TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 18, No. 2, April 2020, pp. 761~769
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v18i2.14747

A robust method for VR-based hand gesture recognition using density-based CNN

Liliana[1], Ji-Hun Chae[2], Joon-Jae Lee[3], Byung-Gook Lee[4]
[1,4]Computer Engineering Dongseo University, South Korea
[1]Informatics Department Petra Christian University, Indonesia
[2,3]School of Computer Engineering Keimyung University, South Korea

Article Info

Article history:

Received Jul 3, 2019
Revised Jan 16, 2020
Accepted Feb 11, 2020

Keywords:

2D image gesture representation
Binary image learning
Density-based CNN
Hand gesture recognition
VR-based physical treatment

ABSTRACT

Many VR-based medical purposes applications have been developed to help patients with mobility decrease caused by accidents, diseases, or other injuries to do physical treatment efficiently. VR-based applications were considered more effective helper for individual physical treatment because of their low-cost equipment and flexibility in time and space, less assistance of a physical therapist. A challenge in developing a VR-based physical treatment was understanding the body part movement accurately and quickly. We proposed a robust pipeline to understanding hand motion accurately. We retrieved our data from movement sensors such as HTC vive and leap motion. Given a sequence position of palm, we represent our data as binary 2D images of gesture shape. Our dataset consisted of 14 kinds of hand gestures recommended by a physiotherapist. Given 33 3D points that were mapped into binary images as input, we trained our proposed density-based CNN. Our CNN model concerned with our input characteristics, having many 'blank block pixels', 'single-pixel thickness' shape and generated as a binary image. Pyramid kernel size applied on the feature extraction part and classification layer using softmax as loss function, have given 97.7% accuracy.

Corresponding Author:

Liliana,
Department of Informatics,
Petra Christian University,
Siwalankerto 121-131 street, Surabaya 60236, Indonesia.
Email: lilian@petra.ac.id

1. INTRODUCTION

Until recently, hand gesture recognition has been developed for various purposes, such as sign language understanding [1-5], Human-Computer Interaction [6], virtual environment interaction [4, 7-9] and controlling using robot [10-12]. Some applications using hand gestures as navigators to walk through a virtual environment [7, 8], virtual keyboard, controller appliances or device inside a certain space [10-12], controller robot surgery and used in medical purposes application such as physical treatment [6].

By taking advantage of current virtual reality (VR) technologies development, many applications that enhance human life, including medical purposes application, have been developed as well [13-15]. Usually, after injury and after stroke patients need physical treatment such as hand and leg motion exercises. On the other hand, VR technologies provide a powerful human-interface interaction [14-15] and audiovisual feedback simulation [13, 15], allow creating new exercises easily and setting the virtual environment flexibly [13, 15-16]. Some researches proved that therapies supported by VR technologies can improve mobility [16-19] and VR interface can simulate the brain better [14]. Even though the first generation of VR sensory devices considered a lack of haptic feedback [18], nowadays, many companies promise fast, accurate and powerful devices [19]. Moreover, VR devices are considered low-cost devices [13, 17] and rich data collection retriever [15-16].

In physical therapy, therapists will design several specific patterns of motion should be exercised by the patient. VR-based physical rehabilitation equipped with motion sensor(s) to sense hand or leg motion performed by a patient. The application needs to find out if the motion is in accordance with the designed pattern of motion [14]. The result of checking the correct gesture will be a response toward the virtual environment [13, 15]. There are two kinds of motion sensors, wearable sensors and camera-based sensors. In case using a camera-based sensor, from frames captured over time, the displacement of human joints position will be considered as human motion [9-11, 20-24]. Motion with certain patterns will be understood as a gesture.

Generally, hand gestures will be categorized as hand pose, hand sequence of movement or hand trajectory, and hand continuous movement [2, 3, 6, 9]. A hand pose is considerably simple, easy to be captured and recognized but not many poses can be represented using one hand or double hand without ambiguity [1, 6-8, 10, 12]. Trajectory gesture consists of several different poses to represent a whole gesture while in continuous gesture, poses and displacement positions or just one joint movement are considered one single gesture. However, trajectory and continuous movement consist of several poses, direction and orientation changing [3, 5, 6, 9, 11, 20]. Since no duration limitation in performing a gesture, it needs duration normalization. Dynamic Time Wrapping (DTW) [25] or define a fix data sampling [9] can be used as solutions for the duration problem.

