ISSN 2347 - 3983
Volume 8. No. 9, September 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter34892020.pdf
https://doi.org/10.30534/ijeter/2020/34892020

Numeric Sign Language Recognition Using Deep Learning

Sang-Hong Lee
Dept. of Computer Science & Engineering, Anyang University, Republic of Korea, shleedosa@gmail.com

ABSTRACT

There are more disabled people living in South Korea than expected, living in a state-of-the-art era that goes beyond the 4G era to the 5G era. In particular, the communication problems experienced by hearing and speech-impaired people are causing discomfort in out-of-home activities and limiting the scope of their activities. The government has decided to implement a neural network that recognizes even some of these signs with the aim of relieving some of the inconveniences experienced by disabled people. We believe that AR (Augmented Reality) and VR (Virtual Reality) can be combined to sufficiently apply them to real life to relieve the discomfort of many hearing and speech impaired people. In this paper, it was basically constructed based on CNN (Convolution Neural Network) and aimed to categorize 10,000 input nodes into 10 classes at the end through synthetic multiplication operations. As the number of images secured is small, the K-Way verification method was used to learn and verify them 300 images were grouped together to verify 5-way.

Key words : Disabled people, Speech-impaired, Numeric sign language, Deep learning.

1. INTRODUCTION

With advances in technology, a number of products related to artificial intelligence are being released. Not only virtual assistant programs like Google Assistant, Apple Siri and Samsung Bixby, but artificial intelligence speakers such as SKT Nugu, KT Gigajini and Kakao Mini, which are also embedded in our daily lives. These products are based on people who are basically able to communicate, so difficulties exist for people with disabilities who are unable to communicate. Especially for deaf people, communication using sign language is inconvenient for ordinary people who do not know sign language [1-6]. Therefore, hearing impaired people are at a great disadvantage [7-8]. According to the survey on the handicapped by the Ministry of Health and Welfare, hearing and speech impaired people have difficulties with their activities outside the home due to difficulties in communicating because of the inconvenience.

2. DATA AND PREPROCESSING

Images were obtained using a laptop webcam. After removing the background, the skin color was detected through YCrCb through Python 3.7.3 and OpenCV, and the remaining skin color was masked black [1]. In order to allow a certain amount of noise, but to eliminate any noise that was not needed, the filter was strongly filtered and the image was treated in black and white tones of (100, 100) the same size in Fig. 1. The final 19050 images were learned by amplifying the data into 150 images with a rotation angle of 20 degrees, a width travel range of 0.2 and a height movement range of 0.2.

