Study of Deep Learning-based Hand Gesture Recognition Toward the Design of a Low-cost Prosthetic Hand

Saikawa Yamato* and *Abderazek* Ben Abdallah

[1]The University of Aizu, School of Computer Science and Engineering, Adaptive Systems Laboratory, Japan

Abstract. Background: The surface EMG (sEMG) signal is inherently noisy and, therefore, not a robust input source for prosthetic systems, especially for fatigue, electrode displacement, and sweat conditions. We propose to address these issues by designing a multi-modal approach that combines vision and EMG empowered with appropriate dataset collection.
Methods: In Frame-based, the machine learning model used for recognition was a 2D-CNN. The data is image data that is input to the model by preparing videos showing 10 patterns of hand gestures along with multiple backgrounds, and dividing these videos into frames. These image data are then pre-processed and input to the machine learning model. The model is then evaluated in terms of the accuracy of hand gesture identification using the test data and the loss value, which represents the error between the expected data and the correct data output. In EMG, the Myo armband is placed on the forearm and the sEMG of 200 (Hz) is measured. There are six patterns of hand gestures in this process. Similar to the images, these sEMG data are preprocessed and input to a machine learning model for classification. The model is evaluated the model by the accuracy of hand gesture identification using the test data and the loss value, precision, recall , F1-score.
Results: The value of the loss function in case of frame-based was 0.0770 and the accuracy was 0.9739 at 1000 epochs of the training data. And the value of the loss function values in the test data were 0.1011 for the loss value and 0.9657 for the accuracy. In the case of EMG, the loss value was 0.931 when the time to maintain the gesture was the longest, and the loss value was 0.171. However, Precision, Recall, and F1-score were not the highest at the longest time for some gestures.
Conclusion: In this paper, we created a hand gesture identification software using Frame-based and sEMG, and measured its accuracy and loss value. For sEMG, we used Precision, Recall, and F1-score to check the metrics of each gesture identification. The frame-based results showed good results in both precision and loss values. sEMG showed an improvement in precision and loss values as the time length increased, but there was a tendency to decrease in some indices. In the future, it is necessary to explore the local relationship between finger and forearm to optimize out learning model.
keywords: Multi-modal, prosthetic, control system

1 Introduction

Hand gestures are a type of non-verbal communication that uses visible body movements to convey important messages. In recent years, hand gesture recognition has received a great deal of attention from the research community for various applications such as advanced driver assistance systems, artificial limbs, and robot control. Therefore, there is a need for accurate and fast classification of hand gestures. One of the methods of gesture recognition is surface electromyography(sEMG). Current research efforts using sEMG include remote control of robotic systems and brain-machine interfaces.[1][2] But the surface sEMG signal is inherently noisy and, therefore, not a robust input source for prosthetic systems, especially for fatigue, electrode displacement, and sweat conditions. We propose to address these issues by designing a multi-modal approach that combines vision and sEMG empowered with appropriate dataset collection.
In this paper, we present a software-based machine learning model for multi-model hand gesture recognition using frame-based images and sEMG signal also evaluated that model and validated the accuracy.

*Corresponding Author: SAIKAWA Yamatoe-mail: s1260072@u-aizu.ac.jp

2 Fundamental Items

2.1 Electromyography(EMG)

Electromyography(EMG) is recording of the weak electrical signals generated when a living organism contracts its muscle fibers as a signal of time and potential.
There are also two types of EMG measurement methods. Those acquired indirectly, by placing EMG sensor on that surface of the skin, are called Surface EMG (sEMG), while those acquired directly, by inserting a needle into the muscle, are called Needle EMG. In this paper, sEMG was used as the data.

