Engineering and Technology Journal
Journal homepage: engtechjournal.org

Hand Gesture Recognition of Static Letters American Sign Language (ASL) Using Deep Learning

Abdulwahab A. Abdulhussein[a*], Firas A. Raheem[b]

[a,b] Affiliation: Control and Systems Engineering Department, University of Technology -Iraq.
eng.wahab86@gmail.com

*Corresponding author.

Submitted: 26/08/2019 Accepted: 25/10/2019 Published: 25/06/2020

K E Y W O R D S

ASL, CNN, Accuracy.

A B S T R A C T

An American Sign Language (ASL) is a complex language. It is depending on the special gesture stander of marks. These marks are represented by hands with assistance by facial expression and body posture. ASL is the main communication language of deaf and people who have hard hearing from North America and other parts of the world. In this paper, Gesture recognition is proposed of static ASL using Deep Learning. The contribution consists of two solutions to the problem. The first one is resized with Bicubic static ASL binary images. Besides that, good recognition results in of detection the boundary hand using the Robert edge detection method. The second solution is to classify the 24 alphabets static characters of ASL using Convolution Neural Network (CNN) and Deep Learning. The classification accuracy equals to 99.3 % and the error of loss function is 0.0002. According to 36 minutes with 15 seconds of elapsed time result and 100 iterations. The training is fast and gives the very good results, in comparison with other related works of CNN, SVM, and ANN for training.

How to cite this article: A. A. Abdulhussein and F. A. Raheem, "Hand gesture recognition of static letters American sign language (ASL) using deep learning," Engineering and Technology Journal, Vol. 38, No. 06, pp. 926-937, 2020.
DOI: https://doi.org/10.30684/etj.v38i6A.533

1. Introduction

In the last decade, most skilled translators are made communication between the deaf and low hearing people. This communication is confirmed by gesture recognition. Gesture recognition is defined as the human body or limbs like fingers, arms, and etc. movement. There are many applications of gesture recognition: Interface between human and computer, video games, polygraph device security, biometric applications and etc. [1, 2]. The static ASL image recognition is implemented using many methods. The moment invariant approach is explained with a little form contribution of feature gesture segmentation. An ASL of hand gestures dataset consists of 2425 images with 5 individuals. The weaknesses of their paper are: the ASL images are rotated and processed by open source tools like ImageMagick. Besides that, there are no curves or results about the error or accuracy [3].

2. Related Work

After gesture recognition of static ASL is done, the classification operation must apply in order to classify the static characters of ASL. The hand gesture boundary is detected using edge technique. The Localized Contour Sequence (LCS) method is presented to classify the static ASL. The classification accuracy is 97.4% [4]. The shape hand algorithm is classified 24 static letters for ASL. Gestured recognition is done depending on the shape method. An accuracy of 180 points of landmark equals to 79.9% [5], Hardware design with the glove sensor using a neural network is introduced. The design is used DAQ 6212. It is useful for interfacing the PC with sensors. The gesture recognition accuracy equals to 90.19% [6]. A neural network is classified as static ASL hand images. The skin color depending on extraction is used for ASL alphabets and digits. The classification accuracy is 73.68 % [7]. Artificial Neural Network (ANN) is chosen to classify the static images ASL. The system consists of several stages like detection, the process of the hand, extracting the features, ANN training, and identification of images. The accuracy is 98 % [8]. The deep neural network learning may be defined as a method of machine learning, that includes networks of neural through more than one of the hidden layers. It has many applications such as recognition of the face, speech, and processing language mission. Deep learning consists of the Convolution Neural Networks (CNNs) and stacked denoising of auto-encoders to recognize the 24 static ASL letters. The accuracy rates are 91.33 % and 92.83 % for data [9]. Alphabet sign language recognition for Peru country is proposed to extract the digital image to decrease the image's noise. In addition, to process the dissimilarity under different illumination. CNN is a classified hand image gesture. The first results for CNN accuracy equal to 95.37 % and the second result for CNN accuracy equal to 96.20 % [10]. In paper [11] Support Vector Machine (SVM) and Artificial Neural Network (ANN) has been used for training. The static ASL alphabet is recognized by (SVM and ANN). The researcher has collected a 100 deep histogram of oriented gradient features per alpha. The accuracy is equal to 94.7 %. In [12] Edge Oriented Histogram (EOH) is recognized as the static ASL alphabet. Their recognition rate has 88.26% within 0.5 seconds. In paper [13] edge oriented histogram (EOH) and multi-class SVM techniques have been used. The average system precision achieved a 93.75% success rate. The precision is used with 64 characteristics.