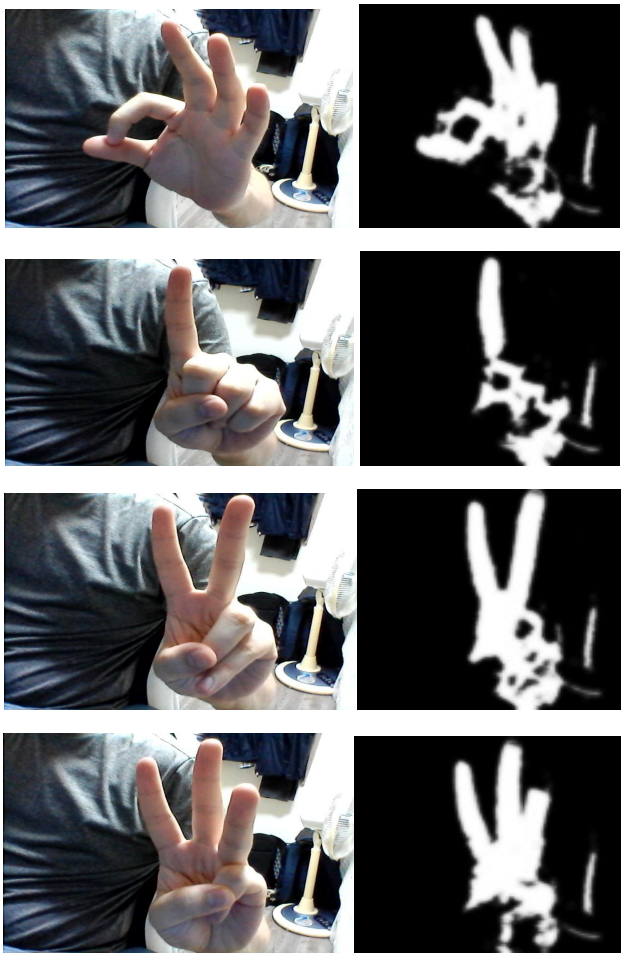

Figure 1: Proposed model

3. CONVOLUTION NEURAL NETWORK (CNN)

CNN achieves two goals of drastically reducing the complexity of models and extracting good features by applying the commonly used convolution operation in video processing or signal processing in Fig. 2 [9-14]. CNN classifies nodes into multiple feature maps due to multiple feature maps (Kernel) extraction, and reduces the size of feature maps through a beam setting in the pooling layer operation.

The VGGNet (VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE – SCALE IMAGE RECOGNITION) model was developed by the Oxford University research team VGG, and was the runner-up model at the 2014 ILSVRC (Imagine Image Recognition Contest). Fig. 3 shows an example of the VGGNet model. As the depth of the model deepened to 11th, 13th, 16th, and 19th layers, the classification error decreased and performance improved.

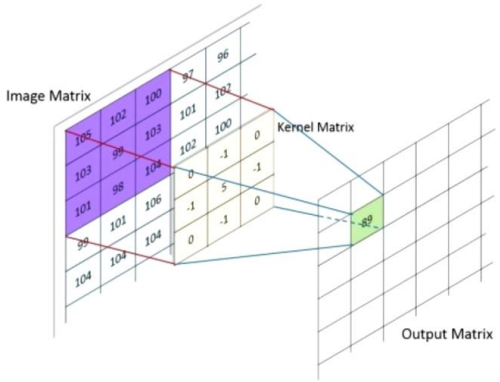

Figure 2: CNN computational method [9]

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 3: VGGNet structure

4. EXPERIMENTAL RESULTS

Table 1 describes the system specifications and software used for numeric sign recognition. In this paper, it was basically constructed based on CNN and aimed to categorize 10,000 input nodes into 10 classes at the end through synthetic multiplication operations. As the number of images secured is small, the K-Way verification method was used to learn and verify them 300 images were grouped together to verify 5-way. The initial artificial neural network model was viewed with FC (Fully Connected) connection at all levels. The learning model is the 10000-512-10 model, which shows the learning and verification accuracy shown in Fig. 4.

Table 1: Explanations about the Measured Data

	Specification
Hardware	CPU : Intel core I7-8550U RAM : 8GB VGA(GPU) : NVIDIA GeForce Mx150 2GB
Operating System	Windows 10 Education x64
Language	Python 3.7.3
IDE	Jupyter Notebook 6.0.0
Tools	Anaconda 1.7.2 NVIDIA CUDA 10.0.130 NVIDIA CUDA tool kit 10 tensorflow-gpu 1.15.0

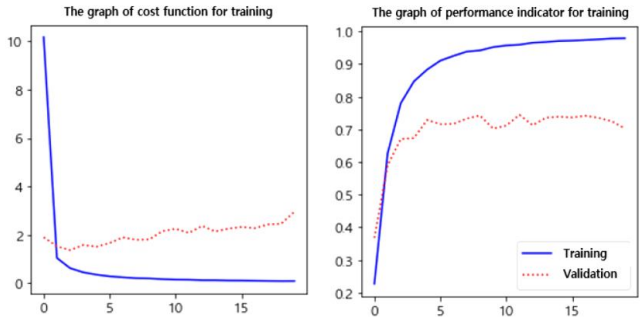

Figure 4: Learning and validation accuracy

FC models did not increase verification accuracy further with further learning. Therefore, as shown in Fig. 5, a large layer was placed to print out as many characteristic maps as possible and to learn them. The verification accuracy showed the performance as shown in Fig. 6, although the 64 batches were studied 50 times as models as shown in Fig. 5. However, the test accuracy was 60 to 80 percent. Rather, it was found that the test accuracy was significantly reduced compared to the verification accuracy due to over time. We looked at the problem of the model in a way that solved it. If the existing VGG-16 was modified, the neural network was further

developed into a model as shown in Fig. 7, and the number of lessons was reduced.