Various techniques have been developed based on what kind of gesture to recognize and what kind of data got from the sensor. Color-based recognizing hand pose try to understand hand' shape, curvature between fingers or how many fingers opened [6, 10, 12]. Color-based data allows a little number of hand motion gestures, such as swap to left or right, push and pull hand [4, 7, 8, 10]. Such gestures can be used to navigate avatars in a virtual environment [4, 7, 8] or control devices inside a room [10-12]. However, to recognize hand movement, color-based data is not enough. It needs depth information to extract a feature vector from the palm area. Yang used HMM [26], Molchanov used HOG [11], some others used spatio-temporal feature [20, 23, 27-29] and others used motion feature [9, 24, 30]. Yet, to recognize more various hand motion gestures, skeleton-based data is better [3, 22, 24, 28]. Using information of all joints' position in a human's hand, palm direction, orientation, rotation while moving can be calculated. The matter in recognizing hand movement is determining the begin-end of a gesture and transform the length-various data into a uniform fix-length vector. De Smedt used fisher vector to represent vectors between 22 joints in hand and hand rotation [3], Lu used palm direction and fingertip angle as feature [5], Yang used tangential angular change over keyframes as feature [26], Liu used palm's displacement information over frames as feature [28] and others took a series of palm position from several frames as 3D data cloud [29, 30].

Using camera-based sensors, such as leap motion and HTC Vive, we face some challenges including various time duration and various orientation and direction performing each gesture. Some users perform a gesture faster, the others slower. The second challenge, users don't always position their hand facing the camera. To overcome the ununiform time duration problem, we adopted Ye and Cheng's idea, sampling a distinct number of points from each whole hand movement tracking [9, 25]. From all 3D points are tracked during a gesture performance, we sample 33 points uniformly. Those 33 points will represent our whole single gesture [9]. Various orientation and direction will be estimated using computer graphics approach.

To answer the need for accurate and real-time response physical treatment application, our research proposed a pipeline to sense and track hand gestures using hand movement sensor and to understand what kind of the performed gesture accurately and quickly. In order to gain a robust hand gesture classification application, we transformed each gesture into binary images and train them using our proposed density-based CNN.

2. RESEARCH METHOD

We propose a pipeline contains two main phases, image of gesture registration and gesture classification as shown in Figure 1.

2.1. Dataset

For our dataset, we collected 14 kinds of gestures designed by a physiotherapist as seen in Figure 2. These gestures are designed to help patients improving their movement ability gradually. Started from one turn rigid movement, continue to more than one movement. For advanced treatment, patients will try to follow a smooth movement, simple and then more complex.

All our gestures consist of one single stroke, a continuous movement. Each gesture is unique, with no similarity shape with 90° left or right rotation. We use the MNIST dataset style, small image 28x28 pixels, centered, black background with white foreground, preserved the gesture shape ratio [31]. Our pipeline will generate a frontal 2D binary image. It means the shape will not be skewed. Palm position toward finger's tip position as orientation and palm position toward the user's eye as direction. We use small resolution images because our gesture shape has 'one-pixel thickness' and sparse (has many 'blank pixel block' on the background part). Based on these conditions, enlarging the image resolution wouldn't give more detail information.

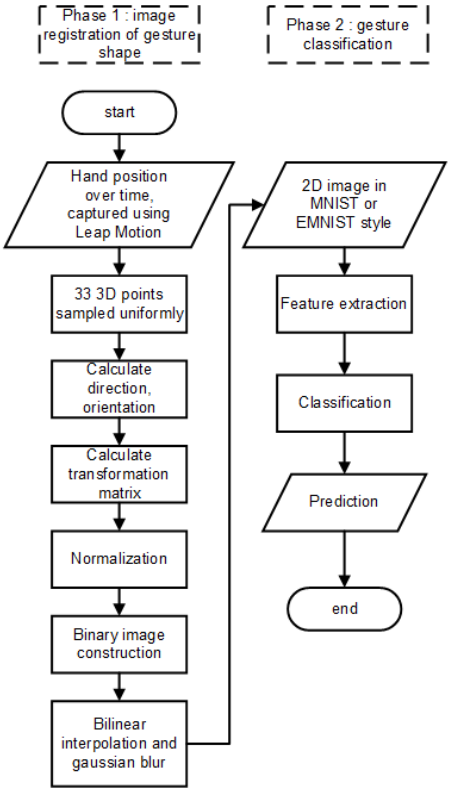
Phase 1 : image registration of gesture shape
Phase 2 : gesture classification
start
Hand position over time, captured using Leap Motion
33 3D points sampled uniformly
Calculate direction, orientation
Calculate transformation matrix
Normalization
Binary image construction
Bilinear interpolation and gaussian blur
2D image in MNIST or EMNIST style
Feature extraction
Classification
Prediction
end

