International Journal of Engineering & Technology, 7 (3.32) (2018) 103-105

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Experimenting Hand-Gesture Image Recognition using Simple Deep Neural Network

Mostafa Alghamdi[1], Tami Alwajeeh[1], Fahad Aljabeer[2], Setiawan Assegaff[3], Rahmat Budiarto[4,*]

[1]*Department of Computer Science, Albaha University, Albaha, Saudi Arabia*
[2]*Information Technology Center, Albaha University, Albaha, Saudi Arabia*
[3]*Department of Information System, STIKOM Dinamika Bangsa, Jambi, Indonesia*
[4]*Department of Computer Information System, Albaha University, Albaha, Saudi Arabia*
**Corresponding Author: (Phone: + 966 55 345 1317; Email: rahmat@bu.edu.sa*

Abstract

Traditionally human interacts with a computer by using keyboard and mouse. Considering person with handicapped from the wrist to the fingertip or amputated wrists or fingertips need alternative way; using voice or hand gesture. This work focuses on the use of hand-gesture image recognition. There are two main issues should be considered; less interactivity in static hand gesture recognition, and less accuracy in dynamic hand gesture recognition. This paper attempts to improve the accuracy of hand-gesture image recognition by experimenting simple deep learning neural network (DLNN). As this work uses a simple DLNN, the relation between the hidden layers is not considered. The number of hidden layers in the proposed architecture of the DLNN for the experiments vary from one to five.
With the aims to understand the effect of the number of neurons in the hidden layers, the DLNN is experimented using different numbers of hidden neurons. Six different types of hand gestures are considered. 800 videos on hand gestures taken from Vision for Intelligent Vehicles and Applications (VIVA) portal are used in the experiment. The data is divided into two; one as training data and another part is for testing. The best result is achieved when the DLNN uses two hidden layers with 250 neurons in the first hidden layer, and 100 neurons in the second hidden layer. The average of the achieved accuracy level is 77.56%. Experimental results also show that the more number of hidden layer causes over-fitting (does not make the recognition better). It is also observed that the increase of hidden layer number and hidden neurons only affect the accuracy of recognition of the trained dataset and does not improve the recognition of untrained dataset. This result is because the interrelation among the hidden layer are not considered.

Keywords: *deep learning; hand gesture images; human computer interface; neural network*

1. Introduction

Traditionally human interacts with a computer by using keyboard and mouse. A person with handicapped from the wrist to the fingertip or amputated wrists or fingertips needs alternative way; using voice or hand gesture. There are two main issues should be considered; less interactivity in Static hand gesture recognition, and less accuracy in Dynamic hand gesture recognition.
This paper attempts to improve the accuracy of hand gesture recognition by experimenting simple deep learning neural network. Deep learning is an approach that models human abstract thinking. One of the main benefits of deep learning over various machine-learning algorithms is its ability to generate new features from limited series of features located in the training dataset. Therefore, deep learning algorithms can create new tasks to solve current ones. Due to its improved data processing models, deep learning generates actionable results when solving data science tasks. While machine learning works only with labeled data, deep learning supports unsupervised learning techniques that allow the system become smarter on its own. The capacity to determine the most important features allows deep learning to provide concise and reliable analysis results.

2. Related Works

Many works on human hand-gesture image recognition and Deep Neural Network methods have already been carried out. For example, Neto, et al. [1] conducted a research about placement for static hand gesture in real-time and continued it by using Artificial Neural Network (ANN). The result of the research showed that the recognition level was very good (99.8% for 10 hand gestures and 96.3% for 30 hand gestures).
Ramjan, et al. [2] introduced a real-time dynamic hand-gesture image detection and recognition. Hand-gesture image was captured from video, extracting objects at the background image then use web camera for tracking object's movement. Then proceed with blurring, grey scaling, Hue Saturation Value (HSV), and blob detection processes. Finally, the image is matched with the image stored in a database to recognize the movement.
The best accuracy level in recognizing human hand-gesture image was 96.87% in a condition of sufficient lighting of the image was completed as reported by Deng [3].
Tang et al. [4] also proposed a static hand posture image recognition in real-time by using a special device called Kinect. The authors implemented Deep Belief Network (DBNN) and Convolutional Neural Network (CNN). The experimental results showed a good performance in the hand-gesture image detection and tracking process (98,063% for DBNN and 93,995% for CNN).