The contributions of the proposed work in this paper can be stated as follows:

1) A new approach has been proposed which represented by a new block diagram structure.
2) Rapidly increasing the accuracy of the overall classifications with decreasing the loss error.
3) Increasing the speed of recognition by using the momentum parameter.

3. The proposed of Hand Gesture Recognition of Static ASL using Deep Learning with CNN

In Figure 1, the proposed static ASL hand gesture process using Deep Neural Learning is explained as below:

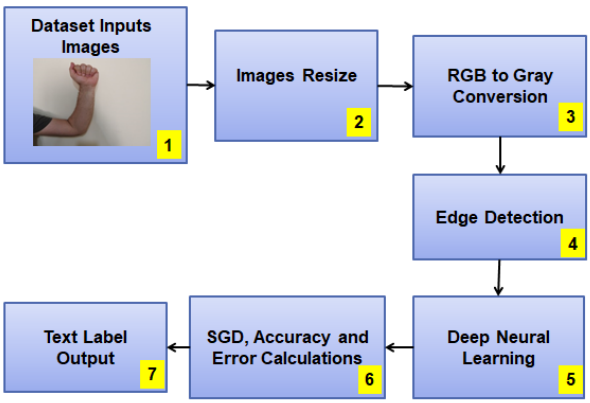

Figure 1: Proposed block diagram of hand gesture recognition of static ASL using deep learning

I. Data inputs images

The first block is loaded and reading each image of the dataset. The Dataset contains 24 static ASL letters and 10 hand images configurations for each letter.

II. Images resize

The second block contains resizing each image into (227 X 227) size of the Bicubic interpolation method. Interpolation is the approach that uses to transfer the digital image from one image accuracy to another. This process is performed by keeping the quality of the image. Besides that, an interpolation algorithm can expand or reduce the number of pixels of the image [14,15]. Bicubic interpolation operation is chosen when the speed does not important in the computation. The pixel $B(r', c')$ is formed with interpolating the nearest 4 x 4 pixels that begins with $A(r, c)$ and ending to $A(r + 2,c + 2)$. The original image has the two-scale factors denoted as Sr and Sc of A.
Sr and Sc are the rows and column scale factors correspondingly as seen in Figure 2.

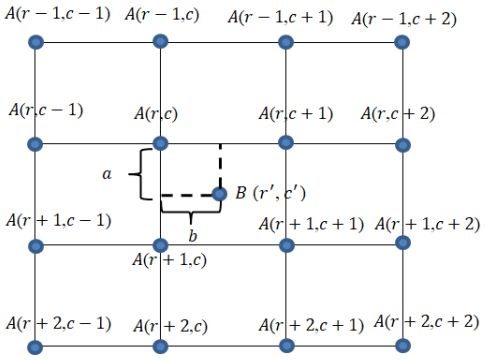

Figure 2: Bicubic Interpolation method [15]

The new scale factors of image B are calculated as Eq. (1) and Eq. (2) [15].

$$Sr = \left(\frac{R}{R'}\right) \quad \ldots. (1)$$

$$Sc = \left(\frac{C}{C'}\right) \quad \ldots. (2)$$

The expression rate of B (r', c') is evaluated as in Eq. (3) for the nearest 16 pixels interpolating [15].

$$B(R', C') = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} A_{ij} \quad \ldots (3)$$

When, A_{ij}presented as the 16 nearest pixel values for $B(r', c')$.The a_{ij} factors are calculated with Lagrange Eq. (4) [15].

$$a_{ij} = a_i \text{ X } a_j \quad \ldots. (4)$$

The a_i and bi are computed as Eq. (5) and Eq. (6)

$$a_i = \prod_{k=0}^{3} \prod_{k \neq i} \frac{(r' - [Sr \times (x\ +\ k)])}{[Sr \times (x + i)] - [Sr \times (x\ +\ k)]} \quad \ldots (5)$$

$$bi = \prod_{k=1}^{3} \prod_{k \neq j} \frac{(c' - [Sc \times (y\ +\ k)])}{[Sc \times (y + i)] - [Sc \times (y\ +\ k)]} \quad \ldots. (6)$$

a_i and b_i are denoted as the ith rows and the jth columns of A respectively. The k value is not equal to i. Temporarily, the values x and y of every row and columns were partitioned with the scale factors of Sr and Sc.

III. RGB to Gray Conversion

In this block, the color images are changed from the RGB images to the gray images with the constant weights using the Eq. (7) below:

$$Gray_{image} = Wr * R + Wg * G + Wb * B \quad (7)$$

$$Wr + Wg + Wb = 1 \quad (8)$$

Where red is sampled to R, green is marked as G, and B is called as blue, Wr, Wg, Wb are the constant-coefficient weights that equal to 0.299, 0.587, and 0.114 for R, G, and B respectively. The Weight is equal to the summation of R, G, and B weights as the equation above [16,17,18].

