A Real-Time System For Recognition Of American Sign Language By Using Deep Learning

Murat Taskiran, Mehmet Killioglu, and Nihan Kahraman
Electronics and Communications Engineering
Yildiz Technical University
Istanbul, Turkey
Email: mrttskrn@yildiz.edu.tr

***Abstract*—Deaf people use sign languages to communicate with other people in the community. Although the sign language is known to hearing-impaired people due to its widespread use among them, it is not known much by other people. In this article, we have developed a real-time sign language recognition system for people who do not know sign language to communicate easily with hearing-impaired people. The sign language used in this paper is American sign language. In this study, the convolutional neural network was trained by using dataset collected in 2011 by Massey University, Institute of Information and Mathematical Sciences, and 100% test accuracy was obtained. After network training is completed, the network model and network weights are recorded for the real-time system. In the real-time system, the skin color is determined for a certain frame for hand use, and the hand gesture is determined using the convex hull algorithm, and the hand gesture is defined in real-time using the registered neural network model and network weights. The accuracy of the real-time system is 98.05%.**

***Keywords*—american sign language; classification; convex hull; convolutional neural network; deep learning; real-time**

I. Introduction

Sign language is a language that provides visual communication and allows individuals with hearing or speech impairments to communicate with each other or with other individuals in the community. According to the World Health Organization, the number of hearing-impaired individuals has recently reached 400 million. For this reason, recent studies have been accelerated to make disabled people communicate more easily. If studies in the literature are examined, Basic Component Analysis is used in the extraction of the feature vector in work done by Mahmoud Zaki and Samir Shaheen in 2011 and Hidden-Markov Model is used as the classifier [1]. In the study conducted by Ching Hua et al. In 2014, motion sensors were used for feature extraction and k-NN and Support Vector Machine (SVM) for classification [2]. In 2015, Cao et al. used the depth comparison feature of Microsoft Kinect to obtain feature vectors and used the random forest and constrained link angle algorithm to classify the obtained vectors [3]. K-Nearest Neighbors (k-NN) Classifier was used in American Sign language recognition by Dewinta Aryanie and Yaya Heriadi in 2015 [4]. In the study made by Jin et al. In 2016, Speeded Up Robust Feature (SURF) algorithm for feature extraction and SVM as classifier were used [5]. The study of Pansare et al.'s work in 2016 is about recognizing the American sign language based on the edge orientation histogram method. In this study, the letters of the alphabet are used, but the digits are not used [6]. In 2016, Truong et al. designed an American sign language translator for text and speech using Haar-cascade algorithm [7]. In this study, only the letters were trained and classified. In another study conducted by Islam and his colleagues in 2016, an attempt was made to create a real-time, high-performance American sign language recognition system in the background of a black environment. For the feature extraction, 'K convex hull' algorithm, which is a combination of K curvature and convex hull algorithm, is proposed. An artificial neural network was used as classifier [8]. In the study of Joshi and his colleagues in 2017, American sign language translator was realized by using edge detection and cross-correlation methodologies [9].

In previous research, the ring projection and wavelet transform were used for feature extraction, and generalized regression neural network was used as classifier [10]. In this article, an attempt was made to design a real-time and high-performance translator for those who do not know the sign language. This study proposes a CNN (Convolutional Neural Network) structure for feature extraction and classifier, and then the hand locating process was applied to construct the real-time system. Skin color detection and convex hull algorithms have been used together in determining hand position. After the detection of the hand location, the part obtained is resized and given to the trained neural network to classify it.

II. Methodology

The flowchart clarifies the proposed system is given in Fig. 1 below. The steps on the flow chart are explained in detail below.

