*Vietnam Journal of Science and Technology* ***58*** *(4) (2020) 514-523*

doi:10.15625/2525-2518/58/4/14742

# TWO-STREAM CONVOLUTIONAL NETWORK FOR DYNAMIC HAND GESTURE RECOGNITION USING CONVOLUTIONAL LONG SHORT-TERM MEMORY NETWORKS

**Phat Nguyen Huu[*], Tien Luong Ngoc**

*School of Electronics and Telecommunications, Hanoi University of Science and Technology,*
*1 Dai Co Viet, Hai Ba Trung, Ha Noi, Viet Nam*

[*]Email: *phat.nguyenhuu@hust.edu.vn*

Received: 29 December 2019; Accepted for publication: 26 June 2020

**Abstract.** Action and gesture recognition provides important information for interaction between human and devices that monitors living, healthcare facilities or entertainment activities in smart homes. Recent years, there are many learning machine models studying to recognize human action and gesture. In this paper, we propose a dynamic hand gesture recognition system based on two stream-convolution network (ConvNet) architecture. Besides, we also modify the method to enhance its performance that is suitable for indoor application. Our contribution is improvement of two stream ConvNet to achieve better performance. We use MobileNet-V2 as an extractor since it has less number of parameters and volume than other convolution networks. The results show that the proposal model improves execution speed and memory resource usage comparing to existing models.

*Keywords:* two stream-ConvNet, spatial stream, temporal stream, dynamic hand gesture recognition, optical flow.

*Classification numbers**:*** 4.2.3, 4.5.3, 4.7.4.

## 1. INTRODUCTION

Dynamic hand gesture recognition is a difficult task in computer vision. There are many researches to propose hand gesture recognition models [1,2]. The authors of [1] proposed a dynamic hand gesture method by combining both deep convolutional neural network (CNN) and Long Short-Term Memory (LSTM). The input data are sequences of 3D hand positions and velocities acquired from infrared sensors called Leap Motion. In the study of [3], the author presented a hand gesture recognition method using Microsoft’s Kinect in real-time. Their system includes detecting and recognizing hand gestures via combining shape, local auto-correlation information and multi-class support vector machine (SVM). The authors of [4] utilized skeleton as input data for model network that is similar to proposed architecture in [1]. The model CNN + LSTM is also used in [5], so that the authors put stacked optical flow into network. In [6, 7], the authors used 3D-CNN architecture to learn spatio-temporal information for hand gesture recognition model. Other studies [1, 8] have applied deep learning model to exploit information from RGB image frames to recognize action in videos; however, those methods still have several disadvantages.

Comparing to image classification tasks indicated in [9, 10] which only use to extract information from RGB images, gesture recognition problem exploits not only information about scene per frames but also temporal aspect. Specifically, each gesture gives ambient related details and previous frames.

Our main aim in this paper is utilizing the state-of-the-art deep learning techniques such as CNN and LSTM based on the newest two-stream ConvNet architecture to recognize dynamic hand gesture in video [11]. The model in study [12] exploited  spatial information as well as temporal information to create feature vectors. Those features are put into two classifiers and fuse by class score fusion block. In this model, the first stream exploits information based on RGB frames and recognize gesture through scenes and second stream utilizes stacked optical flow as input. There are still limited result of this model, because the prediction based on separate frames.

There was an improvement in [11, 13] by applying LSTM since the authors take them into LSTM network after fusing. In the theory, a gesture is recognized based on not only gesture scenes but also relationship among frames. In [11], the authors applied Resnet-101 to extract feature of RGB images as well as stacked optical flow images. However, it did not achieve good performance about execution time and memory resources because of large parameters of Resnet-101.

In this paper, we improve the two-stream ConvNet model to reduce computation time as well as memory resources. The approach is suitable for deploying algorithm into embedded devices instead of performing on expensive computers or cloud-based process.

