

# Recognition system for facial expression by processing images with deep learning neural network

Holman Montiel Ariza<sup>\*1</sup>, Henry Hernández Martínez<sup>2</sup>, Luz Andrea Gaviria Roa<sup>3</sup>

<sup>1</sup>Universidad Distrital Francisco José de Caldas, Facultad Tecnológica,  
Cll 68 D Bis A Sur No. 49F–70, Bogotá D.C., Colombia

<sup>2</sup>Universidad Nacional de Colombia, Departamento de Ingeniería de Sistemas e Industrial,  
Bogotá D.C., Colombia

<sup>3</sup>Fundación Universitaria Panamericana, Facultad de Ingeniería, Bogotá D.C., Colombia

\*Corresponding author, e-mail: [hmontiela@udistrital.edu.co](mailto:hmontiela@udistrital.edu.co)

## Abstract

The recognition systems of patterns in images are mechanisms that filter the information that provides an image to highlight the area of interest for the user. Usually, these mechanisms are based on mathematical transformations that allow the processor to perform interpretations based on the geometry or shape of the image. However, the strategies that implement mathematical transformations are limited, since the effectiveness of these techniques is reduced by changing the morphology or resolution of the image. This paper presents a partial solution to this limitation with a digital image processing technique based on a deep learning neural network (DNN). This technique incorporates a mechanism that allows the DNN to determine the facial expression of a person, based on the segmented information of the image of their face. By segmenting the image and processing its characteristics in parallel, the proposed technique increases the effectiveness of recognizing facial gestures in different images even when modifying their characteristics.

**Keywords:** deep learning neural network, face recognition, processing images, recognition systems of patterns

Copyright © 2019 Universitas Ahmad Dahlan. All rights reserved.

## 1. Introduction

The digital image processing is a methodology to implement a technique that is used in order to improve or restore an image that undergoes some change during its acquisition process. For example, reducing the resolution of a signature when scanning a document. Due to the large number of techniques these are often confused, but they are not the same, since a group of techniques is dedicated to the restoration of images and are used to make assumptions about missing information. The other group contains improvement techniques that allow the user to eliminate or highlight image information to determine areas or regions of interest [1, 2].

The automation of these techniques has allowed the generation of strategies and algorithms for the processing of video or sequences of images in a continuous manner. That is, the same logical structure is used to process large volumes of information, which has encouraged the development of advanced techniques such as artificial vision, closed security systems, the analysis of human gait or the location of geometric patterns [3-8]. Although there is a lot of techniques to process images, these have some limitations, because they depend on the capture device and the resolution of the image to function in an appropriate manner. The lower the resolution of the image, the more mathematical operators, masks or filters used to determine the regions of interest in an image will be even greater. In other words, by increasing the number of operators, the amount of computational resources to determine the information of interest also does [9-11].

This limitation was addressed in different ways in different areas of engineering using intelligent systems or expert systems. One of the commonly used techniques is the Viola Jones algorithm, since it implements a learning algorithm that identifies rosters in sets of images [1, 4]. Although, this technique is easily programmable in devices with a low level of processing, it does not perform interpretations of the mood of the person [12-17]. The problem of identifying states of mind is quite complex and is not a limitation of the algorithm of Viola Jones only, since the identification of feelings from patterns is a subject that is under study for the development of

security systems, medical treatments, among others [18-23]. In this paper there is proposed a strategy to determine moods from facial expression, with a low cost computational algorithm. This algorithm is described in the following sections which are organized as follows: section 2 contains a description of the developed technique. In section 3, there is presented a description of the experiment that was used to test this algorithm. In section 4, the results obtained with the experiment are described. In section 5, there are the conclusions that were built from the results obtained.

