AN ENERGY-EFFICIENT SPIKING CNN IMPLEMENTATION FOR CROSS-PATIENT EPILEPTIC SEIZURE DETECTION

PARSA FARSHADFAR

A THESIS SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF APPLIED SCIENCE

GRADUATE PROGRAM IN
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

September 2021

Abstract

This research aims to develop a data-driven computationally-efficient strategy for automatic cross-patient seizure detection using spatio-temporal features learned from multichannel electroencephalogram (EEG) time-series data. In this approach, we utilize an algorithm that seeks to capture spectral, temporal, and spatial information in order to achieve high generalization. This algorithm's initial step is to convert EEG signals into a series of temporal and multi-spectral pictures. The produced images are then sent into a convolutional neural network (CNN) as inputs. Our convolutional neural network as a deep learning method learns a general spatially irreducible representation of a seizure to improves sensitivity, specificity, and accuracy results comparable to the state-of-the-art results. We overcome the lack of enough data, especially positive samples, created a process to deal with imbalanced datasets, and optimized the network's complexity. In this work, in order to avoid the inherent high computational cost of CNNs while benefiting from their superior classification performance, a neuromorphic computing strategy for seizure prediction called spiking CNN is developed from the traditional CNN method, which is motivated by the energy-efficient spiking neural networks (SNNs) of the human brain. On pre-recorded EEG data from 23 patients, the algorithm's performance is tested (969 hours and 173 seizures in total), which shows sensitivity and specificity of 95.04% and 99.42%, respectively, for patient-specific classification, and 83.02% and 86.31% for cross-patient classification. Results are compared with state-of-the-art, and future steps for hardware implementation and performance improvement are discussed.

Dedication

TO MY WIFE, SORAYA

FOR HER ENDLESS LOVE AND SUPPORT

Acknowledgements

First, I would like to thank my supervisor Professor Hossein Kassiri for his guidance, motivation, and excellent advice. He patiently introduced me to biomedical studies and kindly shared his knowledge with me. His constant incentive and support have made this work possible.

I would like to thank my defense committee for reading my thesis and giving me valuable feedback.

I would like to thank Ali Jalalifar, Mehran Shakerinava, Ali Maleky, Fazel Arasteh, Reza Karimi, and Soroush Sheikh Gargar for helping me in this project and sharing all of their experience with me.

I would also like to acknowledge my colleagues and friends who made a memorable experience for me.

I thank my parents and sister, whom I miss the most, for their supports during these years.

Finally, I owe many thanks to a special person, my wife, Soraya, for her continued and unconditional love, support, and understanding during these two years.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Epilepsy is a neurological disease that may manifest itself in the form of loss of consciousness, jerking of certain body parts, and, in extreme instances, continuous whole-body convulsions. It is a disorder that affects individuals of all ages. It may have a significant effect on the patient's quality of life and may have further social and economic consequences; epilepsy may also end in premature death. Epileptic seizures affect about 70 million individuals worldwide, according to multiple reports [1].

The electroencephalogram (EEG) is a fast and generally cheap method for analyzing and studying brain electrical activity. It is conducted by putting electrodes on the surface of the scalp (a.k.a., scalp EEG (sEEG) recording) or placing them inside the skull (a.k.a., intracranial EEG (iEEG) recording) [2]. Typically, several electrodes are used to record the scalp EEG in a non-invasive manner. This data collection is extensive and includes the synchronized activity of neurons in many regions of the human brain. Compared to other brain monitoring and imaging technologies such as magnetic resonance imaging (MRI) and Positron emission tomography (PET), scalp EEG offers a significantly better temporal resolution (i.e., the update rate). Additionally, compared to iEEG, sEEG has a lower spatial resolution but significantly higher spatial coverage. Therefore, for applications that require real-time (e.g., update rate ~a few milliseconds) monitoring of a large area (>1cm) of the brain with a reasonable resolution (>1mm), sEEG is optimal.

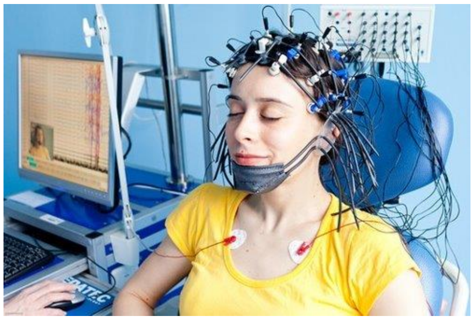


Figure 1-1 A conventional scalp EEG setup [3]

1.1 Seizure Detection

Tools and methods that allow for continuous analysis and evaluation of the appropriate medication are required to manage long-lasting neurological illnesses. Existing pharmaceutical therapies are unable to adequately treat about 30% of epilepsy patients [4]. This is somewhat due to the subjectivity of the information provided to the prescribing physician (e.g., focus, severity, seizure frequency, etc.) or reduction in the effect of the drugs on the patients. Hence, this has led to the growth of technologies that help physicians continuously monitor brain EEG signals and give them an objective assessment of the patients' status, their seizures' focus, and severity. These technologies are often used for presurgical monitoring, responsive neurostimulation, and emergency mental state evaluation.

The scalp EEG is the most commonly utilized method for obtaining epileptic information in order to detect seizure episodes at their onset. These EEG recordings may contain characteristic patterns for seizures that can be detected by trained neurologists, but this task is tedious and expensive and can take several hours for a single patient [5]. Also, according to the news, in some regions of Canada, patients have to wait more than a year for an epilepsy test [6].

In order to improve the likelihood of capturing seizure events, many hours or even days of EEG recordings are needed, which should be analyzed manually by a trained technician. Around 75% of epilepsy patients in poor countries are unable to afford medical diagnosis or treatment [7]. Because of these limitations, there are many research efforts worldwide investigating the possibility of automating for seizure detection and/or prediction [8]–[11]. This includes groups who work on developing portable/wearable hardware for recording EEG signals with medical-grade quality, as well as those who work on algorithms that utilize the recorded and digitized signals from several electrodes to detect an occurring or upcoming seizure.

1.2 Why Machine Learning and Deep Learning

Automated seizure detection methods make use of frequency and time-domain characteristics. However, some of the traditional methodologies have many drawbacks. Seizures pattern data from the EEG are extremely dynamic in nature. Their characteristics vary across individuals and also over time, even within the same patient. As a result, several automated methods are unable to cope with non-stationary EEG signals. Another significant drawback of some automated methods is that EEG data collection is subject to a variety of noise and artifacts, including muscle movements, eye blinks, and ambient noise. That is why nowadays, researchers prefer to use machine learning and deep learning methods, which have a great performance in seizure detection despite of the anomalies happening in the EEG datasets [12]. Figure 1-2 shows an example of a patient-to-patient variation of seizure patterns. Moreover, Figure 1-3 points out an example of a similarity between a seizure and an eye blinking in a patient.

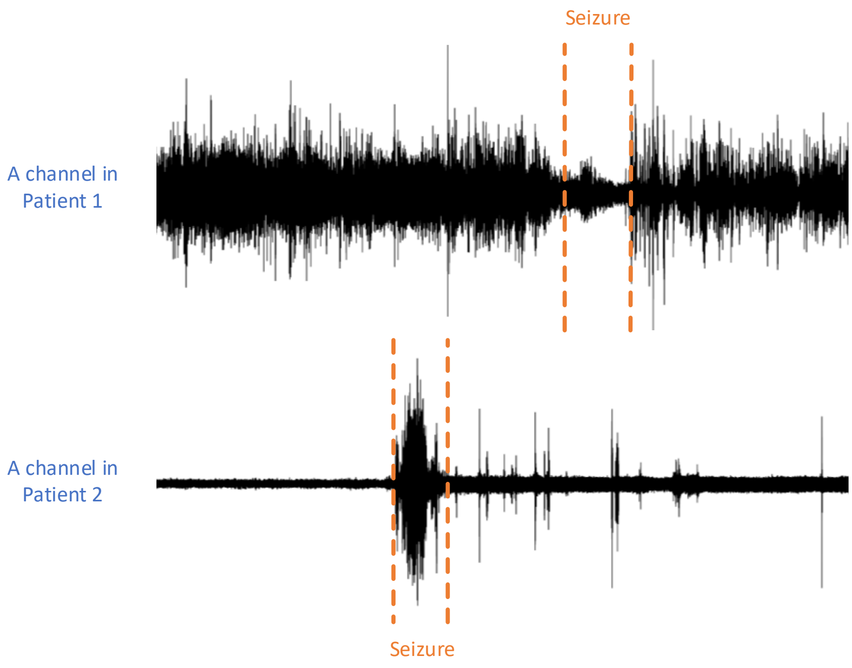


Figure 1-2 An example of a patient-to-patient variation of seizure patterns.

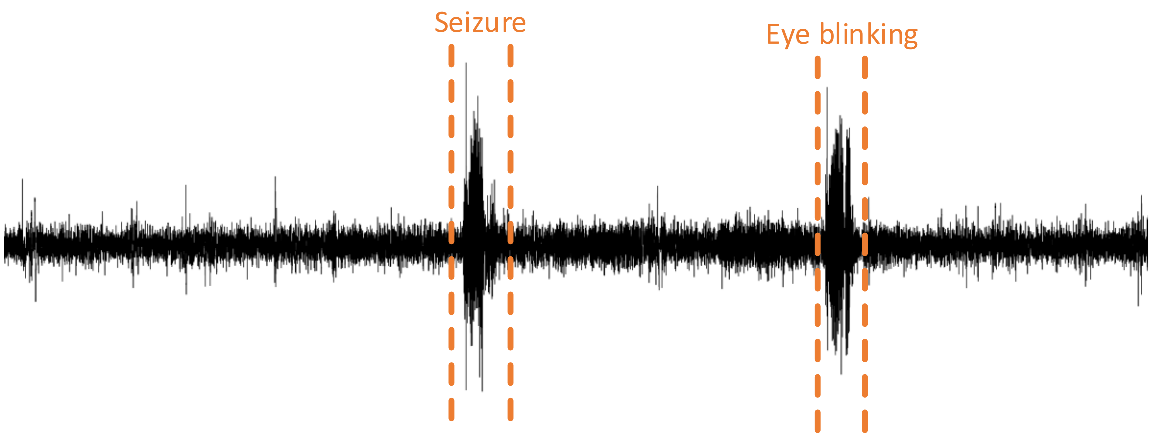


Figure 1-3 An example of the similarity between a seizure and an eye blinking in a patient

1.3 Monitoring vs. Treatment Devices

The long-term need for monitoring and treatments has led to the development of lightweight wearable systems (some even commercialized) with an unnoticeable operation that allows patients to continue their daily routines [13]–[15]. These solutions often include signal processing modules that assist in reducing the amount of data, hence extending the device's battery life. Hardware-embedded signal processing not only improves energy efficiency (wireless communication of data is significantly more power-consuming than processing) but also reduces latency to a decision by bypassing the delays imposed by wireless communication of a large amount of data to a stationary computer and back. This is especially crucial if the decision is for a time-sensitive closed-loop function, such as responsive electrical stimulation after the detection of an epileptic seizure [16]–[20].

1.3.1 Treatment

Patients with drug-resistant epilepsy who are either not surgical candidates or refractory to surgery constitute a large and difficult-to-treat group [21]. Additionally, individuals with drug-resistant epilepsy are at a higher risk of serious consequences associated with inadequate seizure control, such as serious injuries and SUDEP [22]. Improved seizure control is critical for drug-resistant individuals, yet improvements in pharmacological therapy have had little impact on seizure freedom rates during the last four decades [23]. Thus, other ways of treatment are being investigated for individuals who are unable to benefit from reconstructive surgery.

Deep brain stimulation (DBS) as a closed-loop neuro-modulation (feedback-controlled) method would be a surgical therapy option for those with epilepsy who are unable to control their seizures adequately with medication [24]. It entails implanting a pacemaker-like device and electrodes that deliver regular electrical pulses to particular regions of the brain to control epilepsy.

Physicians use patient-specific seizure detection methods like machine learning classifications in the DBS devices to detect seizure onset accurately with low latency. In general, Patient-specific algorithms try to get trained with EEG data for a "specific" patient and then detect the seizure onset accurately and quickly. This accurate, timely detection will help the device with reducing the number of stimulations episodes and consequently the battery life.

1.3.2 Monitoring

Sometimes, there is insufficient time to train the seizure detection algorithm on a specific patient. For instance, a patient presents to a hospital's emergency department; in this case, the patient needs to be urgently monitored, and an expert physician must be notified if a seizure-like activity is detected.

It is evident that monitoring in this case must be a non-invasive procedure and a reasonable amount of delay (e.g., milliseconds, or even seconds) is not a problem, as is the issue with patient-specific treatments where a small delay is critical in the efficacy of the closed-loop responsive treatment. This kind of monitoring requires a cross-patient approach, which implies that the technique must include a generic seizure detection algorithm trained with many different patients' EEG signals that result in acceptable accuracy for a variety of patients.

1.4 Implementation Challenges

SVM algorithm, and conventional methods such as Random Forest, K-means, Fuzzy Clustering, Bayesian Net, and other linear and non-linear classifiers have been shown to yield solid accuracy in the patient-specific methods [8], [25]–[31]. However, since epileptic seizures express themselves differently in different individuals, these methods have not demonstrated an acceptable accuracy in the cross-patient seizure detection as a complex classification [32]. Thus, researchers move toward deep learning algorithms to deal with complex classifications [27].

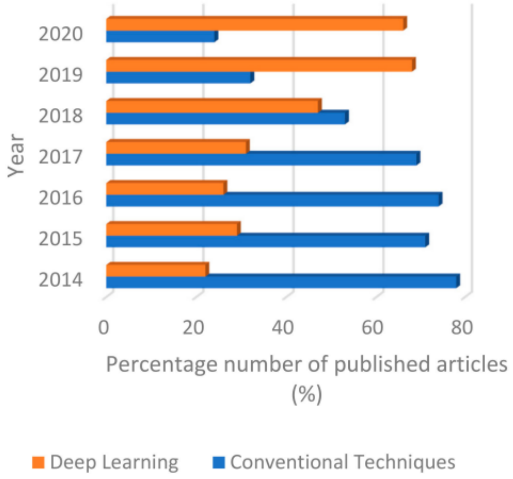


Figure 1-4 Comparison of deep learning and conventional machine learning methods involved in epilepsy studies used by researchers in recent years [27]

Deep learning techniques have been employed for EEG signal processing in the recent few years, with the convolutional neural network (CNN) being the most typical software method because of their attention to spatial features besides the conventional spectral and temporal features extracted from EEG signals, especially for the cross-patient seizure detection methods [12]. However, it is well known that CNNs are compute-intensive, making them difficult to be implemented on wearable medical devices. Because of the fact that computing resources available on medical devices are highly limited, the reduced latency and energy efficiency of embedded processors lead to the loss of accuracy. As a result, only rudimentary algorithms can be implemented on these CPUs for the relatively large number of channels required in a comprehensive EEG system.

1.5 Proposed Design

The human brain is capable of extraordinary perceptual actions while using very little energy. The goal of brain-inspired computing is to create systems that do the same functions as the human brain, which requires both high-precision algorithms and efficient hardware to execute them.

Spiking Neural Network (SNN), as an increasingly popular neuromorphic computing method, is also inspired by the way information is communicated in biological neural networks in the central nervous system of humans, primates, and rodents. The neurons in the SNN do not propagate information at the end of each flow cycle (as in traditional multi-layer perceptron networks), but instead neuron fires only when a membrane potential reaches a certain value, known as the threshold, and sends a signal to neighboring neurons, which significantly change their potentials in response to the signal. It means the network can work asynchronously and is confirmed to be robust [29], [33], [34]. This results in the SNNs being highly energy-efficient and an optimal choice for performing hardware-embedded computations. [35]. It is expected to be the next generation of Artificial Neural networks [36]. SNNs always require sophisticated training approaches such as unsupervised spike-timing-dependent plasticity (STDP) [37], SpikeProp [38], and Tempotron [39]. In this work, to ensure that spatial features of the recorded multi-channel EEG are captured, we proposed to use a Spiking CNN (SCNN) [40]. In doing so, as described in Chapter 3, we first trained a regular CNN to perform seizure detection. This was done twice for two different applications: patient-specific and cross-patient. While our main objective in this work was to develop a seizure classifier for cross-patient detection, the patient-specific model allowed us to compare our model with the myriad of patient-specific classifiers reported in the literature. The cross-patient model's performance was also compared with state-of-the-art, which is a relatively new topic and quite limited in terms of the number of published papers, compared to the patient-specific field. Next, we transformed trained weights in both CNN models into the SNN topology as needed to predict seizures precisely and energy-efficiently.

1.6 Thesis Organization

Spectral energy and phase-locking value (PLV) are the features we extracted from raw EEG data to make the image-based representation of EEGs. Then we needed to train the CNN using these images and transform the trained CNN into spiking CNN. Some encoder functions were also responsible for converting the images to the sequence of spikes before feeding them to the new spiking CNN.

In Chapter 2, we will introduce the EEG dataset used in this work and will discuss different time, frequency, and spatial features of the recorded EEG that we have extracted to be fed to the classifier. We will describe how the features are extracted, what parameters are optimized, and how the extracted features are converted into images that are later fed to an SCNN.

In Chapter 3, we have introduced the deep learning model that we have developed, how different parameters such as stride, padding, activation function, and pooling layers are set/devise, and how the neural network's architecture is designed. Next, the training and testing process were described, and different techniques used for resolving dataset-related issues are explained. This chapter also includes sections on how the developed CNN is converted into an SCNN. In the end, simulation results of both CNN and SCNN, each for the patient-specific and cross-patient problems, are presented for each patient in the dataset and compared to state-of-the-art.

Chapter 4 concludes the thesis and discusses ideas for immediate next steps and long-term plans, and future directions of this project. Figure 1-5 shows a simplified flow diagram of the presented method.

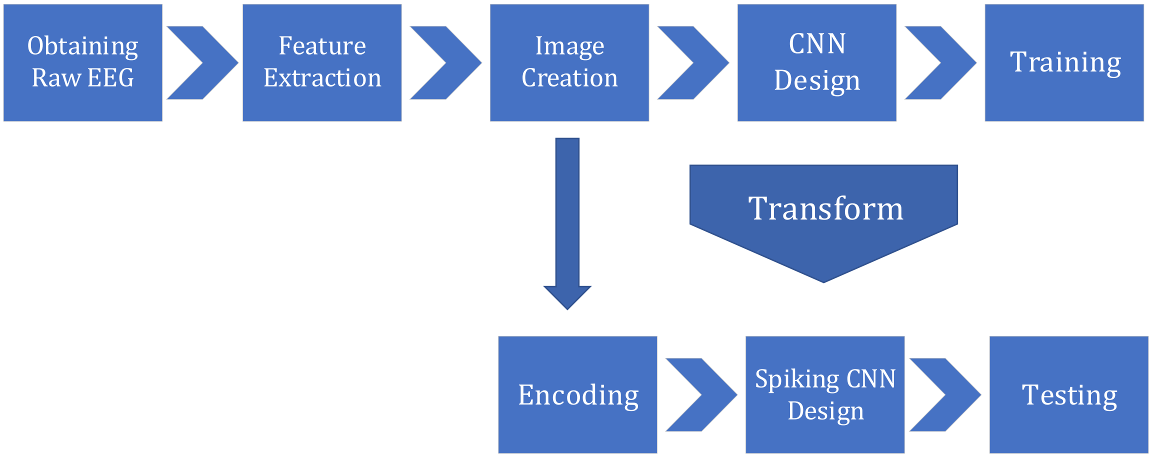


Figure 1-5 Flow diagram of the presented method

Chapter 2

Feature Extraction

To begin, we will discuss the database that will be used and the features that will be extracted from it. Following that, we explain the method of generating images from the features.

