In this situation, the average accuracy is 97.80%, while all other measurements are greater than 97.5%. It was obvious that raising the batch size improved accuracy and all other metrics. For example, as we increased the batch size from 10 to 32 to 48 to 64, the accuracy climbed from 96 percent to 97 percent to 97.8 percent. As demonstrated in Table II, the performance results did not improve as anticipated for the 128×128 resolution, where the accuracy is 97.6 percent and all other performance indicators are above 97 percent. In this situation, the batch size is 48, which is the maximum that our machine can handle.

We did not consider the 2-class cases in this paper because the literature reported very good accuracy close to 100 percent in these cases. As a result, the focus of this work is solely on the more difficult problem, which is the three-class problem: normal (AB) vs epileptic inter-ictal (CD) vs ictal (E). Our experimental results effectively indicated that the proposed CNN classifier can achieve acceptable classification accuracy using adjusted EEG images without any manual EEG signal pre-processing such as noise filtering, feature extraction, and feature reduction methods.

In comparison to the state-of-the-art, Zhang et al. [4] reported an accuracy of 97.4 percent on the three cases, while Ihsan et al [7] reported an accuracy of 96.1 percent when using 1D CNN alone and an accuracy of 99.1 percent when utilizing ensemble learning with voting algorithm. Without the use of any ensemble learning scheme, our results reaches an accuracy of 97.8 percent. We are now exploring several ensemble learning algorithms and anticipate that our findings will improve significantly.

TABLE II
CROSS VALIDATION USING IMAGES OF SIZE 128×128

$Fold$	$Loss$	$Accuracy$	$Precision_M$	$Recall_M$	$F1-score_M$	$auc-ovo_M$
1	0.0322	98.84%	99.04%	98.86%	98.95%	99.98%
2	0.0971	97.16%	97.66%	97.25%	97.44%	99.81%
3	0.0718	97.37%	97.84%	97.81%	97.81%	99.89%
4	0.0645	97.75%	97.91%	97.49%	97.68%	99.89%
5	0.0805	97.30%	97.12%	97.75%	97.41%	99.84%
6	0.0291	99.12%	99.27%	99.18%	99.23%	99.98%
7	0.0719	97.72%	97.96%	97.28%	97.60%	99.85%
8	0.0469	97.96%	98.37%	98.25%	98.28%	99.99%
9	0.3061	95.40%	95.74%	95.03%	95.31%	98.79%
10	0.3358	97.30%	97.83%	96.08%	96.85%	98.85%
average	0.1136	97.59%	97.87%	97.50%	97.66%	99.69%
stdev	0.1115	0.01016	0.009863	0.01230	0.01095	0.004614

V. CONCLUSION AND FUTURE WORK

A 2D CNN classifier is used in this paper to classify three types of EEG signals: normal, ictal, and interictal. Because the number of epilepsy signals was small, we used an augmentation scheme based on a fixed-sized overlapping window. Furthermore, our model was refined utilizing cutting-edge deep learning techniques such as batch normalization, dropout, and a dynamic learning rate scheduler. To properly validate the proposed classifier, the performance was assessed using a 10-fold stratified group cross-validation approach. As a consequence, our CNN classifier attained an average accuracy of 97.8 percent, a precision of 98.15 percent, a recall of 97.63 percent, and a f1-score of 97.86 percent. The key advantage of employing 2D CNN is that the input signal does not need to be processed for feature extraction and feature reduction using any signal processing techniques, and the results exceed state-of-the-art epilepsy detection techniques. This method attempts to simulate an expert neurologist by viewing a 2D image of the EEG signal. The fundamental disadvantage of this strategy is the lack of training data and the need for training-intensive computing, in addition to parameter tweaking. We are currently developing and assessing numerous ensemble learning algorithms in order to improve our outcomes.

REFERENCES

[1] A. Salman, "Epilepsy detection using cnn," https://github.com/adnanalshaikh/cnneeg, 2021.

[2] W. H. Organization, "epilepsy," https://www.who.int/news-room/fact-sheets/detail/epilepsy, accessed: 2021-08-25.