The rest of the article is organized as follows: A brief review about gesture recognition is presented in section 1. The proposed architecture network is discussed in section 2. Section 3 presents the experimental results. Finally, the conclusions and discussion are given in section 4.

## 2. METHODS

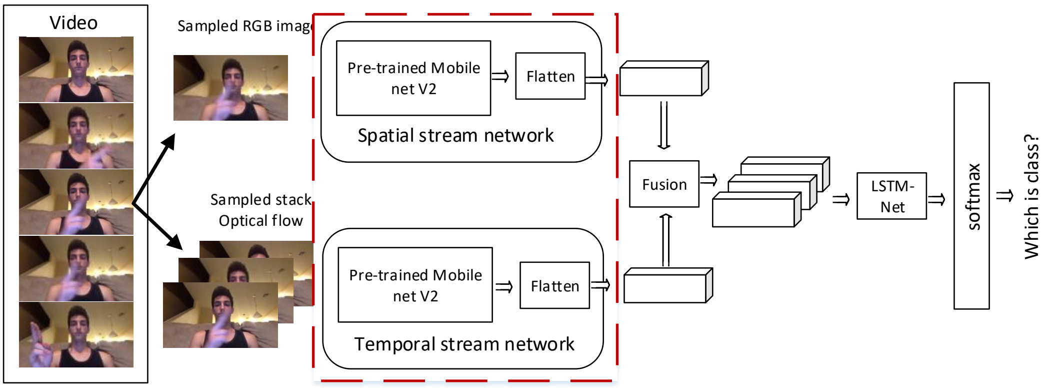

*Figure 1*. Proposed two-stream ConvNet architecture.

In this paper, we propose the model based on two-stream ConvNet and LSTM for dynamic hand gesture recognition in video as shown in Fig. 1 based on [11]. First, we capture RGB image frames and stacked optical flow images into spatial and temporal stream network. We then use them in both networks to train. The feature maps are output by the spatial and temporal stream network. Finally, ConvLSTM is deployed to learn long-term spatiotemporal dependencies. Our contribution is improvement of two stream ConvNet to achieve better performance by using MobileNet-V2 as an extractor that has less number of parameters as well as calculated volume than other state-of-the-art convolution networks.

### 2.1. Feature extraction

The handcrafted feature such as the improved dense trajectories (IDT), and three-dimensional scale-invariant feature transform (SIFT-3D) are constructed and get good performance for activity recognition. However, deep learning networks for activity recognition are gradually occupying the dominant position with the growing capacity of CNN. To solve the problem, the two-stream method is performed for many motion recognition solutions based on RGB and optical stream. Many studies have introduced optical stream for raw RGB frames and achieved considerable improvement in performance in recent years [11,14].

In this architecture, the RGB and optical flow are fed into an extractor block to get feature map. There were many studies to apply CNN in classification task [15, 16]. In [14], the authors designed a two-stream ConvNet architecture using Resnet-101 in extracting feature. Specifically, it is Winner of ILSVRC 2015 (Image Classification, Localization, and Detection). Resnet-101 has an architecture similar to a previous famous network. However, Resnet-101 has many layers that lead to the complex network. It means that the number of parameters as well as calculated volume is high since program execution time and memory resources are large.

MobileNet that published later than Resnet-101 is proposed by authors from Google in 2017. In this network, the authors used a calculus convolution method called “Depthwise Separable Convolution” to reduce size model and calculation complexity. As a result, the model is useful when implemented in mobile and embedded devices. Since we proposed two-stream ConvNet (as shown in Fig. 1), we use MobileNet as an extractor in both stream. Metrics of convolution networks are shown in Table 1.

