IMPROVED DEEP LEARNING ARCHITECTURE WITH BATCH NORMALIZATION FOR EEG SIGNAL PROCESSING

Adenuar Purnomo[1), and Handayani Tjandrasa[2)
[1, 2) Department of Informatics, Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia 60111
e-mail: adenuar.19051@mhs.its.ac.id [1), handatj@its.ac.id[2)

ABSTRACT

Deep learning is commonly used to solve problems such as biomedical problems and many other problems. The most common architecture used to solve those problems is Convolutional Neural Network (CNN) architecture. However, CNN may be prone to overfitting, and the convergence may be slow. One of the methods to overcome the overfitting is batch normalization (BN). BN is commonly used after the convolutional layer. In this research, we proposed a further usage of BN in CNN architecture. BN is not only used after the convolutional layer but also used after the fully connected layer. The proposed architecture is tested to detect types of seizures based on EEG signals. The data used are several sessions of recording signals from many patients. Each recording session produces a recorded EEG signal. EEG signal in each session is first passed through a bandpass filter. Then 26 relevant channels are taken, cut every 2 seconds to be labeled the type of epileptic seizure. The truncated signal is concatenated with the truncated signal from other sessions, divided into two datasets, a large dataset, and a small dataset. Each dataset has four types of seizures. Each dataset is equalized using the undersampling technique. Each dataset is then divided into test and train data to be tested using the proposed architecture. The results show the proposed architecture achieves 46.54% accuracy for the large dataset and 93.33% accuracy for the small dataset. In future studies, the batch normalization parameter will be further investigated to reduce overfitting.

Keywords: *Batch Normalization, CNN, Deep Learning, EEG, Seizure.*

PENINGKATAN ARSITEKTUR *DEEP LEARNING* DENGAN *BATCH NORMALIZATION* UNTUK PEMROSESAN SINYAL EEG

Adenuar Purnomo[1), dan Handayani Tjandrasa[2)
[1, 2) Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia 60111
e-mail: adenuar.19051@mhs.its.ac.id [1), handatj@its.ac.id[2)

ABSTRAK

Deep learning biasanya digunakan untuk memecahkan masalah seperti masalah biomedis, dan banyak masalah lainnya. Arsitektur yang paling umum digunakan untuk menyelesaikan masalah tersebut adalah arsitektur Convolutional Neural Network (CNN). Namun, CNN mungkin cenderung mengalami overfitting, dan konvergensinya mungkin lambat. Salah satu cara untuk mengatasi overfitting tersebut adalah batch normalization (BN). BN biasanya digunakan setelah lapisan konvolusional. Dalam penelitian ini, kami mengusulkan penggunaan BN lebih lanjut dalam arsitektur CNN. BN tidak hanya digunakan setelah lapisan konvolusional tetapi juga digunakan setelah lapisan yang terhubung sepenuhnya. Arsitektur yang diusulkan diuji untuk mendeteksi jenis kejang berdasarkan sinyal EEG. Data yang digunakan adalah beberapa sesi pencatatan sinyal dari banyak pasien. Setiap sesi perekaman menghasilkan sinyal EEG yang direkam. Sinyal EEG di setiap sesi pertama-tama dilewatkan melalui filter bandpass. Kemudian diambil 26 saluran yang relevan, dipotong setiap 2 detik untuk diberi label jenis serangan epilepsi. Sinyal yang terpotong tersebut digabungkan dengan sinyal yang terpotong dari sesi lain, yang dibagi menjadi dua dataset, yaitu dataset besar dan dataset kecil. Setiap set data memiliki empat jenis kejang. Setiap dataset disamakan dengan menggunakan teknik undersampling. Setiap dataset kemudian dibagi menjadi data uji dan latih untuk diuji menggunakan arsitektur yang diusulkan. Hasil uji coba menunjukkan arsitektur yang diusulkan mencapai akurasi 46.54% untuk dataset besar dan akurasi 92.33% untuk dataset kecil. Dalam studi selanjutnya, parameter normalisasi batch akan diselidiki lebih lanjut untuk mengurangi overfitting.

Kata Kunci: *Batch Normalization, CNN, Deep Learning, EEG, Kejang*

.

I. INTRODUCTION

Over the past decade, deep learning, as a sub-field of machine learning, is commonly used to solve problems such as computer vision, natural language processing, biomedical problems, and many other problems. The most common Deep Learning architecture used is Convolutional Neural Network (CNN). Other common Deep Learning architectures such as VGGnet [1] and ResNet [2] have also been commonly used, empowered with the ability to capture sophisticated and hierarchical features of high-dimensional data.

The usage of deep learning in biosignal processing problems have been researched many times. Internet of Things-based learning optimized for seizure prediction was proposed using big EEG data [3]. Signal transforms use empirical mode decomposition and classification using CNN show 98.9% accuracy when classifying between focused and non-focused signals [4]. In the same study, 99.5% accuracy for classifying seizure and non-seizure records, 96.5% for classifying healthy, unfocused, and seizure records, and 95.7% for classifying healthy, focused, and seizure records. The CNN-based model showed the highest accuracy of 96.7% and 97.5% using the Freiburg and CHB-MIT databases [5].