Layer (type)	Output Shape	Param #
conv2d_28 (Conv2D)	(None, 98, 98, 32)	320
conv2d_29 (Conv2D)	(None, 96, 96, 32)	9248
max_pooling2d_13 (MaxPooling	(None, 48, 48, 32)	0
dropout_13 (Dropout)	(None, 48, 48, 32)	0
conv2d_30 (Conv2D)	(None, 46, 46, 64)	18496
conv2d_31 (Conv2D)	(None, 44, 44, 64)	36928
max_pooling2d_14 (MaxPooling	(None, 22, 22, 64)	0
dropout_14 (Dropout)	(None, 22, 22, 64)	0
conv2d_32 (Conv2D)	(None, 20, 20, 128)	73856
conv2d_33 (Conv2D)	(None, 18, 18, 128)	147584
max_pooling2d_15 (MaxPooling	(None, 9, 9, 128)	0
dropout_15 (Dropout)	(None, 9, 9, 128)	0
conv2d_34 (Conv2D)	(None, 7, 7, 256)	295168
conv2d_35 (Conv2D)	(None, 5, 5, 256)	590080
max_pooling2d_16 (MaxPooling	(None, 2, 2, 256)	0
dropout_16 (Dropout)	(None, 2, 2, 256)	0
flatten_1 (Flatten)	(None, 1024)	0
dense_1 (Dense)	(None, 256)	262400
dropout_17 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 10)	2570

Total params: 1,436,650
Trainable params: 1,436,650
Non-trainable params: 0

Figure 5: Initial CNN learning model

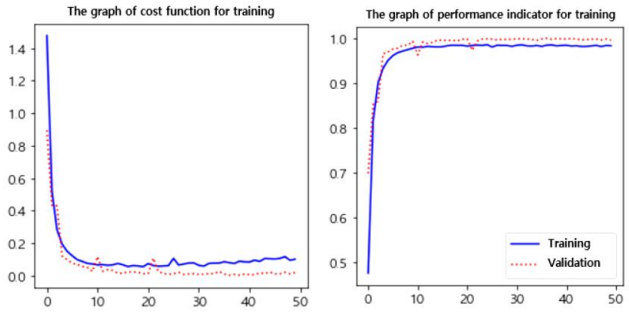

Figure 6: Initial CNN learning model results

However, the model also did not increase its learning accuracy between 0.09 and 0.1 in the first three iterations during the course of study, and the verification accuracy stopped at 0.1. Since neither of the previous models on the 10th layer nor the 16th layer showed good performance here, we tried to reduce the layer to the reverse this time. The final artificial neural network model was based on one output layer in three composite multi-layer Fig. 8. After each composite product, the beam was downscaled with a width of 2 as the maximum pooling layer, and after the pooling layer, a dropout was placed to solve the problem of overfit.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 98, 98, 32)	320
conv2d_2 (Conv2D)	(None, 96, 96, 32)	9248
max_pooling2d_1 (MaxPooling2	(None, 48, 48, 32)	0
dropout_1 (Dropout)	(None, 48, 48, 32)	0
conv2d_3 (Conv2D)	(None, 46, 46, 64)	18496
conv2d_4 (Conv2D)	(None, 44, 44, 64)	36928
max_pooling2d_2 (MaxPooling2	(None, 22, 22, 64)	0
dropout_2 (Dropout)	(None, 22, 22, 64)	0
conv2d_5 (Conv2D)	(None, 20, 20, 128)	73856
conv2d_6 (Conv2D)	(None, 18, 18, 128)	147584
conv2d_7 (Conv2D)	(None, 16, 16, 128)	147584
max_pooling2d_3 (MaxPooling2	(None, 8, 8, 128)	0
dropout_3 (Dropout)	(None, 8, 8, 128)	0
conv2d_8 (Conv2D)	(None, 6, 6, 256)	295168
conv2d_9 (Conv2D)	(None, 4, 4, 256)	590080
conv2d_10 (Conv2D)	(None, 2, 2, 256)	590080
max_pooling2d_4 (MaxPooling2	(None, 1, 1, 256)	0
dropout_4 (Dropout)	(None, 1, 1, 256)	0
flatten_1 (Flatten)	(None, 256)	0
dense_1 (Dense)	(None, 256)	65792
dropout_5 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 64)	16448
dropout_6 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 32)	2080
dropout_7 (Dropout)	(None, 32)	0
dense_4 (Dense)	(None, 10)	330

