

# A Real-Time American Sign Language Recognition System using Convolutional Neural Network for Real Datasets

Rasha Amer Kadhim<sup>1</sup>, Muntadher Khamees<sup>2</sup>

<sup>1</sup>Faculty of Agriculture, University of Diyala, Diyala, Iraq

<sup>2</sup>Faculty of Sciences, University of Diyala, Diyala, Iraq

**Abstract** –In this paper, a real-time ASL recognition system was built with a ConvNet algorithm using real colouring images from a PC camera. The model is the first ASL recognition model to categorize a total of 26 letters, including (J & Z), with two new classes for space and delete, which was explored with new datasets. It was built to contain a wide diversity of attributes like different lightings, skin tones, backgrounds, and a wide variety of situations. The experimental results achieved a high accuracy of about 98.53% for the training and 98.84% for the validation. As well, the system displayed a high accuracy for all the datasets when new test data, which had not been used in the training, were introduced.

**Keywords** – ASL recognition system, deep learning, convolutional neural network (CNNs), classification, real-time

## 1. Introduction

Very few people understand sign language as it is not an international language. This makes it difficult for the majority of hearing communities to communicate with the deaf community.

---

DOI: 10.18421/TEM93-14

<https://doi.org/10.18421/TEM93-14>

**Corresponding author:** Muntadher Khamees,  
Faculty of Sciences, University of Diyala, Diyala, Iraq


**Email:** [alkarawis@gmail.com](mailto:alkarawis@gmail.com)

*Received:* 05 March 2020.

*Revised:* 09 July 2020.

*Accepted:* 16 July 2020.

*Published:* 28 August 2020.

 © 2020 Rasha Amer Kadhim & Muntadher Khamees; published by UIKTEN. This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 4.0 License.

The article is published with Open Access at [www.temjournal.com](http://www.temjournal.com)

The most common method of communication between these two communities is through the use of human based interpretation services.

Notwithstanding, this is costly as it includes human expertise. Although written communication may be an alternative solution, it is cumbersome and useless because the deaf is usually less gifted in writing spoken languages. Also, this method is very slow and impersonal, especially in an emergency such as when an accident occurs, where quick communication is required with a physician and written communication is not always possible.

There are many different sign languages for every language in the world. There are over two hundred languages in utilize around the world today such as the Chinese, Spanish, Irish, British, and American Sign Language (ASL), which is the most common sign language in the world. However, according to Garcia & Viesca [1], “there are only 250,000-500,000 speakers, which significantly limit the number of people that they can easily communicate with”. Therefore, it is necessary to find another way to make communication possible between the majority of hearing communities and the deaf community. Automatic ASL recognition system is a new way of understanding the meaning of deaf signs without needing the help of expert. This technique can be used to translate signs into sounds or texts based on the users’ needs. Signs recognition system is still a difficult problem. Many researchers have tried hard to solve this problem because it requires the detection and recognition of the required information hands poses, hands movements, and human body postures. Besides, signs languages have hundreds of thousands of words, including very alike hands poses, in addition to similarities between some signs.

ASL recognition is divided into two types, namely, recognitions of static and dynamic gestures. This paper is focused on static fingerspelling in ASL language. It is a very significant portion of sign languages recognition because it is used in many situations such as brands, addresses, names, and so

on. The static gestures-based system is still difficult due to visual similarities in different signs. For example, the letters N and M appear to be identical, and are just distinguishable by the situation of the thumb. Likewise, there are enormous variations depending on the viewpoint of the camera. The advantages of a deep learning with CNNs were employed to solve this problem and achieve a real time and accurate sign fingerspelling recognition model.

## 2. Related Works

The first approach in relation to sign language recognition was by Bergh in 2011 [2]. Haar wavelets and database searching were employed to build a hand gesture recognition system. Although this system gives good results, it only considers six classes of gestures. Many types of researches have been carried out on different sign languages from different countries. For example, a BSL recognition model, which understands finger-spelled signs from a video, was built [3]. As Initial, a histogram of gradients (HOG) was used to recognize letters, and then, the system used hidden Markov models (HMM) to recognize words. In another paper, a system was built to recognize sentences made of 3-5 words. Each word ought to be one of 19 signs in their thesaurus. Hidden Markov models have also been used on extracted features [4]. In 2011, a real time American Sign Language recognition model was proposed utilizing Gabor filter and random forest [5]. A dataset of colour and depth images for 24 different alphabets was created. An accuracy of 75% was achieved utilizing both colour and complexity images, and 69% using depth images only. Depth images were only used due to changes in the illumination and differences in the skin pigment. In 2013, a multi-layered random forest was also used to build a real time ASL model [6]. The system recognizes signs through applying random forest classifiers to the combined angle vector. An accuracy of 90% was achieved by testing one of the training images, and an accuracy of 70% was achieved for a new image.