## 2. Materials and Methods

As it was said in the previous section, the techniques of image processing are usually based on mathematical transformations that allow to highlight or indicate regions of interest for the user. At present, it is tried to improve this type of techniques by implementing segmentation and classification methods to reduce the amount of parameters necessary when performing an operation. However, the use and application of classifiers in the process of pattern recognition in an image is a subject under study, due to the large number of applications that this type of technique can have. Among them, the recognition of facial expressions and their relationship with emotions from an image given by the user is an issue in development, since, conventionally, image processing is used to identify parts of the human body and not for understand its operation. This article proposes a contribution to this topic with the development of a character identification system based on a deep learning neural network (DNN), which identifies a person's facial expression and associates it with a feeling that can feel. Finally, structure of the DNN and its previous preparation to develop this work is described below.

### 2.1. Preparation and Processing of a Digital Image

In relation to what was said before, a digital image is constructed with a numerical matrix which represents a two-dimensional image. The dimensions of the matrix vary depending on the resolution of the image and the number of matrices that are used to represent the same scene changes depending on the number of colors. For example, a single binary coefficient matrix is required to represent a black and white image.

There are several ways to obtain a digital image, among them are scanners and digital cameras. An advantage of these devices is that they allow themselves to apply transformations to modify the image before storing it, such as filters to eliminate background light, crop or rotate the scene. However, the processing capacity of these devices is limited, therefore, processing once the acquisition of the image in a computer is the most common [1].

Traditional computer image processing programs allow transformations or beautification of digital images only, due to this, software-based applications have been devised to increase the amount of operations available to users to understand the information that an image provides. Some of these applications allow information to be recovered by reconstructing the image based on assumptions or eliminating characteristics that attenuate the information of interest [2-5].

The information of interest in an image is usually highlighted or extracted by geometric transformations, which allow to indicate patterns or groups of pixels that contain characteristics previously defined by the user. Among the simplest is the system for detecting geometric figures that is based on the number of points a figure can have, and among the most complex are the systems that identify characteristics of an image using intelligent systems.

Processing using intelligent systems requires pre-processing of the image, because the particular characteristics that one wants to find when implementing this system must be pointed out. One way to do this is to use a histogram to teach the intelligent system the relative frequency with which groups of colors appear in an image. The most common way to estimate the values of a histogram is based on: decreasing the number of dimensions of the image by converting its color format to gray scale and estimating the frequencies with the expression shown in (1). Where the dimensions are  $w$  (width) and  $h$  (height),  $n$  represents the gray levels and  $N_{ng}$  is the number of pixels [5].

$$h(ng) = \frac{N_{ng}}{w \cdot h} \quad (1)$$

In this paper, the pre-processing of the image was done in the following way. First, the number of dimensions of the image is reduced by changing the color format to gray scale. Then, the image is segmented into at least four parts that are the face, the mouth and the eyes (the glasses count as one eye). Finally, each segment of the image is transformed into a histogram and stored in an array that will later be processed by the DNN, see Figure 1. The particular characteristics of the DNN and its relation with the pre-processing of the image are described in the following numeral.

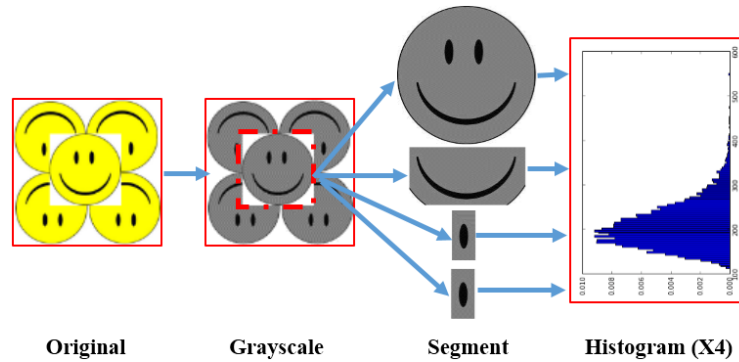


Figure 1. Schematic of image pre-processing

## 2.2. Patterns Recognition with a DNN

Neural networks of deep learning (DNN) are extended models of traditional neural networks, but unlike them DNN generate models to represent large volumes of information in a simple way. Among the most common forms or models to represent groups of data with this type of network are the classifiers and the approximations by regressions. On the one hand, classifiers are models that solve problems of classes in which it is intended to group objects with defined characteristics. On the other hand, regression approximations are numerical representations generated to associate groups of numbers. In both cases, the model is a black box, that is, the DNN estimates an output value from certain input information. However, the user never knows the mathematical expression or form of the classifier that makes up the DNN [14].