Total params: 1,993,994
Trainable params: 1,993,994
Non-trainable params: 0

Figure 7: Modified VGG-16

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 98, 98, 128)	1280
max_pooling2d_1 (MaxPooling2	(None, 49, 49, 128)	0
dropout_1 (Dropout)	(None, 49, 49, 128)	0
conv2d_2 (Conv2D)	(None, 47, 47, 64)	73792
max_pooling2d_2 (MaxPooling2	(None, 23, 23, 64)	0
dropout_2 (Dropout)	(None, 23, 23, 64)	0
conv2d_3 (Conv2D)	(None, 21, 21, 32)	18464
max_pooling2d_3 (MaxPooling2	(None, 10, 10, 32)	0
dropout_3 (Dropout)	(None, 10, 10, 32)	0
flatten_1 (Flatten)	(None, 3200)	0
dropout_4 (Dropout)	(None, 3200)	0
dense_1 (Dense)	(None, 10)	32010

Total params: 125,546
Trainable params: 125,546
Non-trainable params: 0

Figure 8: Final model configuration

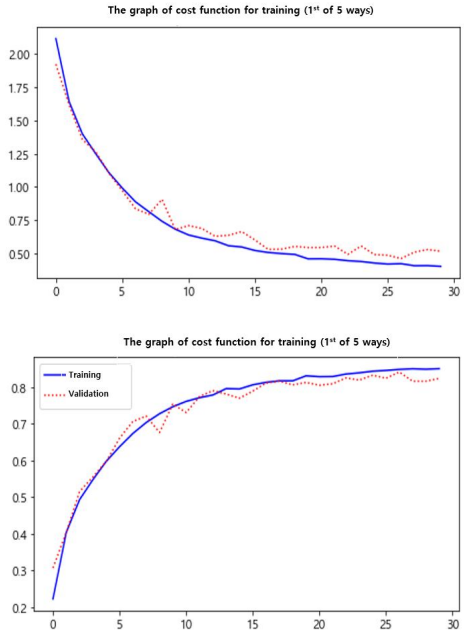

Figure 9: Loss and accuracy of 1st way

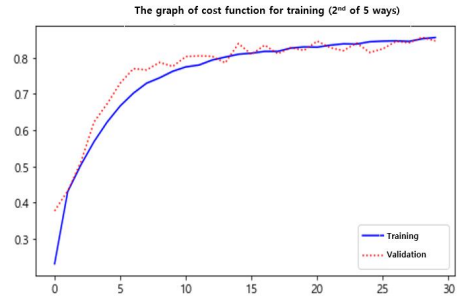

Figure 10: Loss and accuracy of 2nd way

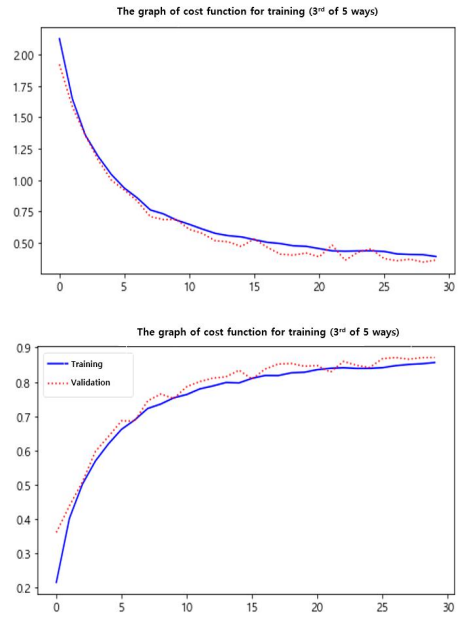

Figure 11: Loss and accuracy of 3rd way

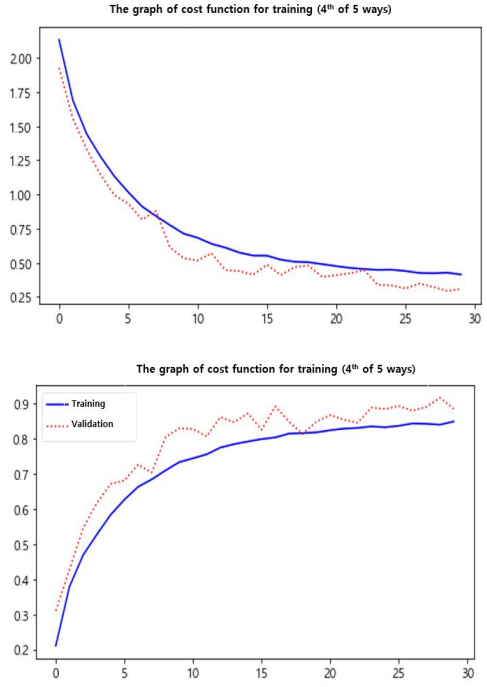

Figure 12: Loss and accuracy of 4th way

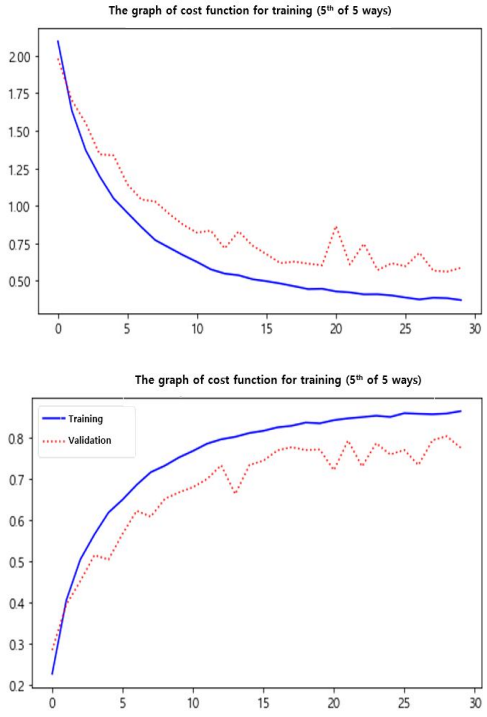

Figure 13: Loss and accuracy of 5th way

We performed 30 learning epochs, and 64 learning pieces were put together. Graphs showing step-by-step loss and accuracy are found to have been learned in a gentle curve, as shown in Fig. 9, Fig. 10, Fig. 11, Fig. 12, and Fig. 13, and some parts that are suspected of being over-suitable. The accuracy of each phase of learning was 85%, 86%, 86%, 84% and 86%, while the accuracy of validation was 82%, 85%, 87%, 89% and 78%. However, in order to check how much of

the suspected overfit occurred and how much of the images worked with a mixture of real-life noise, images that were altered after skin color masking were selected as test data without background removal from additional images that were not used for learning and verification. In each step, the weight value of the last four epochs was obtained and the classification was attempted.

5. CONCLUSION

It is analyzed that the amount of training data set and verification data set and test data set used in this paper is very small compared with other artificial intelligence learning. In artificial intelligence learning, the amount of learning data can make a big difference in terms of accuracy or reliability of the artificial neural network, so it is necessary to secure a better quality data set. In addition, this paper is designed to recognize only numbers through images, but in order to apply them to real life, they should be based on the recognition processing as images and more recognition.