Figure 1. Two stages of gesture recognition pipeline

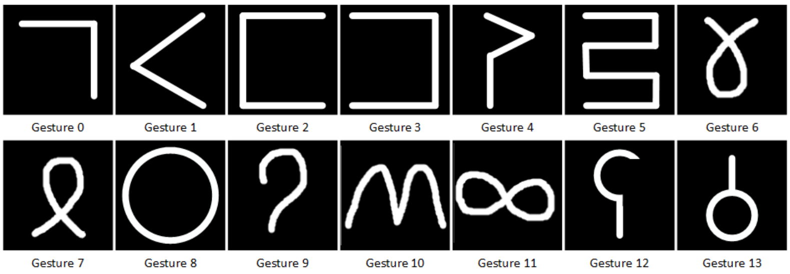

Figure 2. Designed gesture list

2.2. Phase 1: Image Registration of Gesture Shape

Given 33 3D points in the XYZ coordinate, transformation matrix UVN should be calculated to find the fittest plane to those 3D points. N axis is direction, V axis is orientation. First, normal plane or N can be obtained by applying Linear Least Square and Cramer's rule [32]. Given plane equation ax+by+cz+d=0, assuming the z component is always one, the equation becomes ax + by + d = -z. The matrix of all plane equations got from N points is shown in (1):

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ & \cdots & \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ d \end{bmatrix} = \begin{bmatrix} -z_1 \\ -z_2 \\ \cdots \\ -z_n \end{bmatrix} \tag{1}$$

applying linear least squares on (1), it will get (2).

$$\begin{bmatrix} \sum x_i x_i & \sum x_i y_i & \sum x_i \\ \sum y_i x_i & \sum y_i y_i & \sum y_i \\ \sum x_i & \sum y_i & n \end{bmatrix} \begin{bmatrix} a \\ b \\ d \end{bmatrix} = - \begin{bmatrix} \sum x_i z_i \\ \sum y_i z_i \\ \sum z_i \end{bmatrix} \tag{2}$$

Calculating the centroid and subtracting all 3D points with the centroid, x, y and z coordinate in (2) are defined relative to the centroid. Then (2) can be simplified become (3).

$$\begin{bmatrix} \sum x_i x_i & \sum x_i y_i & 0 \\ \sum y_i x_i & \sum y_i y_i & 0 \\ 0 & 0 & N \end{bmatrix} \begin{bmatrix} a \\ b \\ d \end{bmatrix} = - \begin{bmatrix} \sum x_i z_i \\ \sum y_i z_i \\ 0 \end{bmatrix} \tag{3}$$

If the plane is arranged to be through the origin <0,0,0>, then one dimension in (3) can be removed, which relates to *d*. Apply Cramer's rule on that removed dimension matrix gives some linear (4-6). Assuming axis z is removed, normal plane (N axis) will be (7).

$$det = \Sigma\mathrm{xx} * \Sigma\mathrm{yy} - \Sigma\mathrm{xy} * \Sigma\mathrm{xy} \tag{4}$$

$$a = (\Sigma\mathrm{yz} * \Sigma\mathrm{xy} - \Sigma\mathrm{xz} * \Sigma\mathrm{yy})/\mathrm{det} \tag{5}$$

$$b = (\Sigma\mathrm{xy} * \Sigma\mathrm{xz} - \Sigma\mathrm{xx} * \Sigma\mathrm{yz})/\mathrm{det} \tag{6}$$

$$N = [a, b, 1]^T \tag{7}$$

To prevent failure in obtaining normal plane, z component should be assumed to be a non-zero value. The same process is repeated for the non-zero x component and non-zero y component also. Using N vector from the biggest *det* value as direction. Orientation axis can be calculated by predicting the probable orientation, up. In case z component is the non-zero value, y axis <0, 1, 0> will be the probable orientation axis. Then apply (8) to calculate the real orientation axis, V.

$$\boldsymbol{V} = \boldsymbol{up} - \left(\frac{\boldsymbol{up} \cdot \boldsymbol{N}}{|\boldsymbol{N}|}\right) * \mathrm{N} \tag{8}$$