A. Convolutional Neural Network

Traditionally in recognition systems, a classifier structure follows image processing and morphological process units. With the use of deep learning methods, it has become possible to classify in high performance by the minimum number of image processing and morphological processing steps. In this paper, convolutional neural network is used as a fine classifier with Tensorflow and Keras libraries in Python. These libraries work efficiently on powerful modern GPUs (Graphics Processing Units) that allows doing much faster computation and training. In recent years, CNN based classifications and

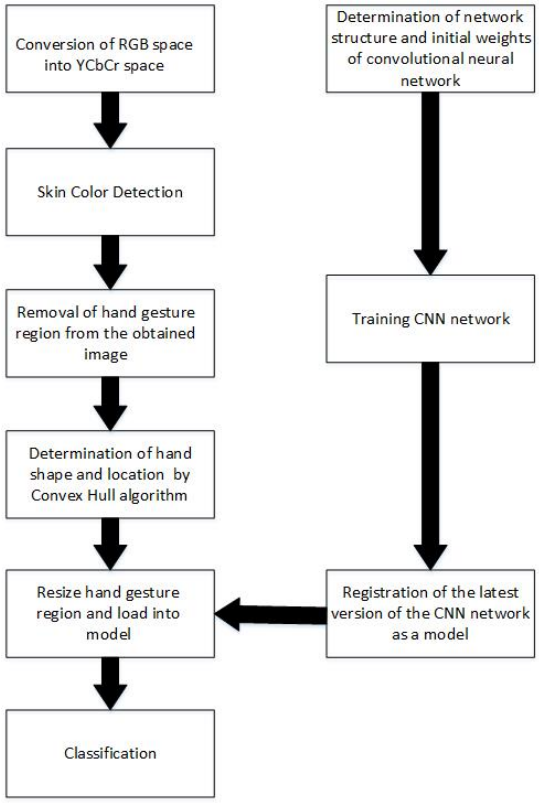

Fig. 1. The flowchart of proposed method

researches are very popular and have proven to be successful in areas like image classification and recognition. Rectified Linear Unit (ReLu) [11] is used as activation function, which makes converge much faster while still presents good quality [12].

B. Training Classifier

Proposed CNN model consists of the input layer, two 2D convolution layers, pooling, flattening and two dense layers as seen in Fig. 2. In the dataset, there are 25 images of cropped images for each hand gesture, in total, 900 images loaded into the program as arrays. Then, each image resized to 28×28 pixels and converted to grayscale image. With the help of Scikit-Learn library, the array is shuffled randomly. Shuffling is needed for splitting array into train and test arrays. After splitting step, the model is created as sequential network and started fitting process. Fitting process ran through all train data, with batch size 120 and epoch number 30. Batch size means how many images will be loaded in every iteration while epoch number means the total cycle that all the images loaded into the neural network for training.

Softmax-based loss function and softmax function given in (1) and (2) respectively, are used in this work. Here, N is the total number of training samples, and C is the total number of classes. It takes a feature vector z for a given training example and squashes its values to a vector of [0,1] valued

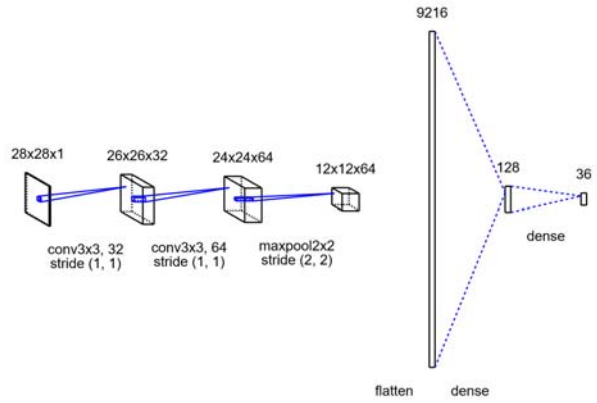

Fig. 2. The architecture of used Convolutional Neural Network

real numbers summing to 1. Equation (1) takes the mean loss for each training example, x_i, to produce the full softmax loss.

$$Loss = \frac{1}{N}\sum_{i=1}^{N} -log(\frac{e^{f_{i,y_i}}}{\sum_{j=1}^{C} e^{f_{i,y}}}) \quad (1)$$

$$f_j(z) = \frac{e^{z_j}}{\sum_{k=1}^{C} e^{z_k}} \quad (2)$$

C. Real-time Application

After training step, the model and weights of neural network loaded into real-time recognition algorithm. The algorithm consists of two parts that run simultaneously for better accuracy. One of the steps is extracting hands bound convex hull points. The other step is classifying hand image with convolutional neural network. When there are similar hand signs, the decision will be made according to those steps results.