The topology or form of the DNN depends on certain parameters defined by the user, among which are the number of entrances, exits, neurons and hidden layers, the form of the activation function and the algorithm of training or reduction of the error [24]. The number of inputs and outputs varies depending on the group of training data, that is, the number of inputs is determined by the independent variables that allow estimating the output value and the number of outputs depends on the number of variables that change depending on the entrance. The number of neurons and hidden layers are stochastic values, see in (2), determined by the user when designing the network, it should be taken into account that increasing the number of neurons ( $\delta$ ) and hidden layers ( $\varepsilon$ ) increases the DNN accuracy and decreases the performance of the processor, since, the amount of numerical calculations is increasing.

$$\{\delta, \varepsilon\} \in \mathbb{Z} \mid \{\delta, \varepsilon\} \geq 0 \quad (2)$$

There is no way to determine the exact number of hidden layers and neurons for each DNN, because each application has a different training data set. This is because each neuron stores a value called weight, which is responsible for modifying the output value of each neuron by increasing or decreasing the input value. In addition, the weight is accompanied by an activation function, which is responsible for limiting the output value of each neuron. Another feature of the weights is that to increase the accuracy of the output value of the DNN, different training algorithms are used, which automatically modify the weights of each neuron to reduce the margin of error between the output of the network and the training data [14, 20, 25].

In this paper it was used a DNN with 1024 entries, one (1) output, 1200 neurons, ten (10) hidden layers, a random function with uniform distribution to assign initial values to the weights, a rectified linear unit activation function, a descendant gradient training function and a function of similarity measurement between the network output and the cosine-based training data. This measure of similarity assumes that the data groups are vectors and their objective is to find an angle between them, that is, the output ( $\alpha$ ) of (3) varies between -1 and 1 (meaning the same) to indicate an degree of correlation between vectors (A and B) and indicates 0 when the vectors are totally different ( $n$  = number of vector components).

$$\alpha = \cos(\theta) = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \tag{3}$$

The training of the network was carried out with a group of samples containing the information of 1000 photographs of segmented faces as mentioned in the previous numeral. The information in these photographs was grouped into a vector where each gray component estimated with the histogram becomes an attribute and each group of four (4) images becomes an instance shown in Table 1. However, when the image can not be fully segmented, a value of zero (0) is assigned to each corresponding attribute.

By grouping the input data, a new attribute was created that is associated to each instance as the emotion that the person feels at the time of capturing the photograph. This instance encodes the emotions with four (4) integers (0 = Neutral, 1 = Happy, 2 = Sad, 3 = Angry), in order to convert this problem of classifying photographs into a polynomial approximation problem. This approach was generated by training the DNN according to the structure of Figure 2 which has 1024 entries generated by each histogram and a user defined output.

Finally, the combination of image segmentation strategies and the generation of the neural network allowed the development of an application. This application allows the user to import groups of images and apply the Haarcascade algorithm to each image to identify the face and each of the regions of interest [5]. Then a geometric transformation was made to extract the regions of interest as individual images and later the routine estimates the histogram of each segment. Once all the images have been encoded, they are grouped in a database and the DNN is trained. The model of the DNN is exported as a function, which works with a graphical interface that allows to import an image or take a photograph to know the mood of the person.

Table 1. Grouping of Segmented Images

		Attributes			
		Face	Mouth	Right eye	Left eye
Instances	$i_1$	$[x_1, x_2, \dots, x_{256}]$	$[x_{257}, x_{258}, \dots, x_{512}]$	$[x_{513}, x_{514}, \dots, x_{768}]$	$[x_{769}, x_{770}, \dots, x_{1024}]$
	$i_2$	$[x_1, x_2, \dots, x_{256}]$	$[x_{257}, x_{258}, \dots, x_{512}]$	$[x_{513}, x_{514}, \dots, x_{768}]$	$[x_{769}, x_{770}, \dots, x_{1024}]$
	$i_3$	$[x_1, x_2, \dots, x_{256}]$	$[x_{257}, x_{258}, \dots, x_{512}]$	$[x_{513}, x_{514}, \dots, x_{768}]$	$[x_{769}, x_{770}, \dots, x_{1024}]$
	...			...	
	$i_n$	$[x_1, x_2, \dots, x_{256}]$	$[x_{257}, x_{258}, \dots, x_{512}]$	$[x_{513}, x_{514}, \dots, x_{768}]$	$[x_{769}, x_{770}, \dots, x_{1024}]$