Further research on the composition of better artificial neural network models and hand shape and motion extraction using object detection should be carried out in order to enhance accuracy of sign language video recognition and translation in the future, in order to secure more quality training data and apply them to real life.

In real life, many deaf people suffer from a lot of discomfort with communication. The recognition and translation of numerical sign language implemented in this paper shows higher performance than the performance in learning and verification accuracy of FC-based artificial neural networks, which were previously the background of the study, in the accuracy that can be obtained when the CNN-based artificial neural network is applied to real life. However, if we supplement the recognition of only numbers, which are the limitations of the neural network we have learned in this paper, with the recognition of other letters and words sentences, we believe that AR and VR can be combined to sufficiently apply them to real life to relieve the discomfort of many hearing and speech impaired people.

ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2019R1F1A1055423).
Corresponding Author: Sang-Hong Lee
(shleedosa@gmail.com)

REFERENCES

1. H. H. Kim, J. R. Cho, **Sign language image recognition system using artificial neural network**, *Journal of The Korea Society of Computer and Information*, Vol.24, pp.193-200, 2019.
2. R. G. Bosworth, C. E. Wright, K. R. Dobkins, **Analysis of the visual spatiotemporal properties of American sign language**, *Vision Research*, Vol.164, pp.34-43, 2019.
3. G. Hodge, L. N. Ferrara, B. D. Anible, **The semiotic diversity of doing reference in a deaf signed language**, *Journal of Pragmatics*, Vol.143, pp.33-53, 2019.
4. K. Suri, R. Gupta, **Continuous sign language recognition from wearable IMUs using deep capsule networks and game theory**, *Computers & Electrical Engineering*, Vol.78, pp.493-503, 2019.
5. H. J. Kim, S. J. Park, and S. K. Lee, **Sign language recognition using motion history volume and hybrid neural networks**, *International Journal of Machine Learning and Computing*, Vol.2, pp.750-753, 2012.
6. H. Takimoto, J. Lee, and A. Kanagawa, **A robust gesture recognition using depth data**, *International Journal of Machine Learning and Computing*, Vol.3, pp.245-249, 2013.
7. J. C. Lv and S. J. Xiao, **Real-time 3D motion recognition of skeleton animation data stream**, *International Journal of Machine Learning and Computing*, Vol.3, pp.430-434, 2013.
8. Amal Fouad, Hossam M. Moftah, Hesham A. Hefny, **MRI Brain Cancer Diagnosis Approach Using Gabor Filter and Support Vector Machine**, *International Journal of Emerging Trends in Engineering Research*, Vol.7, pp.907-914, 2019.
9. Brunch. [Online]. Available: https://brunch.co.kr/@linecard/323.
10. T. Li, Z. T. Zhang, H. Chen, **Predicting the combustion state of rotary kilns using a convolutional recurrent neural network**, *Journal of Process Control*, Vol.84, pp.207-214, 2019.
11. B. Wang, Y. G. Lei, T. Yan, N. P. Li, L. Guo, **Recurrent convolutional neural network: A new framework for remaining useful life prediction of machinery**, *Neurocomputing*, Vol.379, pp.117-129, 2020.
12. E. Cetinic, T. Lipic, S. Grgic, **Learning the principles of art history with convolutional neural networks**, *Pattern Recognition Letters*, Vol.129, pp.56-62, 2020.
13. Abdulmajeed Alsufyani, **Analyses of The P300 Event Related Potentials in EEG Signals Using Shape-based Kernel: Fixed and Random Analyses Approach**, *International Journal of Emerging Trends in Engineering Research*, Vol.8, pp.476-490, 2020.
14. Seok-Woo Jang, Sang-Hong Lee, **Analyses Classification of Epileptic Seizure EEG Based on Fully Connected Neural Network**, *International Journal of Emerging Trends in Engineering Research*, Vol.8, pp.3012-3015, 2020.