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

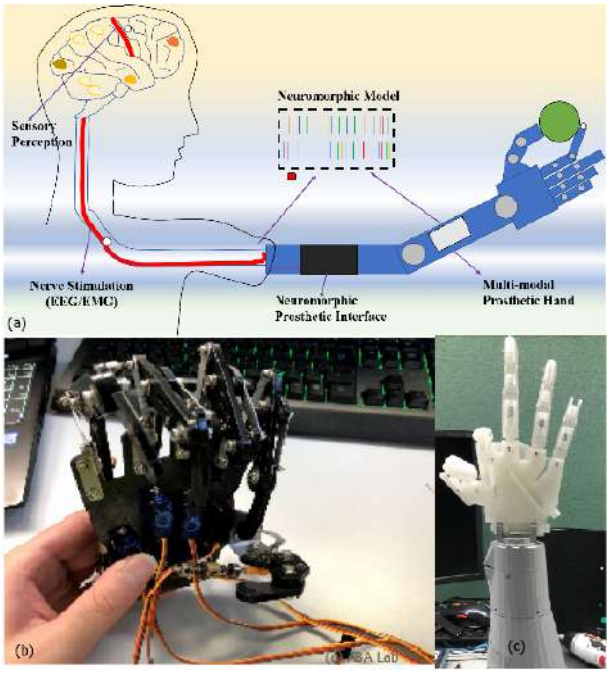

Figure 1. NeuroSys System Overview. (a) The EEG/EMG signals are used as input to the prosthetic hand's controller, (b) Mechanical prosthetic hand for testing, (c) 3D printed prosthetic hand.

2.1.1 Myo armband

The Myo armband is a sEMG sensor developed by Thalmic Labs Inc. The sensor incorporates eight sEMG sensors and one 9-axis inertial measurement unit, which can record hand and arm movements as signals. The time and frequency to be measured can be specified by the user and the recorded data is wirelessly transmitted to the computer by a transmission module.

2.2 Frame-based

Frame-based is image data in which the video is divided into frames. If the image is captured at 30 fps, then 30 images are generated every second. And image data is treated as 3D data, where the x and y-coordinate represents the position of the background, object, etc. that the data, and the z-axis represents the color scale of the data.

2.3 Artificial Neural Network

Artificial Neural network(ANN) is a mathematical model that imitates a neuron. They consist of three types of layers: an input layer, one or more hidden layers, and an output layer. ANN can learn by updating the weights between each node. Therefore, it can be used as a model for machine learning such as pattern recognition. In addition, various models have been proposed for ANN that are suitable for different types of data and conditions. In this paper, we used Convolutional Neural Network(CNN), which is suitable for image recognition and pattern recognition of time series data.

2.3.1 Convolutional Neural Network(CNN)

Convolutional Neural Network (CNN) is a type of ANN proposed based on visual perception of human, which has convolutional layers and pooling layers in addition to the general ANN structure. Each convolutional layer has one or more filters for extracting features of the input data, and performs convolutional operations with them. The pooling layer performs compression of the data output from the convolutional layer. Compression can be done by max pooling, which takes the maximum value in a specified range, or average pooling, which takes the average.
In this paper, we use two types of CNN: 2D-CNN and 1D-CNN. 2D-CNN are CNN with two-dimensional filters and are often used for image and video processing. 1D-CNN are CNN with one-dimensional filters and are often used for time series data.

3 Low-cost Prosthetic Hand Overview

As shown in Figure 1, we investigate advanced prosthetic hands and robot arms with sensorimotor integration and tactile sensing.(Figure 1)The novel prosthetic hand is based on biological signal discrimination with neuromorphic circuits to restore hand function movement for amputations or neurological disorders. Using our neuromorphic circuits and system, we aim to develop solutions to improve the performance and control of upper-limb prosthetics.[3] As a preliminary study to move the robot arm, this paper focuses on the hand. We aim to recognize hand gestures from Frame-based and sEMG of the forearm and reproduce them with the mechanical prosthetic hand for testing and the 3D printed prosthetic hand (Figure 1 (c)).

