Hand Movement Identification Using Single-Stream Spatial Convolutional Neural Networks

Aldi Sidik Permana
Department of Informatics
Universitas Jenderal Achmad Yani
Cimahi, Indonesia
sidika175@gmail.com

Esmeralda Contessa Djamal*
Department of Informatics
Universitas Jenderal Achmad Yani
Cimahi, Indonesia
*esmeralda.contessa@lecture.unjani.ac.id

Fikri Nugraha
Department of Informatics
Universitas Jenderal Achmad Yani
Cimahi, Indonesia
fikri.nugraha@student.unjani.ac.id

Fatan Kasyidi
Department of Informatics
Universitas Jenderal Achmad Yani
Cimahi, Indonesia
fatan.kasyidi@lecture.unjani.ac.id

Abstract— **Human-robot interaction can be through several ways, such as through device control, sounds, brain, and body, or hand gesture. There are two main issues: the ability to adapt to extreme settings and the number of frames processed concerning memory capabilities. Although it is necessary to be careful with the selection of the number of frames so as not to burden the memory, this paper proposed identifying hand gesture of video using Spatial Convolutional Neural Networks (CNN). The sequential image's spatial arrangement is extracted from the frames contained in the video so that each frame can be identified as part of one of the hand movements. The research used VGG16, as CNN architecture is concerned with the depth of learning where there are 13 layers of convolution and three layers of identification. Hand gestures can only be identified into four movements, namely 'right', 'left', 'grab', and 'phone'. Hand gesture identification on the video using Spatial CNN with an initial accuracy of 87.97%, then the second training increased to 98.05%. Accuracy was obtained after training using 5600 training data and 1120 test data, and the improvement occurred after manual noise reduction was performed.**

Keywords— hand gesture identification, video, spatial, convolutional neural networks

I. INTRODUCTION

The latest technological developments allow machines to understand what is instructed by humans in the form of gesture, sound, or just thinking in the brain. So that interaction occurs between humans and robots. The utilization of human and robot interactions is believed to facilitate all aspects of human life. Gesture, as one of the communications, is a collection of images that change each time sequence captured through video. Machines can be trained to learn movements such as hand gestures through machine learning so that it allows robots or computers to recognize what the gesture instructs people. The identification of this gesture is made through a computational model implemented in software.

Video is a collection of images that alternates in sequence in a certain period or better known in units of frames per second (fps). In the video, especially video with individual objects, there is usually a change or gesture between the first frame and the next frames. The development of technology that is overgrowing at this time, especially Deep Learning, allows machines to learn and remember a pattern of movement as certain symbols that give meaning.

The development of computing for machine learning now is using Deep Learning, which enables the identification of images with high accuracy without being extracted first [1]. Better computing must be balanced with better computing devices too. One method of Deep Learning is Convolutional Neural Networks (CNN). CNN is better known as a method for image identification but burdensome hardware performance. Hence, it requires significant resources for maximum performance, so it is necessary to use a Graphics Processing Unit (GPU) [2].

Convolutional Neural Network has a variety of configurations that are adjusted to the dataset and identification used [3]. In identifying, CNN combines with previous methods such as Backpropagation [4] and Support Vector Machine [5]. Use is needed to get weight updates from each identification in the Convolutional Neural Network learning layer [6].

In processing sequential images or videos using CNN, the video cannot be processed entirely but must use input methods such as spatial [7], temporal [8], or both spatial-temporal [9]. Before the feature input method, the video is extracted into a set of frames. Then the frame is processed into CNN using one of three feature input methods.

Some previous studies used CNN in the identification of hand gestures [10], recognition of human action [11], recognition of sign language [12] [13], classification of classical Indian dance movements [14]. Using hand gestures can communicate with computers as an input tool [9], hand motion video of the EgoGesture dataset- which is an egocentric dataset [15].