To extract skin region detection, image color space is converted from RGB to YCbCr using (3).

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.50 \\ 0.50 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} \quad (3)$$

where Y is luma component, C_b and C_r are blue component and red component related to the chroma component of the image. After color space conversion, pixel values are thresholded using a predetermined range to create a mask as $C_b \, \epsilon \, R_{Cb}, and \; C_r \, \epsilon \, R_{Cr}$ where $R_{Cb} = [75, 135]$ and $R_{Cr} = [130, 180]$ [13]. Then, YCbCr image is logically processed with bitwise AND gate with itself using the mask. Afterwards, the image is pre-processed for noise reduction with multiple dilation steps with different kernel size, erosion, median blurring, and thresholding. This step is required because pixels corresponding to hand must be connected to determine hands contours. After preprocessing, the maximum connected area is calculated, and bounding rectangle is extracted. Using OpenCV library functions, convex hull

and convexity defections are calculated. Count of convexity defections let to do a coarse classification.

Convolutional neural network is used for fine classification. The same bounding rectangle is used for the region of interest extraction. After hand extraction, the image is resized to (28×28) pixels, and color space is converted to gray. To able to use model, pixel values are converted from unsigned 8-bit integer to 64-bit float and loaded to model. The model gives results as possibilities of the corresponding letter.

Depending on the results from two parts, a final decision is given as output. These processes are being run in every frame from the camera. Loop time varies between 15 ms and 25 ms, thus meeting the camera speed which captures a new frame in every 30 ms.

III. Results

In the proposed system, the dataset collected in 2011 by the Institute of Information and Mathematical Sciences, Massey University was used to train the CNN network [14]. There are 900 pictures including 25 samples for each of 36 characters consisting of 26 letters and 10 numbers in the dataset. 80% of these pictures were used as training data and 20% were used as test data. Test data were randomly selected each time. The CNN network has achieved maximum test performance in 20 epochs per training. The results of tests performed randomly 5 times at random are given in Table I. The variation of training and test achievements in each epoch is given in Fig. 3. After the network training is completed on the proposed system, the system is made to operate in real-time. For real-time system control, each of 36 characters has been tested 10 times. As a result, 98.05% test performance was achieved in the real-time system. Table II shows the results of real-time system tests performed for each character. For false results, the false decision symbol is given in parentheses.

TABLE I. The results of tests (%) performed randomly 5 times.

Test 1	Test 2	Test 3	Test 4	Test 5
99.44	99.44	**100**	99.44	98.89

IV. Conclusion

The comparison between the studies in the literature and the proposed method is given in Table III. Proposed system has advantage in terms of test accuracy according to these similar studies. Also, the applicability as real-time system is validated. In Fig. 4, test stages are shown, and system achieved high accuracy even with the letters that have similar gestures.

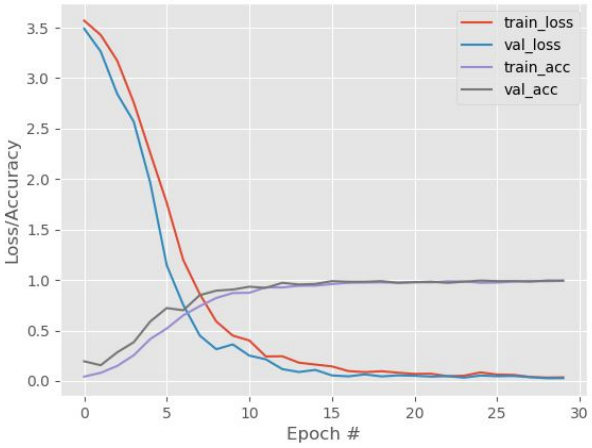

Fig. 3. Training loss and accuracy on American Sign Language

TABLE II. The results of real-time system tests performed for each character.