Acharya et al. [6] predicted seizures using deep CNN and achieved 88.67% accuracy in a recent study. An image-based EEG study using CNN showed a true positive rate of 74.0% between seizures and non-convulsive seizure activity [6]. Seizure prediction using intracranial and scalp EEG signals achieved 81.4% sensitivity using the Freiburg Hospital intracranial EEG dataset. The sensitivity metric achieved 81.2% using the Children's Hospital-MIT-Scalp EEG dataset from Children's Hospital-MIT, and 75,0% sensitivity using the American Epilepsy Society seizure dataset [7].

Tjandrasa et al. classified the EEG signals using a combination of intrinsic mode functions and power spectrum feature extractor, which gave a maximum of 78.6% accuracy for five classes [8]. They also classified the EEG dataset of healthy participants and epilepsy patients using single channel independent component analysis, power spectrum, and linear discriminant analysis. The study obtained a maximum accuracy of 94% for three classes [9].

Many proposed architectures have many kinds of layers which each layer has a different function. For example, LeNet-5 architecture, known as CNN, has convolutional layers, pooling layers, fully connected layers, and the output layer [10][11]. Other than those layers in CNN architecture, there are many layers used to enhance the CNN architecture. One example is the Dropout layer, which overcame overfitting, one of the CNN weaknesses [12]. The other example is the Batch Normalization layer. Batch Normalization (BN) allows us to use much higher learning rates and be less careful about initialization. It also acts as a regularizer, in some cases eliminating the need for Dropout. Applied to a state-of-the-art image classification model, Batch Normalization achieves the same accuracy with 14 times fewer training steps. It beats the original model by a significant margin [13].

In this study, we proposed a further usage of BN in CNN architecture. BN is not only used after the convolutional layer but also used after the fully connected layer. The proposed architecture is tested to detect types of seizures based on EEG signals. The data used are several sessions of recording signals from many patients. Each recording session produces a recorded EEG signal. EEG signal in each session is first passed through a bandpass filter. Then 26 relevant channels are taken, then cut every 2 seconds to be labeled the type of epileptic seizure. The truncated signal is then concatenated with the truncated signal from other sessions, divided into two datasets, a large dataset, and a small dataset. Each dataset has four types of seizures. Each dataset is then equalized using the undersampling technique. Each dataset is then divided into test and train data to be tested using the proposed architecture.

The paper is organized as follows: Section II describes the theory used in this research. Section III explains the whole method used in this research. Section IV explains the discussion from the results and the comparison to other scenarios used. Finally, the conclusions and future work are presented in Section V

II. LITERATURE STUDY

EEG has been researched since before the 20th century. Many studies on EEG started from the P300 speller algorithm, detecting epileptic seizures, sleep monitoring, and much other research. Several EEG studies, such as detection of epileptic seizures [14], sleep monitoring [15], detecting strokes [16], also detecting fatigue while driving [17], are some examples of recent studies.

Several studies on detecting epileptic seizures, such as [18] and [6], also use deep learning to detect them. Meanwhile, research such as [19] and [20] use standard classifiers such as Support Vector Machine (SVM) and Artificial Neural Networks to detect epileptic seizures.

One of the well-known deep learning architectures is CNN architecture. CNN is a deep learning classifier that is widely used in research. Besides being used for signal data [18], CNN is also widely used for image data [21], both research on medical and non-medical data.

The BN^3 architecture is a deep learning architecture in addition to the CNN architecture. This architecture itself first appeared in the research of Liu et al. [12]. Several studies used this architecture as a classifier in their research, such as [22], [23], and [16]. The BN^3 architecture itself was used for the P300 speller algorithm and glioma and stroke detection [16], and other things.

A. Deep Learning

Deep Learning is a subfield of machine learning that deals with algorithms inspired by the brain's structure and the function called neural networks. The adjective "deep" in deep learning comes from the use of multiple layers in the network. The main reason for using deep learning is its empirical effectiveness compared to other approaches [24].

Deep-learning architectures such as deep neural networks, recurrent neural networks, and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, bioinformatics, and medical image analysis, where they have produced results comparable to and in some cases surpassing the human expert performance

B. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a feedforward network proven very successful for image analysis [25]. CNN consists of one or more convolutional layers and is then followed by one or more interconnected layers as in a standard multi-layer neural network. CNN architecture is designed to take advantage of the 2D structure of the input image (or other 2D input such as a signal image).

Since 2006, many methods have been developed to overcome the difficulties encountered in training deep CNNs. Most notably, Krizhevsky et al. proposed a classic CNN architecture. They showed significant improvements upon previous methods on the image classification task. The overall architecture of their method, i.e., AlexNet [26], is similar to LeNet-5 but with a deeper structure. With the success of AlexNet, many works have been proposed to improve its performance. Among them, four representative works are ZFNet [27], VGGNet [1], GoogleNet [28], and ResNet [2]. From the architectures' evolution, a typical trend is that the networks are getting deeper, e.g., ResNet, which won ILSVRC 2015, is about 20 times deeper than AlexNet and eight times deeper than VGGNet. The network can better approximate the target function with increased nonlinearity and get better feature representations by increasing depth. However, it also increases the network's complexity, making the network more difficult to optimize and easier to get overfitting. Along this way, various methods have been proposed to deal with these problems in various aspects.