Other studies recognized hand gestures using the CNN [16]. The research identified hand poses with 96.36% of training accuracy and 95% testing data with grayscale images. But, the sequential image recognition only recognized static hand gestures of the CNN application. Other studies identified hand movements with the Convolutional Neural Network for human and robot space interaction using hand movements [17]. However, this research cannot be done in real-time other than that the data used is static data, given the limitations of the method in sequential image processing. The previous study of hand posture recognition used CNN, which was supported by Gabor filter as a feature extraction method [18]. There were two experimental scenarios carried out, particularly the introduction of hand gestures using CNN from raw images without Gabor filters and from images that had been processed with Gabor filters. Each produced an accuracy of 95.0% and 99.2%. But the study used are png-shaped image data, so that can not be in real-time. Also, recognized images are obtained from a mixture of Kinect RGB cameras and notebook cameras, so they made problems in the quality and quantity of the dataset.

Other studies on dynamic hand gesture recognition using Neural Network and Stochastic gradient-based optimizer that produce accuracy for two conditions of the hand gesture image dataset are good lightning and bad lightning [19]. Each provided an accuracy of 85% and 71.3%. The image used is a photo of the whole hand gestures with the body, so it requires a longer pre-process because they have to segment beforehand to get a picture of the hand gesture. Besides, the experiments conducted focused on hand gestures using histogram-based Motion History Image (HMI) for four gestures, namely swipe left, swipe right, swipe down, and shrink. Classification methods are needed that can handle large datasets to improve accuracy even better.

In other studies about hand gesture recognition using Adapted Convolutional Neural Network (ADCNN) produced an accuracy of 99.73% by relying on data augmentation to get a useful dataset to improve its recognition capabilities [20]. The data used is a hand gesture image dataset, so it is necessary to experiment with a dataset that can handle the introduction of real-time tasks.

This research proposed identifying motion on video using spatial streams CNN in real-time. The spatial stream is a method of how videos are determined by inserting each frame individually into CNN to be extracted. CNN used 13 convolution layers and three fully connected layers. The number of layers being built is intended to match the CNN architecture used, VGG16. The movement was identified with four classes, namely "left", "right", "phone" and "grab".

II. Methods

A. Hand Gesture Identification

The model begins with the stage of data acquisition obtained through a recording scenario with each movement having 70 motion video data with a size of 720 x 480 20 fps. The video data is then extracted to get the frames contained in the video. Then each frame obtained is entered into pre-processing. The movement that occurs in the video in each frame is a feature in video identification using the CNN spatial stream. The identification results are used to increase human, and robot interaction in the form of movements captured through a video camera like in Fig. 1.

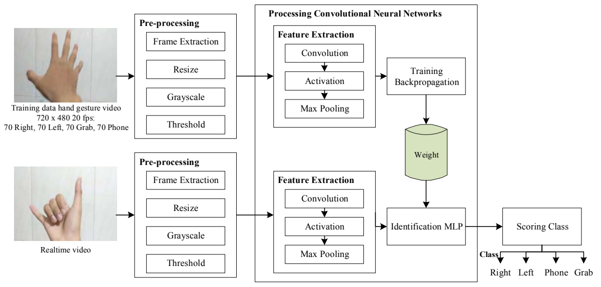


Fig. 1. Hand Gesture Identification Model

The identification system begins with learning using training data with a total of 5600 hand gesture frame data with 1400 hand gesture data frames for each. There are two main processes: pre-processing data and CNN process. The pre-processing is carried out to equalize the characteristics of the data to be processed on CNN, consisting of four stages, namely frame extraction, grayscale, threshold, and resize. After the data goes through the pre-process, then it enters the CNN process to get features and learn from the features obtained using Backpropagation, then the learning results are stored and used in real-time identification of hand gestures on the video.

B. Spatial Stream

A spatial stream is a network that functions to conduct Spatial stream is a network that functions to divide a single unit of video into spatial or frames so that it can be processed on CNN as shown in Fig 2, each frame in the video is processed sequentially to look for features of the frame and identify it [21].