2.1 Database

Datasets are vital in the development of accurate and reliable biological classifiers. There are several EEG datasets that can be used to create automated epileptic seizure detection systems, such as Zenodo [41], Hauz Khas [42], Bern-Barcelona [43], Flint-Hills [42], Kaggle [44], Bonn [45], Freiburg [46], and CHB-MIT [10]. These datasets' signals are captured either intracranially (iEEG) of people and animals or from their scalp (sEEG). In Table 2-1, there is more information about each dataset.

Table 2-1 Details of the popular seizure datasets.

Dataset	Number of Patients	Number of Seizures	Duration	Sampling Frequency (HZ)	Recording Type
Zendo [41]	79 neonates	460	74 minutes/patient	256	sEEG
Hauz Khas [42]	10	NA	NA	200	sEEG
Bern-Barcelona [43]	5	3750	83 hours	512	iEEG
Flint-Hills [42]	10	59	1419 hours	249	iEEG
Kaggle [44]	5 dogs and 2 patients	48	627 hours	400HZ and 5KHZ	iEEG
Bonn [45]	10	NA	39 minutes/patients	173.61	iEEG and sEEG
Freiburg [46]	21	87	708 hours	256	iEEG
CHB-MIT [10]	23	198	969 hours	256	sEEG

2.1.1 Database Detail

We used the Children's Hospital of Boston-Massachusetts Institute of Technology dataset for training and testing our method. This is the largest freely available dataset existing (CHB-MIT Scalp EEG Database [10], [47]). This dataset was first collected to determine the need for surgery by analyzing seizures in patients.

It contains 969 hours of scalp EEG data from 23 individuals, including 18 females and 5 males, with 198 seizures in total. The EEG recordings are split into 24 sets (two sets of EEG recordings which are named CHB01 and CHB21, are for one patient whose second set is 1.5 years after the first one) which were recorded using a sampling rate of 256Hz and 16-bit resolution. The patients' ages range from 1.5 to 22.

An epilepsy specialist has labeled the start and finish times of each seizure, which are utilized for training and testing of our developed algorithm. EEG electrode locations were recorded using the international 10-20 system (23 bipolar channels).

In addition, sometimes during the recordings, the EEG recording devices, got disconnected form the patients' scalp which make anomalies in the EEG signals of the dataset. Hence, we needed to remove those parts from the signals to avoid any misclassifications.

2.1.2 Time Window Sizing and Labeling

In general, the imbalanced number of instances in each class makes the classification tasks more complicated [48]. This issue also presents in seizure detection datasets. The CHB-MIT dataset consists of just 198 seizures, each lasting around one minute in 969 hours. Thus, it suffers from an extreme class imbalance, and few positive samples make the training uncertain; As a result, a data-driven classifier (e.g., an artificial neural network) might get overfit during training.

In order to overcome the problem of the imbalanced dataset and improve the performance, we sliced the EEG recordings into overlapping time windows with 50% overlap [49], as shown in Figure 2-1 with 30-second time windows for the cross-patient method. Also, in the patient-specific method, we used the same approach mentioned in Figure 2-1 with a 4-second window size instead of 30-second ones.

We used this technique to increase the number of instances in the minority class (seizure class). It does, however, result in an increase in the majority class as well. Thus, we needed to use some other techniques besides the overlapping method to reduce the number of instances in the majority class. We explained them in detail in the next chapter to solve the problem of imbalance distribution in the training set.

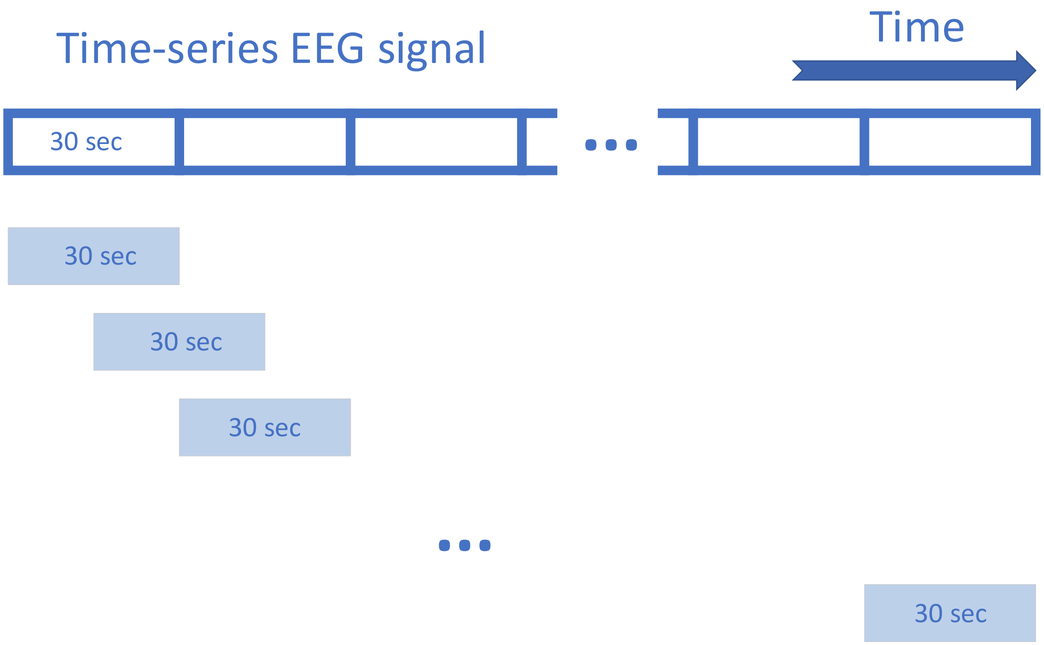


Figure 2-1 30-second windows with 50% overlap for cross-patient method.

Our simulation results, confirmed by many reported in the literature such as Alkanhal et al. [50], have shown that using a longer window size (e.g., 30 seconds) will lead to higher detection accuracy, and this is mainly due to the vulnerability of shorter time windows to various forms of artifacts and noises. However, using a longer time window causes a proportionally longer detection latency. Therefore, for applications where latency is of importance (e.g., when a timely biofeedback must be sent back to the brain immediately after detection to stop a seizure), shorter windows are preferred. As a result, if the algorithm is designed for a responsive closed-loop implantable neuro-stimulator (i.e., a patient-specific algorithm), the window size should be set as a trade-off between delay and accuracy.

Normally, because of the low delay needed in the patient-specific algorithms, 1-to-2 seconds tie windows are preferred to constrain the latency. In this work, we chose to have a 4-second window with 50% overlap for the patient-specific algorithm to ensure a maximum 2-sec latency while minimizing the aforementioned negative implications of short windows.

All in all, by using this overlapping approach, first, we increased the number of windows in order to have more samples from seizure class, and this step will be helpful for the other techniques that we used in the next chapter; second, we could reach half of the delay in seizure detection while having the better performance of using longer time windows [51].

The start and end of each seizure are recorded in a separate text file for each patient in the CHB-MIT dataset, which were used to label the data. The reported timings have a precision equal to one second. Given that our time windows are longer than one second, it is likely that seizure activity only happens during a fraction of the window length. This fraction could be anywhere from 0% to 100%. Also, depending on whether the window falls in the beginning or the end of a seizure episode, we could have both seizure-then-normal and normal-then-seizure situations.

A simple choice would be labeling a window as a "seizure" if the majority (i.e., > 50%) of the window is seizure activity. However, after training our model with this and other choices, the optimal choice turned out to be labeling a window as a seizure, even if it contains one second of seizure activity, no matter it has a seizure-then-normal or normal-then-seizure situation. Furthermore, by labeling in this manner, we will minimize seizure detection delays, which are critical in low-latency patient-specific detection.

Figure 2-2 represents an overview of the above-mentioned labeling method.

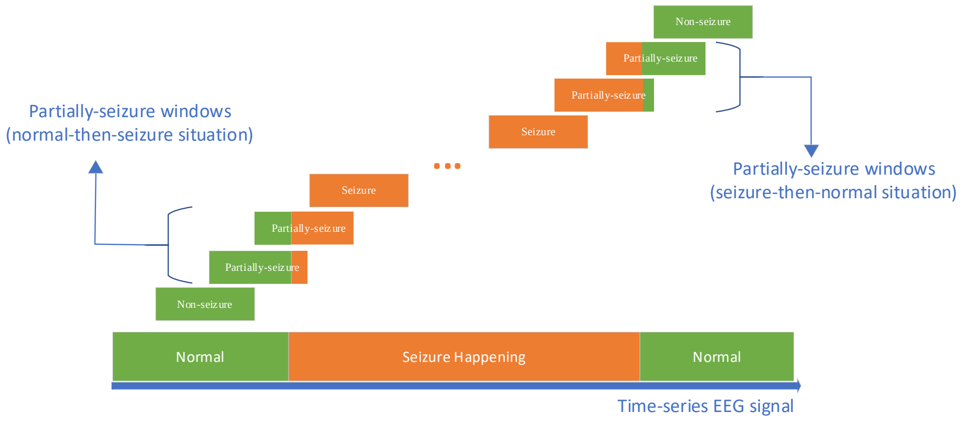


Figure 2-2 all of the partially-seizure windows are considered as seizure windows.

2.1.3 Dataset Proportions

For the patient-specific method, we split each patient's data into three subsets. 70% of the dataset is randomly picked for the train set, 15% of the dataset is randomly chosen for validation, and the remaining 15% is left for the test set, as shown in Figure 2-3.

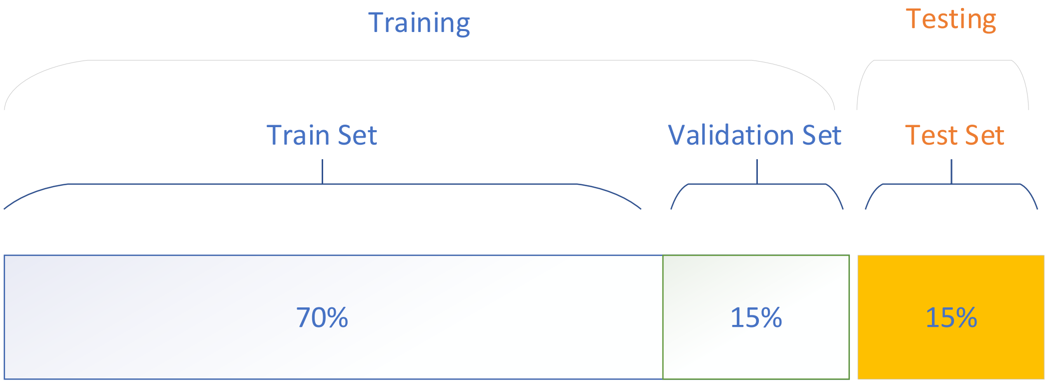


Figure 2-3 Patient-specific dataset proportions

For the cross-patient method, we choose one subject as a test set and then train the model using the other 23 subjects. In addition, we randomly chose 15% of the training set as the validation set to use in our training. Figure 2-4 shows a summary of the proportions in the cross-patient method.

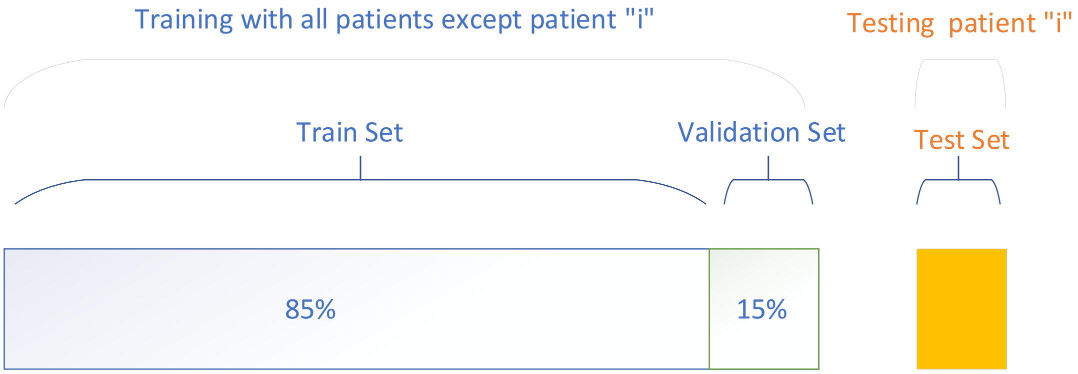


Figure 2-4 cross-patient dataset proportions

We also made sure that for both cross-patient and patient-specific approaches, the subsets are stratified due to the imbalanced dataset we have. It means when we are randomly making our train set, validation set, and test set, the sampling ratio of the seizure class in every subset must be equal to the ratio of that subset to the overall dataset. When the test set accounts for 15% of the entire dataset, the number of seizures time windows (samples) in the test set also stands for 15% of the total number of seizures in the dataset.

2.2 Features

In EEG signal processing and in special seizure detection methods, there are three types of inputs that we can feed to the classifier. Figure 2-5 represents the proportions of different types of inputs across all seizure detection studies. [12]

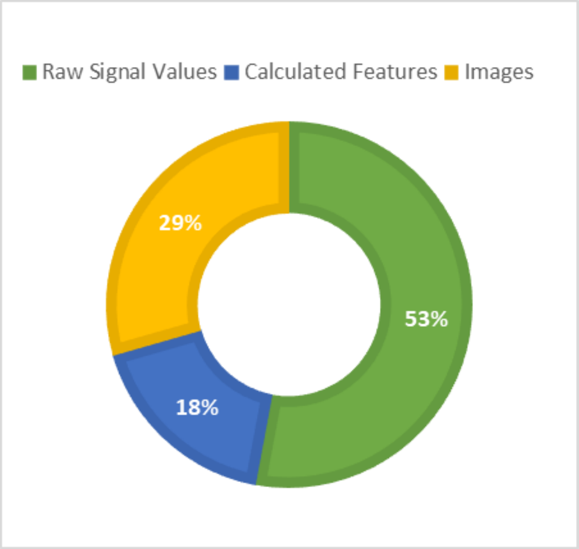


Figure 2-5 The proportions of different types of inputs across all seizure detection studies

Compared to cross-patient detection, developing an accurate classifier is generally a simpler task in most patient-specific studies. As a result, in many cases, reasonably successful classification is achieved by only using the raw recorded signals or simple signal features. From the hardware implementation perspective, this has the advantage of being efficient in terms of required computational resources. However, for the more challenging cross-patient detection, a more sophisticated feature extraction must be employed. The input to the classifier should contain both temporal and spatial features of the recorded EEG signals to increase the chance of accurate detection. Recently, one of the more popular ways to achieve this is to feed images to the classifier. These could be actual snapshots of raw recorded data from all channels for a short (e.g., N=1-2 seconds) period of time that are fed to the classifier every N seconds. It could also be a picture generated from different features extracted from data that is fed to the classifier every N seconds. In this project, first, we extract some features from the raw signals, and then we create images from them to feed our network.

Features represent a distinguishing characteristic or an operative component specified in a pattern section, as well as an identifiable measurement. One of the main objectives of feature extraction is to minimize the loss of important information associated with the signal. Furthermore, feature extraction reduces the resources needed to adequately characterize a large collection of data. When done correctly, feature extraction may decrease the cost of data processing, simplify data implementation, and lessen the need to compress data. [52].

To ensure that we have captured all relevant information required for an accurate seizure detection, we utilized spectral band energy (i.e., a frequency domain feature), phase-locking value (PLV) (i.e., a time-domain feature), and relative intensity of signal energy in different electrodes (i.e., a spatial feature). Each of these features has been proved to obtain high sensitivity (true positive rate) in seizure detection [8], [47], [53].

2.2.1 Spectral Energy

To conduct effective seizure detection, it is essential to extract valuable statistical features from the EEG data. Because the EEG signal is complicated and non-linear in general, it is preferable to employ a non-linear model [54]. The fast Fourier transform (FFT) is used to extract the features of the EEG data in this research. As a result, we used the fast Fourier transform (FFT) technique to convert the time domain signals into frequency domain signals.

Since the number of channels is different for different patients, in this algorithm, we only used the top 19 channels with larger seizure-to-normal average energy for patients (FP1-F7, F7-T7, T7-P7, P7-O1, FP1-F3, F3-C3, C3-P3, P3-O1, FZ-CZ, CZ-PZ, FP2-F4, F4-C4, C4-P4, P4-O2, FP2-F8, F8-T8, T8-P8, P8-O2, and FT9-FT10). Then, FFT is performed on the EEG signal of each channel to extract the signal energy for two bands of 0-7Hz and 7-14Hz. Compared to higher frequencies, it has been shown in literature [11], [50] that the 0-49 Hz band is of the highest importance for seizure detection. We divided this band into three sub-bands of 0-7Hz, 7-14Hz, and 14-49Hz. Later, considering the energy efficiency of the model, our analysis showed no meaningful value for including the 14-49Hz. Therefore, we only used the signal's spectral energy for 0-7Hz and 7-14Hz.

Following bandpass filtering of the raw recorded signals into two mentioned bands for each channel and slicing the signals into many time windows, we calculated the spectral energy of each time window using the following equation,

$$E = \sum_{k=0}^{n} a_k^2 \qquad \text{(Eq 2.1)}$$

Where 'a' is the signal magnitude of the kth sample. For example, the spectral energy of a 4-second time window is equal to the sum of squared values of all 4×256 samples existing in the time window.

Brighter colors in the spectrogram of Figure 2-6 clearly show that two bandwidths chosen from 0 to 14Hz have more energy during the seizure activity and less noise during the normal activity compared to the higher bandwidths. This will make them our determinative frequency domains for the classification.

