draft

by Liliana Liliana

Submission date: 09-Dec-2019 02:30PM (UTC+0700)

Submission ID: 1230435998

File name: obust_method_for_VR_based_hand_gesture_recognition_Liliana.docx (233K)

Word count: 4115

Character count: 23164

A Robust Method for VR-based Hand Gesture Recognition Using Density-based CNN

Liliana[1,2], Ji-Hun Chae[3], Joon-Jae Lee[3], Byung-Gook Lee[1]

[1] Computer Engineering Dongseo University, Busan, South Korea
47 Jurye-ro, Sasang-Gu, Busan 47011, South Korea
[2] Informatic Department Petra Christian University, Surabaya, Indonesia
Siwalankerto 121-131, Surabaya 60236, Indonesia
[3] School of Computer Engineering Keimyung University, Daegu, South Korean
Dalseodaero 675, Dolseo-gu, Daegu 42601, South Korea
e-mail: lilian@petra.ac.id, cowlgns21@naver.com, joonlee@kmu.ac.kr, lbg@dongseo.ac.kr

Abstract

Many VR-based medical purposes application have been developed to help patient with mobility decrease caused by accidents, diseases, or other injury 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 physical therapist assistance. 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. Our dataset consisted 14 kinds of hand motion recommended by a physiotherapist. Given 33 3D points which was mapped into binary images as input, we trained our proposed density-based CNN. Our CNN model concerned on our input characteristics, having many 'blank block pixels', 'single pixel thickness' shape and generated binary image. Pyramid kernel size applied on feature extraction part and classification layer using softmax as loss function, have given around 97% accuracy.

Keywords*: binary image learning, density-based CNN, hand gesture classification, physical treatment, VR-based application*

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 gesture as navigator 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 [16-18] and VR interface can simulate brain better [14]. Even though the first generation of VR sensory devices considered lack of haptic feedback [18], nowadays, many companies promise a fast, accurate and powerful devices [19]. Moreover, VR devices are considered low cost devices [13,17] and rich data collection retriever [15-16].

In a physical therapy, therapist will design several specific patterns of motion should be exercised by patient. VR-based physical rehabilitation equipped with motion sensor 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 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 pattern will be understood as gesture.

Generally, hand gesture 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 using camera-based sensor but not many poses can be represented using one hand or double hand without ambiguity [1,6-8,10,12]. Trajectory movement consists several poses to represent a whole gesture while in continuous movement, poses and displacement positions are considered one single gesture. However, trajectory and continuous movement consist 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 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 gesture, such as swap to left or right, push and pull hand [4, 7, 8, 10]. Such gestures can be used to navigate avatar in virtual environment [4,7,8] or control devices inside 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 gesture, skeleton-based data is better [3,22,24,28]. Using information of all joints' position in human hand, palm direction, orientation, rotation while moving can be calculated. The matter in recognizing hand movement is determining 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 key frames 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 Kinect, 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, user don't always position their hand facing the camera. To overcome 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 of accurate and real-time response physical treatment application, our research proposed a pipeline to sense and track hand gesture 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 a binary images and train them using our proposed density-based CNN. We collected 14 kinds of gesture of a single continuous hand movement recommended by a physiotherapist. Totally, we had 7000 gestures consisted of 14 kind of gestures, performed by 100 persons.

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 gesture designed by a physiotherapist as seen in Figure 2. These gestures are designed to help patient improving their movement ability gradually. Started from one turn rigid movement, continue to more than one movement. For advance treatment, patient will try to follow a smooth movement, simple and then more complex.

All our gestures consist one single stroke, a continuous movement. Each gesture is unique, no similarity shape with 90° left or right rotation. We use 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 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.