After finding the UVN coordinate, a transformation process can be done using (9). u, v and w are coefficients of U, V, N axis. x, y, and z are coefficients on X, Y, and Z axis.

$$[u \quad v \quad w \quad 1] = [x \quad y \quad z \quad 1] \begin{bmatrix} e_{1,1} & e_{1,2} & e_{1,3} & 0 \\ e_{2,1} & e_{2,2} & e_{2,3} & 0 \\ e_{3,1} & e_{3,2} & e_{3,3} & 0 \\ e_{4,1} & e_{4,2} & e_{4,3} & 1 \end{bmatrix} \tag{9}$$

Cramer's Rule is a determinant-based procedure that is used to solve systems of equations without solving all unknown variables. Cramer's Rule allows u, v, w directly calculated using these following vector equations shown in (14). By solving u, v and w variables, all e values on the transformation matrix on (9) can be obtained.

$$D = U \cdot (V \times N) \tag{10}$$

$$D_1 = \vec{t} \cdot (V \times N) \tag{11}$$

$$D_2 = U \cdot (\vec{t} \times N) \tag{12}$$

$$D_1 = U \cdot (V \times \vec{t}) \tag{13}$$

$$u = \frac{D_1}{D}, v = \frac{D_2}{D}, w = \frac{D_3}{D} \tag{14}$$

Let $\vec{t}$ =< 1,0,0 > and use Cramer's rule to calculate $e_{1,1}, e_{1,2}$ *and* $e_{1,3}$. Let $\vec{t}$ =< 0,1,0 > to calculate $e_{2,1}, e_{2,2}$ *and* $e_{2,3}$. Let $\vec{t}$ =< 0,0,1 > to calculate $e_{3,1}, e_{3,2}$ *and* $e_{3,3}$. Finally, let $\vec{t} = < 0,0,0 > -original$ and calculate $e_{4,1}, e_{4,2}$ *and* $e_{4,3}$. Getting all those e values, a transformation matrix is produced. To generate a centered 28 x 28 binary image, a normalization process is needed. First, adjust the ratio of the actual size by divided desired image size, 28 with maximum value between distance in U axis and distance in V axis. (15) is used to adjustment process.

$$ratio = 28/\max(dist_x, dist_y) \tag{15}$$

Multiplying all 2D points with the ratio, finding the center, subtracting with (center - <14, 14>), decimalizing floating values of 2D mapped points into integer pixels position will produce discontinuous line. Bilinear interpolating needed to smoothen them. Figure 3 visualizes all processes in this phase.

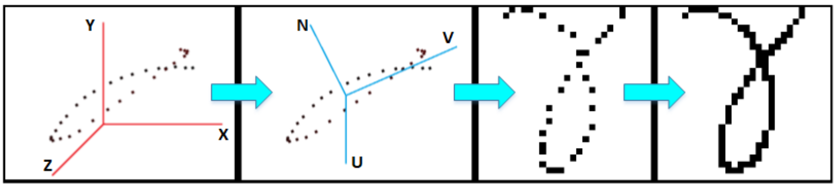

Figure 3. From left to right, 3D points captured from a sensor, calculating the direction and orientation axis of the fittest plane, 2D sparse binary image, after bilinear interpolation

2.3. Phase 2: Hand gesture classification

In this stage, inspired by LeNet-5 that had already proved its success on training a low resolution, small size image dataset that contains single information about simple shapes such as MNIST and EMNIST datasets as published on [31], we proposed our density-based CNN architecture. This density-based CNN architecture has consisted 3 layers for feature extraction and two layers for classification as seen in Figure 4.