*Table 1*. Comparison of metrics of convolution networks.

| No | Network | Accuracy on ImageNet | Number of parameter | Size | Depth |
|---|---|---|---|---|---|
| 1 | VGG-16 [17] | 0.901 | 138,357,544 | 528 MB | 23 |
| 2 | VGG-19 [18] | 0.90 | 143,667,240 | 549 MB | 26 |
| 3 | Inception-V3 [19] | 0.937 | 23,581,784 | 92 MB | 159 |
| 4 | Resnet-101 [16] | 0.938 | 44,675,560 | 171 MB | 101 |
| 5 | Mobilenet-V2[20] | 0.901 | 3,538,984 | 14 MB | 88 |
| 6 | Densenet201 [13] | 0.936 | 20,242,984 | 80 MB | 201 |

### 2.2. CNN and RNN

In [12], the proposed system is two-stream ConvNet. The proposal consists of spatial and temporal stream using RGB and stacking optical flow images as input. However, the proposal has not yet exploited the motion characteristics of object. It means that both RGB and stacked optical flow images are extracted features by CNN to get feature maps followed by a classifier block. In other words, each gesture is only recognized through separating frames since there is not relationship among them. In order to get better performance than the model in [12] the researches in [11, 14] added the LSTM component to memorize the previous information. Specifically, the authors showed an architecture that is combined of CNN and LSTM to perform action recognition task. They did not ignore information gathered from frames since gestures and actions are recognized based on starting frames. Therefore, we use this method to get the best performance in our proposal as shown in Fig. 2.

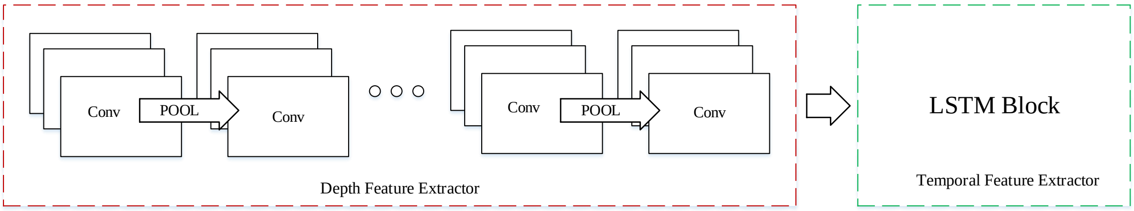

*Figure 2.* The CNN+LSTM architecture.

### 2.3. Two-stream ConvNet

The indicated model in [12] demonstrated by stacking optical flow that can get a high performance in case of limiting data. Recently, two-stream ConvNet architecture becomes popular and is one of the best methods for action and gesture recognition. In this paper, we use both two-stream ConvNet proposed design in [12] and LSTM, as follows.

In our research, we improve the feature extractor in both stream by using Mobilenet-V2 instead of ResNet-101 [11] used.  Fig. 1 show our model with highlight component as our proposal. Two-stream ConvNet is built based on combining both spatial and temporal streams. At first stream, we take RGB images as input and other stream is their stacked optical flow. The input data have to go through a block called extractor which is improved in our study. Using the ConvNet for extractor leads to a better model. Huge parameters and calculation complexity of deep model can lead to low speed execution and take up many memory resources. The authors of [20] showed the number of Mobilenet much smaller than convolution network which are used in [21, 22] whereas there was a significant difference in term accuracy.

The video frames are put into network as shown in Fig. 3. The RGB images and stacked optical flow are yellow and green rectangles, respectively. They are injected into spatial and temporal stream. We then utilize an extractor that belongs to our proposal to get information from images. The receiving feature maps are flattened and fused by fusion block to get a feature vector. This vector is an input for LSTM block.

### 2.4. LSTM

The purpose of the LSTM block is to exploit the information among the frames. The variations among frames within a video may contain additional information that could be useful in determining the human action. One of the most straightforward ways to incorporate and exploit sequences of inputs is RNN. LSTM networks are a modified version of RNN which makes it easier to remember past data in memory. Therefore, the gradient problem of RNN is resolved. LSTM is well suited to classify, process, and predict unknown duration. In this work, we build LSTM block with two layers as shown in Fig. 4.