C. Batch Normalization

Batch Normalization is a technique for training very deep neural networks that standardize each mini-batch layer's inputs. Batch Normalization has the effect of stabilizing the learning process and dramatically reducing the number of training epochs required to train deep networks.

Simply adding Batch Normalization to a network does not take full advantage of our method. To do so, we further changed the network and its training parameters, as follows:

Increase the learning rate. Each training iteration will be slower due to extra normalization calculations and additional parameters when carrying out the learning process. However, the overall training process went much faster.

Remove Dropout. Batch Normalization fulfills some of the same goals as Dropout. Remove Dropout speeds up training without increasing overfitting.

Reduce L2 weight regularization. Batch Normalization improves the accuracy of the held-out validation data.

Accelerate the learning rate decay. In training Inception, the learning rate was decayed exponentially. Because our network trains faster than Inception, we lower the learning rate six times faster.

Shuffle training examples more thoroughly. The within-shard shuffled training data prevents the same examples from always appearing in a mini-batch together. The training data led to about 1% improvements in the validation accuracy.

Reduce the photometric distortions. Because batch-normalized networks train faster and observe each training example fewer times, we let the trainer focus on more "real" images by distorting them less.

A batch normalization normalizes its inputs x_i by calculating the mean μ_B and variance $\sigma_B{}^2$ over a mini-batch and each input channel. Then, it calculates the normalized activations as

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{1}$$

Where ϵ improves numerical stability when the mini-batch variance is very small. To allow for the possibility that inputs with zero mean and unit variance are not optimal for the layer that follows the batch normalization layer, the batch normalization layer further shifts and scales the activations as

$$y_i = \gamma \hat{x}_i + \beta \tag{2}$$

The offset β and scale factor γ (Offset and Scale properties) are learnable parameters updated during network training.

III. METHODOLOGY

This research was completed through several main stages: preprocessing, splitting and balancing data, proposed architecture, training process, and testing process. The research methodology is shown in Fig. 1. The proposed architecture is tested with a dataset. This research data is data derived from the free dataset belonging to TUH

(Temple University Hospital). The dataset is named The TUH EEG Seizure Corpus version 1.5. This dataset is recorded using the International 10-20 Electrode System featuring Modified Combinatorial Nomenclature (MCN), shown in Fig. 2, with a majority sampling rate of 250 Hz. The set consisting of 343 sessions were seizure sessions. There are eight classes in this dataset, namely simple partial seizures (SP), complex partial seizures (CP), focal nonspecific seizures (FN), generalized non-specific seizures (GN), absence seizures (AB), tonic seizures (TN), tonic-clonic seizures (TC), and non-seizures (NS). Each recording session produces a recorded EEG signal. Fig. 3 describes the example of the recorded EEG signal. The x-axis describes the time, and the y-axis describes the signal's amplitude.

A. Preprocessing

The recorded EEG was first passed through a bandpass filter with a cut-off frequency of 0.5-44 Hz and a sample rate of 250 Hz to attenuate the noise. EEG signals in each session passed this filter. Then 26 relevant channels are taken, then cut every 2 seconds to be labeled the type of epileptic seizure. The relevant channels taken are the first 26 channels. Channel 27 and channels after that consisted mostly of blank signals, which can reduce the performance. The truncated signal is then concatenated with the truncated signal from other sessions. Details of the number of the concatenated truncated signals can be seen in Table I.

B. Splitting and balancing

The concatenated truncated signal is divided into two datasets, a large dataset, and a small dataset. The large dataset is a dataset that has an initial class with a sample of more than 10000 data, and the small dataset is a dataset consisting of the rest of the classes. Each dataset has four types of seizures. Each dataset then equalized using the undersampling technique. Each class in each dataset is already equalized, so each dataset's class bias should not happen. The big dataset consisted of NS, SP, CP, and GN types of seizures. The small dataset consisted of FN, AB, TN, and TC types of seizures. The number of data in the small dataset and big dataset after undersampling can be seen in Tables II and III. Each dataset is divided into test and train data to be tested using the proposed architecture with the configuration of 60% training data and 40% testing data.