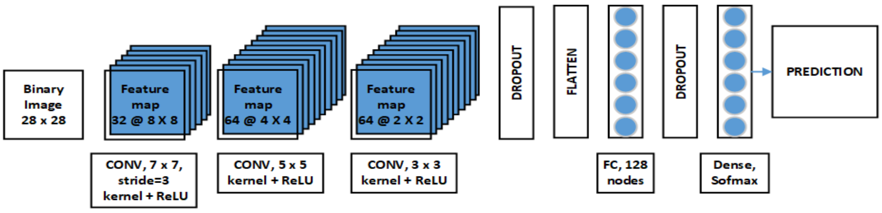

Figure 4. Density-based CNN architecture for hand gesture classification

Our gesture image characteristics are having many 'blank block pixel' as background and 'single-pixel thickness' foreground. So, our binary images considered sparse images. We need to prevent those 'blank block pixel' contributing to the feature maps and boast the dense image block to contribute more. Pyramid kernel size applied to the feature extraction parts will solve this problem. Bigger kernel size on the first layer and getting smaller on the next layer. In the first layer, bigger blank blocks on the background can be eliminated using a big kernel size for the convolution process. As the image size getting smaller, we apply a smaller kernel size for the convolution process. Big size kernel on the first layer will determine which block pixel should contribute more. Not following LeNet-5 architecture which used max-pooling layer, instead of using max pooling, we used large stride (stride = 3) on the first layer convolution process. Because max-pooling will cause blank block pixels near the foreground are calculated as foreground in the next layer. Since our input image size is small, we also need a small model as well. To remove some not significant nodes come from 'blank block pixel', a dropout layer is applied. After that, the output will be flatted into 128 nodes of fully connected layer. We use cross-entropy loss function because we need a probabilistic result. We use 14 nodes for the final layer as the number of gesture shape classes.

2.4. VR application scheme

To implement in VR-based application, we developed a client-server networking scheme as shown in Figure 5. The training part is implemented using python with Keras. The same capturing image implementation as in the client part is used to capture the dataset images. After finishing the training, the weight of that model will be stored on the server and can be accessed by the VR game Content. In the client part, as the application doing loops, the hand controller sensor will capture the user's hand movement and be sent to the server. The server will generate a 2D binary image of the gesture, input it to the density-based CNN and get the prediction. The prediction result will be sent to the client and shown in the application as a response for the user.

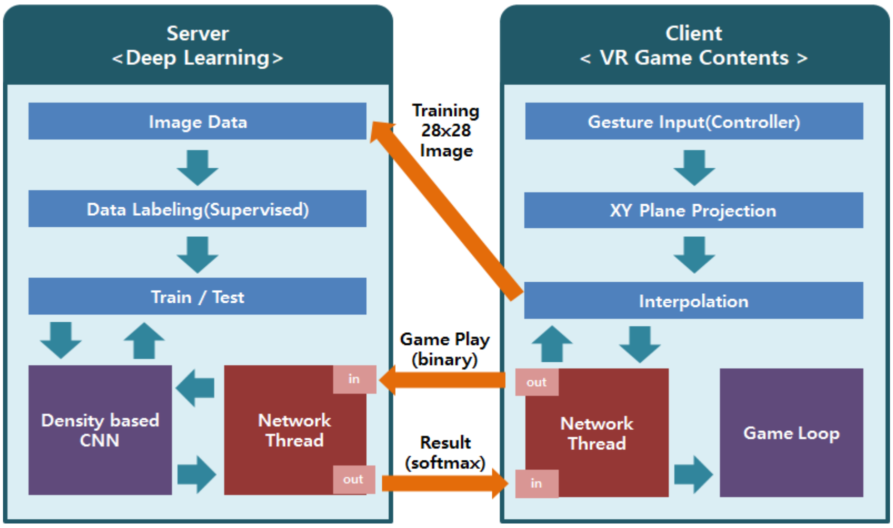

Figure 5. Client-server networking scheme

3. RESULTS AND ANALYSIS

In our experiments, we use 7000 gesture images as the dataset, 4900 for training and 2100 for testing. The goal of our experiment is to measure how far our pipeline suits the problem well. Some excellences of our model are pyramid size kernel applied on CNN layers to avoid blank block pixel contributing in the next layers, remove max-pooling layers and replaced them with convolution stride 3, using binary images dataset, not grayscale image dataset like LeNet 5. Evaluate how suitable the number of layers and number of feature maps of each layer in the CNN part.

We run our model with 600 epochs and 128 images per batch. Comparing our model with LeNet-5 as benchmark model, measuring whether using our pyramid size kernel better than same size kernel for all convolution layers, using three layers in CNN part better than a deeper model, using that number of feature maps on our model's CNN is suitable with our problem well and whether the same model running on grayscale images will make a difference. To obtain grayscale images, we modified our dataset by blurring them using a gaussian blur.

Comparison accuracy between density-based CNN run on binary images, grayscale images and using same size kernels, using deeper layers and applying a fewer number of feature maps and with our benchmark model, LeNet 5 is described in Table 1. Figure 6 shows detail information about the exact value from epoch 30 until 600 with 30 epochs increases.