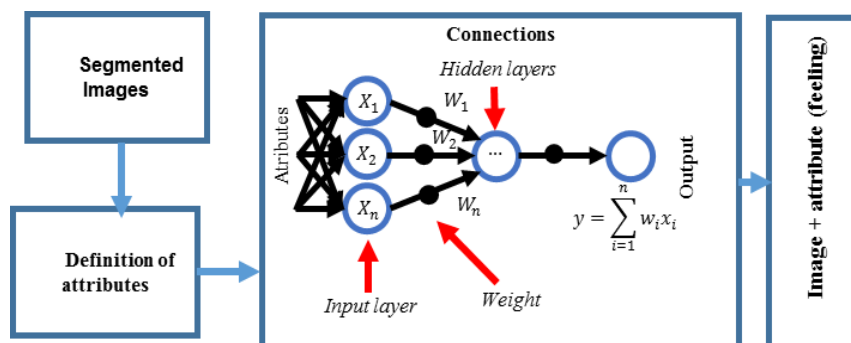


Figure 2. Block diagram of the DNN training process

### Algorithm 1. Application Developed

```













1. DNN Program()
2.     MA←Import image database ().
3.     MA←Apply Haarcascade algorithm (MA).
4.     MS←Segment the images (MA).
5.     HI← Create histograms (MS)
6.     DN← Define DNN.
7.     DN← Train DNN (DN, HI).
8.     Export Trained Model (DN).
     /** Function to evaluate mood ***/
9. Evaluate_Image ()
10.    DN ← Import Trained Model ().
11.    IM ← Read Image ().
12.    IM ← Segment and create Histogram (IM).
13.    SA <- Evaluate Image using DNN (EM, DN).
14.    SA <- Floor Function (SA).
15.    If SA = 0 then Print "Neutral"
16.    Else If SA = 1 then Print "Happy"
17.    Else if SA = 2 then Print "Sad"
18.    Else if SA = 3 then Print "Angry"
19. End DNN

```

### 3. Experiment

The application presented in this paper was made using the libraries KERAS, TENSORFLOW and OPENCV 3.4.0 of PYTHON 3.5.8 in the Eclipse IDE 4.9.0 interpreter with the help of the PyDev third party add-on. Another feature of the application is that it was tested on a computer with an Intel®inside CORE™ i3 processor and 8 GB of RAM. In addition, the application was validated using a set of images available in a repository [26]. The set of images consists of 5026 photographs of people of different ages, gender and race, some of which use accessories such as glasses or monocles to reduce the effectiveness of the recognition algorithm. In addition, each person was photographed several times with an orientation (right sagittal, left sagittal, frontal) and resolution (32x30, 64x60, 120x128) different shown in Table 2. In this paper a set of 923 images was used from the image database that have a label that indicates the mood of the person, a frontal orientation and different resolution. From the 923 images, 600 were used to train the DNN and the rest were used to check the results provided by the DNN once trained.