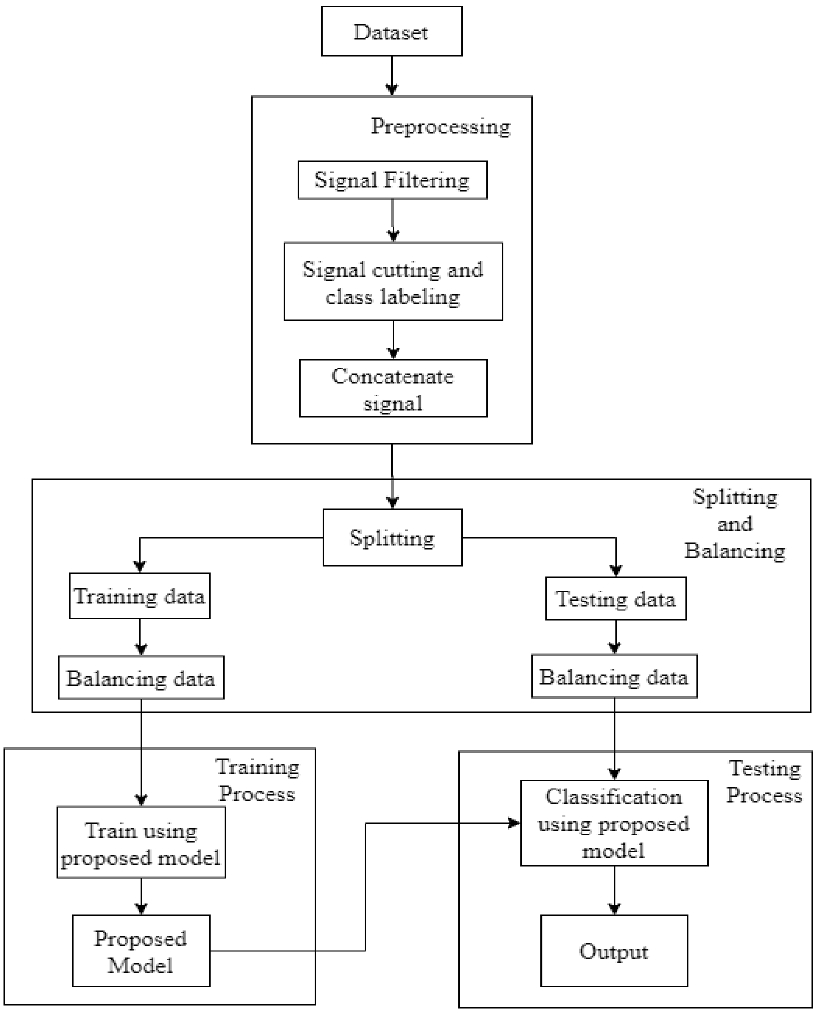

Fig. 1. Research methodology.

TABLE I
DETAILS OF THE NUMBER OF DATA IN DATASETS.

Class	Number of data in the dataset
NS	301171
SP	25784
CP	14215
FN	926
GN	17579
AB	284
TN	254
TC	1824

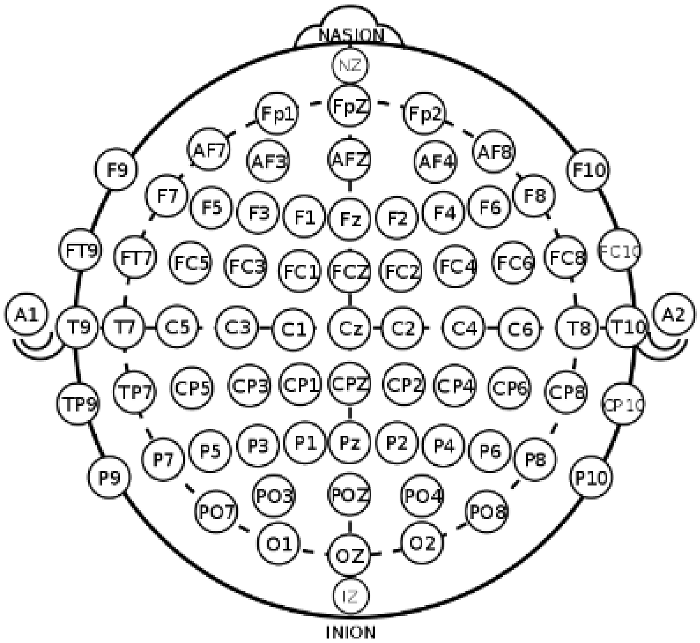

Fig. 2. The International 10–20 Electrode System Featuring Modified Combinatorial Nomenclature (MCN) [29].