Table 1. Comparison accuracy between several models with density-based model

	leNet 5	deep layer model	same size kernel	fewer feature maps	density-based CNN	grayscale image input
Epoch # of the highest accuracy	240	90	180	600	360	180
Highest accuracy	0.963333	0.968000	0.971333	0.951000	0.977333	0.976000

From Figure 6 and Table 1 we can see that using a deeper layer (7 layers) model, the highest accuracy achieved in 90 epochs. It considered the fastest process but it did not gain the highest accuracy. Using the same size kernel on the CNN part or grayscale images can reach high accuracy in 180 epochs. Slower process, better accuracy but still lower than ours. Compare with those models and our density-based CNN model, LeNet 5 which used max-pooling layers reach the lowest accuracy. Using a fewer number of feature maps got the lowest accuracy among others.

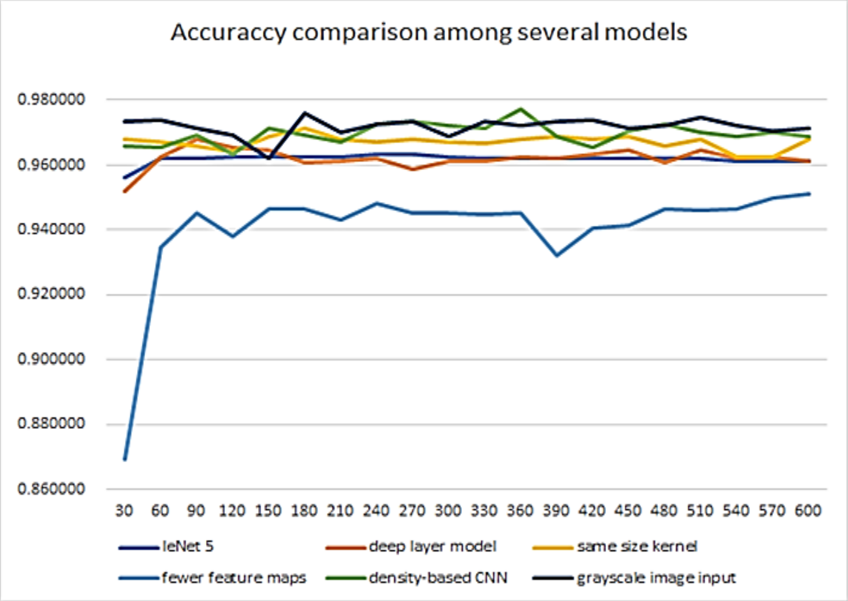

Figure 6. Accuraccy comparison among several models

4. CONCLUSION

Pyramid kernel size works better on binary images than on grayscale images even though our model needed more epochs to get higher accuracy. Since binary images have 'blank block pixel' and 'single-pixel thickness' characteristics, layers with pyramid kernel size and large stride convolution in the first layer accommodated binary images better than max pooling layer (LeNet 5) because they prevent the 'blank block pixel' contributes to the feature maps.

Our pipeline is able to achieve higher accuracy with more epoch than other compared models. Even though other models can achieve their highest accuracy before 300 epochs but got the accuracy decrease after 300 epochs. While our model still got promising increase accuracy after 300 epochs. Binary images versus grayscale or RBG images is not the only reason. Our proposed model suitable for simple various information (only two values), less density image, sparse dots, and unambiguous content image datasets. In this case, transformed 'drawing in the air'-like gesture into 2D images considered as a suitable choice. The only limitation in our system is its lack of sequence information of the gesture because we transformed them into 2D images. For further physical treatment application that needs to train gestures based on their different order of gesture but come out similar 2D images mapping, input sequence gesture will solve that matter better than the input image.

Our proposed networking scheme with gesture classification pipeline can be used generally as long it receives 3D points cloud as input. These 3D points give information about the body's joint movement. Several gesture controllers for VR such as leap motion, kinect and HTC vive support our system with 3D point information. Later, applying the transfer learning scheme, preserve the weight of the CNN part and retrain only the fully connected layers, our density-based CNN with CNN layers using pyramid kernel size will be compatible with other similar datasets.

ACKNOWLEDGEMENTS

This work was supported in part by Institute for Information and Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2018-0-00245, Development of prevention technology against AI dysfunction induced by deception attack). And it was also in part supported by the Dongseo University Research Year.

REFERENCES

[1] L. Abraham, A. Urru, N. Normani, M. P. Wilk, M. Walsh, B. O'Flynn, "Hand Tracking and Gesture Recognition Using Lensless Smart Sensors," in *Sensors*, vol. 18, no. 9, pp. 2834, August 2018.