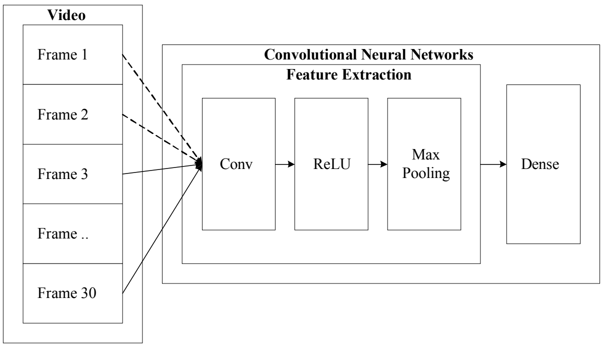


Fig. 2. Spatial Stream Process

C. Visual Geometry Group 16

Visual Geometry Group 16 (VGG16) is a CNN architecture that refers to the depth of layers in obtaining features for more in-depth learning. VGG16 architecture, as shown in Fig. 3. Where the more convolution process, the higher features you get, while the more fully connected layers process, the more weight values are calculated for deeper learning.

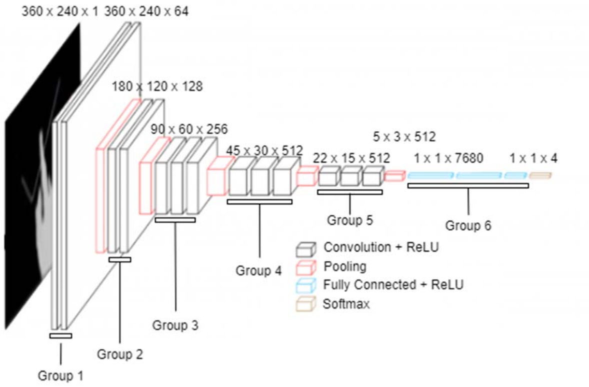


Fig. 3. VGG 16 Architecture

D. Convolutional Neural Network

Convolutional Neural Network is one of the Deep Learning methods that is often used to detect or recognize images. Convolutional Neural Networks can know features well in images through convolution and identification processes consistently [12]. In its architecture, CNN consists of three layers, namely, feature extraction layer, activation layer, and identification layer.

1) Feature Extraction

The feature extraction layer is a layer that converts an image into a feature in the form of numbers that represent that image. The feature extraction layer consists of convolution, activation, and pooling stages.

a) Convolution

The convolution stage is the primary process of CNN, known as a feature detector. The kernel filter moves to all areas of the input image. It performs dot product operations at each position between the kernel filter and the input image portion [22], to produce images with different depths, known as feature maps. The convolution operation can be using (1).

$$FM_{(i_l,j_l)}^{(l,m_l)} = f\left(\sum_{r_l=0}^{k_{h-1}} \sum_{c_l=0}^{k_{w-1}} C_{(r_l,c_l)}^{(l,m_l)} * FM_{((r_l+i_{l-1}),(c_l+j_{l-1}))}^{(l-1)}\right) \tag{1}$$

Where:

FM = feature map
l = index layer/input
m_l = index map of l-layer
i_l = l index row of a feature map layer
j_l = l index column of the feature map layer
f = function
k_w = length filter
k_h = height filter
r_l = l index length filter layer
c_l = l index height filter layer
C = filter convolution

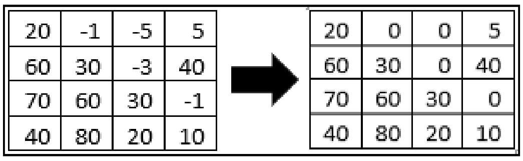
b) Activation

The activation layer is used to eliminate negative values in the image so that the feature extraction process is faster because then the negative value that becomes zero does not require a long time to be processed.