Layer	Input size	Output size
Conv.	64×64×1	62×62×16
Max Pool	62×62×16	31×31×16
Conv.	31×31×16	29×29×32
Max Pool	29×29×32	14×14×32
Conv.	14×14×32	12×12×64
Conv.	12×12×32	10×10×64
Max Pool	10×10×64	5×5×32
FC	1600	128
FC	128	10

Table 1. 2D-CNN for Evaluation

Layer	Input size	Output size
Conv.	10×8	8×64
Conv.	8×64	6×64
Avg. Pool	6×64	3×64
FC	192	6

Table 2. 1D-CNN of N = 10 for Evaluation

Layer	Input size	Output size
Conv.	100×8	98×64
Conv.	98×64	96×64
Avg. Pool	96×64	48×64
Conv.	48×64	46×64
Avg. Pool	46×64	23×64
FC	1472	6

Table 3. 1D-CNN of N = 25, 50, 100 for Evaluation

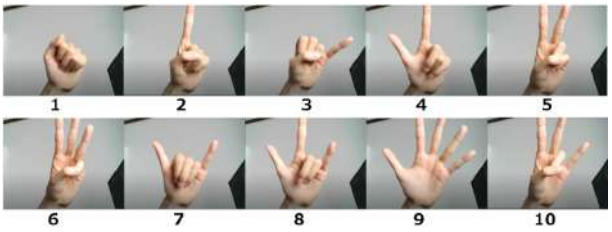

Figure 2.

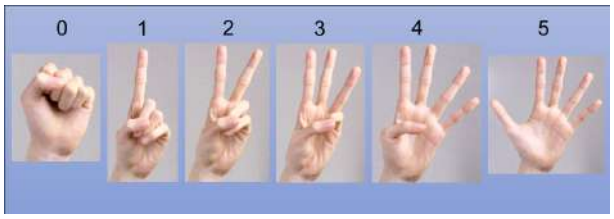

Figure 3.

4 System Design for Recognition

In this paper, we have created two machine learning models for software recognition of what hand gestures indicate from Frame-based and sEMG input data. Therefore, this section indicates a description of how we acquired Frame-based data and a learning model to recognize hand gestures from Frame-based and sEMG. The whole process involves the acquisition of data, pre-processing of those data, and inputting them into the machine learning model. The machine learning models used are 2D-CNN (Table 1) and 1D-CNN (Table 2 and 3). 2D-CNN is used for frame-based recognition, while 1D-CNN is used for sEMG recognition. They were developed using TensorFlow[1] in the Python 3.8.8 environment.

4.1 Learning model

2D-CNN consisting of convolutional layers, maxpooling layers, and a dropout layers[7] is used for analysis and identification. It extracts the features of each Frame-based data and analyzes the patterns. It adopts the Softmax Function in the output layer to output the probability that the input data is each gesture.
1D-CNN consisting of convolutional layers with 1D filters, avaragepooling layers, and a dropout layers is used for analysis and recognition. In convolutional layers, these are set up the convolution operation to extract the features by time for each channel.

4.2 Data Acquisition

4.2.1 Frame-based Acquisition

Firstly, we have prepared a video showing 10 hand gesture patterns with several backgrounds. The 10 hand gesture patterns we used are shown in Figure2. They are shot at 30 fps. We split those videos into frame units to create image data for input to the model. In other words, for a 10-second video, it will be 300 pieces of image data.
3.

4.2.2 sEMG Acquisition

Firstly, the Myo Armband is placed on the forearm. The Myo Armband measures sEMG at 8 locations on a forearm. Therefore, the data acquired has in the form of N × 8. Then, set the frequency and the time to measure in the software that manages the device. The frequency was fixed at 200 (Hz) and the time length was varied according to the data. In addition, sEMG of one gesture is recorded per measurement. During the measurement, the elbow is always extended or lightly bent in order to avoid changes in sEMG due to arm movement as much as possible.
There are six patterns of hand gestures that were measured, and they are shown in Figure

4.3 Data pre-processing

4.3.1 Image pre-processing

To input the data into the machine learning model, we preprocess the data. First, cut off as much of the extra background as possible, focusing on the hand gestures in the image data. This is to allow more information on the hand gestures to be input. Next, resize the image size to 64x64 and change it to gray. This will reduce the data size and thus reduce the time required for training and identification. Finally, perform data augmentation[6], which randomly changes the angle, position, and brightness of each data. This is to prevent overfitting and to increase the versatility of the data. This data augmentation is only used for training the model.