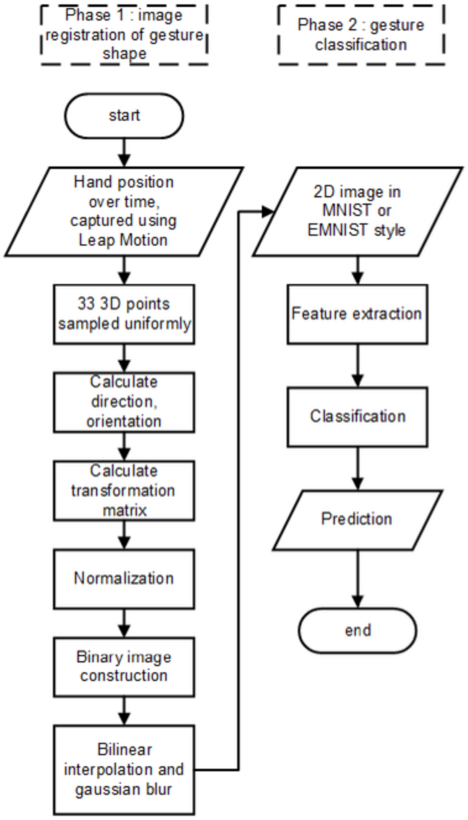
Phase 1 : image registration of gesture shape
Phase 2 : gesture classification
start
Hand position over time, captured using Leap Motion
33 3D points sampled uniformly
Calculate direction, orientation
Calculate transformation matrix
Normalization
Binary image construction
Bilinear interpolation and gaussian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

Figure 1. Two stages of gesture recognition pipeline

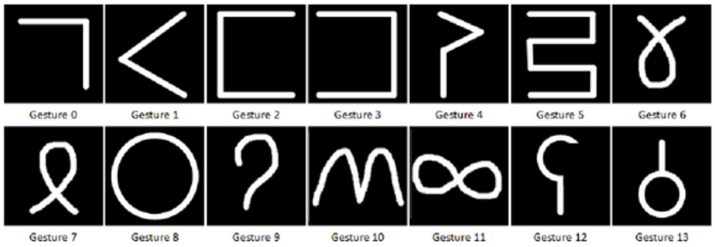

Figure 2. Designed gesture list

2.2. Phase 1: Image of Gesture Shape

Given 33 3D points in 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):

$$\begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ & \cdots & \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ d \end{bmatrix} = \begin{bmatrix} -z_0 \\ -z_1 \\ \cdots \\ -z_n \end{bmatrix} \quad (1)$$

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

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

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

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

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

$$det = \Sigma xx * \Sigma yy - \Sigma xy * \Sigma xy \quad (4)$$

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

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

$$N = [a, b, 1]^T \quad (7)$$

To prevent failure in obtaining normal plane, z component should be assumed to be non-zero value. The same process is repeated for 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.

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

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

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

Cramer's Rule is a determinant-based procedure that is used to solve systems of equations without solving all unknown variable. 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.

$$D = U \cdot (V \times N) \quad (10)$$

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

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

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

$$u = \frac{D_1}{D}, v = \frac{D_2}{D}, w = \frac{D_3}{D} \quad (14)$$

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.

$$ratio = 28/\max(dist_x, dist_y) \quad (15)$$

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

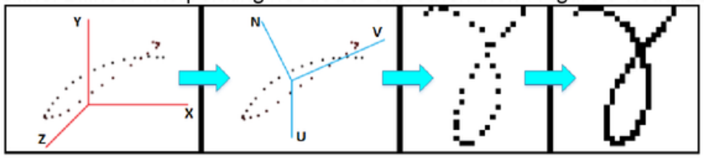

Figure 3. From left to right, 3D points captured from sensor, calculating the direction and orientation axis of the fittest plane, 2D sparse binary image, after bilinear interpolation.

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 contain single information about simple shape such as MNIST and EMNIST datasets as published on [31], we proposed our density-based CNN architecture. This density-based CNN architecture is consisted 3 layers for feature extraction and two layers for classification as seen on Figure 4.

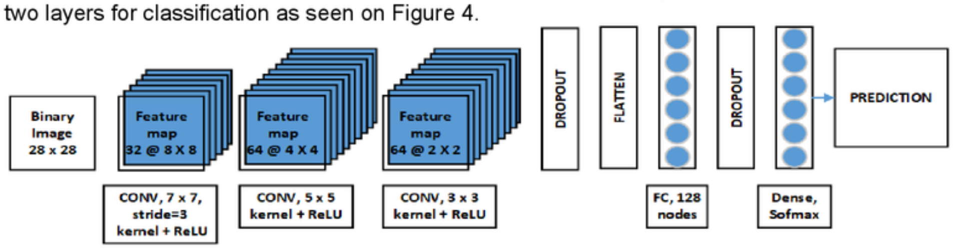

Figure 4. Density-based CNN architecture for hand gesture classification.