$$y(x) = \begin{cases} x, & if\ x \geq 1 \\ 0, & if\ x < 0 \end{cases} \tag{2}$$

Where:

x = Input pixel like Fig. 4



20	-1	-5	5
60	30	-3	40
70	60	30	-1
40	80	20	10

20	0	0	5
60	30	0	40
70	60	30	0
40	80	20	10

Fig. 4. Function Activation ReLU

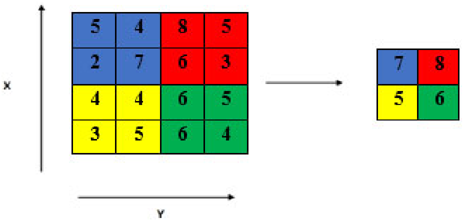
c) Pooling

The pooling step serves to reduce the spatial size and number of parameters in the network as well as speed up computing and control the occurrence of overfitting and produce feature patterns. In the Max Pooling process, it takes the most considerable value of the input layer from the activation function, as shown in Fig. 5.

$$w_t = \frac{w_{(t-1)}-F}{s+1} \tag{3}$$

Where :

W_t = size of wide image pooling
S = *Stride*
F = *filter*



5	4	8	5
2	7	6	3
4	4	6	5
3	5	6	4

7	8
5	6

Fig. 5. Max Pooling

2) Identification

The identification layer uses the Backpropagation Neural Networks method [10]. Input is a feature vector of outputs in the activation and pooling convolution processes which were previously transformed into a dimensional array called flatten. Then the result of Flatten is entered into the Softmax activation function to determine the class by determining the probability value of all classes using (4).

$$Softmax(y_k) = \frac{e^{y_k}}{\sum_{j=1}^{K} e^{y_k}} \tag{4}$$

E. Dataset

The dataset is obtained through recording using a camera with a size of 720 x 480 pixels 20 fps. The record was under normal light conditions and without using anything. The number of videos is 280 videos divided into four classes so that each class has 70 videos.

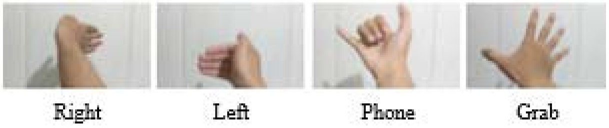


Fig. 6. Datasets

III. Result and Discussion

Data frames have been extracted from the gesture videos as many as 5600 data consists of four classes, each of which contains 1400 data. The video was grouped according to the recording scenario. This data is used for training and testing the CNN model. The data split into two parts, 80% of training data and 20% of test data so that each amount to 4480 and 1120 data.

The experiment used the Keras library with the Python3 programming that was running in the Google Colab platform. It performed a used learning rate of 0.01 until 100 epoch. Meanwhile, the weights of learning were updated by optimizer models such as SGD and Adam, to compare them. Performance is indicated by accuracy and gap between output value and target or called Losses as shown in Table I.

TABLE I. Comparison Optimizer Performance

No	Optimizer	Training Data		Testing Data	
		Losses	Accuracy (%)	Losses	Accuracy (%)
1	SGD	0,3306	90.08	0,3886	87.97
	Adam	0.2292	91.05	0.2324	88.03
2	SGD + noise reduction	0,0015	100.00	0,1605	98.05
	Adam + noise reduction	0,0099	100.00	0,1735	97.16

Table I shows that there was a significant increase in the second training, due to manual noise reduction by removing empty frames where there were no hand objects in the frame. The training used SGD and Adam optimization produces good accuracy where SGD produced 98% accuracy and Adam 97%, as shown in Fig. 7. It was obtained because of manual noise reduction. Before making the noise reduction manually, the results obtained are 87% for SGD and 88% for Adam.

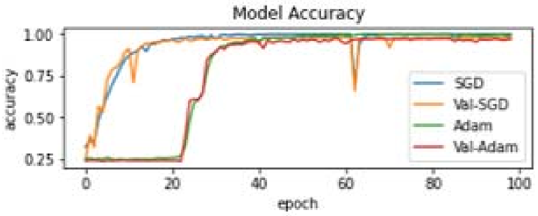


Fig. 7. Accuracy SGD and Adam optimizer

Besides, it can also be seen in Fig 8 that the results of the two optimization models are good, even though the Adam optimization model looks stack at the beginning, but after that, it increases exponentially. It proves that Adam optimizer performance in learning is faster than the SGD.