4.3.2 sEMG pre-processing

Firstly, we merge all the measured sEMG data into one data set. For instance, if there are N x 8 data and M x 8 data, the data becomes (N + M) x 8.
Secondly, The integrated data is then divided into the specified time length to create multiple data. At this time, one data should have only one gesture.

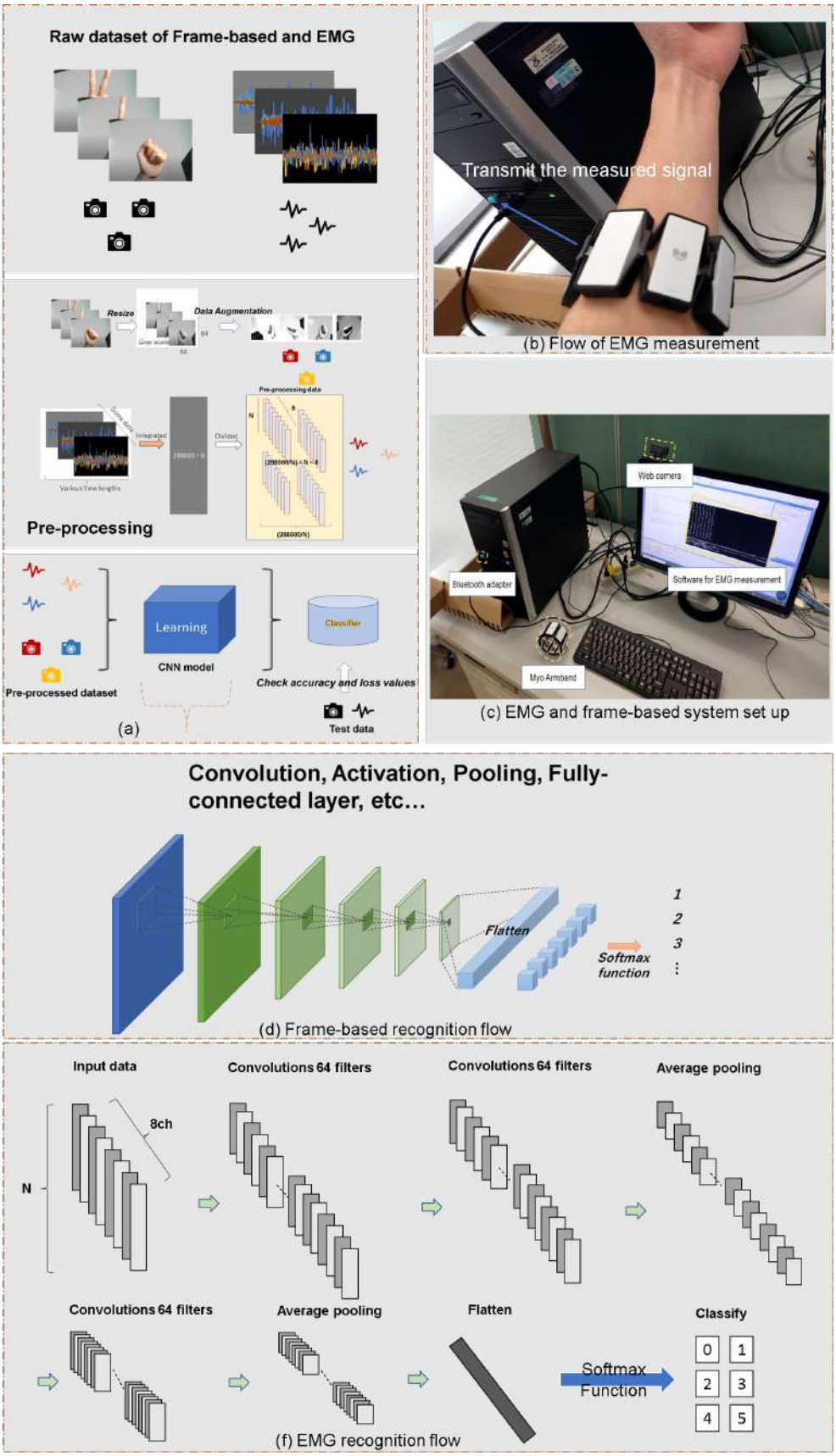

Figure 4. System architecture: (a)Pre-process the raw dataset of EMG and frame-based, the flow of training and classifying ,(b) Method of EMG measurement ,(c)EMG and frame-based system set up ,(d)Frame-based recognition flow ,(d)sEMG recognition flow