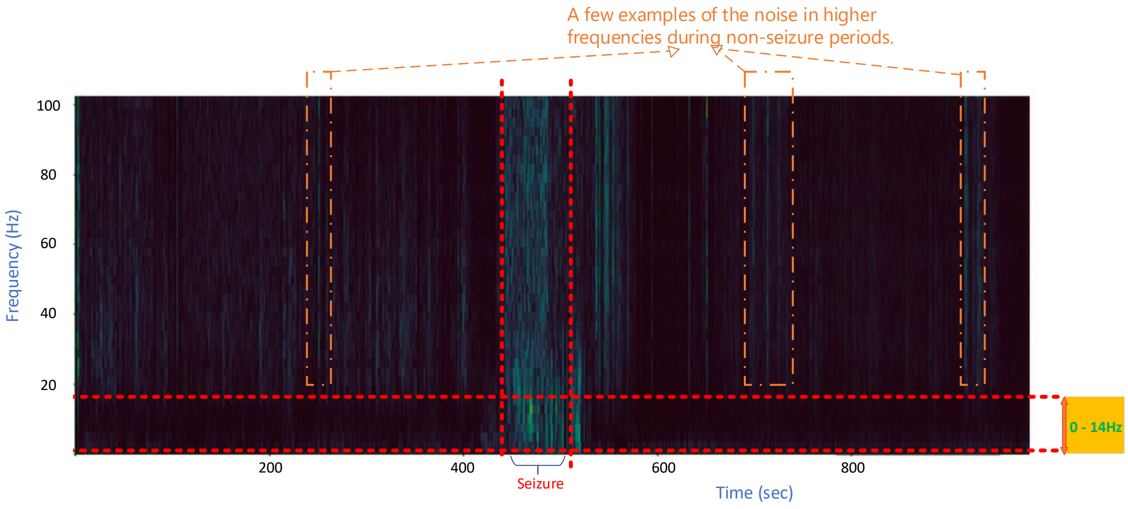


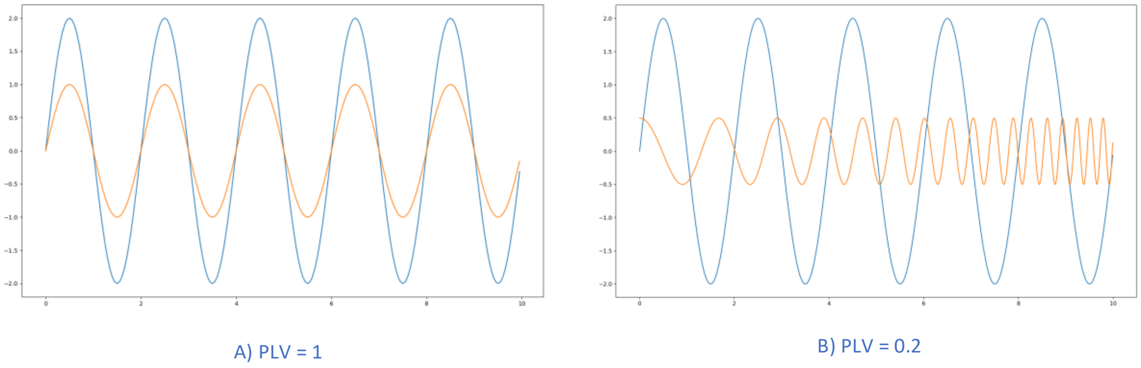
Figure 2-6 Spectrogram of energy for an EEG signal from a channel

2.2.2 Phase Locking Value (PLV)

Phase-locking value (PLV) is the other feature that we used in this project, which unlike spectral band energy, is a bivariate feature (i.e., it uses raw recordings from two different channels, while a univariate feature such as spectral band power is extracted only from a single channel). PLV is a metric of the phase synchrony and correlation between two time-series data, and it ranges from 0 to 1. In order to calculate PLV, we need to find the phase difference ($\Delta\phi_i$) of samples in two signals and feed it to the following equation [20], [53], [55], [56].

$$PLV = \frac{1}{N}\sqrt{(\sum_{i=0}^{N-1} \sin(\Delta\phi_i))^2 + (\sum_{i=0}^{N-1} \cos(\Delta\phi_i))^2} \qquad \text{(Eq 2.2)}$$

A PLV equal to 1 means perfect synchrony, and equal to 0 means no synchrony at all. Figure 2-7 shows two examples of PLV calculated for two signals.



A) PLV = 1

B) PLV = 0.2

Figure 2-7 Examples of PLV calculated for two set of signals.

A) two signals with high synchrony B) two signals with low synchrony

During a seizure episode, different neural clusters across a large area of the brain start to exhibit synchronous behavior. Therefore, as shown in Figure 2-8, the measured phase synchrony, and consequently PLV, of two EEG signals recorded from a pair of electrodes will increase. This promises a useful feature for seizure detection, particularly in the presence of large artifacts that could easily change the spectral band power in each channel. Basically, since PLV is dependent on the signals' phase rather than their magnitude, it is more or less immune to signal magnitude changes due to interferences, artifacts, or noise.

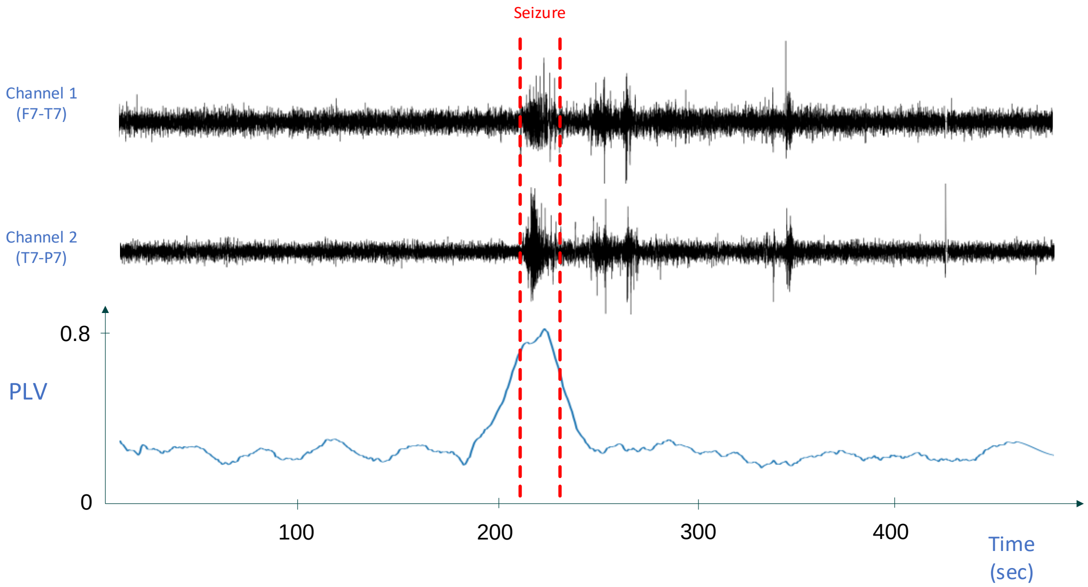


Figure 2-8 Band Energy & PLV during a seizure

We picked the top 16 channels (FP1-F7, F7-T7, T7-P7, P7-O1, FP1-F3, F3-C3, C3-P3, P3-O1, FZ-CZ, FP2-F4, F4-C4, C4-P4, FP2-F8, F8-T8, T8-P8, and FT9-FT10) with the most seizure-to-normal average PLV ratio in our PLV calculations. As a result, in every time window, we get 120 PLV values (i.e., 16C2), indicating the phase synchronization of channels one by one together. An example heatmap for correlations (PLVs) between 16 channels is illustrated in Figure 2-9.

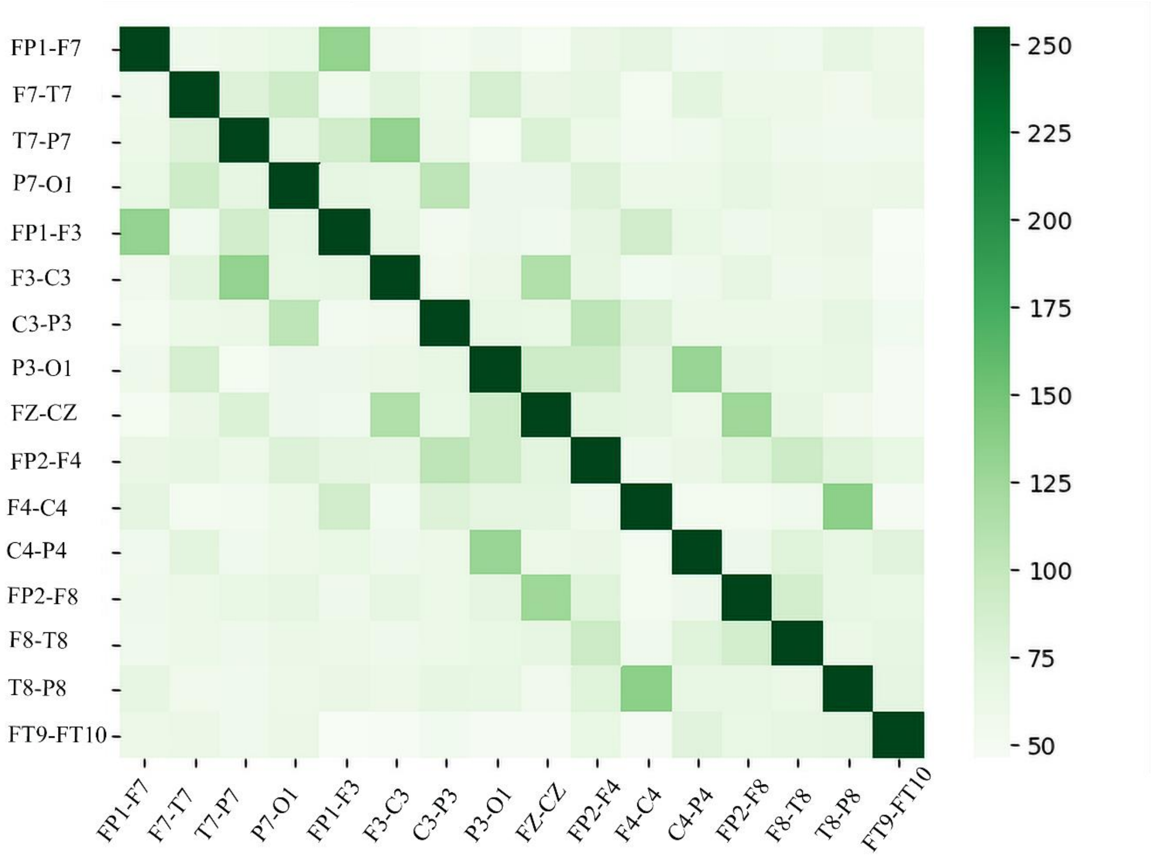


Figure 2-9 Heatmap of the PLV values between 16 channels.

2.2.3 PLV Optimization

Due to instantaneous phase changes in an EEG signal, the PLV value fluctuates significantly over time, particularly during the seizure state, which could easily lead to misclassification, as shown in Figure 2-10. To prevent this, using a windowed averaging filter, we smoothed the PLV values of time windows before creating an image of them and feeding them to the classifier.

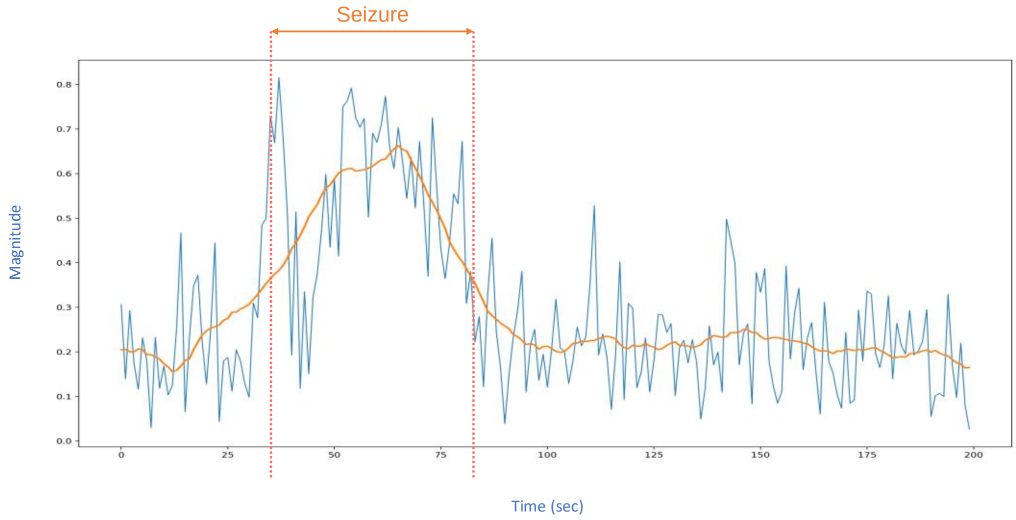


Figure 2-10 PLV Smoothing using moving average filter [8]

2.2.4 Features Correlations

As discussed, the main advantage of using different types of features is to minimize the false positive (i.e., detecting a normal EEG activity as a seizure) rate. While Figure 2-8 showed the advantage of using PLV over the spectral band power in accurate seizure detection, Figure 2-11 shows how PLV values remain high even after the seizure episode is over, which necessitates the use of other features to maintain a low false-positive rate.

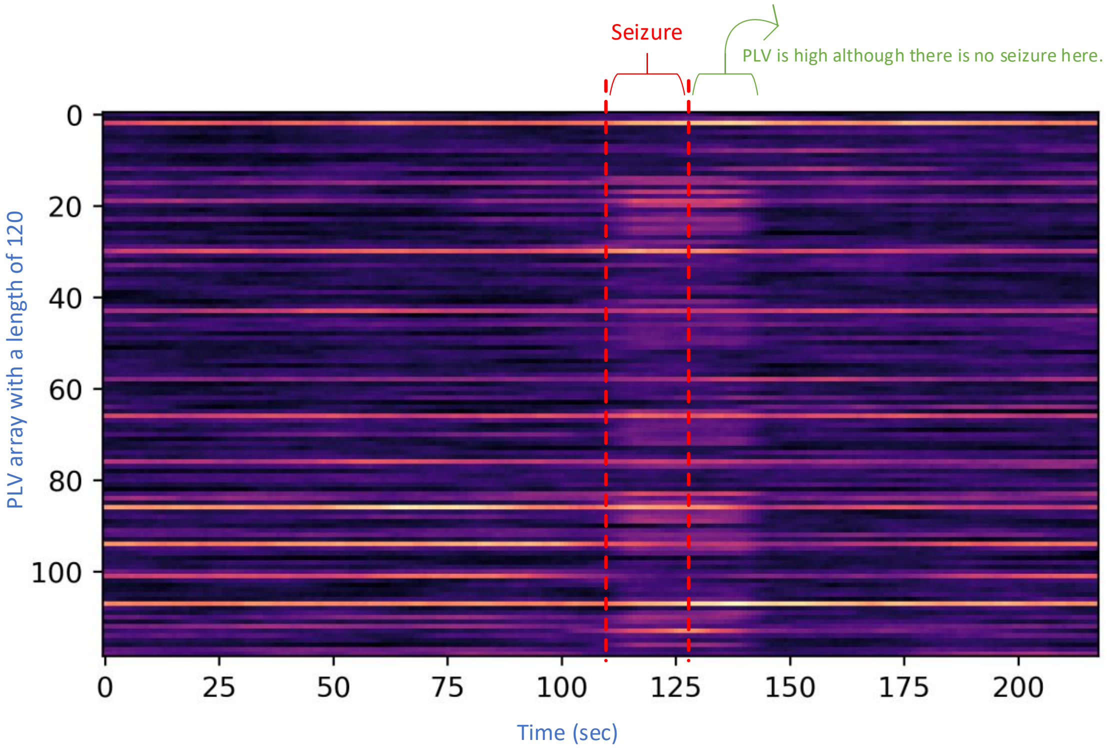


Figure 2-11 A heatmap of PLV shows wrongly high PLV despite the ending of the seizure

Figure 2-12 is the joint distribution plot of PLV and spectral energy. There is no clear correlation evident in this plot, as indicated.

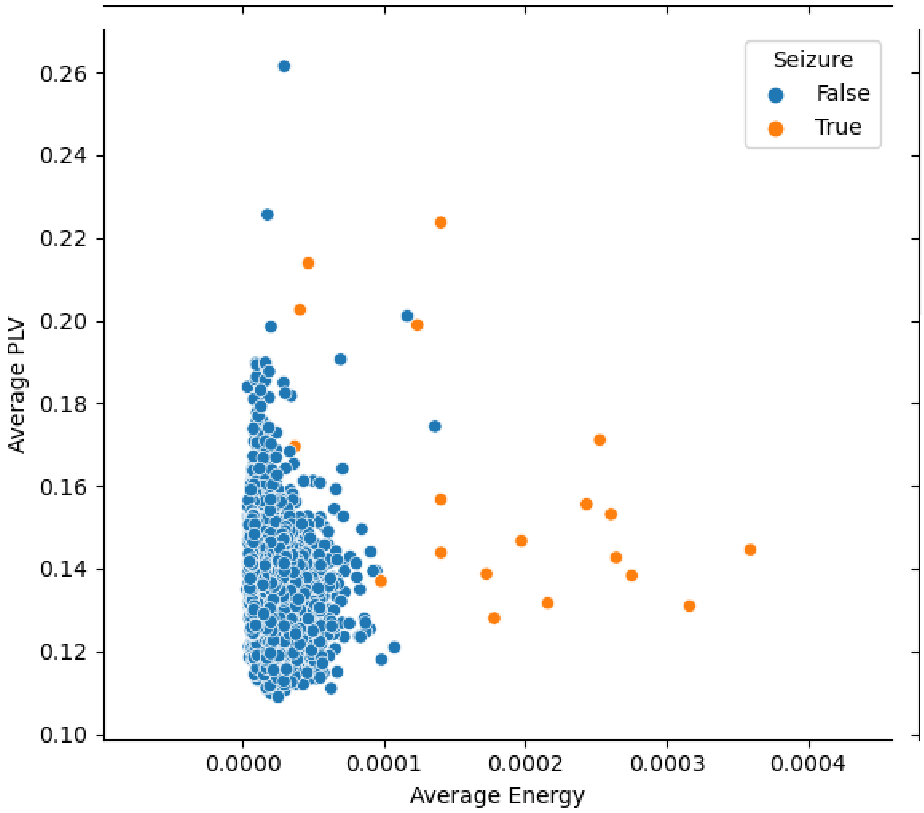


Figure 2-12 Correlation of PLV and spectral energy in patient "23"

Because the two features are proportional to the signal's phase and magnitude, which are by nature orthogonal to each other, we believe that their combination will be significantly more immune to misclassification due to various types of noise and interference.

2.3 Image Creation

Following extraction, the features must be fed to the classifier. While spectral band power carries frequency-domain information and PLV carries the time/phase-domain information of the recorded signal, respectively, the spatial information about the relative location of adjacent electrodes is not embedded in either of these features. Therefore, it is critical to ensure that the spatial information is communicated to the classifier in parallel with the features. This was done in this work by creating an image that mimics the location map of the recording electrodes, in which features extracted from each electrode's signals are overlaid on its location in the picture. Such spatial information also allows for seizure focus localization, if needed.

2.3.1 Normalization of Data

Signal normalization is an important part of patient-independent algorithms that analyze physiological signals and detect characteristics and prominent points automatically. Using the scalp electroencephalogram (sEEG) as an example, the absolute value of the EEG signal varies significantly [57]: with different persons, with age, between different subject states such as sleep or wakefulness, between various areas of the head, and during epileptic seizures. Furthermore, absolute EEG readings may fluctuate over time owing to variations in brain electrical activity as well as the quality of the electrode connected to the scalp.

Automated analysis methods such as seizure detections must use normalized or relative amplitude values to compensate for these variations. The raw data is adjusted by a measure of the average background in order to apply a fixed threshold during signal classification. Of course, there are a variety of approaches that may be used to achieve the required normalization.

We utilized the zero-mean unit-variance normalization technique to normalize the spectral data in this project. Regarding this algorithm, considering the standard deviation formula in Eq 2.3, we need to replace variables x_i by $x'_i = \frac{x_i - \mu}{\sigma}$. As a result, the mean (μ) of variables gets equal to 0 and the variance (σ^2) of them gets equal to 1.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N-1}} \qquad \text{(Eq 2.3)}$$

2.3.2 Image Representation of Energies

The worldwide 10-20 system, which is an internationally accepted technique for defining and applying the position of the scalp electrode and underlying region of the cerebral cortex, is often used to collect EEG data. Furthermore, The "10" and "20" in the worldwide 10-20 system indicates that the actual distance between neighboring electrodes is 10% or 20% of the entire front-back or left-right distance of the skull [58].

