International Journal of Internet, Broadcasting and Communication Vol.13 No.2 208-217 (2021)
http://dx.doi.org/10.7236/IJIBC.2021.13.2.208

IJIBC 21-2-26

Effect of Input Data Video Interval and Input Data Image Similarity on Learning Accuracy in 3D-CNN

Heeil Kim [1] and Yeongjee Chung [2*]

[1]*Master Course, Department of Computer Engineering, Wonkwang University, Korea*
E-mail: heeilkim94@gmail.com
[2]*Professor, Department of Computer and Software Engineering, Wonkwang University, Korea*
E-mail: yeongjee@gmail.com

Abstract

3D-CNN is one of the deep learning techniques for learning time series data. However, these three-dimensional learning can generate many parameters, requiring high performance or having a significant impact on learning speed. We will use these 3D-CNNs to learn hand gesture and find the parameters that showed the highest accuracy, and then analyze how the accuracy of 3D-CNN varies through input data changes without any structural changes in 3D-CNN. First, choose the interval of the input data. This adjusts the ratio of the stop interval to the gesture interval. Secondly, the corresponding interframe mean value is obtained by measuring and normalizing the similarity of images through interclass 2D cross correlation analysis. This experiment demonstrates that changes in input data affect learning accuracy without structural changes in 3D-CNN. In this paper, we proposed two methods for changing input data. Experimental results show that input data can affect the accuracy of the model.

Keywords: *3D-CNN, Gesture recognition, RNN, 2D-Cross correlation.*

1. Introduction

Since Alex Net's record performance improvement in the Imaging Net Challenge [1], many researchers have begun to apply the 2D convolutional neural network (2D-CNN) structure to multiple applications, and to the human behavior recognition field . Typically, there have been attempts to take the 2D-CNN structure as it is and apply it to each image frame. These attempts have been limited because they fail to utilize structurally temporary information in neural networks. Therefore, to solve these structural problems, a combination of convolutional neural network (CNN) and recurrent neural networks (RNN), which learns special features with CNN and learns them with long short-term memory (LSTM). However, these combinations were also limited in performance compared to existing studies, and there was still a structural problem that convolutional filters

Manuscript Received: April. 6, 2021 / Revised: April. 10, 2021 / Accepted: April 15, 2021
*Corresponding Author: yeongjee@gmail.com
Tel: +82-63-850-6887, Fax: +82-63-856-8009
Professor, Department of Computer and Software Engineering, Wonkwang University, Korea

learned only special features. Several methods have been devised to overcome the limitations of these 2D-CNN single structures and to learn more temporal features, one of which has achieved significant results.

3D convolutional neural network (3D-CNN) is an approach using 3D convolution, with convolutional filters all three dimensions. Therefore, the feature map produced by one filter is also three dimensional. Thanks to these structures, 3D-CNN is able to learn Temporal learning of continuous frames from convolutional filters themselves. This structure allows for temporal feature learning for short-terms [2]. However, the 3D-CNN method has the disadvantage of being difficult to proceed with learning because the filter is 3D and there are far more parameters and no pre-trained models such as 2D-CNN.

Because of this, it is burdensome to construct a sufficiently deep structure and has limitations in obtaining high accuracy by transforming the structure of the model. Therefore, in this work, we briefly describe the structure of 3D-CNN and want to analyze the change in the accuracy of the model when only the input data is changed without structural deformation of 3D-CNN. As a method, we transform input data and analyze the accuracy change with two methods: frame interval selection of input data and 2D cross correlation between classes.

2. Related Work

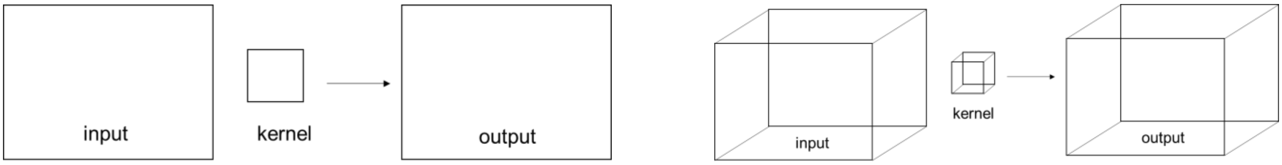
The RNN algorithm is a type of artificial neural network specialized for repetitive and sequential data learning, characterized by an internal circulation structure. Using a circular structure, past learning is reflected in current learning through Weight. The RNN solves the limitations of existing continuous, repetitive, and sequential data learning. It also has the characteristic of enabling the connection between current and past learning and being time dependent. It is mainly used to identify speech waveforms or the front and back components of text [2-3].