Our gesture image characteristics are having many ‘blank block pixel’ as background and ‘single pixel thickness’ foreground. So, our binary images considered as 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 is applied on 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 block on the background can be eliminate using big kernel size for the convolution process. As the image size getting smaller, we apply 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 pixel near the foreground are calculated as foreground in the next layer. Since our input image size is small, we also need 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. Training part is implemented using python with Keras. The same capturing image implementation as in the client part is used for capture the dataset images. After finishing the training, the weight of that model will be stored on server and can be accessed by the VR game Content. In the client part, as the application doing loop, hand controller sensor will capture user’s hand movement and be sent to server. Server will generate 2D binary image of the gesture, input it to the density-based CNN and get the prediction. The prediction result will be sent to client and shown in the application as response for user.

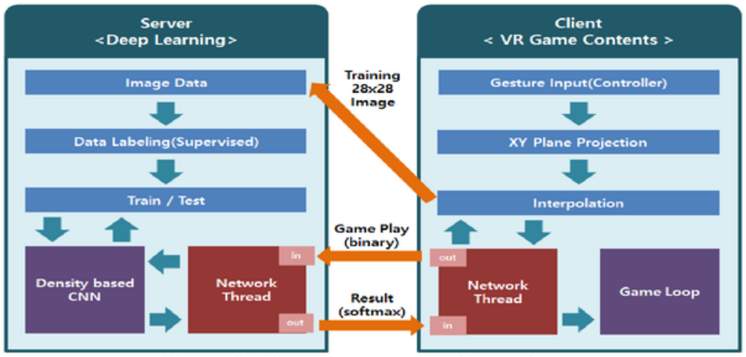

Figure 5. Client Server networking scheme

3. Results and Analysis

Performing the first stage of our pipeline, we got 2D images, transformed 3D points from motion sensors into 2D images with MNIST style, as shown in Figure 6. In our experiments, we use 7000 gesture images as 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 number of layers and number of feature maps of each layers 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 gaussian blur.

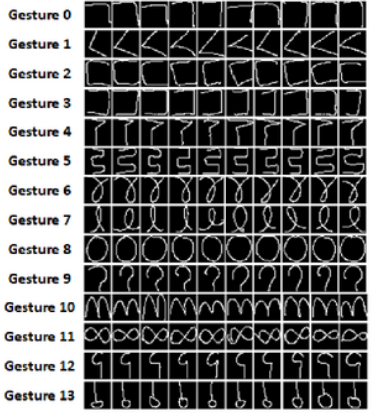

Figure 6. Some sample of hand gesture dataset

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

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

epoch	leNet 5	deep layer model	same size kernel	fewer feature maps	density-based CNN	grayscale image input
30	0.956000	0.952000	0.968000	0.869333	0.966000	0.973333
60	0.962000	0.962667	0.967333	0.934667	0.965333	0.974000
90	0.962000	**0.968000**	0.966000	0.945333	0.969333	0.971333
120	0.962667	0.965333	0.964000	0.938000	0.963333	0.969333
150	0.962667	0.964667	0.968667	0.946667	0.971333	0.962000
180	0.962667	0.960667	**0.971333**	0.946667	0.969333	**0.976000**
210	0.962667	0.961333	0.968000	0.943333	0.967333	0.970000
240	**0.963333**	0.962000	0.967333	0.948000	0.972667	0.972667
270	0.963333	0.958667	0.968000	0.945333	0.973333	0.973333
300	0.962667	0.961333	0.967333	0.945333	0.972000	0.968667
330	0.962000	0.961333	0.966667	0.944667	0.971333	0.973333
360	0.962000	0.962667	0.968000	0.945333	**0.977333**	0.972000
390	0.962000	0.962000	0.968667	0.932000	0.968667	0.973333
420	0.962000	0.963333	0.968000	0.940667	0.965333	0.974000
450	0.962000	0.964667	0.968667	0.941333	0.970667	0.971333
480	0.962000	0.960667	0.966000	0.946667	0.972667	0.972000
510	0.962000	0.964667	0.968000	0.946000	0.970000	0.974667
540	0.961333	0.962000	0.962667	0.946500	0.968667	0.972000
570	0.961333	0.962667	0.962667	0.950000	0.970000	0.970667
600	0.961333	0.961333	0.968000	**0.951000**	0.968667	0.971333

From Table 1 we can see that using a deeper layer (7 layers) model, higest accuracy achieved in 90 epochs. It considered fastest process but it did not gain the highest accuracy. Using same size kernel on the CNN part or grayscale images can reached 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 fewer number of feature maps got the lowest accuracy among others.