IV. Edge detection

It is one of the segment methods. The segment is defined as the extraction of the image into areas. The best approaches are used in this work, is the edge detection approach. In this block, the image edges are extracted and converted from the grayscale to the binary image. In this paper, edge detection is used for the extraction of the fingers edge hand for ASL. An edge segmentation of an image has many operators such as Canny, Prewitt, Robert, and Sobel. They are used for suitable luminous density, giving quick solutions, and processing the online image detection [19].

In this work, Robert edge detection is chosen because it helps to process computer images and vision. Robert operator can estimate the gradient of the image using a discrete approach.

The discrete approach performs by computing the sum of dissimilarities rectangles between transversal pixels that adjusted. It speedily measures the 2D spatial slope of the input image gradient of 2 seeds. In addition, the two kernels (Gx and Gy) are rotated to 90 degrees. This operation is used when the seeds are alternated from one to extra as seen in Figure 3 [20].

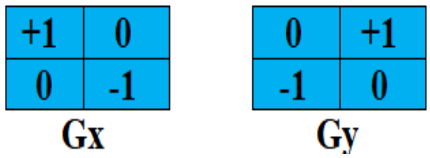

Figure 3: Robert edge detection method with two kernels [19]

V. Deep neural learning

Deep neural network Learning is a part of the machine learning field. The machine learning is a type of Artificial Intelligence as shown in Figure 4 [21,22,23].

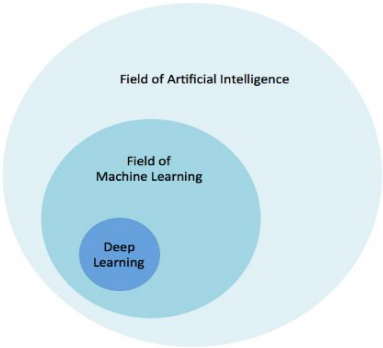

Figure 4: The relationship between AI, deep learning and Machine learning [23]

The deep neural network is used for solution and process the complex problems like recognition of images, natural language and computer vision for large data. In addition, it can classify images features automatically using Deep Learning [21, 23, 24, 25]. Deep Learning mostly uses the CNN algorithm. CNN is a type of Neural Network (NN) with convolution architecture. CNN layers are better than standard NN layers because CNN has a high dimensional filter. The high dimensional filter is convolved with the inputs (images and videos) in this layer. CNN is useful for reducing the memory and the number of network parameters [25, 26]. In this work, one of the most applications of CNN is to recognize the ASL letters for the hearing-impaired and deaf persons.

The proposed Deep Learning with CNN architecture block consists of the following layers with the CNN parameters as shown in Figure 5

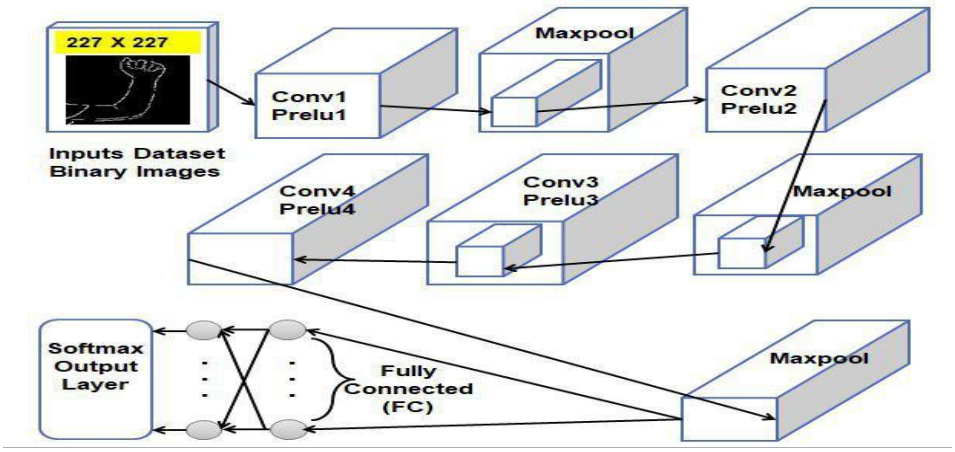

Figure 5: Proposed CNN structure

The CNN layers are explained as below:

1) Convolution layer: is the first layer is used 227x227 pixels binary input images. In this paper, There are four CNN layers as in Figure 5. These images are the outputs of the Edge Detection block. The mathematical equation of convolution layer is expressed as [27,28]:

$$x_j^l = f(\sum_{i \epsilon Mj} (x_i^{l-1} * k_{ij}^l + b_j^l) \quad \ldots (9)$$

Where x_j^l the output images on the current layer are, x_i^{l-1} is the previous output image layer, k_{ij}^l is the convolution kernel (filter) of the presentation layer, and b_j^l are biases of the current layer. Mj represents the selection of input image maps. The convolution operator is marked as the symbol * [27, 28].