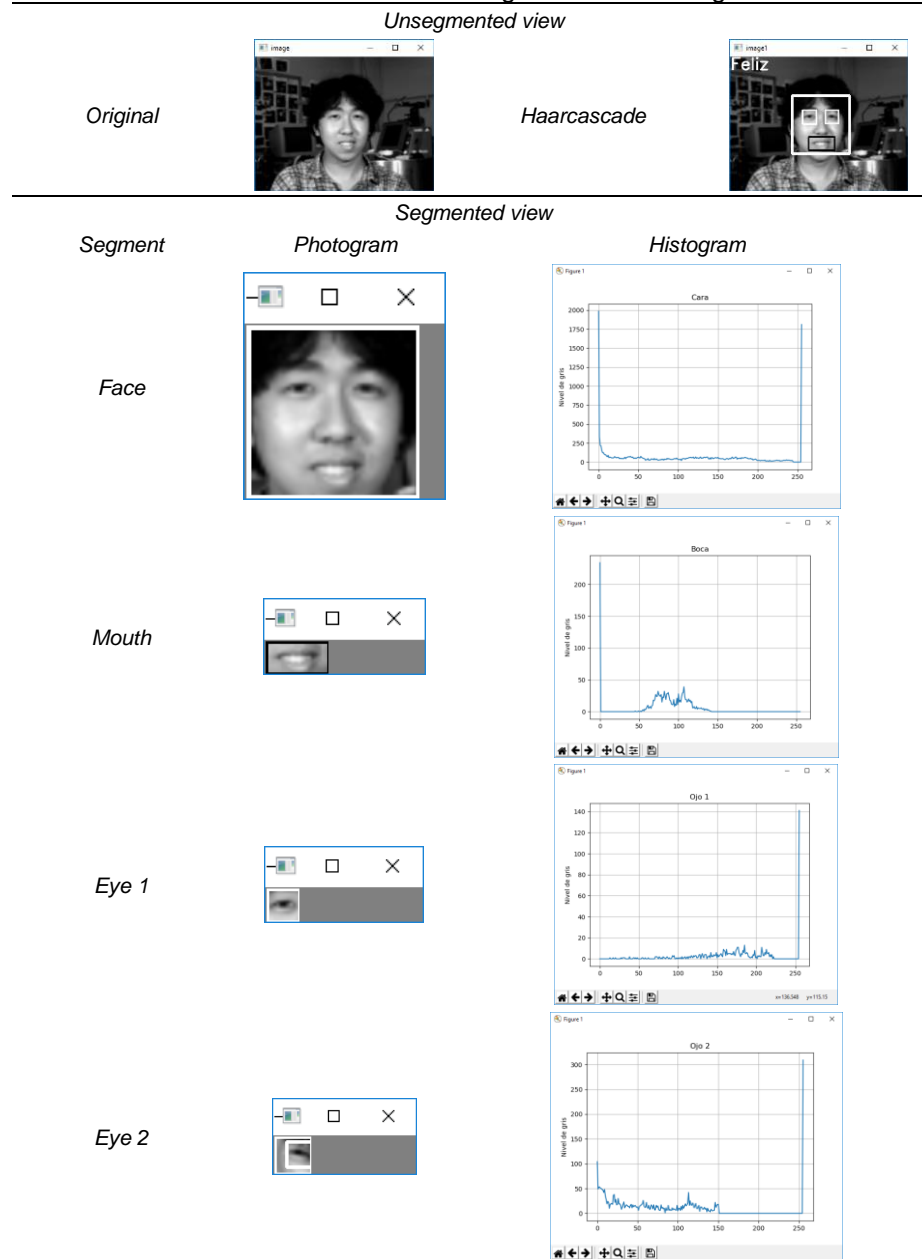
Table 2. Examples of the Photographs of the Base of Grayscale Images

	32x30	64x60	120x128	120x128
Sad				
Angry				
Neutral				

**4. Results**

As mentioned before, the proposed strategy segments the image (regardless of resolution) into several parts and each one becomes a histogram that represents the grayscale of the image, see Table 3. Also, when one creates the histogram can also assign a label that indicates the mood of the person. Finally, several configurations of the neural network were tested to evaluate which is the most appropriate when trying to solve this type of problems. In total, configurations were tested with increasing of one hundred (100) in one hundred (100) in each network training the number of neurons and one (1) in one (1) number of hidden layers. In total 20 DNNs with different topologies were built and their behavior is presented in Figure 3 in which the solid line represents error and the dotted line represents the margin of error. These lines were calculated based on the images from the image database, that is, they are the results when performing 1000 iterations of training and evaluating the DNN with the base of images only.