4. Conclusion

Pyramid kernel size work better on binary images than on grayscale images even though our model needed more epochs to get the 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 able to achieve higher accuracy with more epoch than other compared models. Eventhough 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 sequences information of the gesture because we transformed them into 2D images. For further physical treatment application which need to train gesture based on their different order of gesture but come out similar 2D images mapping, input sequence gesture will solve that matter better than 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 body's joint movement. Several gesture controllers for VR such as leapmotion, kinect and HTC vive support our system with 3D point information. Later, applying transfer learning scheme [33], preserve the weight of CNN part and retrain only the fully connected layers, our density-based CNN with CNN layers using pyramid kernel size will be compatible for other similar datasets.

Acknowledgments

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] Abraham L, Urru A, Normani N, Wilk MP, Walsh M, O'Flynn B. Hand Tracking and Gesture Recognition Using Lensless Smart Sensors. Sensors. 2018; 8(9): 2834.

[2] Cheng H, Yang L, Liu Z. Survey on 3D Hand Gesture Recognition. Circuits and Systems for Video Technology IEEE Transactions on. 2016; 26(9): 1659-1673.

[3] Smedt QD, Wannous H, Vandeborre JP. Skeleton-Based Dynamic Hand Gesture Recognition. Computer Vision and Pattern Recognition Workshops (CVPRW) 2016 IEEE Conference on. 2016: 1206-1214.

[4] Gillies M. What is Movement Interaction in Virtual Reality for?. Proceedings of the 3rd International Symposium on Movement and Computing. 2016.

[5] Lu W, Tong Z, Chu J. Dynamic Hand Gesture Recognition with Leap Motion Controller. IEEE Signal Processing Letters. 2016; 23(9): 1188-1192.

[6] Palacios JM, Sagues C, Montijano E, Llorente S. Human-Computer Interaction Based on Hand Gesture Using RGB-D Sensors. Sensors. 2013; 13: 11842-11860.

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

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

[9] Ye GQ, Corso JJ, Hager GD. Gesture recognition using 3D appearance and motion features. IEEE Transactions on Computer Vision and Pattern Recognition Workshops. 2004: 160-166.

[10] Dinh DL, Kim JT, Kim TS. Hand gesture Recognition and Interface via a Depth Imaging Sensor for Smart Home Appliances. Energy Procedia. 2014; 62: 576-582.

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

[12] Son JP, Sowmya A. Single-Handed Driving System with Kinect. Lecture Notes in Computer Science book series (LNCS). 2013; 8005: 631-639.

[13] Camporesi C, Kallmann M, Han JJ. VR Solutions for Improving Physical Therapy. 2013 IEEE Virtual Reality (VR). 2013: 77-78.

[14] Lupu RG, Irimia D, Ungureanu F, Poboroniuc M, Moldoveanu A. BCI and FES Based Therapy for Stroke Rehabilitation Using VR Facilities. Wireless Communications and Mobile Computing. 2018: 8 pages.

[15] White D, Burdick K, Fulk G, Searleman J and Carroll J. A virtual reality application for stroke patient rehabilitation. IEEE International Conference Mechatronics and Automation. 2005; 2: 1081-1086.

[16] Deutsch JE. Virtual reality and gaming systems to improve walking and mobility for people with musculoskeletal and neuromuscular conditions. Stud Health Technol Inf. 2009; 145: 84-93.

[17] Corbetta D, Imeri F, Gatti R. 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. 2015; 61: 117-124.

[18] Schmid L, Glässel A, Schuster-Amft C. Therapists' Perspective on Virtual Reality Training in Patients after Stroke: A Qualitative Study Reporting Focus Group Results from Three Hospitals. Stroke Research and Treatment. 2016; 2016: 12 pages.