Character	True Result	False Result	Accuracy(%)
0	10	-	100
1	9	1 (Z)	90
2	9	1 (V)	90
3	10	-	100
4	10	-	100
5	10	-	100
6	10	-	100
7	10	-	100
8	10	-	100
9	10	-	100
A	10	-	100
B	10	-	100
C	10	-	100
D	10	-	100
E	10	-	100
F	10	-	100
G	10	-	100
H	10	-	100
I	10	-	100
J	10	-	100
K	10	-	100
L	10	-	100
M	9	1 (N)	90
N	9	1 (M)	90
O	10	-	100
P	10	-	100
Q	10	-	100
R	10	-	100
S	9	1 (T)	90
T	10	-	100
U	10	-	100
V	9	1 (2)	90
W	10	-	100
X	10	-	100
Y	10	-	100
Z	9	1 (1)	90
TOTAL	**355**	**5**	**98.05**

References

[1] M. M. Zaki, S. I. Shaheen, "Sign language recognition using a combination of new vision based features, " *Pattern Recognition Letters*, vol. 32, pp. 572–577, 2011.

[2] C. Chuan, E. Regina, C. Guardino, "American Sign Language Recognition Using Leap Motion Sensor," in *Proc. 13th International Conference on Machine Learning and Applications (ICMLA)*, Detroit, USA, 2014, pp. 541–544.

[3] C. Dong, M. C. Leu, Z. Yin, "American Sign Language alphabet recognition using Microsoft Kinect," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Boston, USA, 2015, pp. 44–52.

[4] D. Aryanie, Y. Heryadi, "American sign language-based finger-spelling recognition using k-Nearest Neighbors classifier," in *Proc. 3rd Inter-*

TABLE III. THE COMPARISON BETWEEN THE STUDIES IN THE LITERATURE AND THE PROPOSED METHOD.

Ref.	**Methods**	**Number of Characters**	**Test Accuracy (%)**	**Real Time System Test Acc. (%)**
[1]	PCA + HMM	Words	89.10	-
[2]	Motion Sensor Information + SVM k-NN	26	72.78/79.83	-
[3]	Depth Comparison + Random Forest	24	90	-
[4]	Full Dimensional Feature + k-NN	8	99.6	-
[5]	SURF + SVM	16	-	97.13
[6]	Edge Orientation Histogram	26	-	88.26
[7]	AdaBoost + HAAR	26	-	98.7
[8]	K Convex Hull + ANN	37	-	94.32
[9]	Edge Detection + Cross Correlation	26	-	94
[10]	Ring Projection and Wavelet Transform + GRNN	36	90.44	-
Prop. Method	**CNN+ Skin Detection and Convex Hull**	**36**	**100**	**98.05**