An American Sign Language alphabets recognition system was first built by localizing hand joint gesture using a hierarchical mode seeking and random forest method [7]. An accuracy of 87% was achieved for the training, and accuracy of 57% when testing new images. In 2013, the Karhunen-Loeve Transform was used to classify gesture images of one hand into 10 classes [8]. These were translated and the axes were rotated to distinguish a modern coordinate model by applying edge detection, hand cropping, and skin filter techniques. An accuracy of (96%) was achieved. Sharma [9] characterized each colour channel after background deduction and noise

elimination using (SVM and k-NN) classifiers, followed by a contour trace. An accuracy of (62.3%) was gained by using (SVM) as a classifier.

Starner, Weaver & Pentland tracked hand movements by using a (3D) glove and an (HMM) model. This model can gain (3D) information from the hands regardless of the spatial direction. An accuracy of (99.2%) was achieved on the test dataset. HMM utilize time series datasets to follow hands movement and recognize them according to where the hand has been [10]. All the researches that have been discussed above used linear classifiers, which are relatively simple and only require attribute extraction and pre-processing to be successful.

Another approach is to use deep learning techniques. This approach was used to build a model that recognizes hands gestures in a continual video stream utilizing DBN models [11]. An accuracy of over 99% was achieved. Another research used a deep learning technique, whereby a feed forward neural network was used to classify a sign. Many image pre-processing methods have been used, such as background subtraction, image normalization, image segmentation and contrast adjustment. In addition, a principal component analysis (PCA) and Gabor filters have been used for feature extraction. An accuracy of 98.5% was achieved with this method [12]. All the works discussed above depended on the extraction of the hand before it is fed to a network. However, a research was done on different sign languages from different countries [13], and this was the most relevant work for the current study. An Italian Sign Language recognition system was built using CNNs to classify 20 Italian gestures. A Microsoft Kinect was applied to full-body images of people, whereby the Kinect was able to capture depth images. Only the depth images were used for training and an accuracy of 91.7% was achieved. However, it was mentioned that the test dataset could be in the training dataset (and/or) the validation dataset [13]. The structural design of the system consisted of two convolutional neural networks, one to extract higher body features and one to extract hand features. The data set, looking at People 2014, was used [14]. The depth map images were also used with data set involving 20 Italian sign motions. The validation was 0.789675, and the final was 0.788804.

Kang, Tripathi & Nguyen built a real time sign finger spelling recognition system using CNNs from the depth map [15]. The authors collected 31,000 depth maps with 1,000 images for each class by using the Creative Senz3D camera. They had 31 various hand signs from 5 various persons. The 31 hand signs included all the fingerspelling of both letter sets and numbers, without (J & Z) letters that require additional external data for the classification

[15]. They utilized hand segmentation and assumed that the hand has to be near to the camera, which helped to make the bounding boxes in the same input size of 256, and 256. For the structural design of the CNNs, they utilized (CaffeNet), three max-pooling layers, three fully-connected layers, and with five layers. They achieved accuracies of 83.58% & 85.49% for the fine tuning and 75.18% & 78.39% for the training. This work utilized only depth map method by using the (Creative-Senz3D) camera, which is expensive and not available to everyone.

Therefore, the proposed system cannot be implemented in a normal PC camera. Real time ASL recognition with CNNs was built using a public dataset known as The ASL Finger Spelling Data sets from Centre for Vision of University of Surrey [1]. The data set includes 24 static signs without the letters (J & Z), which captured in five various sessions, with a similar illumination and background. Tang, lu, Wang, Huang, & li, they build a real time hand posture recognition model utilizing deep learning based CNNs. They utilized open datasets (MSRGesture3D). The dataset contains 12 dynamic ASL gestures collected from 10 subjects. An accuracy of 94.17% was achieved. However, this work utilized only depth map, black background with color images by using the (Kinect -Senz3D) camera, which means they cannot implement the system in normal web-camera. In addition, they do not utilize all the signs, but only 20 motions [16].

This paper differs from previous works in several ways. First, the model created in this paper is the first fingerspelling recognition model to classify a total of 26 letters including (J & Z), in addition, by having 2 classes for space and delete. Second, the method of hand gesture recognition explored by this work required a large dataset for the training. However, all the open datasets of videos or images found in Google tended as made in controlled settings with solid and distinctly coloured clothing, consistent lighting, white or black backgrounds, and high

quality camera resolution (Kinect, Senz3D camera). Therefore, new datasets were built, which included a wider variety of features for example different backgrounds, different skin tones, different lightings, a wide variety of gestures, and to be based on a normal webcam on any personal computer without the need for any high-quality camera resolution. Finally, the system is a re-training system, which means that it started with randomly weights to training the model. The weights were changed to perform the tasks with minimal errors. In this paper, a real time ASL fingerspelling recognition with CNNs was built using real colouring images. Each step of the project will be discussed in the next section.