### 2.5 Fusion

As mentioned above, the two-stream ConvNet recognizes dynamic hand gesture through exploiting information from RGB and stacked optical flow images. Therefore, the feature vectors are fused as input for next block in both stream. There are four types of methods to fuse the feature maps, namely: Sum fusion, Max fusion, concatenation fusion, and Conv fusion as presented in [12]. Conv fusion has the best performance and Max fusion has the worst performance. We adopt the Sum fusion since this strategy has less parameters to compute and the performance is nearly as good as the Conv fusion in our experiment.

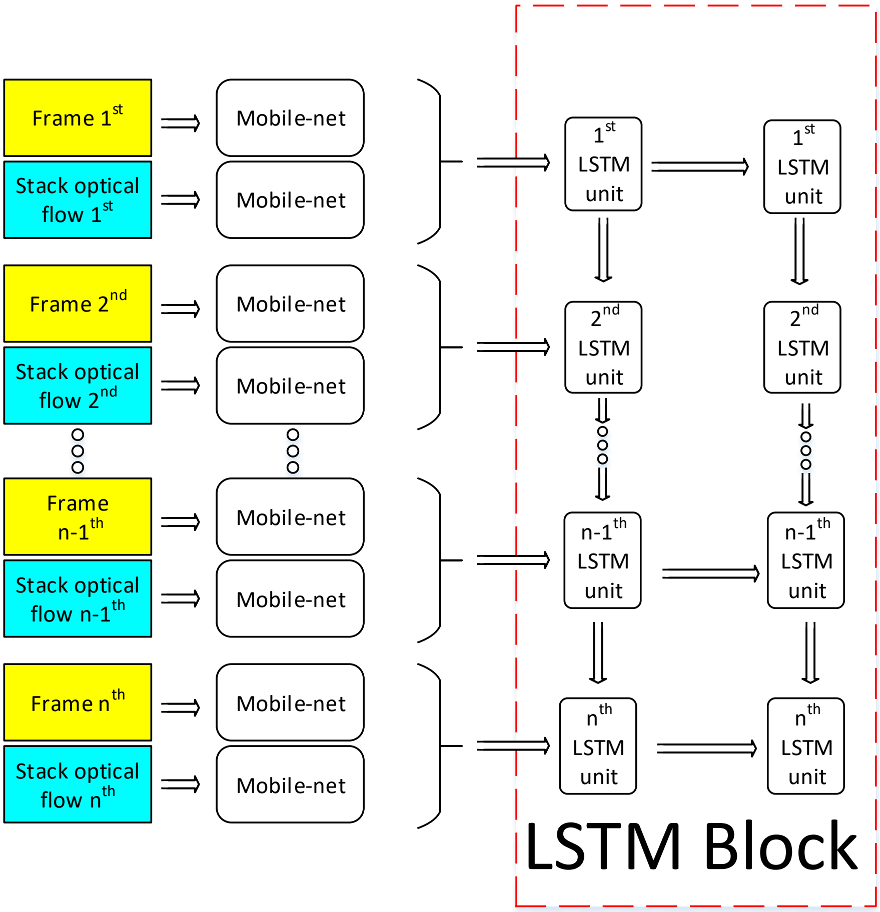

*Figure 3*. Description of the processing flow in the proposed model.

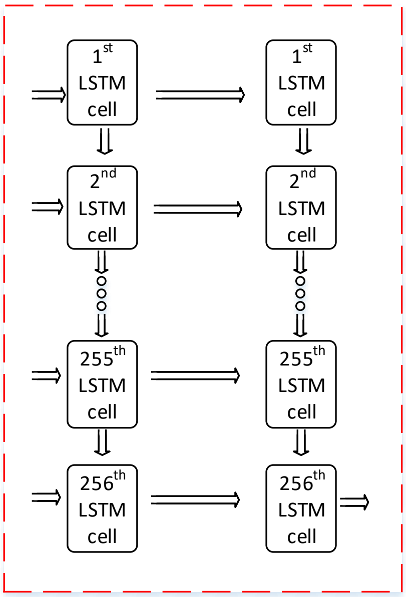

*Figure 4*. LSTM block structure.