The initial stage is to project the 10-20 system electrodes' 3D coordinates into a 2D surface using polar projection and obtaining the 2D coordinates of the electrodes. [59].

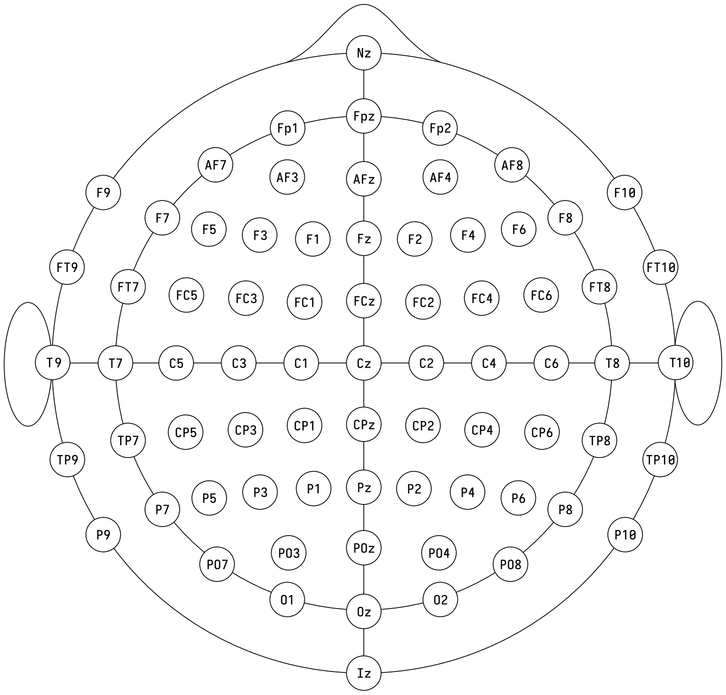


Figure 2-13 2D top-view electrode locations

The spectral energy of all channels for two bands for each time window, each band, we created an image with 16 x 16 pixels and mapped the locations of these electrodes on it. Next, we set the value of the pixel positioned at the center of two electrodes with the value of spectral energy of bipolar channel named with those two electrodes. For example, for the C3-P3 bipolar channel, we found the middle point of C3 and P3 and set its pixel value with the spectral energy of this bipolar channel. Because there are 19 bipolar channels, the described scheme should result in 19 non-zero pixels located at the middle of their corresponding pair of electrodes.

However, we know that the EEG activity is not really zero for all the other pixels. Therefore, to make the image a better representative of reality and to minimize the number of pixels with a 0 value, we applied a Gaussian filter on the image, where the peak happens at the mentioned mid-point between two electrodes and tapers off with a Gaussian behavior as we get away from the peak in all directions.

Finally, for every time window, we created two images, representing a spectral band of 0-7 Hz and 7-14 Hz, respectively, as shown in Figure 2-16.

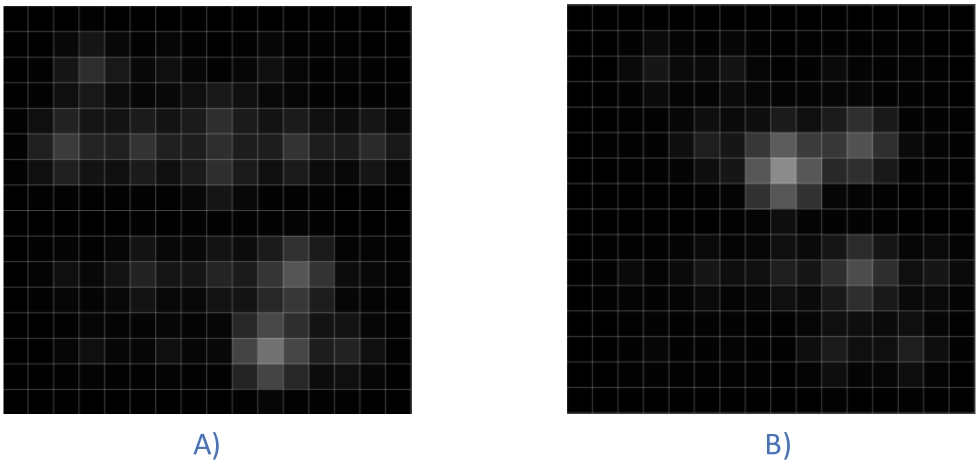


Figure 2-14 Two spectral images for a time window

A) Spectral image of first bandwidth (0-7Hz)   B) Spectral image of Second bandwidth (7-14Hz)

2.3.3 Image Representation of PLV

Similar to the spectral energy, the extracted PLV values for all channel pairs should be represented using an image to be fed to the classifier in parallel with the spectral images. Using the 120 phase-locking values obtained in the feature extraction section, we created a 16×16 image in which each pixel is responsible for the PLV of two indexed channels. For better magnitude resolution, we scaled the PLV values for each pixel from the 0-to-1 range to 0-to-255.

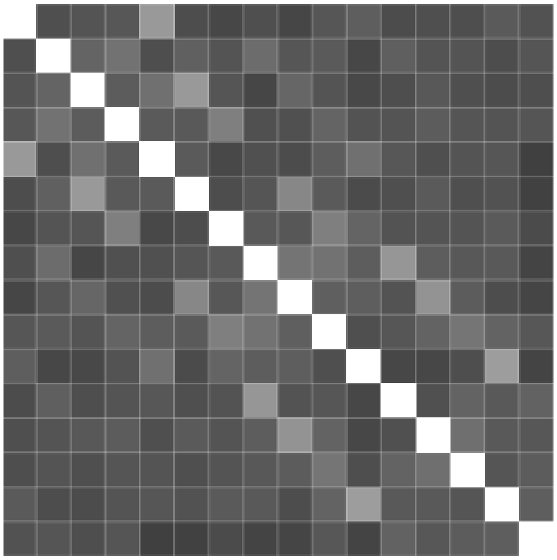


Figure 2-15 An image-based representation of PLV of 16 channels for a time window

2.3.4 Concatenation and Saving

Each time window will include two spectral energy images and one PLV image. We concatenated the three images to a wide one that included all three images. To prevent any loss of accuracy due to an unwanted mixture of extracted features, we ensured that the three images are spaced enough (i.e., a sufficient number of columns with 0 values between them), as seen in Figure 2-16.

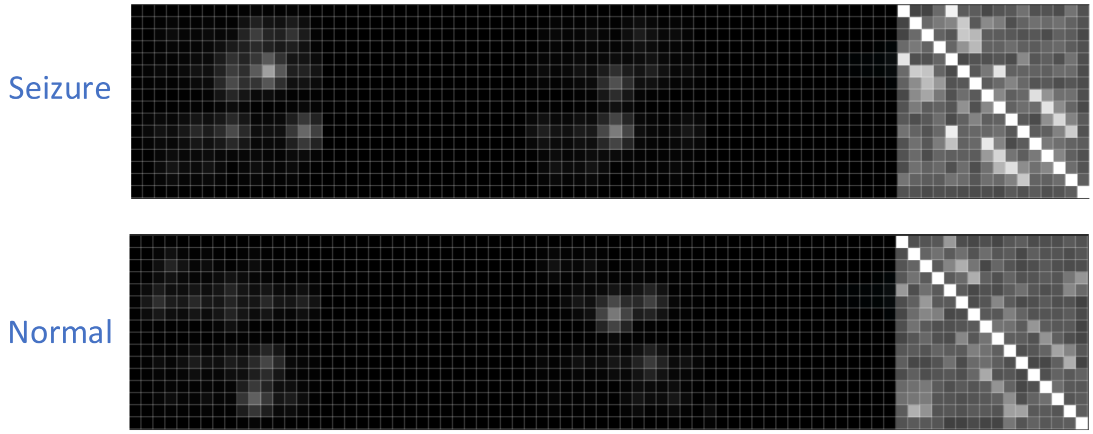


Figure 2-16 two examples of 2D wide image-based representation of EEG using spectral energies and PLV. The size of the images is 16×80.

Chapter 3

Classification

3.1 Convolutional Neural Networks

Convolutional Neural Networks (CNN) are used in a variety of applications. It is arguably the most commonly used deep learning architecture used for seizure detection algorithms [12].

CNN is aimed to learn spatial hierarchies of features automatically and efficaciously using various building blocks divided into two parts: a feature extraction component and a classification component [60], [61]. The convolutional layers and pooling layers serve in the feature extraction component. They extract features from the input images using a combination of linear and non-linear procedures. Fed by these features, the fully connected layers serve as the classifier, calculating the probability of input images belonging to different classes (e.g., seizure and non-seizure) [61].

During training and testing of deep learning CNN models, each input image is passed through a series of convolutional layers, pooling layers, and fully connected layers (FC) to classify an image with probabilistic values between 1 and 0. We utilized it to deal with image representations of spectral, temporal, and spatial information in this project.

3.1.1 Convolutional Layers

A CNN model's primary powerhouse is its convolutional layers. It's not simple to identify important features automatically given just an image and a label. By stacking on top of one other, the convolutional layers learn such complicated features.

Convolution is a mathematical procedure for combining two sets of data. In our instance, a convolution filter (kernel) is used to apply convolution to the input data, resulting in a feature map. It is accomplished by sliding the filter over the input. We conduct element-wise matrix multiplication and then aggregate the results for each place. Finally, the aggregated number is included in the feature map as an output of each layer. The technique is repeated, using several kernels to create some feature maps that represent distinct features of the inputs; different kernels can thus be thought of as different feature extractors.

Researchers typically use 1×1, 3×3, 5×5, or 7×7 filters, depending on the application. Usually, using a larger filter size make the model more complex and therefore increases the accuracy of the classifier. However, it will increase the computations. Furthermore, 1×1 filter is removed from the list of ideal filter sizes for our application since the features retrieved would be local, with no information from adjacent pixels. Finally, considering the trade-off between accuracy and complexity, we used 3×3 filters in our project. This filter size leads to less computation which is an important goal of this project. Also, we would have a shorter training time considering our large dataset in cross-patient training [62], [63].

Figure 3-1 shows an example of convolution of an input image with one of the simple 3×3 kernels available in Pytorch library, which is called edge kernel with a matrix of ($k = \begin{bmatrix} 1 & 0 & 1 \\ 0 & -3 & 0 \\ 1 & 0 & 1 \end{bmatrix}$).

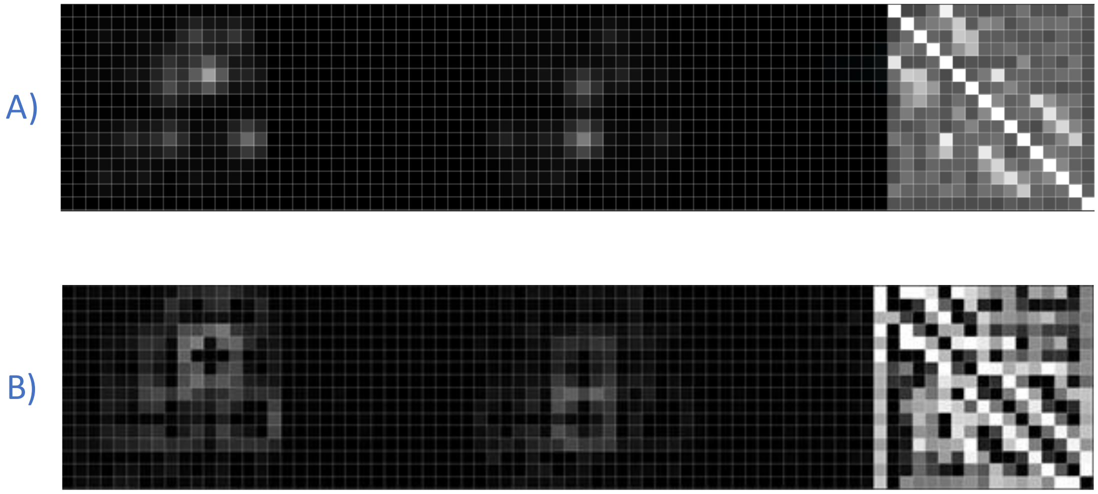


Figure 3-1 A) An example of an input image B) Convolution of the input image with an edge kernel.

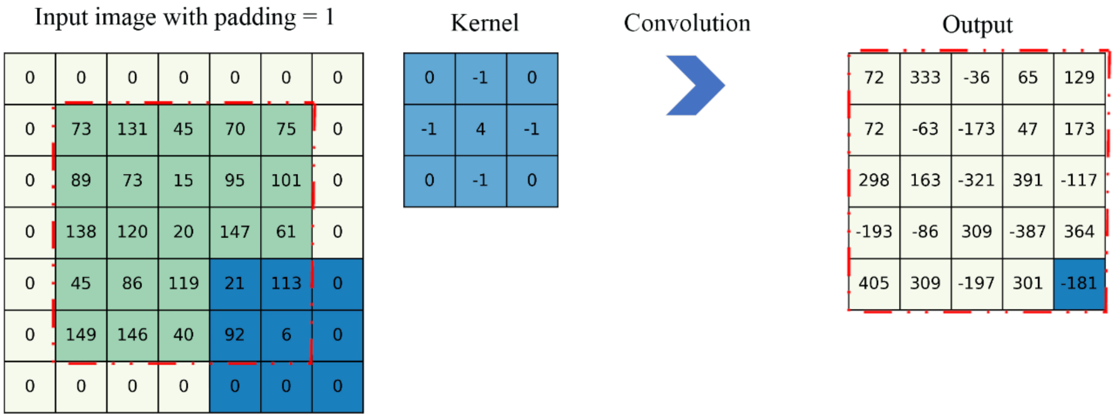
3.1.2 Stride

The stride defines by how many pixels the filter window or pooling window is moved at each step. Large strides result in a smaller overlap between the receptive fields. Hence a smaller feature map will be generated. This comes at the cost of skipping over potentially important spatial features and missing them. Due to the relatively small size of the input images used in this work, going with small strides is still computationally feasible and has the advantage of not missing any important spatial features [64]. As such, we chose minimum stride (i.e., 1) for the convolutional layers and a fairly small stride (i.e., 2) for the pooling layers.

3.1.3 Padding

As described before, a kernel starts at one side of the image and then stepped across the image one stride at a time until reaching the other side. Since the kernel size is larger than 1x1, this means that the pixels at the edge of the input image never get the opportunity to be exposed to the center of the filter. As a result, features located closer to the edges of the input image have a smaller chance of being detected. Padding allows for extending the input image by a few more pixels so that the kernel (filter) starts outside the main input image frame. This will also result in the output image having the same dimensions as the input.

We used padding equal to one pixel for each convolution operation, which is a procedure of symmetrically adding one line of zeros to the border of inputs of convolutional layers. Figure 3-2 shows an example of convolution of one 3×3 kernel and an image with padding.



0	0	0	0	0	0	0
0	73	131	45	70	75	0
0	89	73	15	95	101	0
0	138	120	20	147	61	0
0	45	86	119	21	113	0
0	149	146	40	92	6	0
0	0	0	0	0	0	0

0	-1	0
-1	4	-1
0	-1	0

72	333	-36	65	129
72	-63	-173	47	173
298	163	-321	391	-117
-193	-86	309	-387	364
405	309	-197	301	-181

Figure 3-2 An example of a 3×3 kernel applied to an image with padding equal to 1.

3.1.4 Activation Function

A neural network's activation function is a function that is added to different layers to map nodes' inputs to their corresponding outputs, which help the network learn complex patterns from the input. It is biologically inspired by the activity in our brains, where different inputs cause neurons to fire (or be activated).

In this project, a rectified linear unit (ReLU) activation function follows each convolutional layer of the network. ReLU function is a simple piecewise linear function that returns zero if it gets any negative input and returns the value back if it gets any positive value.

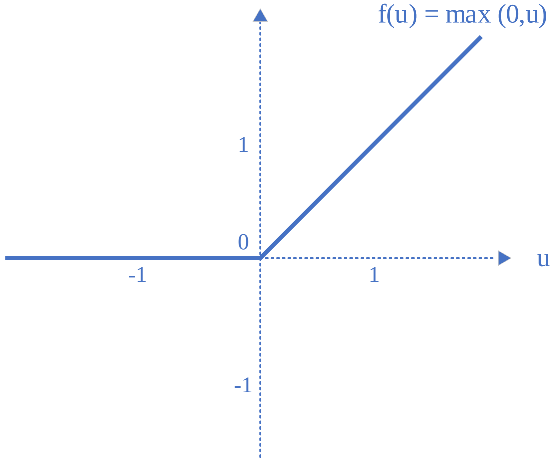


Figure 3-3 ReLU Activation Function

3.1.5 Pooling Layer

Pooling is typically done after one or multiple convolutional layers. The feature map, which is the output of each layer with three dimensions of depth, height, and width, is fed into its following pooling layer as an input, and then they get down-sampled independently by the pooling process, which means a reduction in the width and height of the feature map while maintaining its depth.

There are different kinds of pooling methods in deep learning. We cannot claim that one pooling technique is superior to another in general. The pooling method is usually chosen depending on the data available. Max pooling simply takes the pooling window's maximum value during the slide of the window over the image. Hence, it is helpful when the image's background is dark, and we are just interested in the image's lighter pixels.

While min pooling is exactly the opposite of max-pooling and it takes the minimum value from the set of pixels in the window, and it is usually used when the background is bright and we are interested in the darker pixels. The average pooling technique, on the other hand, calculates the average of pixels in the pooling window. When this pooling technique is employed, it smooths out the picture, so the sharp features may not be seen after that.

Figure 3-4 illustrates different pooling methods applied on images with dark and bright backgrounds.

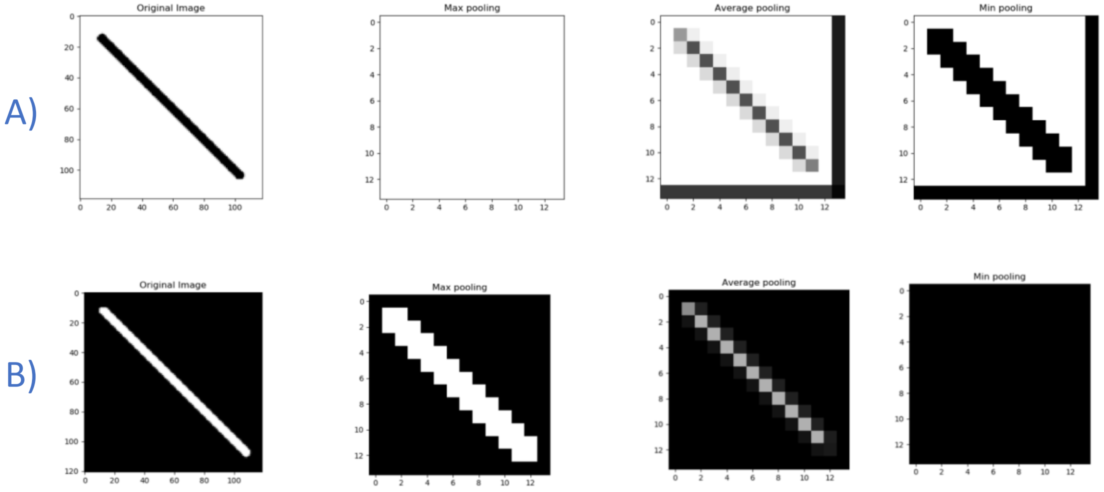


Figure 3-4 two sets of different pooling techniques applied on images with bright background (A) and dark background (B). [65]

Regarding the above-mentioned processes, the pooling layer has three important aspects. It allows us to decrease the number of parameters by reducing the dimension of their inputs, which also reduces computation complexity and training time. It prevents the overfitting of our model to the training set. Also, it helps in extracting suitable features from the images regarding their input images' content.