**Table 3. Process of Generation of the Histograms and the Segments of the Image**



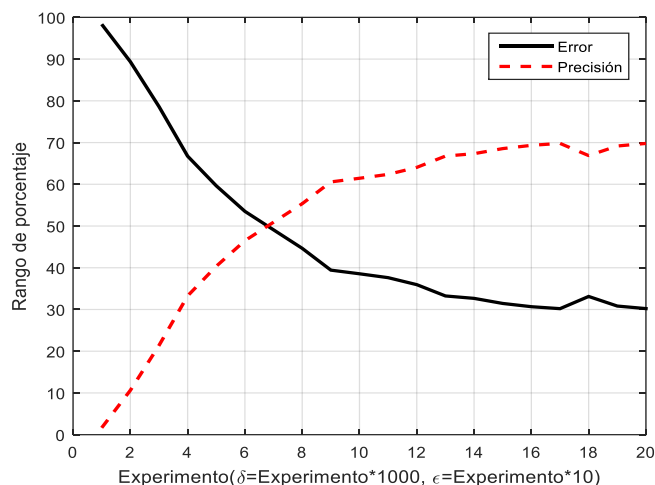


Figure 3. Behavior of the DNN with different configurations

## 5. Conclusion

Figure 3 shows that it is possible for the proposed strategy to reduce the margin of error by estimating what the person in the photograph is feeling with a margin of error close to twenty percent (20%), that is, that of every ten (10) photographs can not predict the facial expression of two (2). This is because in some cases the light affects the detection of the glasses and the Haarcascade algorithm is not perfect and detects more than two eyes or more than one mouth.

One of the advantages of this emotion prediction methodology is that it can be generated for even more attributes, that is, if one has a broader database then could pre-tell more emotions. It can be said that the margin of error is compensated by the flexibility of the DNN, since, compared to traditional classifiers, the prediction by means of polynomials allows establishing a generalized model of DNN to solve this type of problems. In addition, the DNN could be adjusted to different image acquisition elements (cameras or scanners), because during the training the weights are adjusted taking into account images with different resolutions which reduces the effect of the systematic error induced by the environmental conditions.

As can be seen in Figure 3 the DNN arrives at a steady state from training number 13, this means that even if the number of hidden layers or neurons is increased, it is not possible to improve the performance of the DNN. On the one hand, it is possible that, by modifying the functions of activation, optimization and generation of the initial weights, it improves the performance of the DNN. On the other hand, the objective of this paper was to find the most appropriate configuration, so that the DNN is executed with few computational resources and among the evaluated configurations the most adequate was the one presented in section 2.

## References

- [1] Nehru M, Padmavathi S. *Illumination invariant face detection using viola jones algorithm*. 2017 4<sup>th</sup> International Conference on Advanced Computing and Communication Systems (ICACCS). Coimbatore. 2017: 1-4.
- [2] Vikram K, Padmavathi S. *Facial parts detection using Viola Jones algorithm*. 2017 4<sup>th</sup> International Conference on Advanced Computing and Communication Systems (ICACCS). Coimbatore. 2017: 1-4.
- [3] Alyushin MV, Lyubshov AA. *The Viola-Jones algorithm performance enhancement for a person's face recognition task in the long-wave infrared radiation range*. 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EConRus). Moscow. 2018: 1813-1816.
- [4] Shamia D, Chandy DA. *Analyzing the performance of Viola Jones Face Detector on the LDHF database*. International Conference on Signal Processing and Communication (ICSPC). Coimbatore. 2017: 312-315.