Erhan, et al. [5] introduced a method to track an object's location in an image and mark it with a box marker one image at a time. The proposed method uses Deep Convolutional Neural Network as a base for features extraction and model training to predict the location of the box marker. The accuracy for detection was 58.48% and for classification was 77.29%.
Molchanov, et al. [6] worked on recognizing dynamic hand-gestures while driving a car by using the Convolutional Neural Network (CNN). This research utilized the dataset from Vision for Intelligent Vehicles and Applications (VIVA). The result of the research gave an accuracy of 77.5% for classification process.

3. Proposed Method

Figure 1 illustrates the overall architecture of the proposed method. It consists of three sub-systems: Dataset, Pre-processing, and the Deep Neural Network. At the end, the proposed method is used as an interface to assist handicapped person to access applications in a computer. The following sub-sections describes the detail of each sub-system.

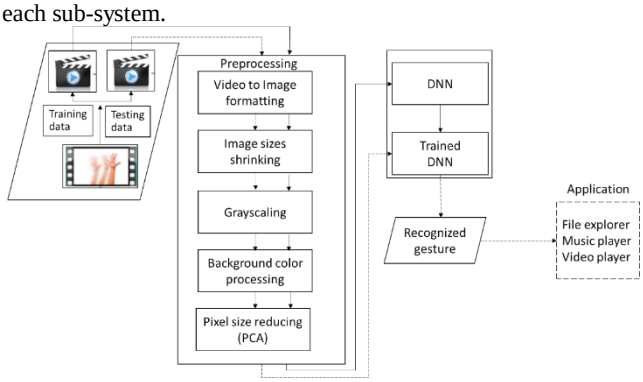

Fig. 1: The overall architecture of the propose method.

3.1. The Dataset

800 videos on hand gestures taken from Vision for Intelligent Vehicles and Applications (VIVA) portal are used in the experiment. They are available at:
http://cvrr.ucsd.edu/vivachallenge/index.php/hands/hand-gestures/.
The data is divided into two, one as training data another part is for testing.

3.2. The Pre-Processing

This paper uses standard procedures in image processing to reduce the image sizes but in the same time preserves the features of the images. In turn, the dimensionality (pixel size) of the selected image procedures decreases speed up the training stage. It starts with converting Video data into image frames. Then shrinking the image using Nearest-Neighbor algorithm, followed by the grey scaling. After that the Frame differencing technique is applied for background color processing, finally, Principle Component Analysis (PCA) is applied to reduce the pixel sizes.
This pre-processing is implemented to minimize the data dimensionality so that it can speed up the training process. The first minimization process will be carried out with six hand gestures and will reduce the pixel size from 4600 to 523.

3.3. The Deep Learning Neural Network

This paper uses simple Deep Learning Neural Network (DLNN). Figure 2 shows the architecture of the DLNN [7].

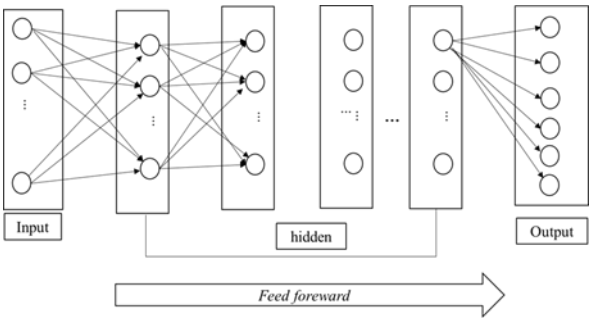

Fig. 2: The architecture of simple DLNN [7].

The DLNN is a Feed-forward Neural Network and trained by backpropagation method. Backpropagation method is one of algorithm, which is used in the training of multilayer perceptron. The procedure is as follows.