5 Evaluation

5.1 Evaluation of Frame-based

5.1.1 Evaluation method for Frame-based

The structure of the model for evaluation is shown in Table 1.
In 2D-CNN for Frame-based, first, A total of 12,587 size image of data are prepared as training data. This data was collected by [5]. And each piece of data is augmented 30 times and added as training data. Then, a 2D-CNN is trained with the number of epochs set to 1000 and the batch size set to 512.
Next, the model is evaluated using test data for the accuracy of hand gesture identification and the loss value, which represents the error between the expected data and the output of the correct data.

5.1.2 Evaluation Result of Frame-based

The accuracy of the 2D-CNN model for identifying frame-based data on the accuracy of training data is 0.9739 and a loss value of training data is 0.0770. The accuracy of the test data is 0.9657, and the loss value of test data is 0.1011. (Figure 5)

5.2 Evaluation of sEMG

5.2.1 Evaluation method for sEMG

The structure of the model for evaluation is shown in Table 2 and Table 3.
For sEMG, first, the data with a total time length of 288000 is divided into the values indicated by the table 4 and 90% of it is used for training data.
Then, a 1D-CNN is trained with the number of epochs set to 3000 and the batch size set to 100. The remaining 10% are used as test data. Then, using the test data, the model is evaluated in terms of the accuracy of hand gesture identification and the loss value, which represents the error between the expected data and the correct data output. In addition to that, we also evaluate each gesture by Precision, Recall, and F1-score.
Precision is a percentage of how correct the gesture predicted by the learned model is, and is calculated as follows using Table 5.

$$TP_i = C_{ii}, \quad FP_i = \sum_{\substack{j=0 \\ j\neq i}}^{5} C_{ij}, \quad FN_i = \sum_{\substack{k=0 \\ k\neq i}}^{5} C_{ki} \tag{1}$$

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \tag{2}$$

Recall is the percentage of what actually output of what should have output, and is calculated by the following formula

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \tag{3}$$

F1-socore is an index that comprehensively evaluates Precision and Recall, and is calculated as follows

$$F1 - socore_i = \frac{2 * Precision_i * Recall_i}{Precision_i + Recall_i} \tag{4}$$

N	Size
10	$28800 \times 10 \times 8$
25	$11520 \times 25 \times 8$
50	$5760 \times 50 \times 8$
100	$2880 \times 100 \times 8$

Table 4. Time length and Input size

True value \ Predicted value	0	1	2	3	4	5
0	C_{00}	C_{01}	C_{02}	C_{03}	C_{04}	C_{05}
1	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}
2	C_{20}	C_{21}	C_{22}	C_{23}	C_{24}	C_{25}
3	C_{30}	C_{31}	C_{32}	C_{33}	C_{44}	C_{45}
5	C_{50}	C_{51}	C_{52}	C_{53}	C_{54}	C_{55}

Table 5. Definition of confusion matrix

5.2.2 Evaluation Result of sEMG

The results of the test accuracy and loss values for each time length are shown in the Figure 6 and Table 6. As the table shows, the test accuracy for N=10 was about 76.5%, but by increasing the length by a factor of 10, the accuracy reached about 93.1%. The value of the loss function was also high at about 0.64 for N=10, but became small at about 0.171 for N=100. Therefore, as the time length increases, the recognition accuracy increases and the value of the loss function becomes smaller.
Table 7 shows the results of Precision. As the table shows, the value of gesture 0 shows 100%, except for N=10. The value for N=10 is also very high at 98.9. Gestures 1 and 5 showed the lowest values when N=10 and the highest values when N=50. Gestures 2, 3, and 4 showed the lowest values when N=10, and the highest values when N=100.
Table 8 shows the results of Recall. As the table shows, the value of gesture 0 shows 100% at N=50,100, and very high values outside of that. All except gesture 5 showed the highest values at N=100 and the lowest values at N=10. Only one, gesture 5, had the highest value at N=50.
Table 9 shows the results of F1-score. All except gesture 5 showed the highest value when N=100 and the lowest value when N=10. Gesture 5 had the highest value when N=50.