2) Nonlinearity

A deep neural network learning is to learn nonlinear mapping. If the nonlinearity absences, a stacked weight network layer are equal to a linear map from the input to the output.

The concept of the rectifier is switched off the output while the negative input. One of the most important and efficient activation functions is a Parametric Rectifier Linear Unit (PReLU). PReLU is the function that improves the fit of the CNN model with closely zero extra computational cost and reduces the risk of overfitting as shown in Figure 6. The activation function described as [26,27,29].

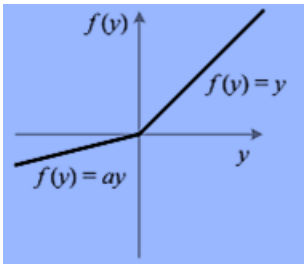

Figure 6: The PReLU activation function [29]

3) Maximum Pooling (Maxpool)

The feature map of the input image is computed by convolution layer. In this layer, the new features are computed with a small neighborhood for the feature map. Pooling operation is useful to reduce the feature dimension, enhance these features, and Avoids overfitting. A final output compact feature still keeps invariant of the rotation, translation, object scale, etc. Maximum pooling is performed downsampling via partitioning the input feature map into non-overlapping rectangular regions with kernel size. Then the output is the maximum of every region [25,27,30]. The maximum pooling example layer as seen in Figure 7.

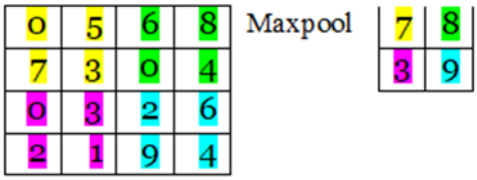

Figure 7: Max pooling example with kernel size 2X2 and stride 2

4) Fully Connected (FC)

The matrix of the feature map is presented as a vector. In this layer, all the layers are fully connected. These features maps are combined with each other in order to create a final CNN model [28].

5) Softmax classifier layer

This softmax layer is connected to the final Fully Connected (FC) layer (feed-forward neural network) in order to predict the image outputs labels. The softmax classifier is used to classify images according to the following probability expression:

$$y_i = \left(\frac{e^{v_i}}{e^{v_1}+e^{v_2}+e^{v_3}+e^{v_4}+\cdots+e^{v_m}}\right) = \frac{e^{v_i}}{\sum_{k=1}^{M} e^{v_k}} \qquad (10)$$

Where, v_i is the sum of the ith weighted output node, and M is the number of output nodes
This equation is applied according to the following condition:

$$\emptyset(v_1) + \emptyset(v_2) + \emptyset(v_3) + \cdots + \emptyset(v_{3M}) = 1 \qquad (11)$$

This condition means that: the softmax function is maintaining the summation for all outputs equal one. In addition, the individual output values range between 0 and 1 [21,31].

VI. SGD, Accuracy and Error Calculations

This section consists of three points:

1) The Stochastic Gradient Descent (SGD)

SGD is a method that evaluated the error of each data training and regulates the weights directly [18]. Back-propagation in deep neural network learning performed by using the SGD method. At a time, taking the one training sample than passing it the neural network.in addition, the error is recorded in each iteration. (SGD) is how to back-propagating the error in order to get better configuration weights. Training is ended one epoch of iteration [22].
If the error is written as:

$$E = \left(\sum_{n} E_n\right) \qquad (12)$$

The SGD approach has updated the weights by using the following form [31]:

$$W_{t+1} = \mu W_t - \mathcal{E}\nabla f(Q_{t+1} - W_{t+1}) \qquad (13)$$

Where $W_t$is the previous weight, W_{t+1} is next weight and $\mathcal{E}$>0 is the learning rate, μ ϵ [0,1] is the momentum coefficient. The benefit of μ is to increase the speed of the system and the convergence parameters. In this work, the μ is chosen 0.9. the update gradient objective ∇f(Q_t) whereQ_t=Q(t+1)-W(t+1).The momentum contributes a calculated gradient at earlier steps, which is weighed in accordance with the convergence parameter. While the updated energy can rapidly bring the solver to the local optimum. The optimum can be overtaken and missed. The momentum method (μ) that we refer to as a classical momentum (CM) is a technique for accelerating gradient descent that accumulates a vector of velocity in the direction of reduction across iterations of the objective function [31,32].