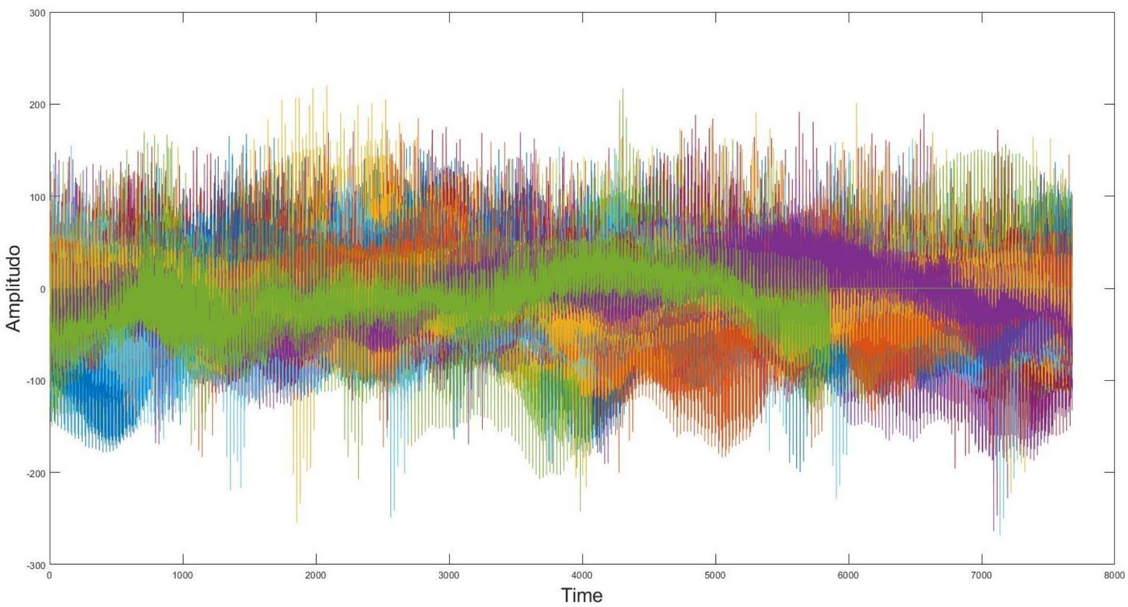

Fig. 3. Input signal from dataset.

C. Proposed architecture

The input data are the signals from the preprocessed data that are converted as if into a 2D-image. Since the input data are images, and CNN is a prominent architecture to process images, CNN-based architecture is used in this research.

TABLE II
NUMBER OF DATA IN THE BIG DATASET.

Class	Number of data before undersampling	Number of data after undersampling
NS	301171	14215
SP	25784	14215
CP	14215	14215
GN	17579	14215

TABLE III
NUMBER OF DATA IN THE SMALL DATASET.

Class	Number of data before undersampling	Number of data after undersampling
FN	926	254
AB	284	254
TN	254	254
TC	1824	254

The proposed architecture used in this research, as seen in Fig. 4, is divided into three sections, the extraction layer after the input layer, the training layer, and the classification layer before the output layer. The extraction layer consists of the Batch Normalization layer, Convolutional layer, ReLU layer, Batch Normalization layer, Max Pooling layer, another Convolutional layer, ReLu layer, Batch Normalization layer, and Max Pooling layer. The extraction layer will extract the features needed from the input layer, classified in the next layer. In this section, the weights that will be trained are at both the Convolutional layer.

The training layer consists of a Fully Connected Layer, Batch Normalization layer, Dropout layer, another Fully Connected Layer, Batch Normalization layer, and Dropout layer. The training layer will weight the features collected from the extraction layer. In this section, the weight that will be trained are at both the Fully Connected layer. The final section is that the classification layer consists of a Fully Connected Layer and Softmax layer. This section classifies the input images into the output classes. Configuration for the proposed architecture is described in the next section.

D. Training Process

The training process aims to create the model to be used in the testing process. The process to train the proposed architecture model is from input data to the extraction layer to the training layer, the classification layer, and the output classes. All the processes in training created a model to be tested in the testing process. The layer configuration of the proposed architecture is as follows: the input layer is 2D-matrix with 26×500 in size, the Convolutional Layer has a filter size of 4×4 with eight filters each layer, Max Pooling Layer has 2×2 pool size, Dropout Layer has a probability of 0.2, first fully connected layer has 32 output size, the second fully connected layer has 16 output size.

Meanwhile, this research's training options are using adam optimizer. With an initial learning rate of 10^{-3}, and for every 100 epoch, the learning rate becomes ten times smaller. The training model is finished when the loss is reached less than 0.1.

E. Testing Process

The trained model is tested, and then the performance is compared to other models in the scenario. The scenario used in the research can be seen in Table IV. The performance metric used to compare architecture is accuracy, precision, and recall. Calculation of accuracy, precision, and recall was defined using the confusion matrix. Fig. 5 Described the confusion matrix. Accuracy, precision, and recall can be calculated using Equations (3), (4), and (5), respectively.

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (5)$$

TABLE IV
COMPARISON OF ARCHITECTURE'S SCENARIO.