Based on the above explanation and our simulation results, we defined the pooling structure for the CNN shown in Figure 3-6. The proposed CNN has a max-pooling layer after the first convolutional layer, and then it has three average pooling layers, one for every two remaining convolutional layers. The max-pooling layer gets the useful brighter pixels from the dark background of the first feature map, and the average pooling layers we smooth the pattern in order not to make the model too much biased to the specific sharp bright pixels and instead keep the general pattern.

3.1.6 Fully Connected Layer

As described previously, to correctly classify the pictures, a fully connected layer learns and utilizes the features generated by the preceding layers.

The fully connected layer takes a 1D vector as its input. Therefore, we convert the last pooling layer's output to a 1D vector through a process called flattening. Flattening converts the 3D output of the last pooling layer to a 1D vector. Figure 3-5 illustrates an overview of the flattening procedure.

Based on empirical results, we chose 1024 neurons for the fully connected layers, because a small number of neurons leads to lower accuracy, and on the other hand, a very large number of neurons needs more training with more inputs to reach an acceptable accuracy.

Figure 3-5 shows the procedure of flattening a 3D array into a 1D array.

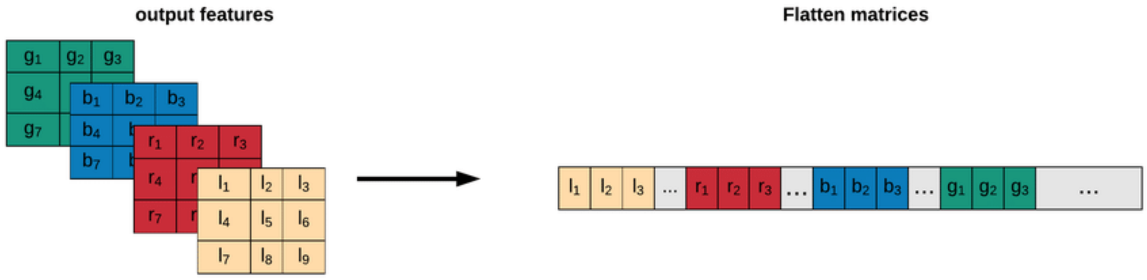


Figure 3-5 Converting 3D input array to a 1D array row using the flattening method. [66]

3.1.7 Proposed Architecture

The proposed architecture of our model is shown in Figure 3-6. The described model is designed similar to the well-known AlexNet, which is employed in large-scale classifications with a large number of inputs in training and with minor differences such as using average pooling and half the number of kernels [67]. It consists of eight convolutional layers, one max-pooling layer, three average pooling layers, and a fully connected layer, plus the input and output layers.

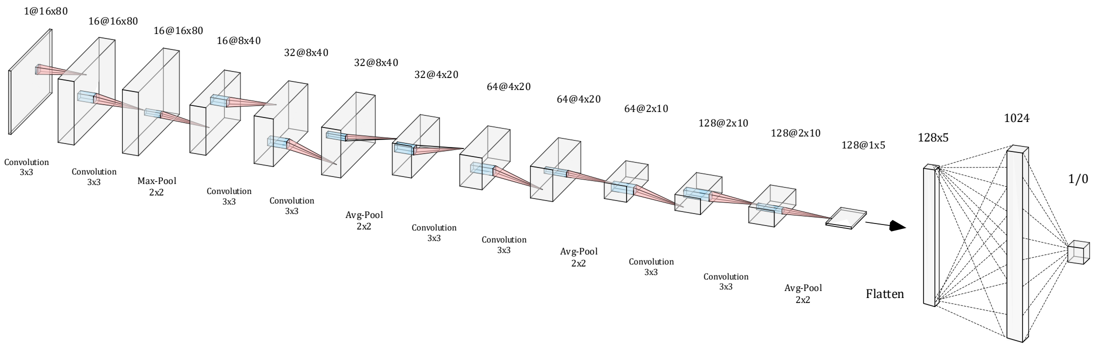


Figure 3-6 The architecture of the proposed CNN model

3.2 Training

As discussed in Chapter 1, motivated by the computational complexity advantage of the spiking architectures, the ultimate goal of this work is to develop a spiking CNN (SCNN). Although there are some methods for training the SCNN directly [34], because of the long training time needed in the direct training approach, we decided to train the conventional CNN and then transform it to SCNN. In this regard, we trained our CNN with the input images. Furthermore, different aspects of training are described in-depth in the following sections.

3.2.1 Imbalanced Dataset

Imbalanced classification is an issue in classifications in which the distribution of instances across classes in the training dataset is not equal. When an imbalanced dataset is used to train a model, the majority class is prioritized throughout the learning process. This implies that the model performs better on the majority class when there are more examples to learn from, but fails to learn critical factors that might aid it in learning minority class due to the insufficient samples.

There are several different methods reported in the literature for learning from imbalanced training datasets, such as data-level methods and algorithm-level methods [68]. The focus of data-level techniques is on balancing distributions in the training set so that it may be used with a standard learning algorithm. Algorithm-level approaches adjust current learning algorithms directly to reduce bias towards the majority class and tune the algorithm to skewed distributions in the training set.

As discussed in section 2.1.2, in this work, we have used overlapping windows during the feature extraction stage, which can be considered as a data-level technique for overcoming the imbalance distribution in the training set. We could increase the number of instances in the minority class using this method. It does, however, lead to a rise in the majority class as well. To address this problem, we use a hybrid sampling technique, which is another data-level technique, and it is detailed in the next subsection.

3.2.1.1 Hybrid Sampling

Random over-sampling simply duplicates instances in the minority class of the training dataset, which is beneficial for balancing the dataset, however, it may lead to overfitting in certain models. On the other hand, random under-sampling removes instances from the majority class to balance the dataset, but it may lead to the loss of crucial information for a model. We developed a hybrid sampling approach in this study, which combines both random over-sampling and random under-sampling [69]. Therefore, by moderate use of each, we can balance the distribution and approximately avoid their negative consequences.

Figure 3-7 illustrates an overview of the hybrid sampling method mentioned above.

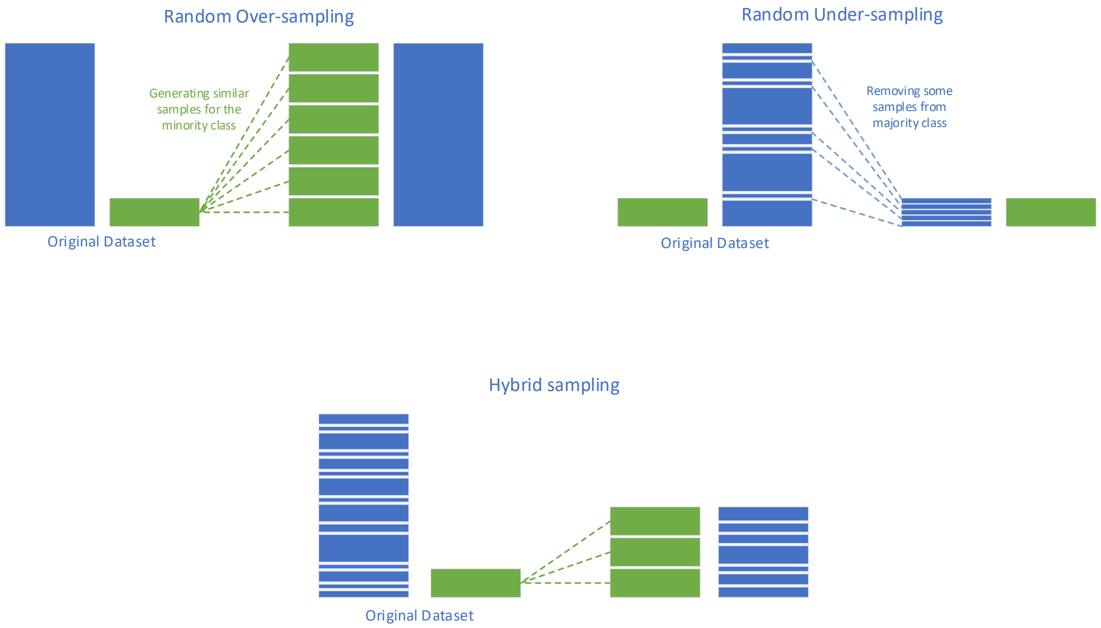


Figure 3-7 An illustration of over-sampling, under-sampling, and hybrid sampling

In the under-sampling part of the hybrid approach, we need to remove some of the non-seizure samples. We could then reduce the seizure to non-seizure samples ratio from about 1/300 to 1/31. However, we made an exception in the under-sampling method to avoid removing the ten samples before the start and after the end of each seizure. Although these are non-seizure samples, our simulation results, as well as the results published in the literature, suggest that they are more important than the other non-seizure samples, and removing them can have a negative effect on the training [50]. For the over-sampling part of the hybrid approach, we generated new seizure samples by duplicating the available seizure samples, which resulted in the ratio changing from 1/31 to 1/7. This means that the dataset was transformed in a way that in every eight samples, we have on average seven non-seizure samples and one seizure sample.

3.2.1.2 Weighted Loss Function

The loss function in deep learning is the distance between the ground truth and the model's estimated output. The learning algorithm's objective is to reduce the error generated during training by the loss function. Since our classification is binary and there are only two classes, a binary cross-entropy loss function is utilized to determine the distance between an estimated output value and its true value [70].

Weighted loss functions are supposed to improve the learning process by supplementing the loss function with a weight array corresponding to coefficients of the learning errors for different classes in the dataset. The goal is to push deep learning models into focusing on situations in which they make greater learning errors, such as misclassification of minority classes [71]. Hence, in this research, we provide a weight array to the loss function, and it will multiply the errors of minority class with a ratio of the number of instances in the non-seizure class to the number of instances in the seizures class, which is equal to 7 after the hybrid sampling.

3.2.2 Optimizer and Batch Size

One of the main objectives throughout the training phase is to determine the optimal weight values that will result in the least amount of loss. Optimization algorithms are in charge of minimizing losses and delivering the most feasible precise results. We used the Adam optimizer, which includes an adaptive learning rate and is optimized for deep neural network training. it has been proved to have a high rate of convergence when it is used to train neural networks with a learning factor of 0.001 [72].

It is commonly known that smaller batch sizes result in quicker training dynamics compared to larger batch sizes. However, this statement has limitations, a batch size equal to 1 often performs very badly due to overfitting, and It is widely accepted that the optimal batch size is a trade-off between 1 and the full training dataset to have a faster convergence besides avoiding the overfitting. Also, the batch size is often dependent on the characteristics of the dataset and the model [73]. In this project, based on empirical results, we chose a batch size equal to 128 to have a reliable performance in training.

3.2.3 Early Stopping

A common challenge in the training of neural networks is determining how long the model should be trained. Insufficient training will result in the model underfitting for the train and test sets. Excessive training, on the other hand, results in the model being overfitted to the train set and performing poorly on the test set [74].

An effective solution to this is called early stopping, which simply is to train on the training dataset but to stop training when performance on the validation set begins to deteriorate. [74].

As discussed in section 2.1.3, the validation set is a portion of the dataset (e.g., 15%) that is kept aside for monitoring the model's performance throughout training. This validation set is not utilized in the model's training, and it is just for monitoring the accuracy. Additionally, for this purpose, it is typical to monitor the loss on a validation dataset. As part of the training process, the model's loss on the train set will be monitored as well.

Epoch is one complete pass through the train set during the training. In CNN methods, it is required to train the model multiple epochs to reach the optimal weights in the model. At the end of each epoch, the model's performance on the train set is assessed against the validation set, which results in an extra computational cost during training. This may be mitigated by assessing the model less often, for example, every 2, 5, or 10 training epochs in studies that need a high number of epochs. In this project, for both cross-patient and patient-specific training, the number of epochs is equal to 9 on average; thus, checking the validation loss in every epoch would not have a huge computational cost.

We monitored the validation loss and training loss at the conclusion of each epoch and then compared them to their values from the previous epoch. Following that, if the validation loss increases while the training loss decreases, we ceased training; otherwise, we saved the model and continued training as long as it was beneficial to the model 's accuracy.

3.3 Transforming to SCNN and Encoding

As discussed in Chapter 1, neuromorphic computing models inspired by biology are being studied in an attempt to replicate the computational efficiency and innate potential for event-based asynchronous processing of the human brain while conducting classification. In this section, the transformation of CNN to SCNN, encoding, and testing are discussed in detail.

3.3.1 Transforming to SCNN

A neuron is the elementary block of the nervous system in the human brain. Neurons transmit some signals like an electrical impulse known as an action potential (spike) through the nerves. Neuronal dynamics can be thought of as a combination of an integrating process and a mechanism that activates action potentials above a certain threshold voltage. Indeed, firing events are generally described in experiments when the membrane potential of a neuron exceeds a certain threshold value. A formal threshold voltage (V_{th}) is defined as a minimum voltage for spike initiation. Also, there is reset voltage which represents the voltage of neurons after each spike firing. In short, we say a neuron fires a spike when the voltage V_t (which contains the cumulative voltage of all inputs) exceeds V_{th} from below. Integrate-and-fire models is a neuron model in which action potentials are described as events [34], [75].

Figure 3-8 illustrates an overview of the different processes happening in the Integrate-and-fire neurons.

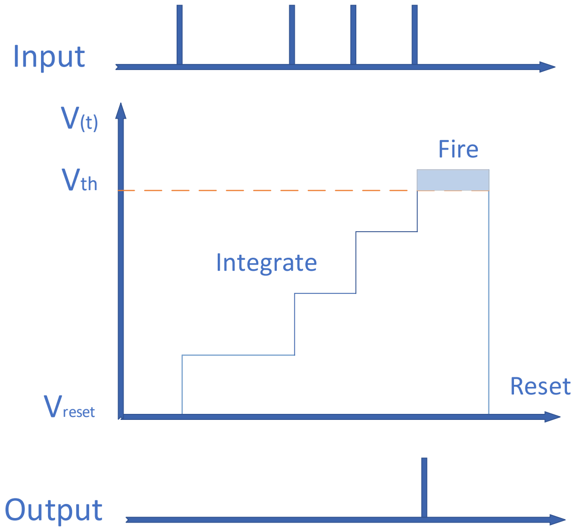


Figure 3-8 an overview of integrating, fire and reset in the Integrate-and-fire neurons

Furthermore, Figure 3-9 shows a model of the integrate-and-fire neuron in a spiking convolutional neural network.

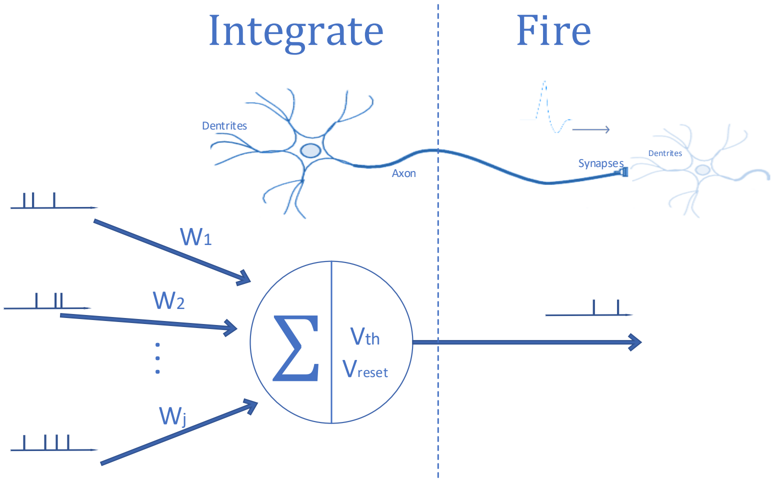


Figure 3-9 Model of an integrate-and-fire neuron used in convolutional layers and fully connected layers of SINABS library.

We converted the CNN and its weights into an SCNN using the SINABS library, which is a publicly available library for this purpose in the Pytorch framework [68]. Both SCNN and CNN networks have the same structure; Therefore, the trained CNN weights (32-bit floating-point numbers) of each layer can be mapped directly on SCNN. We didn’t utilize any activation function at the last output layer. This is mainly because, after the conversion of CNN to SCNN, we only have a series of binary values representing the actual spikes in the neural network of a human's brain. Furthermore, the activation function of the integrate-and-fire neurons employed in the spiking model works similarly to ReLU. Hence, there is no need for ReLU as well.

The resting potential of all neurons in all layers is reset to zero to make the model hardware-friendly. The data communicated across the network has a single bit magnitude (i.e., 0 or 1) and asynchronous timing (i.e., the bit-stream density is not consistent). Using the single-bit data flowing through the network allows for replacing the energy-consuming floating-point multiplications with energy-efficient adding operations in all layers. This results in the SCNN requiring far less computational resources than CNN, making it more hardware-friendly.

3.3.2 Encoding

The encoder converts the 2D input images made of floating-point values into 3D time-dependent arrays of binary values. Given a 2D input image with size of [Height, Width], the output would have to be [Time Step, Height, Width] consisting of 0 or 1. According to the SINABS library [76], there is a one-millisecond time distance between each 2D array with the size of [Height, Width] during the process of feeding the inputs to SCNN.

In order to create a sequence of spikes in the time domain with a one-millisecond time distance between them, first, the pixel magnitude of images were scaled to the 0-to-1 range, and then the encoder function generates a random matrix of the same width and height as the input images using uniform random values at each time step. The produced random numbers, which are between 0 and 1, get compared to every value of the pixels in the original input images. The associated spike value will be set to 1 if the random number is less than the corresponding value in the original image; else, the spike value will be set to 0. The encoder reduces the size of processed values in the images from 32-bit floating points to a series of 1-bit numbers (spike train) while maintaining the primary characteristics of inputs, which allow us to achieve our energy-efficiency goal [77], [78].

Figure 3-10 represents the overall structure of the proposed SCNN.