## 3. EXPERIMENTS

### 3.1. Dataset

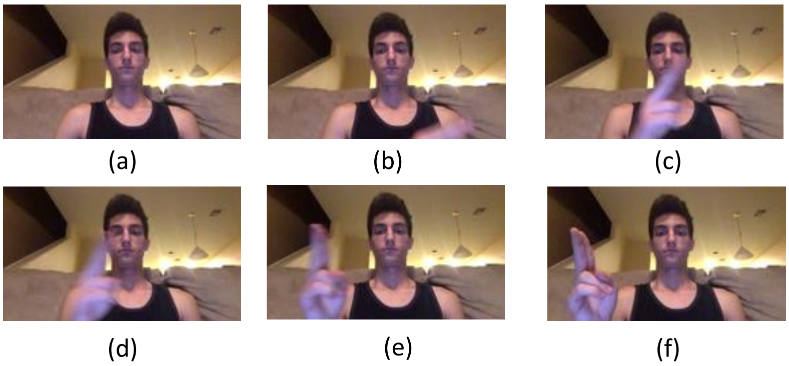

*Figure 5*. Description of several RGB images from Jester Dataset. (a) 1st frame, (b) 12th frame, (c) 20th frame, (d) 26th frame, (e) 30th frame, and (f) 36th frame.

The dynamic hand gesture 6/25 20BN-jester Dataset V1 [23] was selected as the database that is one of few dynamic hand gesture datasets as shown in Figs. 5, 6, and 7. To get the optical flow image, there are two common kinds of algorithm for optical flow extracting Brox and TV-L1. In this work, we select TV-L1 to create optical flow that is slightly better than Brox. We use both RGB and optical flow images as input to two-stream ConvNets.

*Table 2*. Class name and the number of data per class.

| Class | Swiping Down | Swiping Right | Swiping Left | Sliding Two Fingers Up | Sliding Two Fingers Right | Stop Sign |
|---|---|---|---|---|---|---|
| Number | 240 | 240 | 240 | 240 | 240 | 240 |

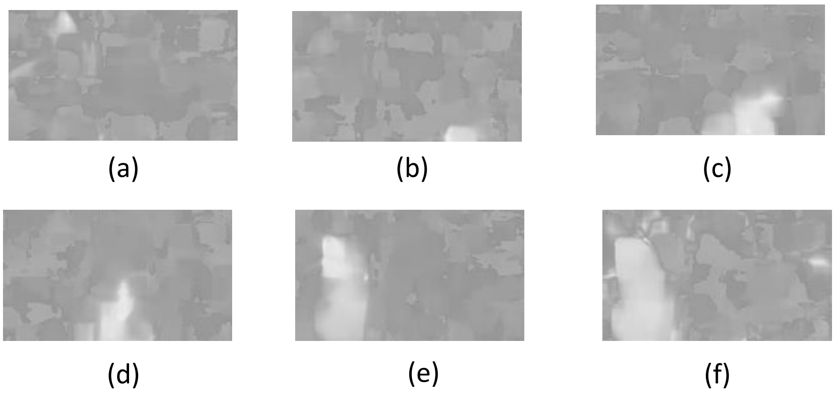

*Figure 6*. Description of several images of stacked optical flows (a) 1st frame, (b) 12th frame, (c) 20th frame, (d) 26th frame, (e) 30th frame, (f) 36th frame.

The collected dataset is divided into 60 %, 20 %, 20 % for training, validation and testing, respectively with the number of class and class name as shown in Tab. 2.