Architecture	Extraction Layer	Training Layer	Total Layer
Proposed Architecture	With Batch Normalization	With Batch Normalization	19
Architecture A	With Batch Normalization	Without Batch Normalization	17
Architecture B	Without Batch Normalization	With Batch Normalization	16
Architecture C	Without Batch Normalization	Without Batch Normalization	14

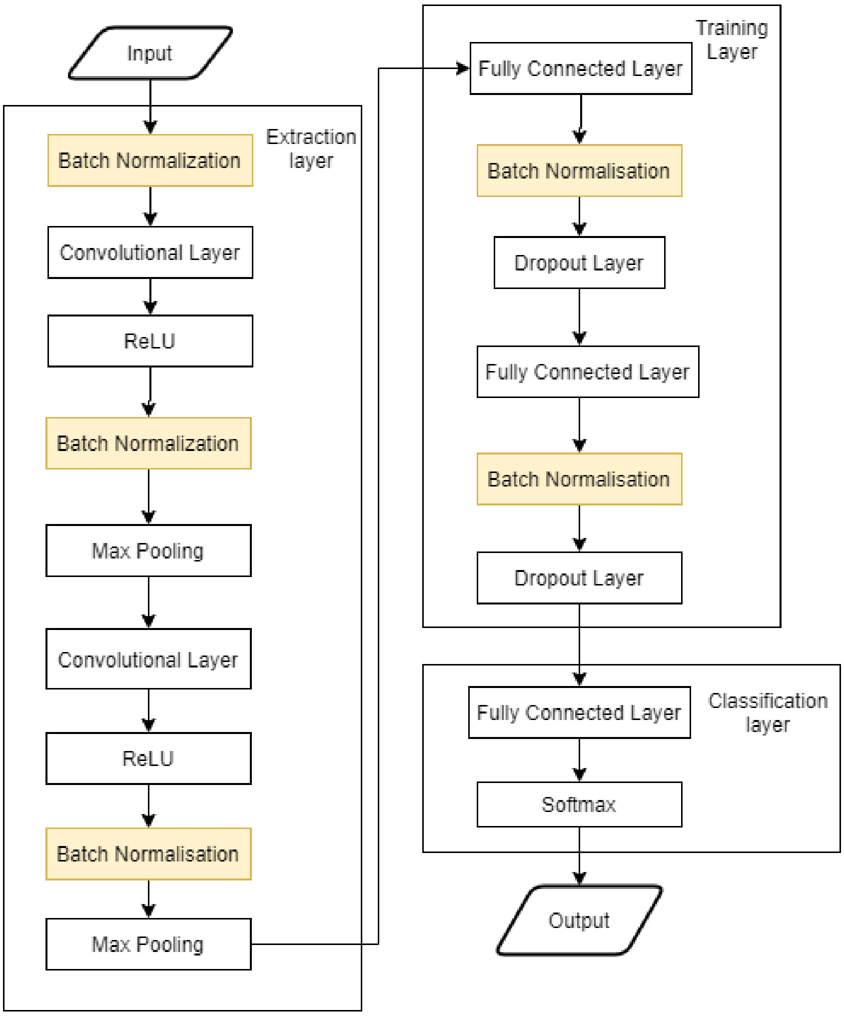

Fig. 4. Proposed architecture.

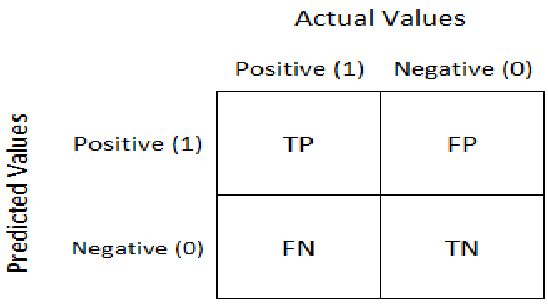

Fig. 5. Confusion matrix.

IV. RESULTS AND DISCUSSION

Fig. 6 is one of the examples of data after the preprocessing. The signal data is converted as if it is an image to be trained in the proposed architecture. The training dataset and testing dataset are trained in each architecture in this research scenario. So there are eight models in this research to be tested. The testing process gives the accuracy, precision, and recall of each class. The precision and recall of each class are then summed and divided so that the averaged precision and recall can be calculated. Table V shows that the proposed architecture excels in all the big and small datasets' research metrics. The proposed architecture's model gave 6.9% better accuracy than Architecture A's model, 4.9% better accuracy than Architecture B's model, and gave 3.9% better accuracy than Architecture C's model in the small dataset. For the big dataset, the accuracy metric proposed architecture's model gave 6.2% better accuracy than Architecture A's model, 6.2% better accuracy than Architecture B's model. The proposed architecture even has 19.1% better accuracy than Architecture C's model.