[2] H. Cheng, L. Yang and Z. Liu, "Survey on 3D Hand Gesture Recognition," *in IEEE Transactions on Circuits and Systems for Video Technology,* vol. 26, no. 9, pp. 1659-1673, September 2016.

[3] Q. De Smedt, H. Wannous and J. Vandeborre, "Skeleton-Based Dynamic Hand Gesture Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),* Las Vegas, NV, pp. 1206-1214, 2016.

[4] M. Gillies, "What is Movement Interaction in Virtual Reality for?," *MOCO '16: Proceedings of the 3rd International Symposium on Movement and Computin*g, pp. 1-4, July 2016.

[5] W. Lu, Z. Tong and J. Chu, "Dynamic Hand Gesture Recognition With Leap Motion Controller," *IEEE Signal Processing Letters*, vol. 23, no. 9, pp. 1188-1192, September 2016.

[6] J.M. Palacios, C. Sagues, E. Montijano, S. Llorente, "Human-Computer Interaction Based on Hand Gesture Using RGB-D Sensors," *Sensors*, vol. 13, no. 9, pp. 11842-11860, September 2013

[7] C. Khundam, "First person movement control with palm normal and hand gesture interaction in virtual reality," *2015 12th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, Songkhla, pp. 325-330, 2015.

[8] F. Zhang, S. Chu, R. Pan, N. Ji and L. Xi, "Double hand-gesture interaction for walk-through in VR environment," *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, Wuhan, pp. 539-544, 2017.

[9] Guangqi Ye, J. J. Corso and G. D. Hager, "Gesture Recognition Using 3D Appearance and Motion Features," *2004 Conference on Computer Vision and Pattern Recognition Workshop*, Washington, DC, USA, pp. 160-166, 2004.

[10] D. L. Dinh, J. T. Kim, T. S. Kim, "Hand gesture Recognition and Interface via a Depth Imaging Sensor for Smart Home Appliances," in *Energy Procedia,* vol. 62, pp. 576-582, 2014.

[11] P. Molchanov, S. Gupta, K. Kim and K. Pulli, "Multi-sensor system for driver's hand-gesture recognition," *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, Ljubljana, pp. 1-8, 2015.

[12] J. P. Son, A. Sowmya, "Single-Handed Driving System with Kinect," *International Conference on Human-Computer Interaction HCI 2013,* vol 8005, pp. 631-639, July 2013.

[13] C. Camporesi, M. Kallmann and J. J. Han, "VR solutions for improving physical therapy," *2013 IEEE Virtual Reality (VR)*, Lake Buena Vista, FL, pp. 77-78, 2013.

[14] R. G. Lupu, D. C. Irimia, F. Ungureanu, M. S. Poboroniuc, A. M., "BCI and FES Based Therapy for Stroke Rehabilitation Using VR Facilities," in *Wireless Communications and Mobile Computing*, vol. 2018, no. 4, April 2018.

[15] D. White, K. Burdick, G. Fulk, J. Searleman and J. Carroll, "A virtual reality application for stroke patient rehabilitation," *in CyberPsychology & Behavior*, vol. 9, no. 6, pp. 1081-1086, December 2006.

[16] J. E. Deutsch, "Virtual reality and gaming systems to improve walking and mobility for people with musculoskeletal and neuromuscular conditions," in *Studies in health technology and informatics*, vol. 145, pp. 84-93, Febuary 2009.

[17] D. Corbetta, F. Imeri, R. Gatti, "Rehabilitation that incorporates virtual reality is more effective than standard rehabilitation for improving walking speed, balance and mobility after stroke: a systematic review," *Journal of Physiotherapy*, vol. 61, no. 3, pp. 117-124, July 2015.

[18] L. Schmid, A. Glässel, C. Schuster-Amft, "Therapists' Perspective on Virtual Reality Training in Patients after Stroke: A Qualitative Study Reporting Focus Group Results from Three Hospitals," *Stroke Research and Treatment* vol. 2016, no. 2, pp. 1-12, December 2016.

[19] "Documentation", Retrieved from https://developer.leapmotion.com/documentation/, 1st Feb 2020.

[20] P. Wang, W. Li, S. Liu, Z. Gao, C. Tang and P. Ogunbona, "Large-scale Isolated Gesture Recognition using Convolutional Neural Networks," *2016 23rd International Conference on Pattern Recognition (ICPR),* Cancun, pp. 7-12, 2016.