2) Accuracy

The performance of this work is measured by the accuracy rate. An accuracy equation of 24 static letters of ASL is computed as [33]:

$$Accuracy\ (\%) = \left(\frac{correct\ output}{total\ number\ of\ input}\right) * 100 \qquad (14)$$

3) Error

The measurement of the Deep neural network learning of CNN model error is called the cost error or loss function as described in Eq. (15)

$$J = \sum_{i=1}^{M}\{-d_i\ ln(y_i) - (1-d_i)ln(1-y_i)\} \qquad (15)$$

Where y_i is the output value of softmax layer, d_i is the right output value of the training data, and M is the output node number [21].

VII. Text Label Output

It contains the text label output of each image.

The flowchart proposed of static hand gesture is sketched as in Figure 8.

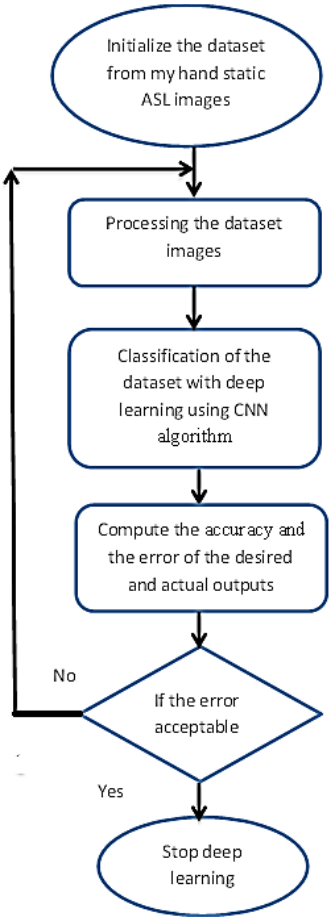

Figure 8: The proposed flowchart of work

4. Simulation Results

The simulation results of the proposed work for recognizing the ASL static 24 letters using CNN are presented as shown in the figures below:

The static recognition of similar two-hand configurations for the letters (A) and (E) are simulated respectively as shown in Figures 9 and 10.

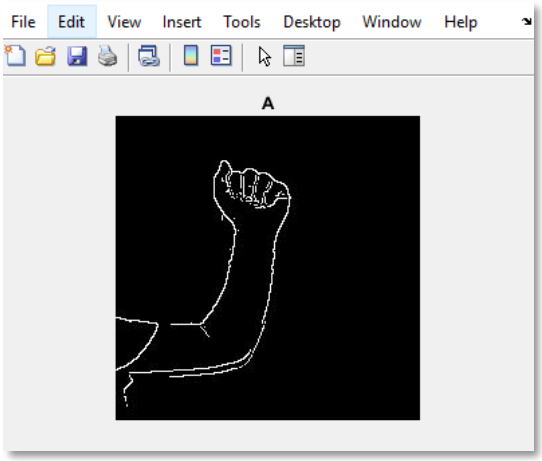

Figure 9: Recognition of letter A

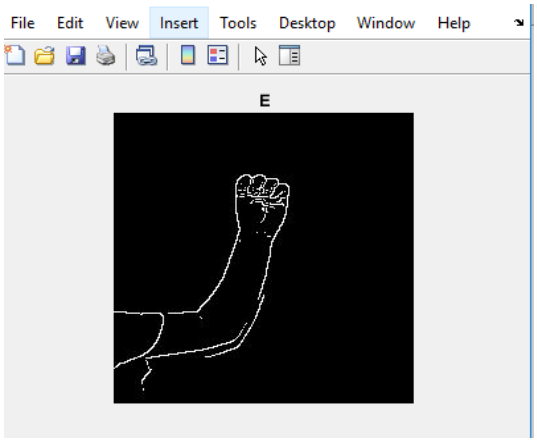

Figure 10: Recognition of letter E

The static recognition of similar two-hand configurations for the letters (M) and (N) are simulated respectively as shown in Figures 11 and 12.