It is known that RNNs gradually reduce the gradient during backpropagation when the distance between the relevant information and the point where it is used is significantly reduced. This is called a vanishing gradient problem. It is LSTM that is designed to overcome this problem. LSTM is a structure that adds cell-state to the hidden state of an RNN. Similar to the method of RNNs calculated over time, but with differences in the computational method of the hidden layer. LSTM consists of four stages. Step 1 uses a sigmoid function to select the data to delete. Step 2 uses sigmoid and hyperbolic tangential functions (Tanh) to determine whether new data are stored in the LSTM cell phase. Step 3 updates the cell state and in the final step determines the output value from the Tanh function and cell state that passed through the sigmoid function. In addition, there are several other studies for video recognition, such as Two-Stream Net using two streams and Inflated 3D ConvNet (I3D), which converts 2D ConvNet into 3D ConvNet [5-7].

3. Train Hand Gesture With 3D-CNN

3.1 3D-CNN Architecture

3D-CNN is one of the most used models in video modeling. With 3D CNNs, video data can be learned without the loss of spatiotemporal information. Compared to 2D-CNN, 3D-CNN create temporal information due to 3D convolution and 3D pooling operations. That is, operations performed only spatially by 2D-CNN can be performed over space-time in 3D-CNN. Figure 1 shows a relative comparison of the concept of a single convolution. Although 2D convolution only produces one image because it is recognized as a channel even when applied to multiple images, 3D convolution preserves temporal information of input signal and the calculation results is volume. Figure 1(a) shows the input image, kernel, and output form of a two-dimensional convolution, and Figure 1(b) shows the input image, kernel, and output form of a three-dimensional convolution. [8-9].



(a) Two-dimensional **(b) Three-dimensional**

Figure 1. 2D and 3D convolution operations.

3.2 Data Set

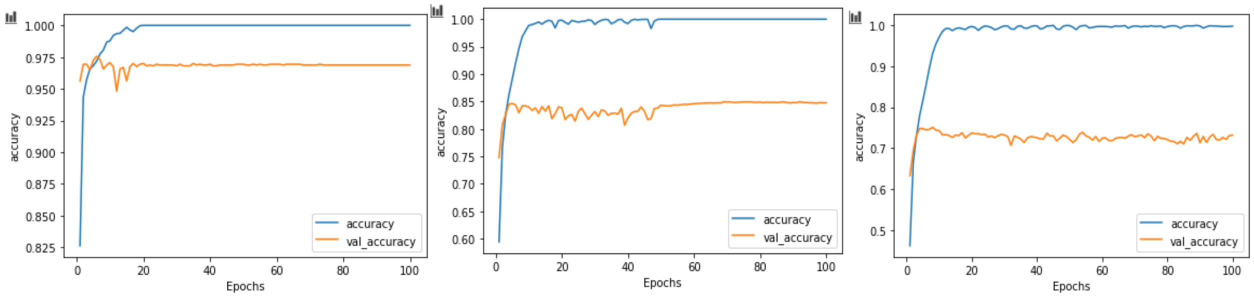
We used a public video clip dataset called 200B-jester Dataset V1 [10]. The 20BN-Jester Dataset V1 consists of 148,092 video clips, extracted and used only eight of the 27 labels. 2000 clips were used for each class and 30 frames were extracted for each clip. Table 1 shows a list of gestures we used.

Table 1. Gesture list

Labels
No Gesture
Sliding Two Fingers Left
Sliding Two Fingers Right
Stop Sign
Swiping Left
Swiping right
Thumb Up
Thumb Down

3.3. Train Hand Gesture

As a result of progressing from one class to ten classes, the accretion was close to 99% for two classes, but the accuracy decreased as the number of classes increased. We trained gesture by varying the number of classes, running rates, and batch sizes to find parameters showing high accuracy. First, we analyzed how the number of classes used in learning affects accuracy. Learning was conducted by dividing into two classes, four classes, and eight classes. Tables 2 and Figure 2 shows the learning accuracy of the model. Learning accuracy decreased dramatically as the number of classes increased.