[3] Q. Yuan, W. Zhou, Y. Liu, and J. Wang, "Epileptic seizure detection with linear and nonlinear features," *Epilepsy & Behavior*, vol. 24, no. 4, pp. 415–421, 2012.

[4] T. Zhang, W. Chen, and M. Li, "Epileptic seizure detection with linear and nonlinear features," *Biomedical Signal Processing and Control*, vol. 31, pp. 550–559, 2017.

[5] Y. Li, Z. Yu, Y. Chen, C. Yang, Y. Li, X. A. Li, and B. Li, "Automatic seizure detection using fully convolutional nested lstm," *Int. J. Neural Syst*, vol. 30, no. 4, 2020.

[6] C. Gómez, P. Arbeláez, and e. a. M. Navarrete, "Automatic seizure detection based on imaged-eeg signals through fully convolutional networks," *Sci Rep*, vol. 10, no. 4, 2020.

[7] I. Ullah, M. Hussain, E. ul Haq 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, 2018.

[8] L. Guo, D. Rivero, J. Dorado, J. R. Rabunal, and A. Pazos, "Automatic epileptic seizure detection in eegs based on line length feature and artificial neural networks," *Journal of neuroscience methods*, vol. 191, no. 1, pp. 101–109, 2010.

[9] P. Swami, T. K. Gandhi, B. K.Panigrahi, M. Tripathi, and S. Ananda, "Expert systems with applications," *Journal of neuroscience methods*, vol. 56, pp. 116–130, 2016.

[10] A. B. Gardner, A. M. Krieger, G. Vachtsevanos, and B. Litt, "One-class novelty detection for seizure analysis from intracranial eeg," *Journal of Machine Learning Research*, vol. 7, no. 37, pp. 1025–1044, June 2006.

[11] A. Shoeb and J. Guttag, "Application of machine learning to epileptic seizure," in *27th International Conference on Machine Learning*, June 2010, p. 975–982.

[12] N. Nicolaou and J. Georgiou, "Detection of epileptic electroencephalogram based on permutation entropy and support vector machines," *Expert Systems with Applications*, vol. 39, no. 1, pp. 202–209, Jan. 2012.

[13] Y. U. Khan, N. Rafiuddin, and O. Farooq, "Automated seizure detection in scalp eeg using multiple wavelet scales," in *2012 IEEE International Conference on Signal Processing, Computing and Control*, march, June 2012.

[14] D. Ma, S. Yuan, J. Shang, J. Liu, L. Dai, X. Kong, and F. Xu, "The automatic detection of seizure based on tensor distance and bayesian linear discriminant analysis," *International Journal of Neural Systems*, vol. 31, no. 5, pp. 1025–1044, 2021.

[15] U. R. Acharya, F. Molinari, S. V. Sree, S. Chattopadhyay, K.-H. Ng, and J. S.Surigh, "Automated diagnosis of epileptic eeg using entropies," *Biomedical Signal Processing and Control*, vol. 7, no. 4, pp. 401–408, 2012.

[16] S. Nasehi and H. Pourghassem, "Patient-specific epileptic seizure onset detection algorithm based on spectral features and ipsonn classifier," in *2013 International Conference on Communication Systems and Network Technologies*. march: IEEE, April 2013.

[17] Z. Lasefr, S. Shiva, V. Ayyalasomayajula, and K. Elleithy, "Epilepsy seizure detection using eeg signals," in *2017 IEEE 8th Annual ubiquitous Computing, Electronics and Mobile Communication Conference*. IEEE, 2017.

[18] C. Ieracitano, N. Mammone, A. Hussain, and F. C. Morabito, "A novel multi-modal machine learning based approach for automatic classification of eeg recordings in dementia," vol. 123, pp. 176–190, 2020.

[19] K. Patel, C.-P. Chua, S. Fau, and C. J. Bleakley, "Low power real-time seizure detection for ambulatory eeg," *2009 3rd International Conference on Pervasive Computing Technologies for Healthcare*, 2009.

[20] R. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. 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*, vol. 64, no. 6, 2001.

[21] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing and Management*, vol. 45, p. 427–437, 2009.

[22] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," 2017.

[23] J. Jordan, "setting the learning rate of your neural network," https://www.jeremyjordan.me/nn-learning-rate/, accessed: 2021-08-25.