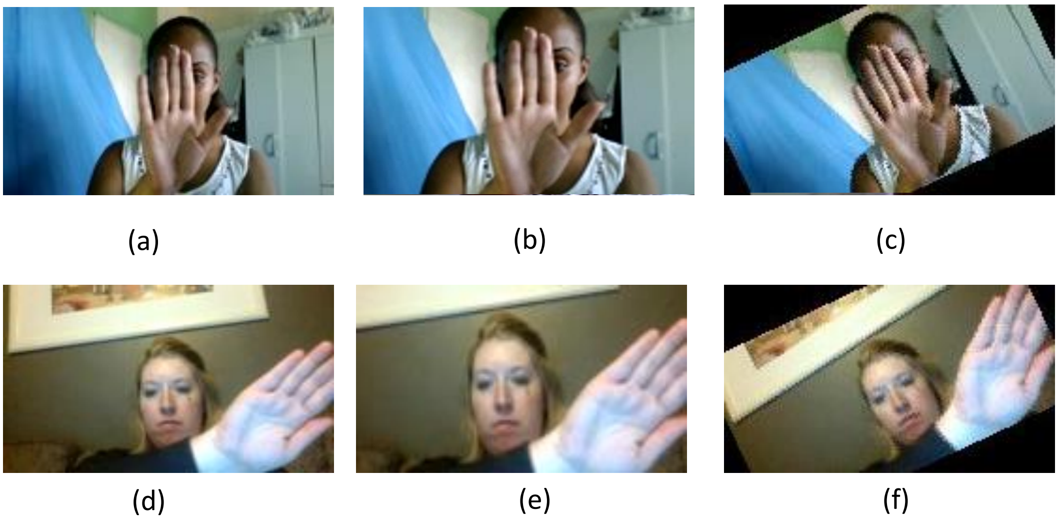

*Figure 7*. Description of several images after augmentation: (a) and (d) original images, (b) and (e) Zoom augmentation, (c) and (f) Rotation augmentation.

### 3.2. Data normalization

Data normalization is one of the most important techniques in machine learning. In this paper, we normalize the input images into [0, 1]. We use the standardized method according to the formula:

$$x_i^{'} = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}, \tag{1}$$

where $x_i$ and $x_i^{'}$ in turn are the initial characteristic values and the standardized characteristic values, respectively. $\min(x_i)$ and $\max(x_i)$ are the maximum and minimum value of the i$^{th}$ characteristic.

### 3.3. Training

We train model with 50 epochs, mini-batch size = 16, Adam optimizer with parameter of lr = 0.01, p = 0.95. Input images are resized 227×227 in accordance with Pre-trained MobileNet. We chose timesteps = 32 with LSTM since the number of frame per gesture is from 29 to 32. We use “model checkpoint” in Keras library to save the model weights for training process since there are accuracy improvement comparing with previous epoch. The system will save model weight when accuracy is improved. During training process, we use several augmentation methods (Rotation, Zooming) in order to create data diversity. Therefore, the number of data after augmentation are 864 video for training process. The augmentation method helps to avoid the over-fitting problem.

### 3.4. Results

Figure 8 compares the accuracy and loss value of the proposal model based on the training and evaluation dataset. Figure 8 (b) shows that speed of loss function is pretty good and stable.