[21] Y. Zhou, G. Jiang, Y. Lin, "A Novel finger and hand pose estimation technique for real-time hand gesture recognition," *Pattern Recognition*, vol. 49, pp. 102-114, August 2015.

[22] M.F. Kassim, M.N.H. Mohd, "Food Intake Gesture Monitoring System Based-on Depth Sensor", *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 2, pp. 470-476, June 2019.

[23] M. Galinium, J. Yapri, J. Purnama, "Markerness motion Capture for 3D Human Model Animation Using Depth Camera", *TELKOMNIKA Telecommunication Electronics Computing and Control,* vol. 17, no. 3, pp. 1300-1309, June 2019.

[24] L. Tian, N. M. Thalmann, D. Thalmann, J. Zheng, "Nature Grasping by a Cable-driven Under-actuated Anthropomorphic Robotic Hand," *TELKOMNIKA Telecommunication Electronics Computing and Control*, vol. 17, no. 1, pp. 1-7, Febuary 2019.

[25] H. Cheng, J. Luo and X. Chen, "A windowed dynamic time warping approach for 3D continuous hand gesture recognition," *2014 IEEE International Conference on Multimedia and Expo (ICME)*, Chengdu, pp. 1-6, 2014.

[26] C. Yang, D. K. Han, H. Ko, "Continuous hand gesture recognition based on trajectory shape information (in press)," in *Pattern Recognition Letters*, vol. 99, pp. 39-47, May 2017.

[27] O. Köpüklü, A. Gunduz, N. Kose, G. Rigoll, "Real-time Hand Gesture Detection and Classification Using Convolutional Neural Networks," *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019),* Lille, France, pp. 1-8, 2019.

[28] F. Liu, W. Zeng, C. Yuan, Q. Wang, Y. Wang, “Kinect-based Hand Gesture Recognition Using Trajectory Information, Hand Motion Dynamics and Neural Networks,” *Artificial Intelligent Review*, vol. 52, no. 1, pp. 563–583, April 2019.

[29] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree and J. Kautz, "Online Detection and Classification of Dynamic Hand Gestures with Recurrent 3D Convolutional Neural Networks," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* Las Vegas, NV, pp. 4207-4215, 2016.

[30] Ho-Joon Kim, J. S. Lee and J. Park, "Dynamic hand gesture recognition using a CNN model with 3D receptive fields," *2008 International Conference on Neural Networks and Signal Processing,* Nanjing, pp. 14-19, 2008.

[31] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[32] D. Eberly, “Least Squares Fitting of Data by Linear or Quadratic Structures”, Geometric Tools, July 1999.

BIOGRAPHIES OF AUTHORS

Liliana She receives her B.S from Surabaya University, Surabaya, Indonesia in 2002 and her M.Eng from Dongseo University, Busan, South Korea in 2009. Now she is pursuing her Doctoral degree in Dongseo Univerity, Busan, South Korea. She has been working as Lecturer at Petra Christian University since 2003. Her interest fields are Computer Graphics, Computer Vision and Intelligent Systems.

Ji-Hun Chae He received his B.S, M.S in Computer Engineering from Keimyung University, Daegu, South Korea, in 2016 and 2018 respectively. He is currently a research engineer at Virnect research Institute. His research interests include Computer Vision, Image Processing, Computer Graphics and Gesture Recognition.

Joon-Jae Lee He received his B.S., M.S., and Ph.D. in Electronic Engineering from the Kyungpook National University, Daegu, South Korea, in 1986, 1990, and 1994, respectively. From March 1995 to August 2007, he was with the Computer Engineering faculty at the Dongseo University, Busan, South Korea. He is currently a full professor of the Department of Game Mobile Contents, Keimyung University. He was a visiting scholar at the Georgia Institute of Technology, Atlanta, from 1998 to 1999, funded by the Korea Science and Engineering Foundation (KOSEF). He also worked for PARMI Corporation as a research and development manager for 1 year from 2000 to 2001. His main research interests include image processing, three-dimensional computer vision, and fingerprint recognition.

Byung-Gook Lee He received his B.S. in Mathematics from Yonsei University, Korea, in 1987, and his M.S. and Ph.D. in Applied Mathematics from Korea Advanced Institute of Science and Technology (KAIST) in 1989 and 1993, respectively. He worked at the DACOM Corp. R&D Center as a senior engineer from March 1993 to February 1995. He has been working at Dongseo University, Korea, since 1995 and is currently a full professor with the Division of Computer Information Engineering. His research interests include computer graphics, computer-aided geometric design, and image processing.