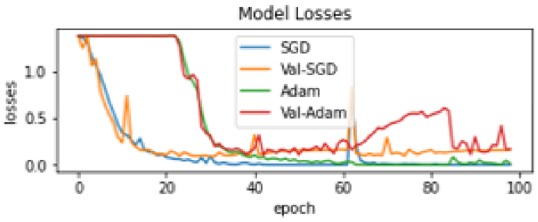
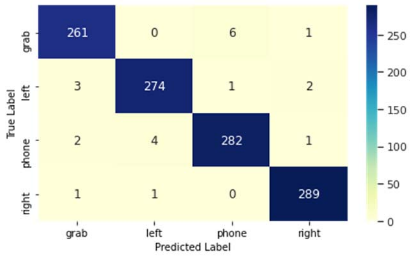


Fig. 8. Losses SGD and Adam optimizer

The confusion matrix can be seen in Fig 9. The row labels on the side is the actual class, while the column labels below are the predicted classes.



	grab	left	phone	right
grab	261	0	6	1
left	3	274	1	2
phone	2	4	282	1
right	1	1	0	289

Fig. 9. Confusion Matrix SGD Optimizer

At the time of testing with 1120 data, the most errors were found in the grab and phone class, where there were seven cases not identified for the correct class. However, there is a class that can be adequately recognized, particularly the right class with the fewest prediction errors.

TABLE II. PRECISION, RECALL, F1-SCORE AND SUPPORT

	Precision (%)	Recall (%)	F1-Score (%)	Support
Right	97	97	97	268
Left	96	98	97	280
Phone	96	96	96	289
Grab	99	97	98	291

Table II shows the precision, recall, and F1 score in each class that above 96%. The best precision obtained in “grab” class testing is 98% recognition, while the smallest is 96% “phone” and “left” class because a lot of “phone” and “left” class data is identified against other classes.

IV. CONCLUSION

The results showed that the training using the VGG16 architecture with SGD and Adam optimization went well. The VGG16 architecture in this study is proven to have an adequate level of accuracy even though it uses the VGG16 configuration in general.

This study conducted an experiment by manually reducing noise by eliminating empty frames, which significantly improved learning. It was evidenced by an increase in accuracy from 88% to 98%. We can also find that the learning process using Adam gradually and steadily strengthens well even at the start of the stack but increases exponentially. The models are due to adaptive updating and continuous weighing of error values. Using SGD and Adam optimizer, even in standard configurations without over-turning, can solve overfitting problems.

Further research can be done by making variations on the spatial single-stream model, by experimenting with changes in overlapping variations to get a continuous connection between frames using traditional overlapping or by using Recurrent Neural Networks deep learning.

ACKNOWLEDGMENT

Thanks to the Ministry of Technology Research for the financial support provided for this research in the “Penelitian Terapan Unggulan Perguruan Tinggi” 2020 grant with number B/87/E3/RA.00/2020.

REFERENCES

[1] O. K. Oyedotun and A. Khashman, “Deep learning in vision-based static hand gesture recognition,” *Neural Computing and Applications*, vol. 28, no. 12, pp. 3941–3951, 2017.

[2] Z. Zhao, L. Song, R. Xie, and X. Yang, “GPU accelerated high-quality video/image super-resolution,” *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, pp. 2–5, 2016.

[3] W. Zhao, S. Du, and W. J. Emery, “Object-Based Convolutional Neural Network for High-Resolution Imagery Classification,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 7, pp. 3386–3396, 2017.

[4] J. E. Zafra and R. D. Hernández, “Comparison between Backpropagation and CNN for the Recognition of Traffic Signs,” *International Journal of Applied Engineering Research*, vol. 12, no. 17, pp. 6814–6820, 2017.

[5] A. F. Agarap, “An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification,” *Computer Science, Computer Vision and Pattern Recognition*, pp. 5–8, 2017.