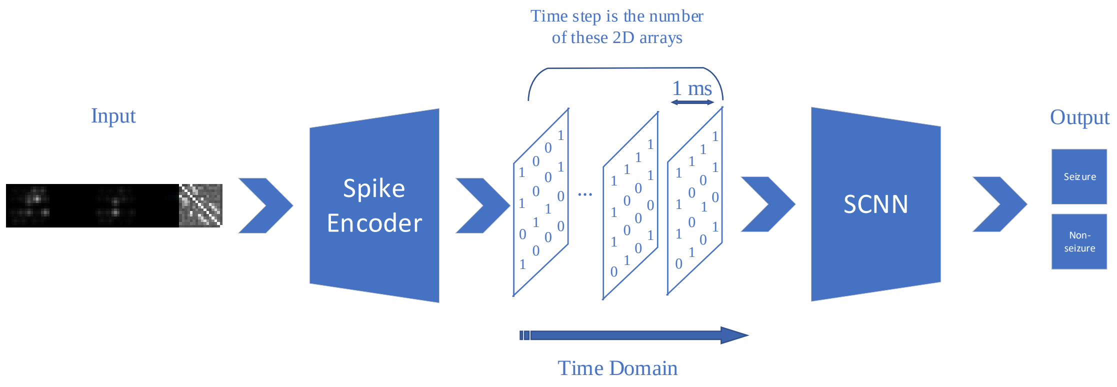


Figure 3-10 The overall structure of SCNN

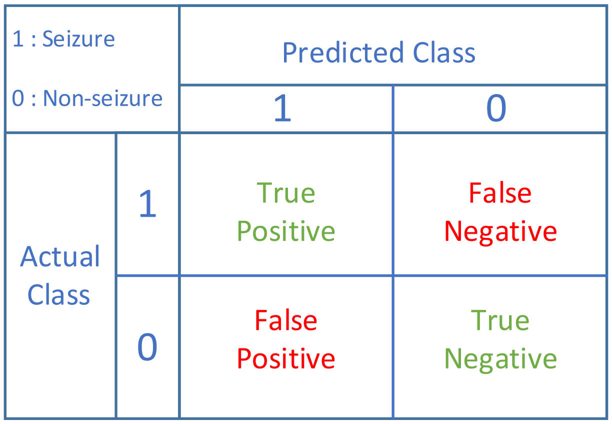
In conclusion, we used the training set and validation set to train the normal CNN, and then the trained weights in the layers of the model got mapped on a corresponding SCNN with the same architecture. To evaluate the performance of the SCNN, we need to calculate the accuracy of the test set. In the testing stage, the input data is first transformed into spike sequences using a spike encoder. Then, the spike sequences are fed to the SCNN.

3.4 Results

In this section, our project benchmarks are compared with the benchmarks of the other research in both patient-specific and cross-patient aspects. A noteworthy point to mention is that in this research, not only do we focus on the classification accuracy, but also, we had to consider the energy efficiency of the model.

3.4.1 Classification Metrics

A confusion matrix summarizes the classification problem's predicted outputs. It informs us not only about the errors which the classifier makes but also about the types of errors that are made. Figure 3-11 demonstrates an overview of a confusion matrix with a visual summary of true positive, true negative, false negative, and false negative.



1 : Seizure 0 : Non-seizure		Predicted Class	
		1	0
Actual Class	1	True Positive	False Negative
	0	False Positive	True Negative

Figure 3-11 Confusion Matrix

The most commonly used quantitative metrics used for evaluating the performance of a seizure classifier are sensitivity and specificity. Sensitivity, also known as recall or true positive rate (TPR), is a measure that reflects the proportion of seizure instances (positives) that the model properly classifies as seizures. The sensitivity equation is as follows:

$$Sensitivity = \frac{True\ Positive}{True\ Positive + False\ Negative} \qquad \text{(Eq 3.1)}$$

Specificity, also known as true negative rate (TNR), is the ratio of non-seizure instances properly identified as non-seizure by the model. Its equation is as follows:

$$Specificity = \frac{True\ Negative}{True\ Negative + False\ Positive} \qquad \text{(Eq 3.2)}$$

False alarm per hour is another metric used in the evaluation of cross-patient seizure detection models. It has been defined differently in various kinds of literature. In the state-of-the-art cross-patient seizure detection papers, it is considered as the average number of false positives per hour. Also, as an exception, the successive false positives in the seizure detection method are considered as only one false positive [11], [50], [79].

Latency is another metric that is important for closed-loop systems where the detection results are used for generating bio-feedback stimulations to the brain. Therefore, it is an important metric for implantable patient-specific applications. However, latency depends on the computational resources used in the seizure detector system, and therefore, the latency of an algorithm has a device-to-device variation.

The computational cost of each classification is one of the important hardware-related metrics that must be considered when trying to maximize sensitivity and specificity. In this research, we tried to compare the computation cost of one classification of a time window in our model with state-of-the-art cross-patient classifiers.

In the SCNN method, one of the main parameters affecting the accuracy, latency, and computational cost is the time step. Technically, using a greater time step in the encoding function leads to a higher accuracy; however, it linearly increases the latency and the energy consumption of the classifier. In this project, we chose a time step equal to 10 in the encoder function as a trade-off between the above-mentioned metrics. Figure 3-12 shows the effect of using different time steps on the accuracy of the classifier.

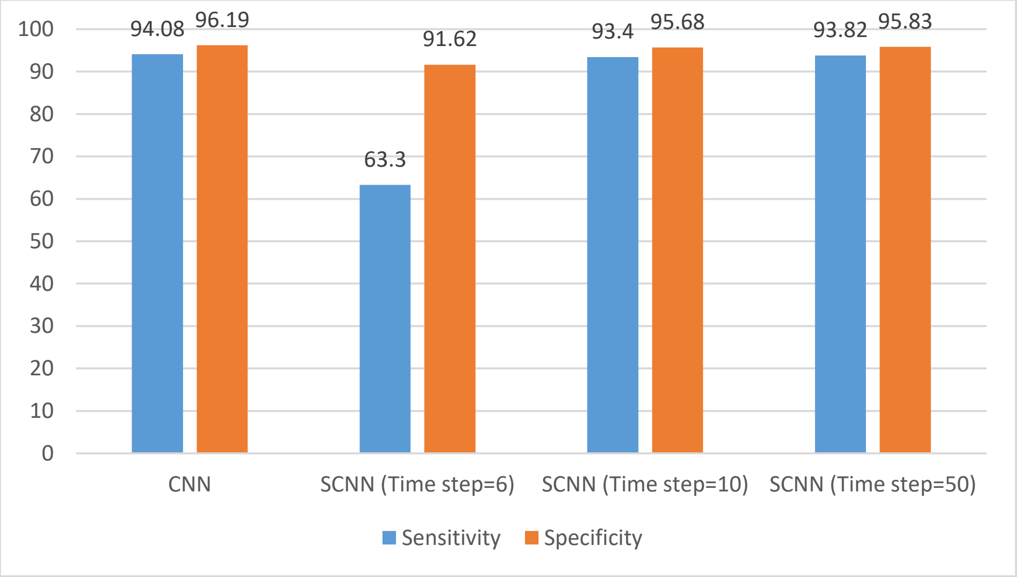


Figure 3-12 A comparison of cross-patient model's accuracy using different time steps in the encoder for patient "2"

3.4.2 Results and Comparison

The main focus of this work was the cross-patient SCNN method, and almost all the optimizations have been done in this regard. However, in this subsection, the results of different methods developed in this work are represented. We implemented two different neural network models, a conventional CNN and an SCNN. Each of these is trained and tested for patient-specific and cross-patient detection. Also, the proportion of test set, train set, and validation sets are explained in detail in 2.1.3.

3.4.2.1 Patient-specific CNN

The patient-specific CNN algorithm was evaluated using a test set proportion of each patient of the pre-recorded EEG from the CHB-MIT database. Our results show that it had an average detection sensitivity of 96.2% and a specificity of 99.51%, respectively. Figure 3-13 and Table 3-1 show the sensitivity and specificity of the proposed patient-specific CNN.

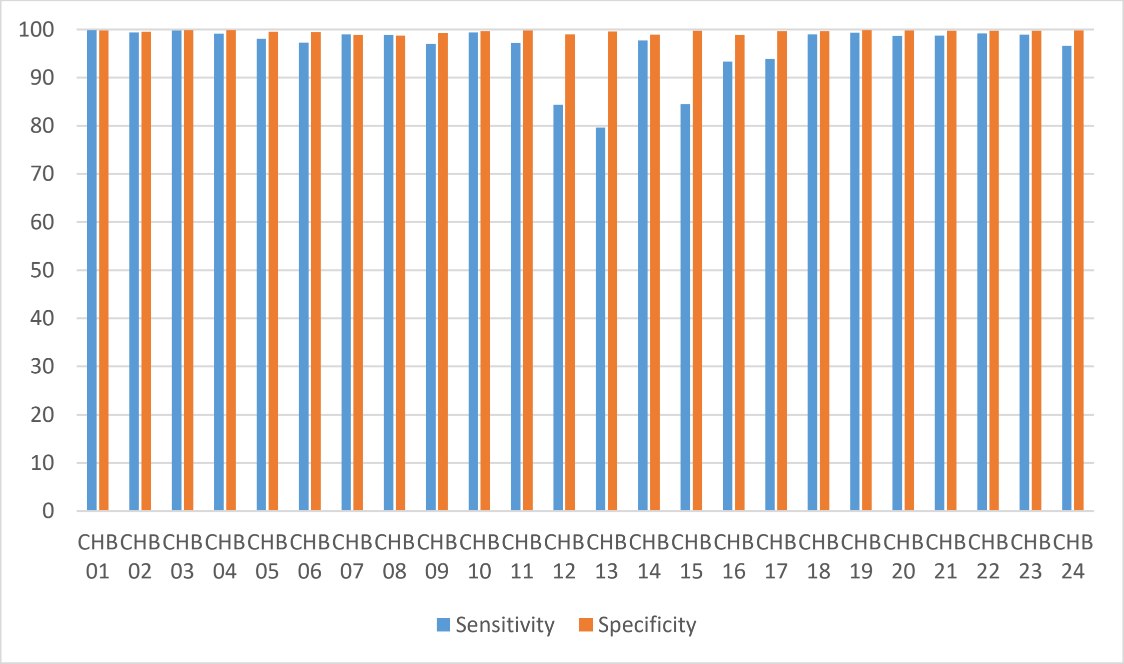


Figure 3-13 Sensitivity and specificity of the patient-specific CNN

Table 3-1 Sensitivity and specificity of the patient-specific CNN

	Sensitivity %	Specificity %
CHB 01	99.87	99.76
CHB 02	99.42	99.52
CHB 03	99.8	99.83
CHB 04	99.11	99.84
CHB 05	98.07	99.52
CHB 06	97.23	99.43
CHB 07	98.97	98.85
CHB 08	98.86	98.74
CHB 09	97.02	99.25
CHB 10	99.4	99.67
CHB 11	97.22	99.82
CHB 12	84.34	99.01
CHB 13	79.64	99.6
CHB 14	97.71	98.92
CHB 15	84.47	99.7
CHB 16	93.35	98.88
CHB 17	93.89	99.65
CHB 18	98.96	99.68
CHB 19	99.32	99.86
CHB 20	98.68	99.81
CHB 21	98.71	99.73
CHB 22	99.19	99.69
CHB 23	98.94	99.7
CHB 24	96.61	99.81
Average	96.20	99.51

3.4.2.2 Cross-patient CNN

We evaluated the performance of the cross-patient CNN algorithm using the CHB-MIT database. As it was explained in section 2.1.3, in order to test each patient in the cross-patient method, we train the model using the other 22 patients (one patient had two sets, and the second set was recorded a long time after the first set). Our results show that it had an average detection sensitivity of 85.88% and a specificity of 86.87%, respectively. Figure 3-14 and Table 3-2 show the sensitivity and specificity of the proposed cross-patient CNN.

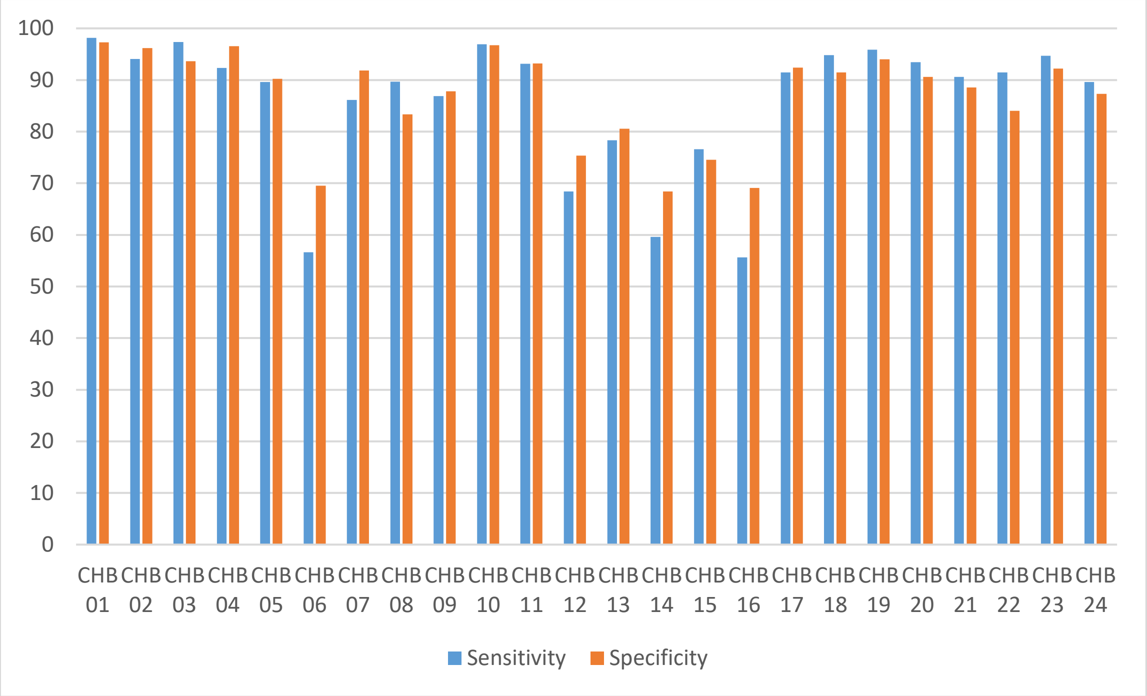


Figure 3-14 Sensitivity and specificity of the cross-patient CNN

In both patient-specific and cross-patient methods, There are some patients (e.g., patients 12, 13, 15) that our model could not have a good detection accuracy in their dataset. The state-of-the-art works also have reported these anomalies for these patients [11], [50]. This may be because of the noises and variations in EEG monitoring devices' characteristics, or maybe the insufficient number of seizures in the CHB-MIT dataset, which brings about inefficient training.

Table 3-2 Sensitivity and specificity of the cross-patient CNN

	Sensitivity	Specificity
CHB 01	98.14	97.31
CHB 02	94.08	96.19
CHB 03	97.36	93.63
CHB 04	92.33	96.56
CHB 05	89.62	90.24
CHB 06	56.6	69.5
CHB 07	86.11	91.82
CHB 08	89.68	83.36
CHB 09	86.87	87.83
CHB 10	96.93	96.71
CHB 11	93.11	93.22
CHB 12	68.41	75.36
CHB 13	78.31	80.55
CHB 14	59.6	68.42
CHB 15	76.58	74.54
CHB 16	55.62	69.07
CHB 17	91.45	92.41
CHB 18	94.82	91.45
CHB 19	95.89	94.02
CHB 20	93.46	90.62
CHB 21	90.6	88.54
CHB 22	91.44	84
CHB 23	94.67	92.23
CHB 24	89.61	87.34
Average	85.88	86.87

3.4.2.3 Patient-specific SCNN

After transforming CNN to SCNN, the patient-specific SCNN method was evaluated using a test set of each patient’s EEG data from the CHB-MIT database. Our results show that it had an average detection sensitivity of 95.04% and a specificity of 99.42%, respectively.

Figure 3-15 and Table 3-3 show the sensitivity and specificity of the proposed patient-specific SCNN.

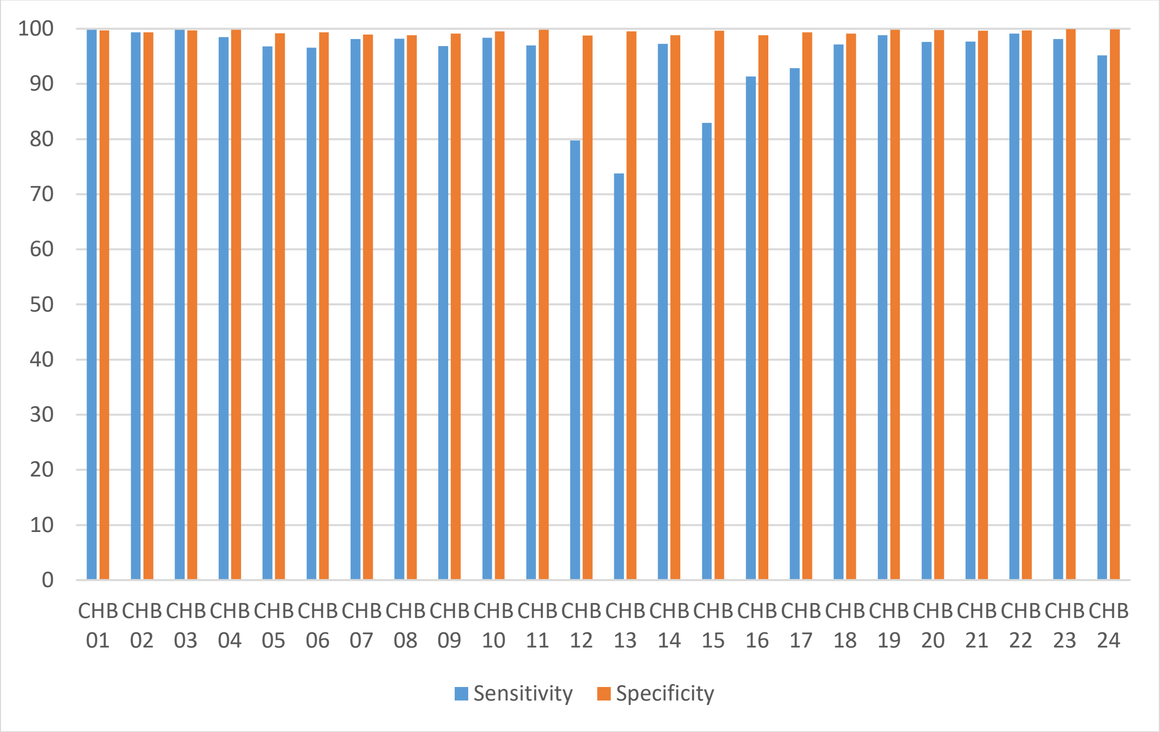


Figure 3-15 Sensitivity and specificity of the patient-specific SCNN

Table 3-3 Sensitivity and specificity of the patient-specific SCNN

	Sensitivity	**Specificity**
CHB 01	99.82	99.71
CHB 02	99.37	99.38
CHB 03	99.81	99.72
CHB 04	98.51	99.8
CHB 05	96.78	99.21
CHB 06	96.56	99.35
CHB 07	98.12	98.93
CHB 08	98.22	98.86
CHB 09	96.86	99.1
CHB 10	98.35	99.52
CHB 11	96.97	99.82
CHB 12	79.73	98.75
CHB 13	73.74	99.51
CHB 14	97.27	98.84
CHB 15	82.94	99.64
CHB 16	91.37	98.82
CHB 17	92.84	99.33
CHB 18	97.16	99.12
CHB 19	98.85	99.81
CHB 20	97.59	99.79
CHB 21	97.66	99.65
CHB 22	99.1	99.69
CHB 23	98.14	99.93
CHB 24	95.15	99.89
Average	95.04	99.42

Table 2-1 compares the result of our patient-specific CNN and SCNN with state-of-the-art.