1. Initialization
Initially, every weight that connects each neuron is given a random value and distributed equally with small deviation using (1).

$$[-2.55/F_i, 2.55/F_i] \quad (1)$$

Where : F_i is the number of input from neuron i in the neural network.

2. Activation (*Feed-forward*)
Activation process or feed-forward put all the input into the neural network. The activation of the neural network is done by using input $x_1(p)$, $x_2(p)$, ..., $x_n(p)$ with the desired output are $y_{d1}(p)$, $y_{d2}(p)$, ..., $y_{dn}(p)$, p is the sum of iteration done.

- The actual output of every neuron in the hidden layer is calculated using (2).

$$y_i(p) = sigmoid\left[\sum_{i=1}^{n} x_i(p) w_{ij}(p)\right] \quad (2)$$

Where n is the number of *input* from neuron j in the hidden layer.

- *sigmoid* is the activation function.
- The value of the output of every neuron in the output layer is calculated using (3).

$$y_k(p) = sigmoid\left[\sum_{j=1}^{m} x_{jk}(p).w_{jk}(p)\right] \quad (3)$$

Where m is the number of input in neuron k in the output layer and *sigmoid* is the activation function.

3. Weight training
Each weight in the neural network is updated using backpropagation to the errors in the output layer.

- *Error* in every neuron in the output layer is calculated by (4).

$$\delta_k(p) = y_k(p) - y_{dk}(p) \quad (4)$$

Then the correction of the weight is counted using (5).

$$\Delta\omega_{jk}(p) = \alpha y_i(p).\delta_k(p) - \mu\Delta w_{jk}(p-1) \quad (5)$$

Where α is a learning rate that defines the speed of learning the backpropagation algorithm. μ is momentum rate that defines the size of the weight update.

The update for every weight which is connected to the neuron in the output layer is done by using (6).

$$\omega_{jk}(p+1) = \omega_{jk}(p) - \Delta w_{jk}(p) \quad (6)$$

- *Error* in every neuron in the hidden layer is calculated using (7).

$$\delta_j(p) = \left[\sum_{k=1}^{m} \delta_k(p).\omega_{jk}(p)\right].y_j(p).(1-y_j(p)) \quad (7)$$

Then the correction of the weight is calculated using (8).

$$\Delta\omega_{ij}(p) = \alpha x_i(p).\delta_{kj}(p) - \mu.\Delta w_{ij}(p-1) \quad (8)$$

The updated value of every weight that is connected to the neuron in the hidden layer is computed using (9).

$$\omega_{ij}(p+1) = \omega_{ij}(p) - \Delta w_{ij}(p) \quad (9)$$

4. Iteration

The training process of backpropagation algorithm is completed if the error criteria is already achieved as required.

4. Experimental Set Up, Results & Discussion

4.1. Experimental Set Up

Table 1 shows training parameters for DNN in the experiments.

Table 1: Training parameters for the DLNN.

Parameters	Experiment number				
	1	2	3	4	5
hidden layer #	2	2	4	4	4
neuron in 1st hidden layer	200	250	300	350	350
neuron in 2nd hidden layer	100	100	150	200	200
neuron in 3rd hidden layer	-	-	75	100	125
neuron in 4th hidden layer	-	-	75	75	75

The number of nodes in the input layer and in the output layer is 523 and 6 respectively. The six (6) output represent 6 hand gestures: move from top to bottom, bottom to top, left to right, right to left, back to front and front to back. The learning rate is 0.08, momentum rate is 0, number of epoch is 1250, and error threshold is 0.01.

The experiments are carried out using the following hardware specifications: PC with Processor: Intel Core i7, RAM: 8GB, O/S: Windows 8.1, 64 bit and Java programming language.

4.2. Experimental Results and Discussion

Figure 3 depicts the he graph of the DLNN training on five experiments. The error level of the training process fall below 0.01. It means that the DLNN can achieve a low error level for various number of the hidden layers and hidden neurons.

The trained DLNN is tested using both the training dataset and the testing dataset in the experiment, to detect the hand gestures. The experiments are repeated 10 times each.