[6] J. Zhang, K. Shao, and X. Luo, “Small sample image recognition using improved Convolutional Neural Network,” *Journal of Visual Communication and Image Representation*, vol. 55, no. July, pp. 640–647, 2018.

[7] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, “Spatial as deep: Spatial CNN for traffic scene understanding,” *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pp. 7276–7283, 2018.

[8] L. Jing, X. Yang, and Y. Tian, “Video you only look once: Overall temporal convolutions for action recognition,” *Journal of Visual Communication and Image Representation*, vol. 52, no. January, pp. 58–65, 2018.

[9] S. Yan, Y. Xiong, and D. Lin, “Spatial-temporal graph convolutional networks for skeleton-based action recognition,” *32nd AAAI*

Conference on Artificial Intelligence, AAAI 2018, pp. 7444–7452, 2018.

[10] A. A. Alani, G. Cosma, A. Taherkhani, and T. M. McGinnity, "Hand gesture recognition using an adapted convolutional neural network with data augmentation," *2018 4th International Conference on Information Management, ICIM 2018*, pp. 5–12, 2018.

[11] H.-H. Pham, L. Khoudour, A. Crouzil, P. Zegers, and S. A. Velastin, "Video-based human action recognition using deep learning: a review," *Archivo preprint*, pp. 1–34, 2015.

[12] M. R. Islam, U. K. Mitu, R. A. Bhuiyan, and J. Shin, "Hand gesture feature extraction using deep convolutional neural network for recognizing American sign language," *2018 4th International Conference on Frontiers of Signal Processing, ICFSP 2018*, no. September, pp. 115–119, 2018.

[13] F. Nugraha and E. C. Djamal, "Video Recognition of American Sign Language Using Two-Stream Convolution Neural Networks," *International Conference on Electrical Engineering and Informatics*, 2019.

[14] P. V. V. Kishore *et al.*, "Indian Classical Dance Action Identification and Classification with Convolutional Neural Networks," *Advances in Multimedia*, vol. 2018, January 2018.

[15] C. Cao, Y. Zhang, Y. Wu, H. Lu, and J. Cheng, "Egocentric Gesture Recognition Using Recurrent 3D Convolutional Neural Networks with Spatiotemporal Transformer Modules," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-Octob, pp. 3783–3791, 2017.

[16] R. Patel, J. Dhakad, K. Desai, T. Gupta, and S. Correia, "Hand gesture recognition system using convolutional neural networks," *2018 4th International Conference on Computing Communication and Automation, ICCCA 2018*, pp. 1–6, 2018.

[17] G. W. B, L. Ren, and J. S. Dai, "Static Hand Gesture Recognition with Parallel CNNs for Space Human-Robot Interaction," *International Conference on Intelligent Robotics and Applications*, vol. 1, no. October, pp. 71–83, 2017.

[18] H. Qassim, A. Verma, and D. Feinzimer, "Compressed residual-VGG16 CNN model for big data places image recognition," *2018 IEEE 8th Annual Computing and Communication Workshop and Conference, CCWC 2018*, vol. 2018-Janua, pp. 169–175, 2018.

[19] M. I. N. P. Munasinghe, "Dynamic Hand Gesture Recognition Using Computer Vision and Neural Networks," *2018 3rd International Conference for Convergence in Technology, I2CT 2018*, January, 2018.

[20] Y. Hu, M. Lu, and X. Lu, "Spatial-Temporal Fusion Convolutional Neural Network for Simulated Driving Behavior Recognition," *2018 15th International Conference on Control, Automation, Robotics and Vision, ICARCV 2018*, pp. 1271–1277, 2018.

[21] M. Dyrmann, H. Karstoft, and H. S. Midtiby, "Plant species classification using deep convolutional neural network," *Biosystems Engineering*, vol. 151, no. 2005, pp. 72–80, 2016.

[22] Y. Wang, M. Long, J. Wang, and P. S. Yu, "Spatiotemporal pyramid network for video action recognition," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2097–2106, 2017.