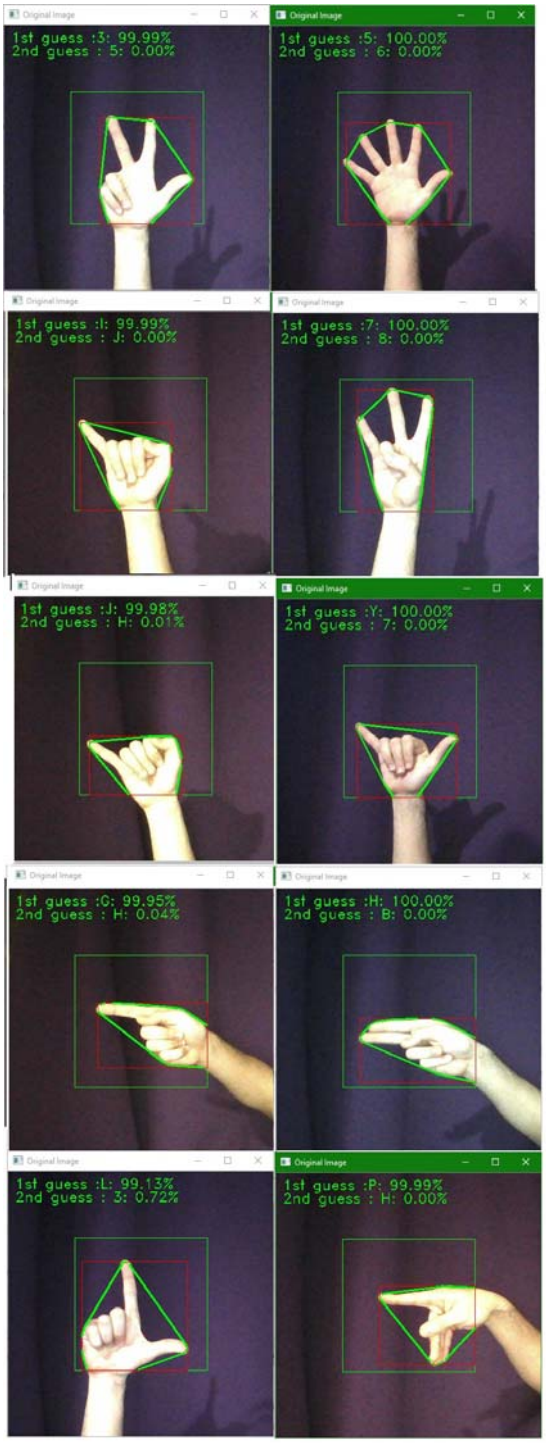

Fig. 4. Real-time system test outputs. '3', '5', 'I', '7', 'J', 'Y', 'G', 'H', 'L', 'P', respectively.

national Conference on Information and Communication Technology (ICoICT), Nusa Dua, Bali, 2015, pp. 533–536.

[5] C. M. Jin, Z. Omar, M. H. Jaward, "A mobile application of American sign language translation via image processing algorithms," in *Proc. IEEE Region 10 Symposium (TENSYMP)*, Bali, Indonesia, 2016, pp. 104–109.

[6] J. R. Pansare, M. Ingle, "Vision-based approach for American Sign Language recognition using Edge Orientation Histogram," in *Proc. International Conference on Image, Vision and Computing (ICIVC)*, Portsmouth, UK, 2016, pp. 86–90.

[7] V. N. T. Truong, C. Yang, Q. Tran, "A translator for American sign language to text and speech," in *Proc. IEEE 5th Global Conference on Consumer Electronics*, Kyoto, Japan, 2016, pp. 1–2.

[8] M. M. Islam, S. Siddiqua, J. Afnan, "Real time Hand Gesture Recognition using different algorithms based on American Sign Language," in *Proc. IEEE International Conference on Imaging, Vision Pattern Recognition (icIVPR)*, Dhaka, Bangladesh, 2017, pp. 1–6.

[9] A. Joshi, H. Sierra, E. Arzuaga, "American sign language translation using edge detection and cross correlation," in *Proc. IEEE Colombian Conference on Communications and Computing (COLCOM)*, Cartagena, Colombia, 2017, pp. 1–6.

[10] M. Taskiran, S. Cimen, Z. G. Cam Taskiran, "The novel method for recognition of american sign language with ring projection and discrete wavelet transform, " *World Journal of Engineering Research and Technology (WJERT)*, vol. 4, no. 1, pp. 92–101, 2018.

[11] V. Nair, G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, Haifa, Israel, 2010, pp. 807–814.

[12] A. Krizhevsky, I.Sutskever , G. E.inton , "Imagenet classification with deep convolutional neural networks ," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[13] S. L. Phung, A. Bouzerdoum, D. Chai, "A novel skin color model in ycbcr color space and its application to human face detection," in *Proceedings IEEE 2002 International Conference On Image Processing,* , NY, USA, 2002, Vol. 1, pp. I–I.

[14] A. L. C. Barczak, N. H. Reyes, M. Abastillas, A. Piccio, T. Susnjak, "A new 2D static hand gesture colour image dataset for asl gestures," in *Massey University,* , 2011.