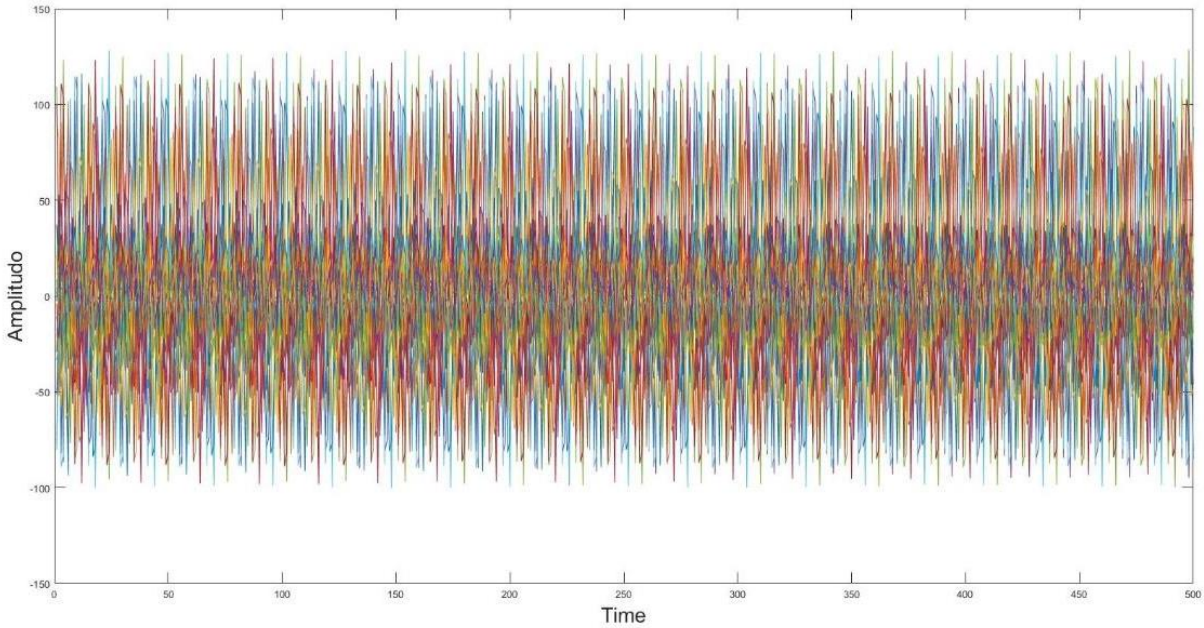

Fig. 6. Preprocessed data.

TABLE V
PERFORMANCE OF THE PROPOSED ARCHITECTURE COMPARED TO OTHER ARCHITECTURE.

Dataset	Architecture	Accuracy(%)	Averaged Precision(%)	Averaged Recall(%)
Small	Proposed Architecture	**92.327**	**92.325**	**92.625**
Small	Architecture A	85.396	85.4	85.025
Small	Architecture B	87.376	87.375	88.3
Small	Architecture C	88.366	88.35	88.65
Big	Proposed Architecture	**46.535**	**46.38**	**47.6**
Big	Architecture A	40.296	40.3	40.025
Big	Architecture B	44.926	44.95	46.05
Big	Architecture C	27.383	27.375	27.475

The proposed architecture's model has 6.9%, 4.9%, and 3.9% better than Architecture A, B, and C's model in the small dataset's precision metric. For the big dataset, the proposed architecture's model has 6%, 1.4%, and 19% better precision than Architecture A, B, and C's model. The same with the recall metric, in the small dataset, the proposed architecture's model gave 7.6%, 4.3%, and 3.9% better recall than Architecture A, B, and C's model. The proposed architecture's model has 7.5%, 1.5%, and 20.1% better recall in the big dataset in the big dataset than Architecture A, B, and C's model. Architecture C has the second-best performance in the small dataset, followed by Architecture B, and the worst is Architecture A. But for the big dataset, Architecture C has the worst performance, and followed by Architecture A, and then Architecture B,

From the results, we can see that the Batch Normalization layer usage in both the extraction and training layers can produce the highest performance. It produced better results than usage in one of the layers only and without the Batch Normalization layer. The Batch Normalization layer usage makes the training model less susceptible to overfitting, especially in the large dataset. In the small dataset, the Batch Normalization layer utilization does not significantly affect performance; in some cases, it can reduce performance, although it makes the training process faster.

All the architectures used in this research failed to have more than 50% accuracy, averaged precision, and averaged recall for the big dataset. The small dataset results contradicted it, with each metric having more than 85% for each architecture. One of the reasons is because the big dataset is from many people. So that from the same type of seizure, there are some kinds of differences in the signal recorded. The differences are not that big for a small dataset. Because the undersampling technique used in this research is randomly picked, so the undersampling technique also contributed to the big dataset's low matric results. The other reason is that the parameter used in the architecture in this research is not fined tuned. The parameter also contributed to why the model is too overfitting to the big dataset's train data.

V. CONCLUSION

We can conclude from this research that the proposed architecture has better performance than the other architecture tested. Although it failed to reach more than 80% performance in the big dataset, the proposed architecture improved the existing deep learning architecture. In future research, each layer's parameter should be tuned, especially the batch normalization layer. Also, use the better undersampling method that reduces the randomness that causes the difference between signals from the same type.

REFERENCES

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016.

[3] M. Hosseini, D. Pompili, K. Elisevich, and H. Soltanian-Zadeh, "Optimized Deep Learning for EEG Big Data and Seizure Prediction BCI via Internet of Things," *IEEE Trans. Big Data*, vol. 3, no. 4, pp. 392–404, Dec. 2017.

[4] A. H. Ansari, P. J. Cherian, and H. H. Sciences, "Neonatal Seizure Detection Using Deep Convolutional Neural Networks," in *Proc. IEEE 27th International Workshop on Machine Learning for Signal Processing*, 2017.