6 Discussion

The overall accuracy tends to increase as the time increases, but for gesture 0, the accuracy was 100% or very close to 100% at all times. Therefore, the hand grasping action like gesture 0 causes the muscles in the forearm to contract the most. And gesture 5 had the lowest of the three indices compared to the other gestures. Therefore, it can be seen that the hand-opening motion such as gesture 5 causes the least contraction of the forearm muscles.
However, even when some fingers are grasped, such as gestures 1, 2, and 3, a high percentage of the gestures are

N	Test Accuracy	Test loss	Training Accuracy	Training loss
10	0.756	0.640	0.816	0.485
25	0.860	0.418	0.917	0.219
50	0.898	0.320	0.966	0.093
100	0.931	0.171	0.989	0.039

Table 6. Accuracy and loss value of sEMG

N	Gesture 0	Gesture 1	Gesture 2	Gesture 3	Gesture 4	Gesture 5	Avg.
10	0.989	0.758	0.687	0.649	0.690	0.740	0.752
25	1.	0.889	0.839	0.766	0.830	0.836	0.858
50	1.	0.940	0.921	0.798	0.822	0.931	0.902
100	1.	0.918	0.98	0.907	0.885	0.889	0.930

Table 7. Precision of sEMG

N	Gesture 0	Gesture 1	Gesture 2	Gesture 3	Gesture 4	Gesture 5	Avg.
10	0.985	0.808	0.71	0.624	0.725	0.662	0.752
25	0.995	0.907	0.79	0.774	0.853	0.826	0.858
50	1	0.918	0.911	0.83	0.863	0.879	0.9
100	1	0.978	0.925	0.867	0.982	0.8	0.925

Table 8. Recall of sEMG

N	Gesture 0	Gesture 1	Gesture 2	Gesture 3	Gesture 4	Gesture 5	Avg.
10	0.987	0.782	0.698	0.636	0.707	0.699	0.752
25	0.997	0.898	0.81	0.77	0.841	0.831	0.858
50	1	0.929	0.916	0.814	0.842	0.904	0.901
100	1	0.947	0.951	0.886	0.931	0.842	0.926