Table 3-4 Comparison with the state-of-the-art algorithms for patient-specific seizure detection

Authors	Year	Classifier	Dataset	Sensitivity%	Specificity%
Karimi et al. [8]	2020	SVM	CHB-MIT	96.87	99.95
Thara et al.[80]	2019	DNN	CHB-MIT	98.17	94.93
Rahib et al. [81]	2020	CNN	CHB-MIT	96.67	98.33
Gao et al. [82]	2020	CNN	CHB-MIT	92.6	97.1
Tian et al. [83]	2021	S-CNN	CHB-MIT (Patients 1,5,6,8,10,14,22)	95.1	99.2
This Work	2021	S-CNN	CHB-MIT (Patients 1,5,6,8,10,14,22)	98.01	99.31
This Work	2021	CNN	CHB-MIT	96.2	99.51
This Work	2021	S-CNN	CHB-MIT	95.04	99.42

As it is shown in Table 3-4, while the model was not optimized for a patient-specific application, its performance is competitive to state-of-the-art. However, as evident from the table, such a performance has been achieved using data-driven conventional machine learning methods such as support vector machines (SVMs) which have a significantly lower computational cost (e.g., [8]).

3.4.2.4 Cross-patient SCNN

Because of the patient-to-patient variations of seizures happening in different people's EEG, there are only a few deep learning research that could lead to acceptable results for the cross-patient seizure detection. However, deep learning algorithms are extremely power-consuming because of the large number of multiplication existing in their algorithm [84].

The cross-patient SCNN algorithm was evaluated on test sets made from the pre-recorded EEG from the CHB-MIT database. It was determined that the detection had an average sensitivity of 83.02% and a specificity of 86.31%, respectively. Figure 3-14 and Table 3-2 show the sensitivity and specificity of the proposed cross-patient SCNN.

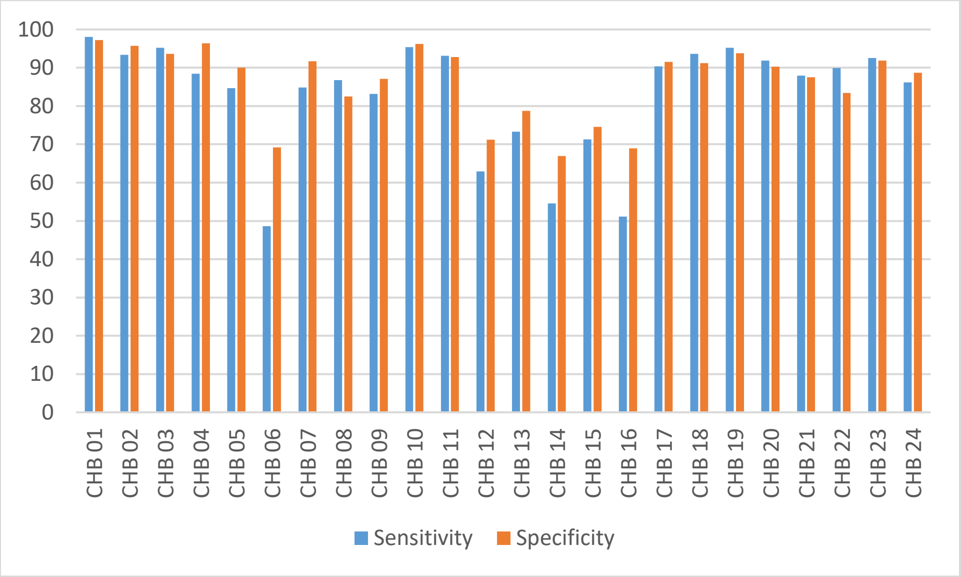


Figure 3-16 Sensitivity and specificity of the cross-patient SCNN

Table 3-5 Sensitivity and specificity of the cross-patient SCNN

	Sensitivity	Specificity
CHB 01	98.03	97.24
CHB 02	93.4	95.68
CHB 03	95.22	93.65
CHB 04	88.43	96.42
CHB 05	84.65	90.03
CHB 06	48.63	69.19
CHB 07	84.82	91.71
CHB 08	86.73	82.53
CHB 09	83.2	87.13
CHB 10	95.38	96.2
CHB 11	93.11	92.81
CHB 12	62.93	71.2
CHB 13	73.28	78.74
CHB 14	54.56	66.98
CHB 15	71.33	74.54
CHB 16	51.11	68.97
CHB 17	90.33	91.56
CHB 18	93.6	91.24
CHB 19	95.2	93.82
CHB 20	91.87	90.32
CHB 21	87.92	87.5
CHB 22	89.92	83.44
CHB 23	92.54	91.83
CHB 24	86.2	88.73
Average	83.02	86.31

Furthermore, Table 3-6 compares the cross-patient results of this work with state-of-the-art. Additionally, although the number of multiplications and additions was not reported in their studies, we measured them in order to compare the energy consumption of our results to theirs.

Table 3-6 Comparison with the state-of-the-art algorithms for EEG-based cross-patient seizure detection

Authors	**Year**	**Classifier**	**Dataset**	**Sensitivity %**	**Specificity %**	**False Alarm (/hour)**	**Number of ADD**	**Number of MUL**	**Least Latency**	**Energy Consumption In 45nm tech**
Thodorof et al. [11]	2016	CNN + RNN	CHB-MIT	85.16	83.21	0.8	13.7 M	6.9 M	30s	37.86 μj
Alkanhal et al. [50]	2018	CNN	CHB-MIT	87.95	86.5	0.75	20.4 M	10.3 M	30s	56.47 uj
This Work	2021	CNN	CHB-MIT	85.88	86.87	0.75	34.1 M	17.2 M	15s	94 uj
This work	2021	SCNN	CHB-MIT	83.02	86.31	0.77	341 M	0	15s	10.2 μj

Using a library in python called torch stat, we calculated the number of additions through our model for each classification which was 34.1M. Thus, since the time step in our project is equal to 10, we needed to multiply the number of additions by the time step value (10) to obtain the total number of additions in one image classification. Furthermore, in order to compare the energy efficiency of our work with others, First, we estimated the number of multiplications and additions in one classification of each model in Table 3-6 using the torch stat library. Then using the estimated energy cost of multiplications and additions for integer and floating-point numbers in 0.9 volts and 45nm technology proposed by Horowitz [35], we calculated the approximate energy consumption of those models.

The energy consumption of conventional CNN in our work is high. Because, unlike the other CNN methods, we were unable to use some of the regularization techniques such as the dropout layer responsible for randomly setting layers’ input to zero and batch-normalization technique for normalization of the layers' inputs, which both are supposed to improve performance and avoid overfitting in CNN. Therefore, we had to increase the number of pooling layers and convolutional layers to compensate for this issue. This problem is because of the neuromorphic nature of the model inspired by biology and the binary values of data representing spike trains flowing in the SCNN. With this in mind, comparing the proposed SCNN with the other CNNs, it has been confirmed that while the sensitivity and specificity might have been degraded when we migrated from CNN to SCNN, the computational cost is improved significantly.

Figure 3-17 shows a comparison of the energy cost per classification between our work and the state-of-the-art, which represents at least a 75% reduction in energy cost compared to the other works.

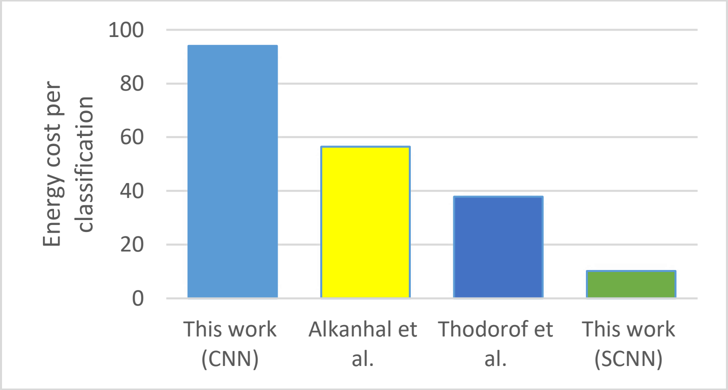


Figure 3-17 Comparison of energy cost per classification between our work and the state-of-the-art

Chapter 4

Conclusions and Future Works

4.1 Conclusion

We presented the design, development, and characterization of energy-efficient spiking neural network for seizure detection. The developed SCNN learns temporal, spatial, and spectral features from multi-channel EEG signals, and is tolerant to both the variations between seizures and among different patients, as well as the noise inherent in EEG data. Having the image data from spectral energy and phase-locking value features made us able to use CNN to do the classification, as opposed to standard EEG analysis techniques that ignore some spatial information. An additional benefit of leveraging spatial information of recorded EEG data is that it enables seizure localization, which is particularly beneficial for pre-surgical monitoring applications.

The suggested architecture's performance demonstrated that utilizing a multi-second time window may be used instead of a recurrent neural network, thus reducing the overall design complexity. Additionally, we addressed the issue of imbalanced datasets by selecting the proper methods such as overlapping windows, hybrid sampling, and weighted loss function. However, the variance of the predicted distribution continued to be high. Indeed, by training on a small portion of the non-seizure class and testing on a much larger collection of non-seizure samples, minor parameter changes may have a substantial effect on the accuracy.

Automation of this procedure may benefit patients with epilepsy in terms of diagnosis, monitoring, and therapy planning. The method may be especially advantageous in underdeveloped countries where access to neurology is limited.

While cross-patient studies had positive outcomes, the sensitivity for some patients was poor. Their seizure pattern was most likely different from what was provided in the training set, emphasizing the need for more data as a training set. This is a challenge underlying the development of deep learning methods with limited datasets.

In the spiking CNN, the spike encoder converts retrieved images to spike trains which can be evaluated in SCNN. To maintain high accuracy while minimizing computing resource consumption in training, we first built and trained the CNN and then mapped obtained optimized weights to Spiking CNN. As demonstrated by the results, the proposed approach not only achieves high sensitivity but also manages to reduce computation complexity by at least 73% compared to the state-of-the-art.

Table 2-1 shows a summary of the average sensitivity and specificity in all proposed models.

Table 4-1 A summary of the average sensitivity and specificity in all proposed models

		Average Sensitivity	**Average Specificity**
CNN	Patient-specific	96.2	99.51
	Cross-patient	85.88	86.87
SCNN	Patient-specific	95.04	99.42
	Cross-patient	83.02	86.31

4.2 Future Works

Currently, the barriers preventing researchers from progressing in seizure detection systems, especially the cross-patient models, are as follows:

First, many clinical datasets include only selected segments of EEG data, which itself is incompatible with real-world applications that require detection from real-time EEG signals, and clinical datasets are typically not publicly available.

Second, although the quantity of datasets in this area is vast, they cannot be effectively merged owing to differences in frequency sampling or other characteristics of EEG devices, resulting in a limited amount of total useable data for training a model.

Finally, deep learning models need enormous computing resources, which are not widely accessible. The researchers must work in this direction in order to interfere with epileptic seizures using energy-efficient technology, thus assisting patients at any time and in any location.

From the algorithmic perspective, deep learning has made significant progress toward human-level accuracy performance on a variety of recognition tasks [85]. On the hardware side, combining neuromorphic and traditional methods offers the potential of a new generation of embedded, real-time systems, but first needs resolving fundamental structural and operational gaps between the spiking algorithms and contemporary hardware designs. Neuromorphic computing systems learn and infer in real-time at low power levels. They need computational units that are logically similar to the brain's neurons. Each neuron in the SNN has the ability to fire independently, and when it does, it transmits pulsed signals to other neurons in the network, which directly affect their electrical states [86].

Despite decades of study, relatively few published papers have shown that today's neuromorphic devices can do quantitative computations. Now, this is beginning to change with the introduction of Intel's Loihi neuromorphic processor, which is intended to handle a wide variety of spiking neural networks at a size, speed, and set of features comparable to those of state-of-the-art conventional computing architectures [87]. This kind of processors not only can be used for biomedical purposes like seizure detection, But also can be employed for any other kind of machine learning and artificial intelligence methods.

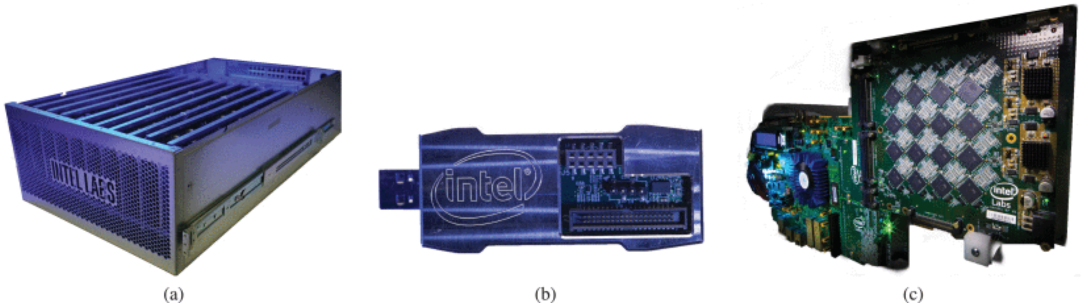


Figure 4-1 (a) Pohoiki Springs large-scale arrangement with about 768 Loihi chips. (b) Kapoho Bay USB system with two Loihi chips, event-based camera interfaces, and sensors. (c) Nahuku board interfaced with the Intel Arria 10 FPGA with 32-chip expansion. [87]

4.2.1 Future of This Project

One of the main constraints in the training of the cross-patient algorithm is the insufficient number of seizures in the CHB-MIT dataset which led to relatively lower accuracy compared to the patient-specific methods. Thus, it would be a good idea to use other available datasets such as the Freiburg dataset, which has the same 256Hz sampling rate [46], to extend our training dataset for the cross-patient method. However, we must normalize the values of both datasets together to reach a better performance. Lastly, We can use different approaches to direct future development on this project.

First, we can implement the S-CNN model on an FPGA using the Caffe library in Python [88]. This framework can convert the model to a Verilog-based model compatible with FPGAs. FPGAs are especially helpful for developing application-specific integrated circuits (ASICs) with minimum hardware usage. However, they operate in the digital domain and are just a simulation of spiking networks, not a true neuromorphic processor like Loihi. Moreover, we can integrate our product with analog electronic circuits on the same chip to be able to conduct both EEG processing and recording together and eventually, leading to in-vivo testing of the prototype on humans.

Second, using frameworks such as Caffe2 or TensorflowLite, the deep learning trained models can be used on the android and ios smartphones for inference [89]. Then it will be possible for us to do the classifications based on the input EEG data receiving from an energy-efficient Bluetooth connection with the EEG recording device.

Third, we can use cloud-based deep learning inference using cloud platforms such as Amazon AWS or NVIDIA's inference [90]. Again, we can conduct this approach using a smartphone to collect real-time raw EEG data with a Bluetooth connection from a recording device. Then send it to the cloud platform for feature extraction and classification, and finally receive the resulting predicted label as an inference from the server. Moreover, an alarm can be set for patient, patient’s family, and a doctor.

References

[1] S. A and T. S, “The Epidemiology of Global Epilepsy,” *Neurol. Clin.*, vol. 34, no. 4, pp. 837–847, Nov. 2016, doi: 10.1016/J.NCL.2016.06.015.

[2] L. J. Greenfield, J. D. Geyer, and P. R. Carney, “Reading EEGs : a practical approach,” p. 444, 2020.

[3] A. J. Casson, “Wearable EEG and beyond,” *Biomed. Eng. Lett.*, vol. 9, no. 1, p. 53, Feb. 2019, doi: 10.1007/S13534-018-00093-6.

[4] L. Forsgren, E. Beghi, A. Õun, and M. Sillanpää, “The epidemiology of epilepsy in Europe - A systematic review,” *European Journal of Neurology*, vol. 12, no. 4. John Wiley & Sons, Ltd, pp. 245–253, Apr. 01, 2005, doi: 10.1111/j.1468-1331.2004.00992.x.

[5] M. F, A. RG, E. CE, and L. K, “Seizure prediction: the long and winding road,” *Brain*, vol. 130, no. Pt 2, pp. 314–333, 2007, doi: 10.1093/BRAIN/AWL241.

[6] “Patients wait as long as a year for epilepsy test | London Free Press.” https://lfpress.com/news/local-news/patients-wait-as-long-as-a-year-for-epilepsy-test (accessed Sep. 05, 2021).

[7] J. Echauz *et al.*, “Monitoring, signal analysis, and control of epileptic seizures: A

paradigm in brain research," *2007 Mediterr. Conf. Control Autom. MED*, 2007, doi: 10.1109/MED.2007.4433785.

[8] M. R. Karimi and H. Kassiri, "A Multi-Feature Nonlinear-SVM Seizure Detection Algorithm with Patient-Specific Channel Selection and Feature Customization," Sep. 2020, pp. 1–5, doi: 10.1109/iscas45731.2020.9180729.

[9] A. Subasi, "Application of adaptive neuro-fuzzy inference system for epileptic seizure detection using wavelet feature extraction," *Comput. Biol. Med.*, vol. 37, no. 2, pp. 227–244, Feb. 2007, doi: 10.1016/J.COMPBIOMED.2005.12.003.

[10] A. H. Shoeb, "Application of Machine Learning to Epileptic Seizure Onset Detection and Treatment MASS NSl OF TECHNOLOGY," Massachusetts Institute of Technology, 2009.

[11] P. Thodoroff, J. Pineau, and A. Lim, "Learning Robust Features using Deep Learning for Automatic Seizure Detection," Jul. 2016, Accessed: Aug. 24, 2021. [Online]. Available: https://arxiv.org/abs/1608.00220v1.

[12] A. Craik, Y. He, and J. L. Contreras-Vidal, "Deep learning for electroencephalogram (EEG) classification tasks: a review," *J. Neural Eng.*, vol. 16, no. 3, p. 031001, Apr. 2019, doi: 10.1088/1741-2552/AB0AB5.

[13] A. Dabbaghian, T. Yousefi, S. Z. Fatmi, P. Shafia, and H. Kassiri, "A 9.2-gram Fully-Flexible Wireless Ambulatory EEG Monitoring and Diagnostics Headband

with Analog Motion Artifact Detection and Compensation," *IEEE Trans. Biomed. Circuits Syst.*, 2019, doi: 10.1109/TBCAS.2019.2936327.

[14] M. G. Bleichner and S. Debener, "Concealed, unobtrusive ear-centered EEG acquisition: Ceegrids for transparent EEG," *Front. Hum. Neurosci.*, vol. 11, p. 163, Apr. 2017, doi: 10.3389/fnhum.2017.00163.

[15] Y. Gu *et al.*, "Comparison between scalp EEG and behind-the-ear EEG for development of a wearable seizure detection system for patients with focal epilepsy," *Sensors (Switzerland)*, vol. 18, no. 1, Jan. 2018, doi: 10.3390/s18010029.

[16] H. Kassiri *et al.*, "Closed-Loop Neurostimulators: A Survey and A Seizure-Predicting Design Example for Intractable Epilepsy Treatment," *IEEE Trans. Biomed. Circuits Syst.*, vol. 11, no. 5, pp. 1026–1040, Oct. 2017, doi: 10.1109/TBCAS.2017.2694638.

[17] M. T. Salam, H. Kassiri, R. Genov, and J. L. Perez Velazquez, "Rapid brief feedback intracerebral stimulation based on real-time desynchronization detection preceding seizures stops the generation of convulsive paroxysms," *Epilepsia*, vol. 56, no. 8, pp. 1227–1238, Aug. 2015, doi: 10.1111/epi.13064.

[18] H. Kassiri *et al.*, "All-wireless 64-channel 0.013mm2/ch closed-loop neurostimulator with rail-to-rail DC offset removal," in *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, Mar. 2017, vol. 60, pp. 452–453, doi: 10.1109/ISSCC.2017.7870456.