[19] ``Leap Motion's software and hardware platform brings your bare hands directly into virtual and augmented reality,'' Retrieved from https://www.leapmotion.com/technology

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

[21] Zhou Y, Jiang G, Lin Y. A Novel finger and hand pose estimation technique for real-time hand gesture recognition. Pattern Recognition. 2016; 49: 102-114

[22] Kassim MF, Mohd MNH. Food Intake Gesture Monitoring System Based-on Depth Sensor. Bulletin of Electrical Engineering and Informatics. 2019; 8(2): 470-476

[23] Galinium M, Yapri J, Purnama J. Markerness motion Capture for 3D Human Model Animation Using Depth Camera. Telkomnika. 2019; 17(3): 1300-1309

[24] Tian L, Thalmann NM, Thalmann D, Zheng J. Nature Grasping by a Cable-driven Under-actuated Anthropomorphic Robotic Hand. Telkomnika. 2019; 17(1): 1-7

[25] Cheng H, Luo J, Chen X. A windowed dynamic time warping approach for 3D continuous hand gesture recognition. IEEE ICME. 2014: 1-6.

[26] Yang C, Han DK, Ko H. Continuous hand gesture recognition based on trajectory shape information (in press). Pattern Recognition Letters. 2017; 99: 39-47

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

[28] Liu F, Zeng W, Yuan C, Wang Q, Wang Y. Kinect-based Hand Gesture Recognition Using Trajectory Information, Hand Motion Dynamics and Neural Networks. Artificial Intelligent Review. 2019; 52(1): 563–583

[29] Molchanov P, Yang X, Gupta S, Kim K, Tyree S, Kautz J. 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, 2016: 4207-4215.

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

[31] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998; 86(11): 2278-2324.

[32] Eberly D. Least Squares Fitting of Data by Linear or Quadratic Structures. Geometric Tools 1999. Retrieved from https://www.geometrictools.com/Documentation/LeastSquaresFitting.pdf

[33] Torrey L, Shavlik J. Transfer Learning. In: Soria E, Martin J, Magdalena R, Martinez M, Serrano A. Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques., Hershey: Information Science Reference - Imprint of: IGI Publishing. 2009: 242-264.

draft

ORIGINALITY REPORT

6%	3%	4%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	KiHwan Kim, HoonJae Lee. "Proposal of Multi-channel Operation Technique Using PingPong256", 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), 2018 Publication	1%
2	www.css.tayloru.edu Internet Source	1%
3	Submitted to Multimedia University Student Paper	1%
4	Hendry Raharjo, Min Xie, Thong Ngee Goh, Aarnout C. Brombacher. "A Methodology to Improve Higher Education Quality using the Quality Function Deployment and Analytic Hierarchy Process", Total Quality Management & Business Excellence, 2007 Publication	<1%
5	Al Bakri, A.M. Mustafa, A.R. Rafiza, D. Hardjito,	

	H. Kamarudin, and I. Khairul Nizar. "Characterization of LUSI Mud Volcano as Geopolymer Raw Material", Advanced Materials Research, 2012. Publication	<1%
6	Submitted to Sonoma State University Student Paper	<1%
7	Jinting Wu, Kang Li, Xiaoguang Zhao, Min Tan. "Chapter 22 Unfamiliar Dynamic Hand Gestures Recognition Based on Zero-Shot Learning", Springer Science and Business Media LLC, 2018 Publication	<1%
8	Alvin Poernomo, Dae-Ki Kang. "Content-Aware Convolutional Neural Network for Object Recognition Task", International journal of advanced smart convergence, 2016 Publication	<1%
9	"Advances in Multimedia Information Processing – PCM 2017", Springer Science and Business Media LLC, 2018 Publication	<1%
10	"Communications, Signal Processing, and Systems", Springer Science and Business Media LLC, 2019 Publication	<1%
	onlinelibrary.wiley.com	

11	Internet Source	<1%
12	orchard.wccnet.org Internet Source	<1%
13	blog.vinceliu.com Internet Source	<1%
14	Submitted to Loughborough University Student Paper	<1%
15	Submitted to Imperial College of Science, Technology and Medicine Student Paper	<1%
16	Stephen Wang-Cheung Lam. "Extensions of the general polar value based control point specification method in constructing tensor product B-spline surfaces", Computers & Graphics, 2000 Publication	<1%

Exclude quotes	On	Exclude matches	< 5 words
Exclude bibliography	On		