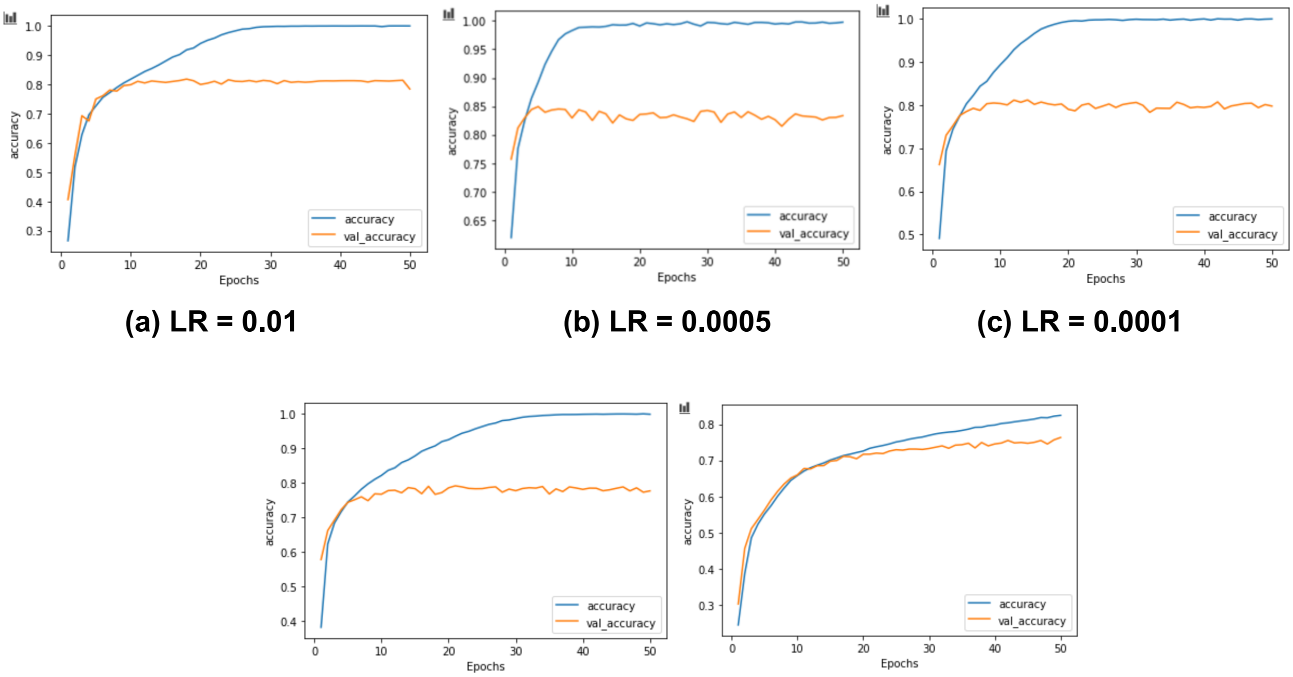
(a) 2 classes **(b) 6 Classes** **(c) 8 Classes**

Figure 2. Accuracy by number of classes

Table 2. Accuracy by number of classes

Number of classes	Accuracy
2	0.9688
6	0.8470
8	0.7318

Next, we analyze the effect of learning rate (LR) on accuracy during learning. LR was studied in five stages from 0.01 to 0.00001. Figure 3 and Table 3 shows the learning results by LR. Figure 3(b) shows learning results showing the highest accuracy, and Figure 3(e) shows learning results with the lowest accuracy. Figures 3(b), 3(c), 3(d), and 3(e) shows that the accuracy seems to increase as LR increases, but the accuracy decreases when *LR = 0.01*.