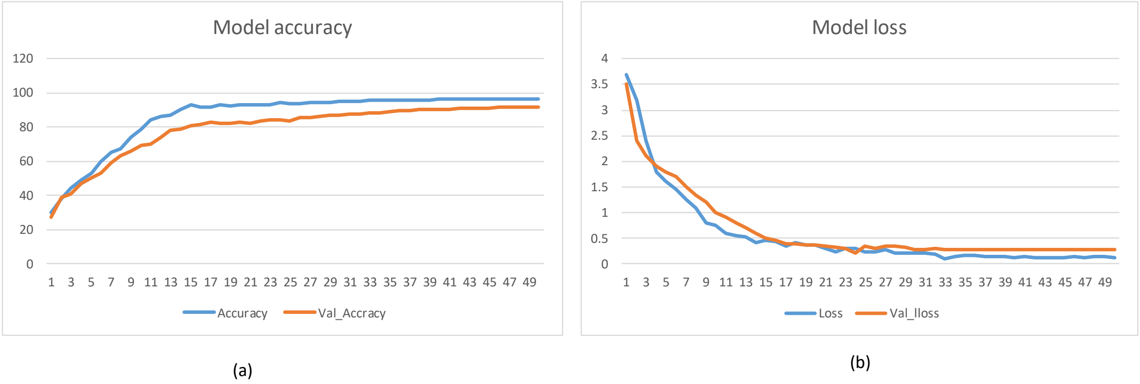

*Figure 8*. (a) The accuracy model for train and validation dataset. (b) The model loss for train and validation dataset.

*Table 3*. Comparison of results among other methods.

| Method | Number-of parameters | Size (Megabyte) | Accuracy (%) | Average execution time (s/gesture) |
|---|---|---|---|---|
| MobileNet-V1 | 22.4 | 196 | 90.86 | 0.873 |
| **Our proposal** | **24.4** | **249** | **91.25** | **0.792** |
| VGG-16 | 37.5 | 149 | 92.54 | 2.512 |
| InceptionV3 | 50.2 M | 251 | 91.86 | 0.890 |
| Exception | 79.8 | 515 | 93.37 | 1.625 |
| Resnet 101 | 135.8 M | 931 | 92.39 | 2.791 |

From Tab. 3, it is clear that there is not much accuracy difference between our proposed model and existing models whereas size and time execution model are of great difference. Specifically, the time execution and size in the latest architecture using Resnet-101 are 931 MB (Megabyte) and 2.791 (seconds/ gesture) while our proposal has 249 MB and 0.792 (seconds/gesture). Therefore, our proposed model has less than about three times of size, and execution speed of one gesture is from 28 to 36 frames. The execution speed of a model usually depends on the number of parameters of the model. However, it also depends on the computational complexity that is determined by its architecture. By improving the architecture of the model, we will reduce its computational complexity and execution speed. This problem was demonstrated by the using MobileNet V2 network [11] comparing with its predecessors.

## 4. CONCLUSIONS

Generally, ConvNets with two-stream of the optical flow and original RGB have been widely used in activity as well as gesture recognition. The method of two-stream ConvNets and RNN has been proved competitively. In this paper, we researched existing approaches and proposed the model based on two-stream ConvNet architecture and MobileNet to improve its performance. Comparing with existing models, MobileNet is a lightweight network that uses depthwise separable convolution  to deepen the network and reduce its parameters. The result experiment demonstrated that the proposed model improves execution speed and memory resource. In the future, we will collect more gesture images that would increase the accuracy of detecting as well as tracking objects for real applications on wireless sensor networks.

***Acknowledgement.*** This research was supported by Hanoi University of Science and Technology and Ministry of Science and Technology under the project No. B2020-BKA-06, 103/QD-BGDT signed on 13/01/2020.