Table 9. F1-score of sEMG

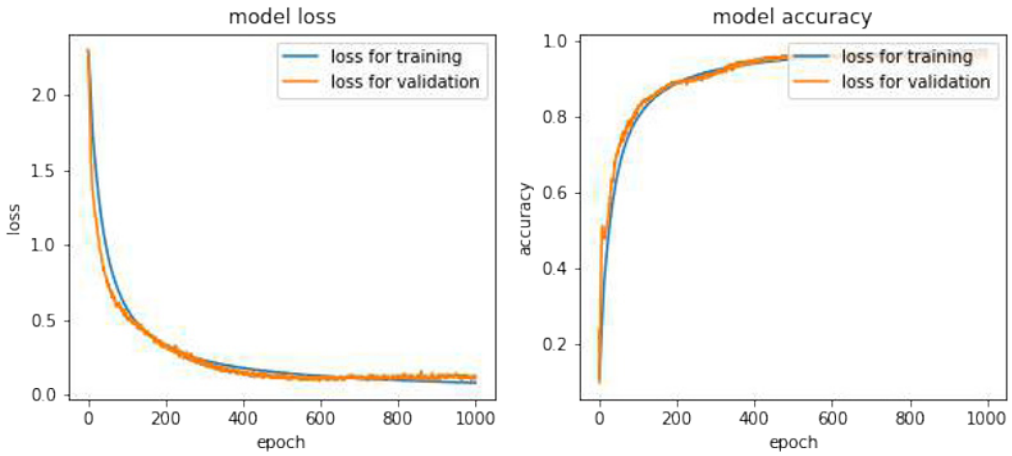

Figure 5. Graph of accuracy and loss value for 2D-CNN

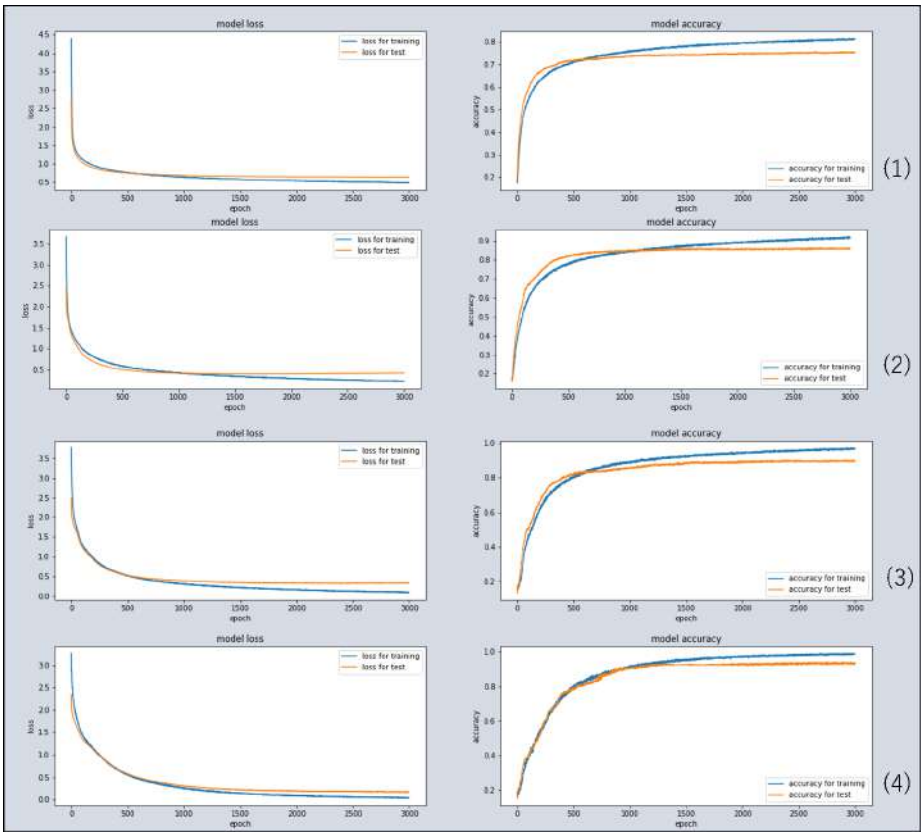

Figure 6. Graph of accuracy and loss value for 1D-CNN (1)N=10, (2)N=25, (3)N=50, (4)N=100