(a) LR = 0.01 **(b) LR = 0.0005** **(c) LR = 0.0001**

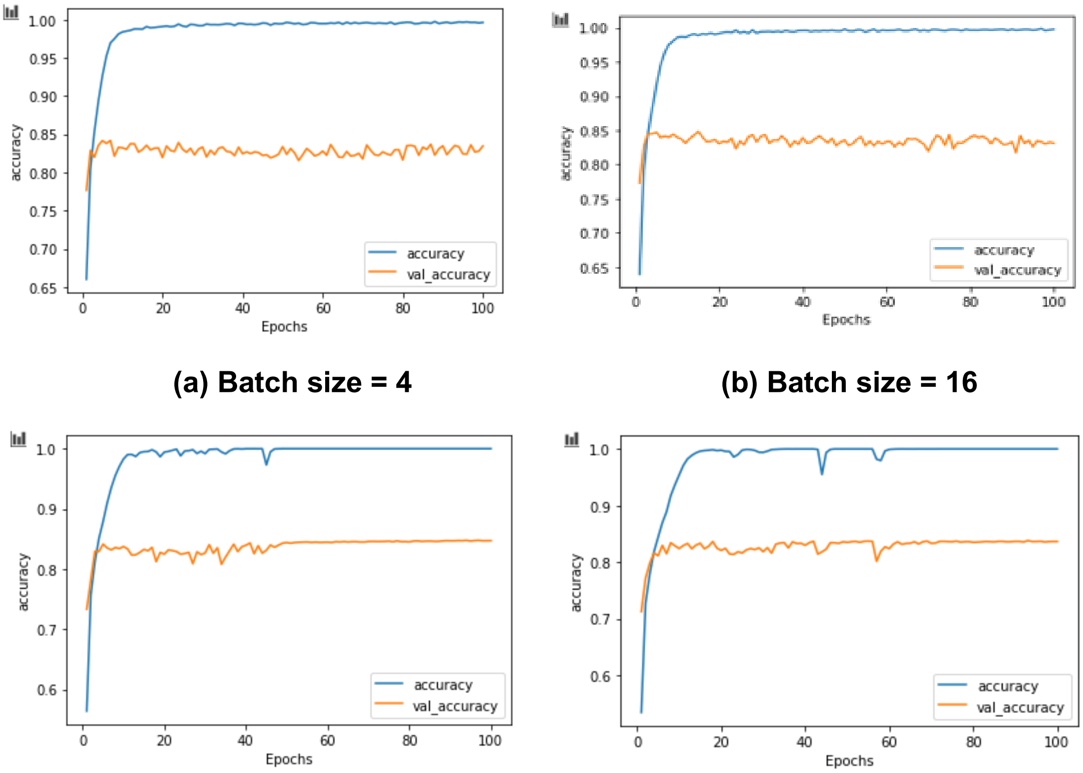
(d) LR = 0.00005 **(e) LR = 0.00001**

Figure 3. Accuracy by learning rate

Table 3. Accuracy by learning rate

Learning rate	Accuracy
0.01	0.7843
0.0005	0.8333
0.0001	0.7975
0.00005	0.7760
0.00001	0.7632

Finally, we analyze the effect of batch size on accuracy. It had the highest accuracy when the batch size was 64, and only tested up to batch size 128 to avoid out of memory (OOM) error. Table 4 and Figure 4 show the learning results by batch size.



(a) Batch size = 4

(b) Batch size = 16

(c) Batch size = 64

(d) Batch size = 128

Figure 4. Accuracy by batch size

Table 4. Accuracy by batch size

Batch size	Accuracy
8	0.8345
16	0.8312
64	0.8470
128	0.8363

4. Methods for selecting input data.

We approach the effect of influencing the accuracy of the model from two aspects without any deformation of a separate 3D-CNN structure. The first is input image tuning through data interval selection, and the second is 2D cross correlation to measure the similarity between data and analyze the effect of these data on learning accuracy.

4.1. Through Frame Intervals

For 20BN-jester Dataset V1, each class consists of a video clip and a frame. The frame is divided into stop shift-stop intervals, assuming that all clips follow the same structure as shown in Figure 5. We selected input intervals from n frames in three ways. The input frame is a center 30 frame with half the stop interval and half the motion interval, and a front and rear 30 frames with many stop intervals. Learning accuracy was detected the highest in the 30 frames in the middle of the video, and the front and back sections of the video were found to be less accurate than the middle section.