## REFERENCES

1. Naguri C. R. and Bunescu R. C. - Recognition of Dynamic Hand Gestures from 3D Motion Data Using LSTM and CNN Architectures, 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, 2017, pp. 1130-1133.
2. Wang H. and Schmid C. - Action Recognition with Improved Trajectories, IEEE International Conference on Computer Vision, Sydney, 2013, pp. 3551-3558.
3. Ngoc T.N. - Real-Time Hand Gesture Recognition, Journal of Computer and Cybernetics **29** (3) (2013) 232-240.
4. Lai K. and Yanushkevich S. N. - CNN+RNN Depth and Skeleton based Dynamic Hand Gesture Recognition, 24th International Conference on Pattern Recognition (ICPR), Beijing, 2018, pp. 3451-3456.
5. Ding X., Xu C. and Yan Q. - A Video Gesture Processing Method Based on Convolution and Long Short-Term Memory Network, IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), Chengdu, 2019, pp. 383-388.
6. Molchanov P., Gupta S., Kim K. and Kautz J. - Hand gesture recognition with 3D convolutional neural networks, IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Boston, 2015, pp. 1-7.
7. Zhang W. and Wang J. - Dynamic Hand Gesture Recognition Based on 3D Convolutional Neural Network Models, IEEE 16th International Conference on Networking, Sensing and Control (ICNSC), Banff, 2019, pp. 224-229.
8. Karpathy A., Toderici G., Shetty S., Leung T., Sukthankar R. and Fei-Fei L. - Large-Scale Video Classification with Convolutional Neural Networks, IEEE Conference on Computer Vision and Pattern Recognition, Columbus, 2014, pp. 1725-1732.
9. Krizhevsky A., Sutskever I., and Hinton G. E. - ImageNet Classification with Deep Convolutional Neural Networks, Advances in Neural Information Processing Systems **25** (2) (2012) 1-9.
10. Sultana F., Sufian A., and Dutta P. - Advancements in Image Classification using Convolutional Neural Network, 4th International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), Kolkata, 2018, pp. 122-129.
11. Ye W., Cheng J., Yang F. and Xu Y. - Two-Stream Convolutional Network for Improving Activity Recognition Using Convolutional Long Short-Term Memory Networks, IEEE Access **7** (2019) 67772-67780.

12. Simonyan K. and Zisserman A. - Two-stream convolutional networks for action recognition in videos, Proceeding of the Advances in Neural Information Processing Systems (NIPS), 2014, pp. 568-576.
13. Huang G., Liu Z., Maaten V. D. L., and Weinberger K. Q. - Densely Connected Convolutional Networks, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, 2017, pp. 2261-2269.
14. Ma C. Y., Chen M. H., Kira Z., AlRegib G. - TS-LSTM and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition, Signal Processing: Image Communication **71** (2019) 76-87.
15. Guo T., Dong J., Li H. and Gao Y. - Simple convolutional neural network on image classification, IEEE 2nd International Conf. on Big Data Analysis (ICBDA), Beijing, 2017, pp. 721-724.
16. He K., Zhang X., Ren S. and Sun J. - Deep Residual Learning for Image Recognition, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2016, pp. 770-778.
17. Simonyan K. and Zisserman A. - Very Deep Convolutional Networks for Large-Scale Image Recognition, 3rd International Conf. on Learning Representations (ICLR2015), USA, 2015, pp. 1-14.
18. Wang L., Xiong Y., Wang Z., Qiao Y., Lin D., Tang X., Gool L. V. - Temporal Segment Networks: Towards Good Practices for Deep Action Recognition, 14th European Conference, Amsterdam, 2016, pp. 20-36.
19. Szegedy C., Vanhoucke V., Ioffe S., Shlens J., Wojna Z. - Rethinking the Inception Architecture for Computer Vision, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2016, pp. 2818-2826.
20. Howard A. G., Zhu M., Chen B., and Kalenichenko D. - MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, Computer Vision and Pattern Recognition, 2017, pp. 1-10.
21. Sun L., Jia K., Yeung D. Y, Shi B. E. - Human Action Recognition Using Factorized Spatio-Temporal Convolutional Networks, IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 4597-4605.
22. Donahue J., Hendricks L. A, Rohrbach M., Venugopalan S., Guadarrama S., Saenko K., Darrell T. - Long-Term Recurrent Convolutional Networks for Visual Recognition and Description, IEEE Transactions on Pattern Analysis and Machine Intelligence **39** (4) (2017) 677-691.
23. Materzynska J., Berger G., Bax I. and Memisevic R. - The Jester Dataset: A Large-Scale Video Dataset of Human Gestures, 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea (South), 2019, pp. 2874-2882.