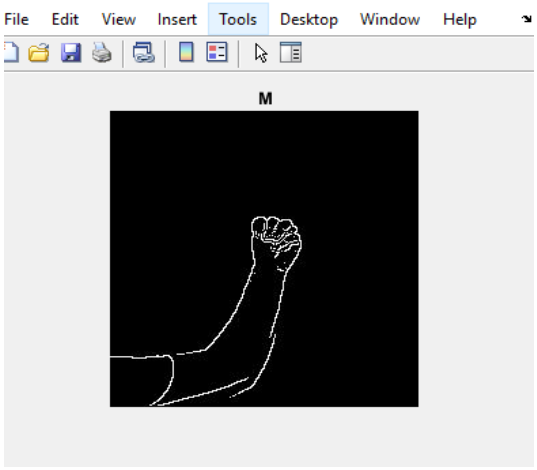

Figure 11: Recognition of letter M

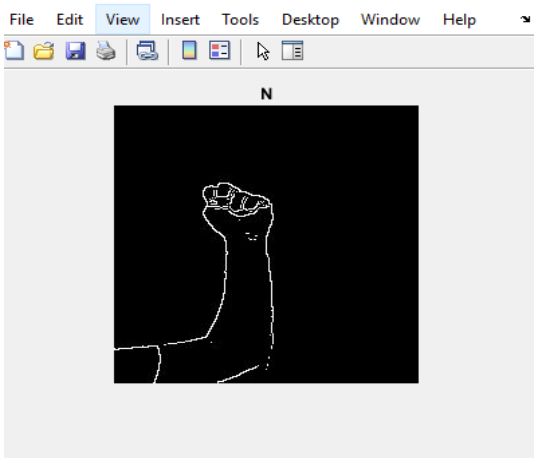

Figure 12: Recognition of letter N

The static recognition of similar two-hand configurations for the letters (S) and (T) are simulated respectively as shown in Figures 13 and 14.

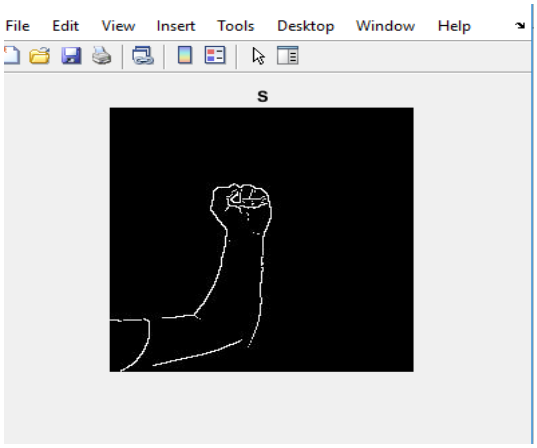

Figure 13: Recognition of letter S

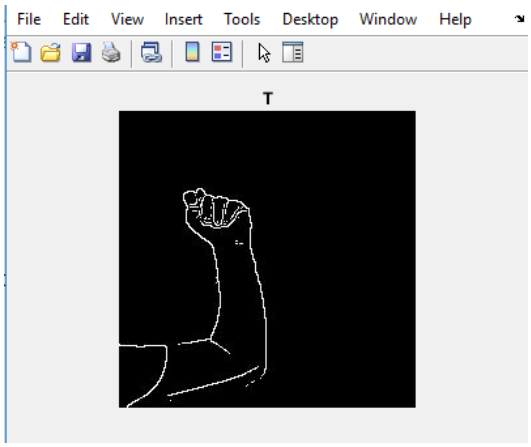

Figure 14: Recognition of letter T

The static recognition of similar two-hand configurations for the letters (R) and (U) are simulated respectively as shown in Figures 15 and 16.

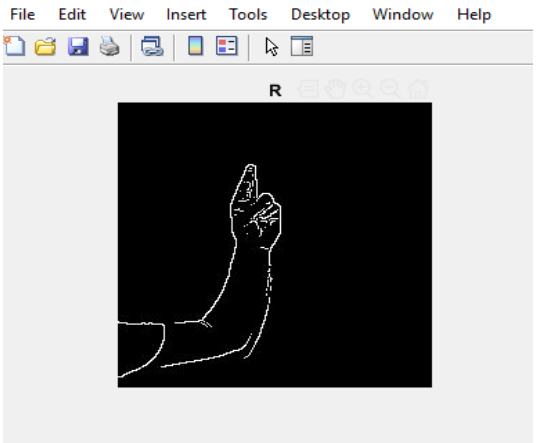

Figure 15: Recognition of letter R

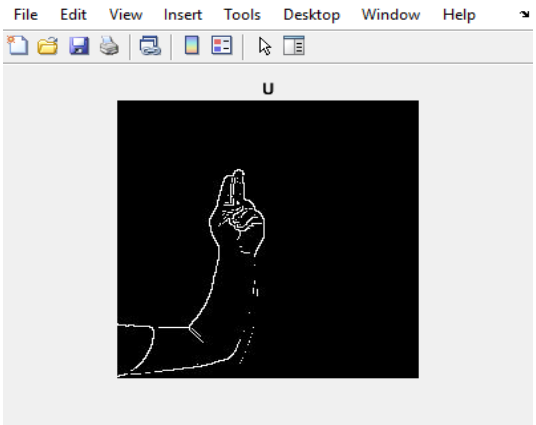

Figure 16: Recognition of letter U

Training progress of loss and accuracy is computed according to the ex (14, 15).
In Figure 17 the accuracy is reached to 99.3 % and the loss equals 0.0002 after 300 iterations for 100 epochs. The elapsed time is 36 minutes with 15 seconds.