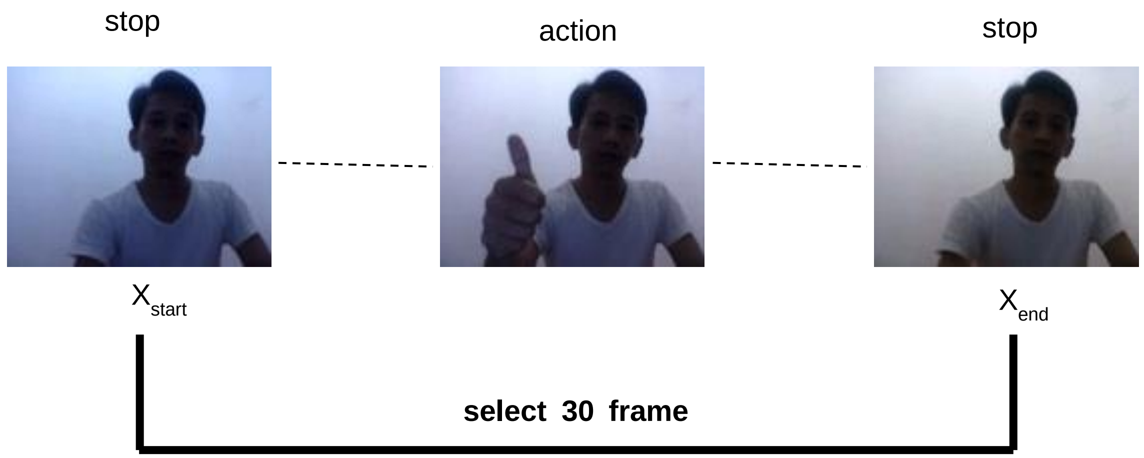


Figure 5. Input data structure

Therefore, this means that the accuracy can vary depending on how many stationary and action intervals appear within a single learning. If the stop section and the action section appear similar, it can be confirmed that the accuracy is the highest, and if one section is larger than the other, the accuracy is lower. For example, if the stop interval is 15 frames out of 30 frames and the action interval is 15 frames out of 30 frames, it will be higher than if the stop interval is 1 frame and the action interval is 29 frames. This means that the accuracy of the model can be made better or worse by selecting the starting point of the input interval when learning 3D-CNN.

4.2. Through 2D Cross Correlation Between Input Data

We measured the similarity between input images through a two-dimensional cross correlation analysis that analyzes how similar they are between the data [11], and we normalized the values detected through the analysis and converted them to values between 0 and 1. Figures 7, 8, and 9 are two-dimensional reciprocal analysis graphs of the same image, similar image, and different image, respectively. The 2D cross correlation are calculated by (1):

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} h[u,v]F[i+u,j+v] \tag{1}$$

In Equation (1), *i* is width of frame, *j* is height of frame. The *u* and *v* is coordinate vales of the kernel *h*. *h[u,v]* is the prescription for weights in the linear combination. A negative number exists in the value of k because the base point in the formula is the center of the image. In the actual calculation, the calculation is based on the origin so that no negative numbers are produced.

$$g[i,j]_{normalization} = \frac{g[i,j]-g[i,j]_{min}}{g[i,j]_{max}-g[i,j]_{min}} \tag{2}$$

We were able to see that the highest point is calculated close to the point, as shown in Figure 7, if the same image is computed in 2D cross correlation. This is because for the same image, the middle part of the image is the most similar to all pixels. In addition, the more different the images, the higher the correlation value overall. We performed the above calculations among frames of corresponding indexes between input data and measured the similarity between images. In addition, we found that the higher the similarity between images, the slightly lower the learning accuracy if all parameters were fixed and learned after measuring only the similarity between images.