- [5] Djamaluddin D, Indrabulan T, Andani, Indrabayu, Sidehabi SW. *The simulation of vehicle counting system for traffic surveillance using Viola Jones method*. 2014 Makassar International Conference on Electrical Engineering and Informatics (MICEEI). Makassar. 2014: 130-135.
- [6] Guojin C, Yongning L, Miaofen Z, Wanqiang W. *The image auto-focusing method based on artificial neural networks*. 2010 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications. Taranto. 2010: 138-141.
- [7] Omer MK, Sheta OE, Adrees MS, Stiawan D, Riyadi MA, Budiarto R. Deep Neural Network for Heart Disease Medical Prescription Expert System. *Indonesian Journal of Electrical Engineering and Informatics*. 2018; 6(2): 217-224.
- [8] Zhang D, Han X, Deng C. Review on the research and practice of deep learning and reinforcement learning in smart grids. *Journal of Power and Energy Systems*. 2018; 4(3): 362-370.
- [9] Guojin C, Miaofen Z, Honghao Y, Yan L. *Application of Neural Networks in Image Definition Recognition*. 2007 IEEE International Conference on Signal Processing and Communications. Dubai. 2007: 1207-1210.
- [10] Kim HI, Lee SH, Ro YM. *Face image assessment learned with objective and relative face image qualities for improved face recognition*. 2015 IEEE International Conference on Image Processing (ICIP). Quebec. 2015: 4027-4031.
- [11] P Srinivasa, RK Nadesh, NC Senthil. Robust Face Recognition Using Enhanced Local Binary Pattern. *Bulletin of Electrical Engineering and Informatics*. 2018; 7(1): 96-101.
- [12] Sudhakar K, Nithyanandam P. An Accurate Facial Component Detection Using Gabor Filter. *Bulletin of Electrical Engineering and Informatics*. 2017; 6(3): 287-294.
- [13] Baltrušaitis T, Robinson P, Morency LP. *OpenFace: An open source facial behavior analysis toolkit*. 2016 IEEE Winter Conference on Applications of Computer Vision (WACV). Lake Placid. 2016: 1-10.
- [14] Feng R, Leung CS, Sum J, Xiao Y. Properties and Performance of Imperfect Dual Neural Network-based k-WTA Networks. *IEEE Transactions on Neural Networks and Learning Systems*. 2015; 26(9): 2188-2193.
- [15] Chu WS, De la Torre F, Cohn JF. Selective Transfer Machine for Personalized Facial Expression Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2017; 39(3): 529-545.
- [16] Zen G, Porzi L, Sangineto E, Ricci E, Sebe N. Learning Personalized Models for Facial Expression Analysis and Gesture Recognition. *IEEE Transactions on Multimedia*. 2016; 18(4): 775-788.
- [17] Zhiqi Y. *Gesture learning and recognition based on the Chebyshev polynomial neural network*. 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference. Chongqing. 2016: 931-934.
- [18] Lee D, Lee J. Equilibrium-based support vector machine for semisupervised classification. *IEEE Trans. Neural Netw.* 2007; 18(2): 578-583.
- [19] Terrence J. *The Deep Learning Revolution*. The MIT Press. 2018: 1-10.
- [20] Vigneswaran KR, Vinayakumar R, Soman KP, Poornachandran P. *Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security*. 2018 9<sup>th</sup> International Conference on Computing, Communication and Networking Technologies (ICCCNT). Bangalore. 2018: 1-6.
- [21] Saad MM, Jamil N, Hamzah R. Evaluation of Support Vector Machine and Decision Tree for Emotion Recognition of Malay Folklores. *Bulletin of Electrical Engineering and Informatics*. 2018; 7(3): 479-486.
- [22] Povoda L. *Sentiment analysis based on Support Vector Machine and Big Data*. 39<sup>th</sup> IEEE International Conference on Telecommunications and Signal Processing (TSP). Vienna. 2016: 543-545.
- [23] Kaur H, Mangat V. *A survey of sentiment analysis techniques*. 2017 International Conference on Ini-SMAC (IoT in Social, Mobile, Analytics and Cloud (I-SMAC)). India. 2017: 921-925.
- [24] Adege AB. *Applying Deep Neural Network (DNN) for large-scale indoor localization using feed-forward neural network (FFNN) algorithm*. 2018 IEEE International Conference on Applied System Invention (ICASI). Chiba. 2018: 814-817.
- [25] Zegers P, Sundareshan MK. Trajectory generation and modulation using dynamic neural networks. *IEEE Transactions on Neural Networks*. 2003; 14(3): 520-533.
- [26] D Dua, K Taniskidou. UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science. 2017.