[5] R. San-segundo, M. Gil-martín, L. F. D. Haro-enríquez, and J. Manuel, "Classification of epileptic EEG recordings using signal transforms and," *Comput. Biol. Med.*, vol. 109, pp. 148–158, 2019.

[6] M. Zhou *et al.*, "Epileptic Seizure Detection Based on EEG Signals and CNN," *Front Neuroinform*, vol. 12, no. 95, pp. 1–14, 2018.

[7] A. Emami, N. Kunii, T. Matsuo, T. Shinozaki, and K. Kawai, "NeuroImage: Clinical Seizure detection by convolutional neural network-based analysis of scalp electroencephalography plot images," *NeuroImage Clin.*, vol. 22, no. May 2018, p. 101684, 2019.

[8] H. Tjandrasa, S. Djanali, and F. X. Arunanto, "Feature extraction using combination of intrinsic mode functions and power spectrum for EEG signal classification," *Proc. - 2016 9th Int. Congr. Image Signal Process. Biomed. Eng. Informatics, CISP-BMEI 2016*, pp. 1498–1502, 2017.

[9] H. Tjandrasa and S. Djanali, "Classification of EEG signals using single channel independent component analysis, power spectrum, and linear discriminant analysis," in *Lecture Notes in Electrical Engineering*, 2016, vol. 387, pp. 259–268.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," in *IEEE Proceedings*, 1998.

[11] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object recognition with gradient-based learning," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1681, pp. 319–345, 1999.

[12] M. Liu, W. Wu, Z. Gu, Z. Yu, F. F. Qi, and Y. Li, "Deep learning based on Batch Normalization for P300 signal detection," *Neurocomputing*, vol. 275, pp. 288–297, 2018.

[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn. ICML 2015*, vol. 1, pp. 448–456, 2015.

[14] N. D. Truong, A. D. Nguyen, L. Kuhlmann, M. R. Bonyadi, and J. Yang, "A Generalised Seizure Prediction with Convolutional Neural Networks for Intracranial and Scalp Electroencephalogram Data Analysis," *arXiv*, pp. 1–8, 2017.

[15] L. Yin, C. Zhang, and Z. Cui, "Experimental research on real-time acquisition and monitoring of wearable EEG based on TGAM module," *Comput. Commun.*, vol. 151, pp. 76–85, 2020.

[16] J. Amin, M. Sharif, M. A. Anjum, M. Raza, and S. A. C. Bukhari, "Convolutional neural network with batch normalization for glioma and stroke lesion detection using MRI," *Cogn. Syst. Res.*, vol. 59, pp. 304–311, 2020.

[17] D. Jing, D. Liu, S. Zhang, and Z. Guo, "Fatigue driving detection method based on EEG analysis in low-voltage and hypoxia plateau environment," *Int. J. Transp. Sci. Technol.*, vol. 9, no. 4, 2020.

[18] S. Raghu, N. Sriraam, Y. Temel, S. V. Rao, and P. L. Kubben, "EEG based multi-class seizure type classification using convolutional neural network and transfer learning," *Neural Networks*, vol. 124, pp. 202–212, 2020.

[19] S. Raghu, N. Sriraam, and P. L. Kubben, "Automated detection of epileptic seizures using successive decomposition index and support vector machine classifier in long-term EEG," *Neural Comput. Appl.*, vol. 1, 2019.

[20] V. Srinivasan, C. Eswaran, and N. Sriraam, "Approximate Entropy-Based Epileptic EEG Detection Using Artificial Neural Networks," *IEEE Trans. Inf. Technol. Biomed.*, vol. 11, no. 3, pp. 288–295, May 2007.

[21] D. Lin, F. Lin, Y. Lv, F. Cai, and D. Cao, "Chinese Character CAPTCHA Recognition and performance estimation via deep neural network," *Neurocomputing*, vol. 288, pp. 11–19, 2018.

[22] D. Macêdo, C. Zanchettin, A. L. I. Oliveira, and T. Ludermir, "Enhancing batch normalized convolutional networks using displaced rectifier linear units: A systematic comparative study," *Expert Syst. Appl.*, vol. 124, pp. 271–281, 2019.

[23] J. Wang, S. Li, Z. An, X. Jiang, W. Qian, and S. Ji, "Batch-normalized deep neural networks for achieving fast intelligent fault diagnosis of machines," *Neurocomputing*, vol. 329, pp. 53–65, 2019.

[24] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, "Chapter 10 - Deep learning," in *Machine Learning: A Constrained-based Approach*, Morgan Kaufmann, 2017.

[25] Q. Zhang, M. Zhang, T. Chen, Z. Sun, Y. Ma, and B. Yu, "Recent advances in convolutional neural network acceleration," *Neurocomputing*, vol. 323, pp. 37–51, 2019.

[26] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[27] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," *CoRR*, vol. abs/1311.2, 2013.

[28] C. Szegedy *et al.*, "Going deeper with convolutions," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 1–9, 2015.

[29] H. Tjandrasa, *Klasifikasi Sinyal EEG dan Aplikasinya*. Surabaya: ITS Press, 2017.