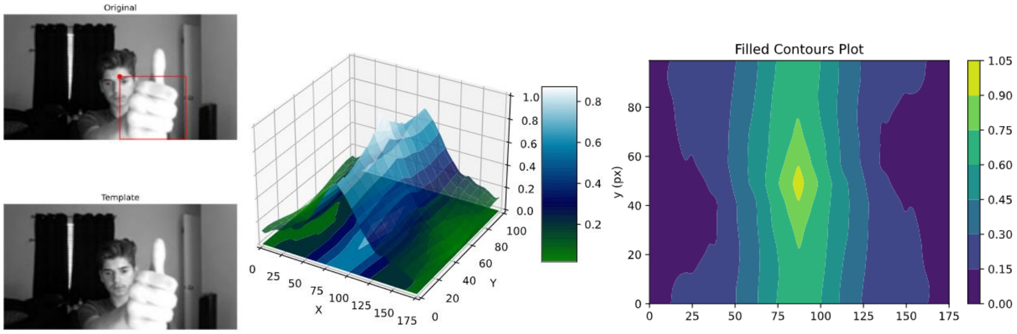


Figure 7. Same image

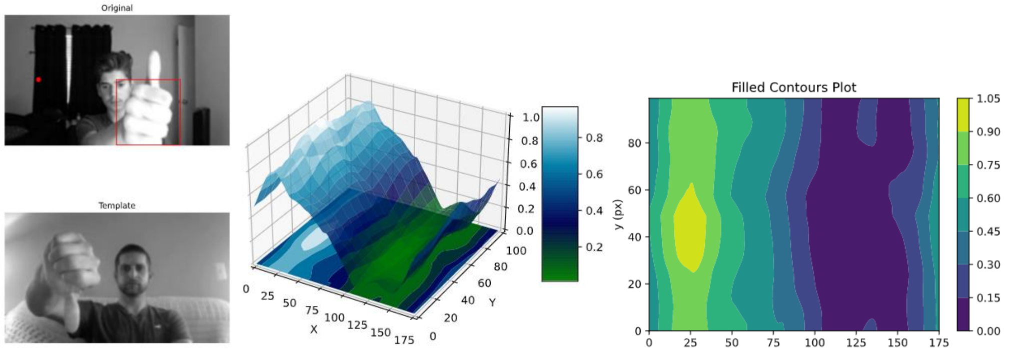


Figure 8. Similar Image

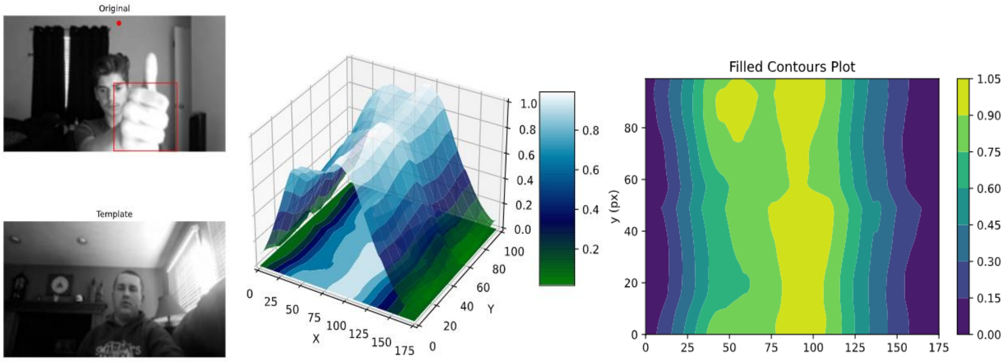
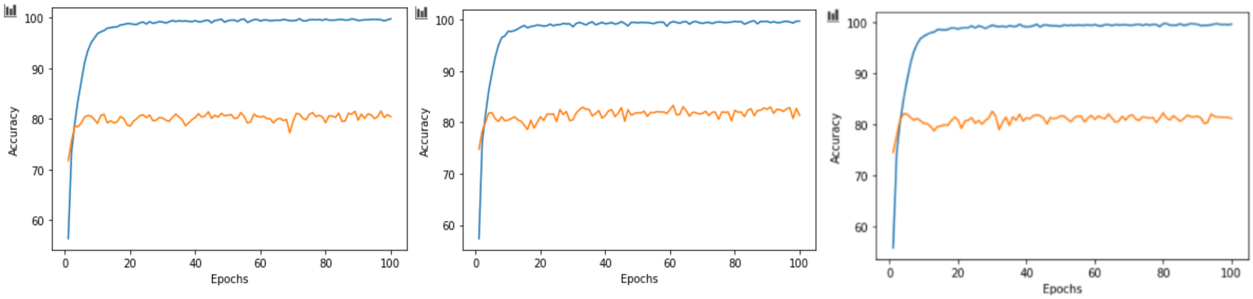


Figure 9. Difference Image

5. Result

The experimental results are as follows. When the data were changed by the method presented in 4.1. it was detected with the highest accuracy in the 30 frame interval in the middle 30 frame. This means that the ratio of motion to stationary motion to recognition within the video interval of the learning data affects the learning accuracy. In this experiment, we found that the learning accuracy decreased when the proportion of either interval increased, and the accuracy difference was approximately 4%. Table 5 and Figure 6 shows the learning accuracy by input frame interval.