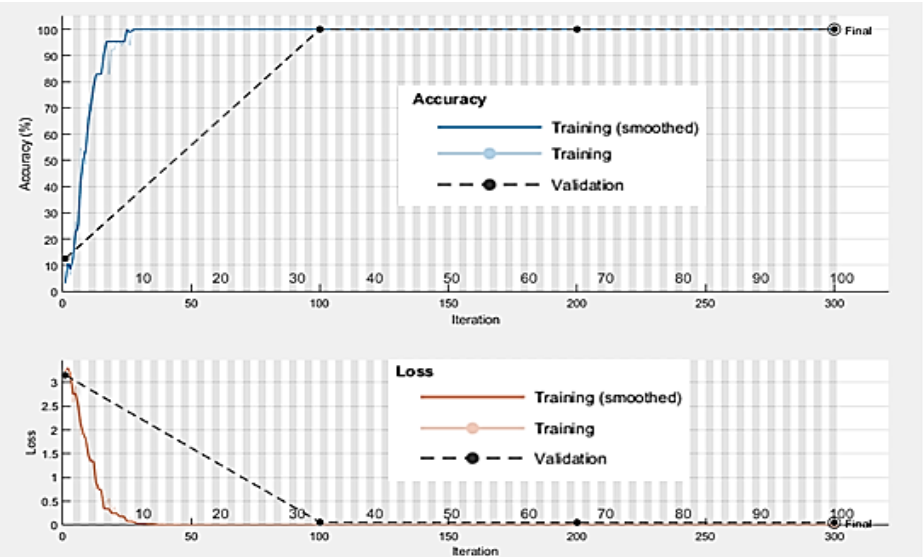

Figure 17: Accuracy and loss functions results

5. Conclusion

This paper presents the proposed gesture recognition of 24 static letters for ASL using Deep Learning. The 227 X 227 RGB images are resized with the Bicubic interpolation method. The binary images of ASL hand edges are extracted using edge detection. These images (240 image dataset of 24 letters with 10 individuals) are trained with four Convolution Neural Networks (CNNs). The accuracy is computed and equals to 99.3 %. Our approach has excellent results of classifications in comparison with other related works. The loss error was 0.0002. The training was relatively fast, due to the small elapsed time. The static ASL letters that have similar hand configurations are better recognized compared to the related work results. Moreover, The deep learning process of the images which represent the static letters has a high accuracy percentage of gesture recognition.

References

[1] P.A. Nanivadekar and V. Kulkarni, "Gesture recognition: a revolutionary tool," *International Journal of Technological Advancement and Research*, Vol. 3 Issue. 3, 2013.

[2] N. Patel and S. JingHe, "A survey on hand gesture recognition techniques, methods and tools," *International Journal of Research in Advent Technology*, Vol. 6, No. 6, 2018.

[3] A.L.C. Barczak, N.H. Reyes, M. Abastillas, A. Piccio, and T. Susnjak, "A new 2D static hand gesture colour image dataset for ASL gestures," *Res. Lett. Inf. Math. Sci*., Vol. 15, pp. 12-20, 2011.

[4] A. Julka and S. Bhargava, "A static hand gesture recognition based on local contour sequence," *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 3, Issue 7, 2013.

[5] A.K. Gautam and A. Kaushik, "American sign language recognition system using image processing method ," *International Journal on Computer Science and Engineering (IJCSE)*, Vol. 9 No.07, 2017.

[6] F.A. Raheema and H.A. Raheem, "ASL recognition quality analysis based on sensory gloves and MLP neural network," *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, 2018.

[7] S. Biradar, and A.M. Tuppad, "A static hand gesture classification system for american sign language (ASL) finger spelling and digits, *"International Journal of Lastest Trends in Engineering and Technology (IJTET)*, Vol. 7, Issue 1, 2016.

[8] T.N. Nguyen, H.H. Huynh, and J. Meunier, "Static hand gesture recognition using artificial neural network," *Journal of Image and Graphics*, Vol. 1, No.1, 2013.

[9] O.K. Oyedotun and A. Khashman, "Deep learning in vision-based static hand gesture recognition ," *Springer, Neural Computing and Applications*, 2016.