[19] M. R. Pazhouhandeh, H. Kassiri, A. Shoukry, I. Wesspapir, P. Carlen, and R. Genov, "Artifact-Tolerant Opamp-Less Delta-Modulated Bidirectional Neuro-Interface," in *IEEE Symposium on VLSI Circuits, Digest of Technical Papers*, Oct. 2018, vol. 2018-June, pp. 127–128, doi: 10.1109/VLSIC.2018.8502286.

[20] H. Kassiri *et al.*, "Rail-to-rail-input dual-radio 64-channel closed-loop neurostimulator," *IEEE J. Solid-State Circuits*, vol. 52, no. 11, pp. 2793–2810, Nov. 2017, doi: 10.1109/JSSC.2017.2749426.

[21] S. U. Schuele and H. O. Lüders, "Intractable epilepsy: management and therapeutic alternatives," *Lancet Neurol.*, vol. 7, no. 6, pp. 514–524, Jun. 2008, doi: 10.1016/S1474-4422(08)70108-X.

[22] B. PN, F. D, and A. H. W, "The descriptive epidemiology of epilepsy-a review," *Epilepsy Res.*, vol. 85, no. 1, pp. 31–45, Jul. 2009, doi: 10.1016/J.EPLEPSYRES.2009.03.003.

[23] C. Z, B. MJ, L. D, and K. P, "Treatment Outcomes in Patients With Newly Diagnosed Epilepsy Treated With Established and New Antiepileptic Drugs: A 30-Year Longitudinal Cohort Study," *JAMA Neurol.*, vol. 75, no. 3, pp. 279–286, Mar. 2018, doi: 10.1001/JAMANEUROL.2017.3949.

[24] N. Zangiabadi, L. D. Ladino, F. Sina, J. P. Orozco-Hernández, A. Carter, and J. F. Téllez-Zenteno, "Deep Brain Stimulation and Drug-Resistant Epilepsy: A Review of the Literature," *Front. Neurol.*, vol. 10, no. JUN, p. 601, 2019, doi:

10.3389/FNEUR.2019.00601.

[25] A. Gabara, R. Yousri, D. Hamdy, M. H. Zakhari, and H. Mostafa, “Patient Specific Epileptic Seizures Prediction based on Support Vector Machine,” *Proc. Int. Conf. Microelectron. ICM*, vol. 2020-December, Dec. 2020, doi: 10.1109/ICM50269.2020.9331776.

[26] H.-T. Shiao *et al.*, “SVM-Based System for Prediction of Epileptic Seizures from iEEG Signal,” *IEEE Trans. Biomed. Eng.*, vol. 64, no. 5, p. 1011, May 2017, doi: 10.1109/TBME.2016.2586475.

[27] S. Saminu *et al.*, “A Recent Investigation on Detection and Classification of Epileptic Seizure Techniques Using EEG Signal,” *Brain Sci. 2021, Vol. 11, Page 668*, vol. 11, no. 5, p. 668, May 2021, doi: 10.3390/BRAINSCI11050668.

[28] U. Orhan, M. Hekim, and M. Ozer, “EEG signals classification using the K-means clustering and a multilayer perceptron neural network model,” *Expert Syst. Appl.*, vol. 38, no. 10, pp. 13475–13481, Sep. 2011, doi: 10.1016/J.ESWA.2011.04.149.

[29] K. Roy, A. Jaiswal, and P. Panda, “Towards spike-based machine intelligence with neuromorphic computing,” *Nature*, vol. 575, no. 7784, pp. 607–617, Nov. 2019, doi: 10.1038/s41586-019-1677-2.

[30] E. Pippa *et al.*, “Improving classification of epileptic and non-epileptic EEG events by feature selection,” *Neurocomputing*, vol. 171, pp. 576–585, Jan. 2016, doi:

10.1016/J.NEUCOM.2015.06.071.

[31] K. D. Tzimourta *et al.*, “A robust methodology for classification of epileptic seizures in EEG signals,” *Heal. Technol. 2018 92*, vol. 9, no. 2, pp. 135–142, Sep. 2018, doi: 10.1007/S12553-018-0265-Z.

[32] K. Saab, J. Dunnmon, C. Ré, D. Rubin, and C. Lee-Messer, “Weak supervision as an efficient approach for automated seizure detection in electroencephalography,” *npj Digit. Med. 2020 31*, vol. 3, no. 1, pp. 1–12, Apr. 2020, doi: 10.1038/s41746-020-0264-0.

[33] N. K. Kasabov, “NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data,” *Neural Networks*, vol. 52, pp. 62–76, Apr. 2014, doi: 10.1016/j.neunet.2014.01.006.

[34] T. Van Pottelbergh, G. Drion, and R. Sepulchre, “Robust modulation of integrate-and-fire models,” *Neural Comput.*, vol. 30, no. 4, pp. 987–1011, Sep. 2017, doi: 10.1162/neco_a_01065.

[35] M. Horowitz, “1.1 Computing’s energy problem (and what we can do about it),” *Dig. Tech. Pap. - IEEE Int. Solid-State Circuits Conf.*, vol. 57, pp. 10–14, 2014, doi: 10.1109/ISSCC.2014.6757323.

[36] S. Ghosh-Dastidar and H. Adeli, “Spiking neural networks,” *Int. J. Neural Syst.*, vol. 19, no. 4, pp. 295–308, Nov. 2009, doi: 10.1142/S0129065709002002.

[37] P. U. Diehl and M. Cook, “Unsupervised learning of digit recognition using spike-timing-dependent plasticity,” *Front. Comput. Neurosci.*, vol. 9, no. AUGUST, p. 99, Aug. 2015, doi: 10.3389/fncom.2015.00099.

[38] S. B. Shrestha and Q. Song, “Robustness to training disturbances in spikeprop learning,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 7, pp. 3126–3139, Jul. 2018, doi: 10.1109/TNNLS.2017.2713125.

[39] L. R. Iyer and Y. Chua, “Classifying Neuromorphic Datasets with Tempotron and Spike Timing Dependent Plasticity,” Jul. 2020, doi: 10.1109/IJCNN48605.2020.9207474.

[40] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, “Deep learning in spiking neural networks,” *Neural Networks*, vol. 111. Elsevier Ltd, pp. 47–63, Mar. 01, 2019, doi: 10.1016/j.neunet.2018.12.002.

[41] N. J. Stevenson, K. Tapani, L. Lauronen, and S. Vanhatalo, “A dataset of neonatal EEG recordings with seizure annotations,” *Sci. Data 2019 61*, vol. 6, no. 1, pp. 1–8, Mar. 2019, doi: 10.1038/sdata.2019.39.

[42] E. Bou Assi, D. K. Nguyen, S. Rihana, and M. Sawan, “Towards accurate prediction of epileptic seizures: A review,” *Biomed. Signal Process. Control*, vol. 34, pp. 144–157, Apr. 2017, doi: 10.1016/J.BSPC.2017.02.001.

[43] R. G. Andrzejak, K. Schindler, and C. Rummel, “Nonrandomness, nonlinear

dependence, and nonstationarity of electroencephalographic recordings from epilepsy patients,” *Phys. Rev. E,* vol. 86, no. 4, p. 046206, Oct. 2012, doi: 10.1103/PhysRevE.86.046206.

[44] “American Epilepsy Society Seizure Prediction Challenge | Kaggle.” https://www.kaggle.com/c/seizure-prediction (accessed Aug. 09, 2021).

[45] A. RG, L. K, M. F, R. C, D. P, and E. CE, “Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: dependence on recording region and brain state,” *Phys. Rev. E. Stat. Nonlin. Soft Matter Phys.*, vol. 64, no. 6 Pt 1, p. 8, 2001, doi: 10.1103/PHYSREVE.64.061907.

[46] IhleMatthias *et al.*, “EPILEPSIAE - A European epilepsy database,” *Comput. Methods Programs Biomed.*, vol. 106, no. 3, pp. 127–138, Jun. 2012, doi: 10.1016/J.CMPB.2010.08.011.

[47] A. L. Goldberger *et al.*, “PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals.,” *Circulation*, vol. 101, no. 23, 2000, doi: 10.1161/01.cir.101.23.e215.

[48] P. Branco, L. Torgo, and R. P. Ribeiro, “A survey of predictive modeling on imbalanced domains,” *ACM Computing Surveys*, vol. 49, no. 2. Association for Computing Machinery, Aug. 01, 2016, doi: 10.1145/2907070.

[49] Y. Xu, J. Yang, S. Zhao, H. Wu, and M. Sawan, “An End-to-End Deep Learning

Approach for Epileptic Seizure Prediction," *Proc. - 2020 IEEE Int. Conf. Artif. Intell. Circuits Syst. AICAS 2020*, pp. 266–270, Aug. 2020, doi: 10.1109/AICAS48895.2020.9073988.

[50] I. Alkanhal, B. V. K. V. Kumar, and M. Savvides, "Automatic Seizure Detection via an Optimized Image-Based Deep Feature Learning," *Proc. - 17th IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2018*, pp. 536–540, Jan. 2019, doi: 10.1109/ICMLA.2018.00086.

[51] K. D. Tzimourta *et al.*, "Evaluation of window size in classification of epileptic short-term EEG signals using a Brain Computer Interface software," *Eng. Technol. Appl. Sci. Res.*, vol. 8, no. 4, pp. 3093–3097, Aug. 2018, doi: 10.48084/etasr.2031.

[52] A. S. Al-Fahoum and A. A. Al-Fraihat, "Methods of EEG Signal Features Extraction Using Linear Analysis in Frequency and Time-Frequency Domains," *ISRN Neurosci.*, vol. 2014, pp. 1–7, Feb. 2014, doi: 10.1155/2014/730218.

[53] H. Kassiri *et al.*, "Battery-less Tri-band-Radio Neuro-monitor and Responsive Neurostimulator for Diagnostics and Treatment of Neurological Disorders," *IEEE J. Solid-State Circuits*, vol. 51, no. 5, pp. 1274–1289, May 2016, doi: 10.1109/JSSC.2016.2528999.

[54] F. WJ and S. CA, "Spatial EEG patterns, non-linear dynamics and perception: the neo-Sherringtonian view," *Brain Res.*, vol. 357, no. 3, pp. 147–175, 1985, doi: 10.1016/0165-0173(85)90022-0.

[55] M. H. Myers, A. Padmanabha, G. Hossain, A. L. de J. Curry, and C. D. Blaha, "Seizure Prediction and Detection via Phase and Amplitude Lock Values," *Front. Hum. Neurosci.*, vol. 10, no. MAR2016, Mar. 2016, doi: 10.3389/FNHUM.2016.00080.

[56] D. Cho, B. Min, J. Kim, and B. Lee, "EEG-Based Prediction of Epileptic Seizures Using Phase Synchronization Elicited from Noise-Assisted Multivariate Empirical Mode Decomposition," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 8, pp. 1309–1318, Aug. 2017, doi: 10.1109/TNSRE.2016.2618937.

[57] P. E. M. Smith and S. J. Wallace, "Clinicians' guide to epilepsy," p. 276, 2001.

[58] G. M. Rojas, C. Alvarez, C. E. Montoya, M. de la Iglesia-Vayá, J. E. Cisternas, and M. Gálvez, "Study of Resting-State Functional Connectivity Networks Using EEG Electrodes Position As Seed," *Front. Neurosci.*, vol. 0, no. APR, p. 235, Apr. 2018, doi: 10.3389/FNINS.2018.00235.

[59] J. P. Snyder, "Map projections: A working manual," *Prof. Pap.*, 1987, doi: 10.3133/PP1395.

[60] J. Ren, M. Green, and X. Huang, "From traditional to deep learning: Fault diagnosis for autonomous vehicles," *Learn. Control*, pp. 205–219, 2021, doi: 10.1016/B978-0-12-822314-7.00013-4.

[61] S. Shajun Nisha and M. Nagoor Meeral, "Applications of deep learning in

biomedical engineering," *Handb. Deep Learn. Biomed. Eng.*, pp. 245–270, 2021, doi: 10.1016/B978-0-12-823014-5.00008-9.

[62] S. Han *et al.*, "Optimizing Filter Size in Convolutional Neural Networks for Facial Action Unit Recognition."

[63] L. Du *et al.*, "A Reconfigurable Streaming Deep Convolutional Neural Network Accelerator for Internet of Things," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 65, no. 1, pp. 198–208, Jan. 2018, doi: 10.1109/TCSI.2017.2735490.

[64] J. Yepez and S. B. Ko, "Stride 2 1-D, 2-D, and 3-D Winograd for Convolutional Neural Networks," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 28, no. 4, pp. 853–863, Apr. 2020, doi: 10.1109/TVLSI.2019.2961602.

[65] "Everything about Pooling layers and different types of Pooling." https://iq.opengenus.org/pooling-layers/ (accessed Aug. 20, 2021).

[66] M. Loukadakis, J. Cano, and M. O'Boyle, "Accelerating Deep Neural Networks on Low Power Heterogeneous Architectures," *undefined*, 2018.

[67] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, Sep. 2014, Accessed: Aug. 19, 2021. [Online]. Available: https://arxiv.org/abs/1409.1556v6.

[68] J. L. Leevy, T. M. Khoshgoftaar, R. A. Bauder, and N. Seliya, "A survey on

addressing high-class imbalance in big data," *J. Big Data 2018 51*, vol. 5, no. 1, pp. 1–30, Nov. 2018, doi: 10.1186/S40537-018-0151-6.

[69] C. Seiffert, T. M. Khoshgoftaar, and J. Van Hulse, "Hybrid sampling for imbalanced data," *2008 IEEE Int. Conf. Inf. Reuse Integr. IEEE IRI-2008*, pp. 202–207, 2008, doi: 10.1109/IRI.2008.4583030.

[70] A. Usha Ruby, P. Theerthagiri, I. Jeena Jacob, and Y. Vamsidhar, "Binary cross entropy with deep learning technique for image classification," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 4, pp. 5393–5397, Jul. 2020, doi: 10.30534/IJATCSE/2020/175942020.

[71] D. Rengasamy, M. Jafari, B. Rothwell, X. Chen, and G. P. Figueredo, "Deep Learning with Dynamically Weighted Loss Function for Sensor-Based Prognostics and Health Management," *Sensors (Basel).*, vol. 20, no. 3, Feb. 2020, doi: 10.3390/S20030723.

[72] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, Dec. 2014, Accessed: Aug. 22, 2021. [Online]. Available: https://arxiv.org/abs/1412.6980v9.

[73] "Effect of batch size on training dynamics | by Kevin Shen | Mini Distill | Medium." https://medium.com/mini-distill/effect-of-batch-size-on-training-dynamics-21c14f7a716e (accessed Aug. 22, 2021).

[74] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning requires rethinking generalization,” *Commun. ACM*, vol. 64, no. 3, pp. 107–115, Nov. 2016, Accessed: Aug. 23, 2021. [Online]. Available: https://arxiv.org/abs/1611.03530v2.

[75] B. AN, “A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input,” *Biol. Cybern.*, vol. 95, no. 1, pp. 1–19, Jul. 2006, doi: 10.1007/S00422-006-0068-6.

[76] “Welcome to SINABS’s documentation! — sinabs 0.2.1.dev64 documentation.” https://sinabs.ai/ (accessed Aug. 23, 2021).

[77] E. Satuvuori, M. Mulansky, A. Daffertshofer, and T. Kreuz, “Using spike train distances to identify the most discriminative neuronal subpopulation,” *J. Neurosci. Methods*, vol. 308, pp. 354–365, Oct. 2018, doi: 10.1016/J.JNEUMETH.2018.09.008.

[78] D. Sihn and S.-P. Kim, “A Spike Train Distance Robust to Firing Rate Changes Based on the Earth Mover’s Distance,” *Front. Comput. Neurosci.*, vol. 0, p. 82, Dec. 2019, doi: 10.3389/FNCOM.2019.00082.

[79] W. SB, S. ML, E. RG, and G. AJ, “Seizure detection: evaluation of the Reveal algorithm,” *Clin. Neurophysiol.*, vol. 115, no. 10, pp. 2280–2291, Oct. 2004, doi: 10.1016/J.CLINPH.2004.05.018.

[80] T. D. K. Thara, P. S. Prema, and F. Xiong, “Auto-detection of epileptic seizure events using deep neural network with different feature scaling techniques,” *Pattern Recognit. Lett.*, vol. 128, pp. 544–550, Dec. 2019, doi: 10.1016/J.PATREC.2019.10.029.

[81] R. Abiyev, M. Arslan, J. B. Idoko, B. Sekeroglu, and A. Ilhan, “Identification of Epileptic EEG Signals Using Convolutional Neural Networks,” *Appl. Sci. 2020, Vol. 10, Page 4089*, vol. 10, no. 12, p. 4089, Jun. 2020, doi: 10.3390/APP10124089.

[82] Y. Gao, B. Gao, Q. Chen, J. Liu, and Y. Zhang, “Deep Convolutional Neural Network-Based Epileptic Electroencephalogram (EEG) Signal Classification,” *Front. Neurol.*, vol. 0, p. 375, 2020, doi: 10.3389/FNEUR.2020.00375.

[83] F. Tian, J. Yang, S. Zhao, and M. Sawan, “A new neuromorphic computing approach for epileptic seizure prediction,” *Proc. - IEEE Int. Symp. Circuits Syst.*, vol. 2021-May, 2021, doi: 10.1109/ISCAS51556.2021.9401560.

[84] E. García-Martín, C. F. Rodrigues, G. Riley, and H. Grahn, “Estimation of energy consumption in machine learning,” *J. Parallel Distrib. Comput.*, vol. 134, pp. 75–88, Dec. 2019, doi: 10.1016/J.JPDC.2019.07.007.

[85] C. Szegedy *et al.*, “Going Deeper with Convolutions,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June-2015, pp. 1–9, Sep. 2014, Accessed: Aug. 29, 2021. [Online]. Available: https://arxiv.org/abs/1409.4842v1.

[86] S. K. Esser *et al.*, “Convolutional networks for fast, energy-efficient neuromorphic computing,” *Proc. Natl. Acad. Sci.*, vol. 113, no. 41, pp. 11441–11446, Oct. 2016, doi: 10.1073/PNAS.1604850113.

[87] M. Davies *et al.*, “Advancing Neuromorphic Computing with Loihi: A Survey of Results and Outlook,” *Proc. IEEE*, vol. 109, no. 5, pp. 911–934, May 2021, doi: 10.1109/JPROC.2021.3067593.

[88] R. Di Cecco, G. Lacey, J. Vasiljevic, P. Chow, G. Taylor, and S. Areibi, “Caffeinated FPGAs: FPGA framework for convolutional neural networks,” *Proc. 2016 Int. Conf. Field-Programmable Technol. FPT 2016*, pp. 265–268, May 2017, doi: 10.1109/FPT.2016.7929549.

[89] Y. Deng, “Deep learning on mobile devices: a review,” vol. 10993, p. 109930A, May 2019, doi: 10.1117/12.2518469.

[90] T. Guo, “Cloud-based or On-device: An Empirical Study of Mobile Deep Inference,” *Proc. - 2018 IEEE Int. Conf. Cloud Eng. IC2E 2018*, pp. 184–190, Jul. 2017, Accessed: Aug. 29, 2021. [Online]. Available: https://arxiv.org/abs/1707.04610v2.