The results are shown in Table 2. The best result is achieved in the second experiment where the DLNN uses two hidden layers with the number of neurons of 250 in the first hidden layer, and 100 neurons in the second hidden layer. The average of the achieved accuracy level is 77.56%. Table 2 shows also that the more number of hidden layer causes over-fitting (does not make the recognition better).

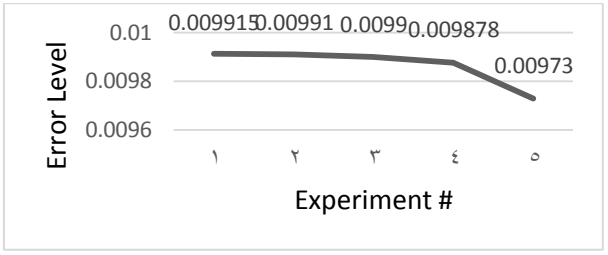

Fig. 3: Training results.

From the experiments, it is also observed that the increase of hidden layer number and hidden neurons only affect the accuracy of recognition of the trained dataset but did not improve the recognition of untrained dataset. This result is due to the fact that the interrelation among the hidden layer is not taken into account.

Table 2: Testing results.

Hand gestures	Experiment number				
	1	2	3	4	5
Top- bottom	71.487%	78.88%	75.1%	74.78%	73.07%
Bottom-top	76.86%	80.67%	84.62%	81.17%	82.26%
Right- left	86.21%	92.77%	81.91%	92.77%	85.80%
Left- right	85.59%	88.63%	88.66%	81.27%	81.27%
Back- front	66.77%	56.65%	59.01%	63.06%	48.04%
Front- back	41.24%	67.77%	56.54%	55.65%	72.34%
Average	71.36%	77.56%	74.20%	74.78%	73.76%

5. Conclusion

Simple Deep Learning Neural Network is able to recognize six hand gestures. The DLNN using two hidden layers with number of neuron in first and second hidden layer is 250 and 100, respectively gave the best accuracy of 77.56%. Increasing the number of hidden layers does not improve the accuracy, due to not considering any inter-relation between hidden layers.

Experimenting the hand-gestures video/image recognition in real-time using more proper DLNNs, such as Convolution Neural Network using Tensorflow is currently authors' ongoing and future works.

References

[1] Neto P, Pereira D, Pires JN, & Moreira AP,"Real-time and Continuous Hand Gesture Spotting: An Approach Based On Artificial Neural Networks" , *Proceedings of the IEEE International Conference on Robotic and Automation (ICRA)* 6-10 May 2013, Karlsruhe, Germany, pp. 178-183, http://dx.doi.org/10.1109/ICRA.2013.6630573.

[2] Ramjan MR, Sandip RM, Uttam PS. & Srimant WS, (2014), Dynamic hand gesture recognition and detection for real time using human computer interaction. *International Journal of Advance Research in Computer Science and Management Studies (IJARCSMS)*, 2(3): 425-430.

[3] Deng L, & Yu D, Deep Learning Methods and Applications. 978-1-60198-814-0. Now: Netherland, (2014).

[4] Tang A, Lu K, Wang Y, Huang J, & Li H, (2013), A real-time hand posture recognition system using deep neural networks. *ACM Transactions on Intelligent Systems and Technology*, 9(4): 39:1-21.

[5] Erhan D, Szegedy C, Toshev A, & Anguelov D, "Scalable Object Detection using Deep Neural Networks", *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition*, 23-28 June 2014, Washington DC, USA, pp. 2155 – 2162, http://dx.doi.org/10.1109/CVPR.2014.276.

[6] Molchanov P, Gupta S, Kim K, & Kautz J, "Hand Gesture Recognition With 3D Convolutional Neural Networks", *Proceedings of the 28th IEEE Workshop on Computer Vision and Pattern recognition*, 7-12 June 2015, Boston MA, USA, pp.1-7, http://dx.doi.org/ 10.1109/CVPRW.2015.7301342.

[7] Negnevitsky M, *Artificial Intelligence: A guide to intelligent systems*. 2nd Edition. Pearson Education Limited: Upper Saddle River, (2005).