[10]. J.L. Flores C.E. Gladys Cutipa, R.L. Enciso, "Application of convolutional neural networks for static hand gestures recognition under different invariant features," IEEE, 2017

[11] J. Bamwend and M. Özerdem, "Recognition of static hand gesture with using ANN and SVM"," *Dicle University Journal of Engineering*, 2019

[12] J. Pansare and M. Ingle, "Vision-based approach for American sign language recognition using edge orientation histogram," International Conference on Image, Vision and Computing, 2016

[13] S. Nagarajan and T. Subashini," Static Hand Gesture Recognition for Sign Language Alphabets using Edge Oriented Histogram and Multi Class SVM", *International Journal of Computer Applications*, Vol. 82, 2013.

[14].A. Prajapati, S. Naik, and S. Mehta, "Evaluation of different image interpolation algorithms ," *International Journal of Computer Applications (0975 – 8887),*Vol. 58, No. 12, 2012.

[15] M.B. Hisham, S.N. Yaakob, R.A. Raof, A.B. Nazren, and N.M. Wafi, "An analysis of performance for commonly used interpolation method ," American Scientific Publishers Advanced Science Letters, United States of America, 2015.

[16] S.A. Alrubaie and A.H. Hameed, "Dynamic weights equations for converting grayscale image to RGB image," *Journal of University of Babylon for Pure and Applied Sciences*,Vol. 26, No.8, 2018.

[17] C. Saravanan, "Color image to grayscale image conversion," IEEE, Second International Conference on Computer Engineering and Applications, pp. 196-199, 2010.

[18] S.J. Pise, "An outlet for a creative mind, thinkquest," Springer Science & Business Media, Proceedings of the First International Conference on Contours of Computing Technology, 2011.

[19] P. Selvakumar and S. Hariganesh "The performance analysis of edge detection algorithms for image processing," International Conference on Computing Technologies and Intelligent Data Engineering, 2016.

[20] S.M. Sharef, F.A. Rahem, and S.S. Jouma'a, "Implementation of fuzzy logic techniques in detecting edges for noisy images," The Second Engineering Conference of Control, Computers and Mechatronics Engineering (ECCCM2), pp. 154-162, 2014.

[21] P. Kim, "MATLAB deep learning ITH machine learning, neural networks and artificial intelligence, " APress, 2017.

[22] S. Skansi, "Introduction to deep learning from logical calculus to artificial intelligence," Springer, 2018.

[23].A. Gibson and J. Patterson, "Deep Learning," O'Reilly Media, Inc., 2017.

[24] G.S. Chadha, E. Meydani, and A. Schwung, "Regularizing neural networks with gradient monitoring," Springer, Recent Advances in Big Data and Deep Learning: Proceedings of the INNS BDDL, Sestri Levante, Geneva, Italy, 2019.

[25] S. Gong, C.L. J,B. Z.Y. Li, and H. Dong, "Advanced image and video processing using MATLAB," Springer, 2019.

[26] S. Khan, H. Rahmani, S.A. Shah, and M. Bennamoun, "A guide to convolutional neural networks for computer vision," Morgan & Claypool Publishers series, 2018.

[27] M.Z. Alom, T.M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. Esesn, and A.S, "The history began from AlexNet: a comprehensive survey on deep learning approaches," Cornell University, arXiv.org, cs, arXiv:1803.01164, Computer Science, Computer Vision and Pattern Recognition, 2018.

[28] S. Ameen and S. Vadera, "A convolutional neural network to classify american sign language finger spelling from depth and colour images," John Wiley & Sons, Ltd., 2017.

[29] K. H. Zhang and S.R Sun, "Delving deep into rectifiers: surpassing human-level performance on image net classification," Cornell University, arXiv:1502.01852v1, Computer Science, Computer Vision and Pattern Recognition, 2015.

[30] J. Nagi, F. Ducatelle, G.A. DiCaro, D. Ciresan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, and L.M. Gambardella, "Max-pooling convolutional neural networks for vision-based hand gesture recognition," IEEE International Conference on Signal and Image Processing Applications (ICSIPA2011), 2011.

[31] C. M. Bishop, "Pattern recognition and machine learning," Springer, New York, NY, 2006.

[32] I. Sutskever, J. Martens, G. Dahl, and G.Hinton," On the importance of initialization and momentum in deep learning," Proceedings of the 30 th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013.

[33] W. Tangsuksant, S. Adhan, and C. Pintavirooj, "American sign language recognition by using 3D geometric invariant feature and ANN classification ," IEEE, The Biomedical Engineering International Conference, 2014.