(a) Front 30 frame **(b) Middle 30 frame** **(c) Rear 30 frame**

Figure 6. Accuracy by frame intervals

Table 5. Accuracy by frame intervals

Frame	Accuracy
Front 30 frame	0.8138
Middle 30 frame	0.8470
Rear 30 frame	0.8113

The results learned by the method presented in 4.2. show that classes with high inter-frame similarity have a fine reduction in accuracy compared to those with low inter-class similarity. We calculated a set of three types of gestures with the method of 4.2. and separated the gestures into high similarity, medium similarity, and low similarity. Table 6 shows the learning accuracy by image similarity. There was no significant difference in accuracy between medium and high similarity, and learning accuracy was high in low similarity.

Table 6. Accuracy by image similarity

Gesture set	Similarity	Accuracy
Stop Sign-Thumb Up	High	0.9238
Stop Sign-Swiping Left	Medium	0.9370
Stop Sign-No Gesture	Low	0.9513

5. Conclusion

We analyzed the impact of input data selection on model accuracy. We found that the learning accuracy of the model increased when clear actions appeared in the input data interval, and that the accuracy decreased if the input data interval was narrow or too wide. In addition, we find that the higher the similarity between the data of the class being input upon learning, the lower the accuracy of the model. In this paper, we show that when there are performance limitations or a deeper model cannot be designed, the accuracy can be increased by changing the input data interval of 3D-CNN or by input class-specific correlation analysis. However, while this study has selected data assuming that all data follow the stop-action-stop cycle, there is a limitation in that all data may not follow the stop-action-stop cycle. There is also a limitation that larger resolution of images can have a bad effect on learning time because correlation requires a lot of computation and accuracy depends not only on the similarity between data but also on the absolute quality of the input data. Future work will study the technique of selecting frames by detecting data change intervals to overcome these limitations and how to adjust image resolution to make them change less in learning time while conducting correlation analysis.

Acknowledgement

This paper was supported by Wonkwang University, Iksan Korea in 2021.

References

[1] K. Alex, S. Ilya and E. T. Geoffrey , “ImageNet Classification with Deep Convolutional Neural Networks,” *Advances in neural information processing systems , Communications of the ACM,* vol. 60, no. 6, pp: 84–90, May 2017.
DOI: https://doi.org/10.1145/3065386

[2] S. G. Choi, and W. Xu, “A Study on Person Re-Identification System using Enhanced RNN,*” The journal of the institute of internet, broadcasting and communication(JIIBC)***,** v.17 no.2, pp. 15–23, Apr. 2017.
DOI: https://doi.org/10.7236/JIIBC.2017.17.2.15

[3] T. Du , B. Lubomir, F. Rob , T. Lorenzo and P. Manohar, “Learning Spatiotemporal Features with 3D Convolutional Networks,*” Proceedings of the IEEE International Conference on Computer Vision (ICCV),* pp: 4489-4497, Oct 2015.
DOI: https://doi.org/10.1109/iccv.2015.510

[4] K. Cho, B. V. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation,*” Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP),* 2014.
DOI: https://doi.org/10.3115/v1/d14-1179

[5] Hochreiter and Schmidhuber, “LONG SHORT-TERM MEMORY,” 1997.
DOI: https://doi.org/10.1162/neco.1997.9.8.1735

[6] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-Scale Video Classification with Convolutional Neural Networks,*" 2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014.
DOI: https://doi.org/10.1109/cvpr.2014.223

[7] J. Carreira and A. Zisserman, "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset,*" 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* Jul. 2017.
DOI: https://doi.org/10.1109/cvpr.2017.502

[8] K. Yang, R. Li, P. Qiao, Q. Wang, D. Li, and Y. Dou, "Temporal Pyramid Relation Network for Video-Based Gesture Recognition," *2018 25th IEEE International Conference on Image Processing (ICIP),* Oct. 2018.
DOI: https://doi.org/10.1109/icip.2018.8451700

[9] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional Two-Stream Network Fusion for Video Action Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* Jun. 2016.
DOI: https://doi.org/10.1109/cvpr.2016.213

[10] TwentyBN, "jester dataset: a hand gesture dataset," https://www.twentybn.com/datasets/jester, 2017.

[11] J. David, "Correlation and Convolution", Class Notes for CMSC 426, 2005.