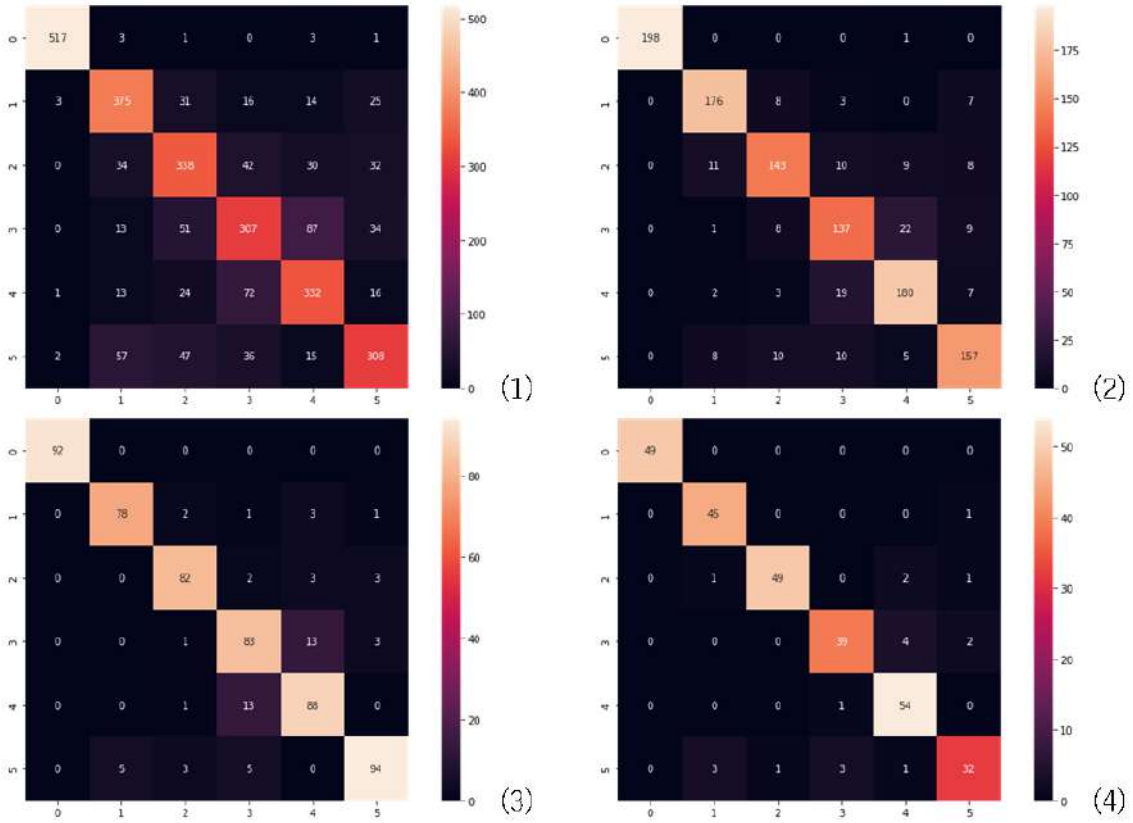

Figure 7. Confusion matrix (1)N=10, (2)N=25, (3)N=50, (4)N=100

misclassified as gesture 5, as shown by the ConfusionMatrix. On the other hand, gesture 4 was less misrecognized than 1, 2, and 3, even though the hand was the most open. We suggested that the tension state of the thumb may be deeply related to the forearm.
Therefore, it is possible to create a more accurate learning model by looking for areas in the arm that are significantly related to other fingers and measuring the sEMG of hand gestures from there.

7 Conclusion and Future Work

In this paper, we created a software to identify hand gestures using Frame-based and sEMG, and measured its accuracy and the loss value. In addition, we used Precision, Recall, and F1-score for sEMG to check the index of each gesture identification. The frame-based results showed good results in terms of both accuracy and loss values. sEGM showed that the accuracy and loss values improved as the time length increased, but only Gesture 5 showed a decreasing trend in terms of other indicators. For gesture 0, it was also observed that the time length did not matter.
In the future, we need to explore the local relationship between fingers and forearms to create an even better learning model and we then need to create one integrated machine learning model using these two sets of data.

References

[1] Ray Antonius and Hendra Tjahyadi. Electromyography Gesture Identification Using CNN-RNN Neural Network for Controlling Quadcopters. *Journal of Physics*,July(2021).

[2] Andrea Sarasola-Sanz,Nerea Irastorza-Landa,Eduardo López-Larraz,Carlos Bibián,Florian Helmhold, Doris Broetz,iels Birbaumer,Ander Ramos-Murguialday. A hybrid brain-machine interface based on EEG and EMG activity for the motor rehabilitation of stroke patients. *International Conference on Rehabilitation Robotics (ICORR)* ,July(2017).

[3] Division of Computer Engineering Ben Abdallah Laboratory *http://web-ext.u-aizu.ac.jp/misc/benablab/projects.html*.

[4] TensorFlow *https://www.tensorflow.org/?hl=ja*.

[5] Naoto Ageishi. Design of Hand Gesture Recognition based on Deep Neural for Deep Learning. ,(2020).

[6] Connor Shorten and Taghi M. Khoshgoftaar. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data* ,volume 6, July (2019).

[7] Shaofeng Cai, Yao Shu, Gang Chen, Beng Chin Ooi, Wei Wang, Meihui Zhang. Effective and Efficient Dropout for Deep Convolutional Neural Networks for Deep Learning. *IEEE* ,July(2020).