### 3. Methodology

#### 3.1. System Architecture

In this research, a real-time ASL fingerspelling recognition was built with CNNs algorithm using real colouring images. It comprised a total of 26 alphabets, including J and Z, in addition to two classes for space and delete. This system was divided into three phases, and the first phase represents the collection of data. The methods of Hand-Gesture recognition explored by this research required a large dataset for training, so it has been decided to build new datasets that included a wider variety of features such as different lightings, different skin tones, different backgrounds, and a wide variety of situations. The second phase was a multi-class recognition with CNN, while the last phase was the writing system, which represented the communication between the computer and the user. This system facilitates the communication between the majority of hearing communities and the deaf community. It is an input system that uses a PC camera. The flowchart in Figure1 shows the architecture of the proposed system.

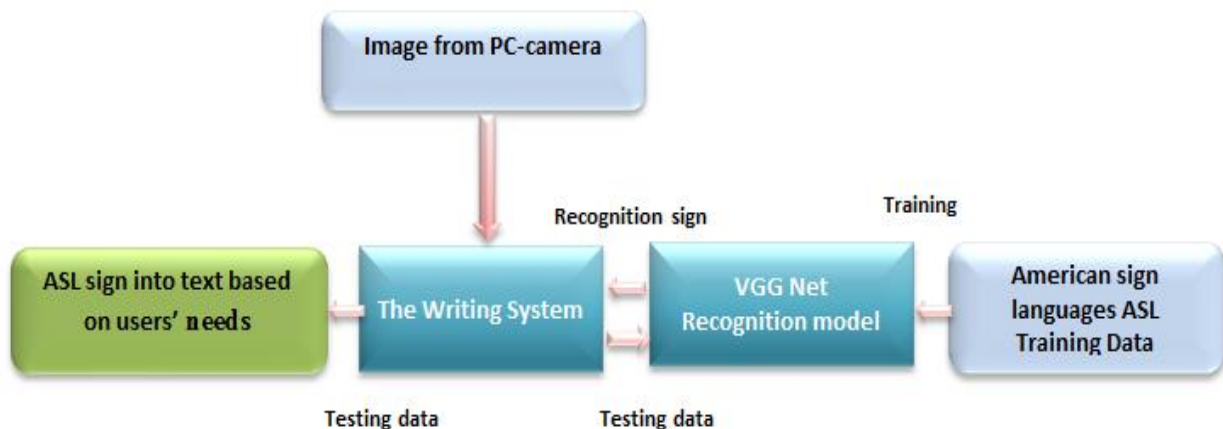


Figure 1. System architecture

### 3.2. Multi-Class-Recognition with CNNs

This system moved away from the feature-based methods utilized in most of the works discussed above. A multi-classification system was built and based on convolutional neural networks (CNNs) with a deep learning data structure that had been used for multi-class recognition. Each ASL sign was represented as an individual category. The classifier results would be one of 28 classes starting from 0 to 27. The CNN architecture used was the VGG\_Net [17], which is very deep convolutional network architecture for high-scale image recognition. The proposed re-training model started the training by initializing the network weights randomly, and then, the weights were updated to perform the tasks with fewer mistakes. The weights of the network were saved to be loaded again as the initial weights for new experiments known as fine-tuning. The VGG blocks from the TFlearn (TFlearn Development Team Github, 2017) original web were used in this work.

### 3.3. Training

For the multi-class recognition system, 28 classes of static fingerspelling in American Sign Language (ASL) from the images that had been collected were used. All the images were scaled to 224 x 224 pixels, and then, normalized to be fed to the VGG\_Net. The path of the images, with the label of each one, was saved into the text file. The images were converted into the NumPy array form (No. of Images, 32, 32, 3) by utilizing TFlearn data to feed them to the system. For training the model, the total training datasets used were 61.614, with around at least 2200 for each class. For the validation datasets, about 0.30 from the training datasets were used, so that in total comprising 43,120 was used for the training, and 18,480 for the validation datasets. The shuffle=True means the training datasets will be shuffled each time before splitting the validation dataset and before training it. Data on 2816 images were kept to test the mode for around 100 images for each class. It took more than seven days to train the model.

## 4. Experiments

### 4.1. Dataset and Parameters

The writing system was built using a Python framework, which took one image of a whole user with the sign every 5 seconds before sending it to the multi-class recognition model to be recognized. The system works as a printing system because it prints the ASL sign and converts it into text. To collect a good dataset, a Python program was written, which took one image of 640 x 480 pixels every 1 second. Help was obtained from six volunteers, each of whom did all the signs in a different background,

different lighting, and different viewpoint of the sign. A total of 61.614 images were collected for 28 classes, comprised of 26 alphabets, including J and Z, as well as two classes for space and delete. Table 1 shows the number of classes and how many images were included.

Each image of all the signs was done in a different background, different lighting, and different viewpoint of the sign. Previously, the letters J and Z involved motion. To solve this case, each one of the three different viewpoints was taken for each movement. Namely, the image for the starting movement of the letter was followed by the image for the middle movement, and another for the last movement.

Softmax (activation function) was used for the hyper parameter tuning, while the learning rate was (0.0001), the epoch number was 500, the dropout rate was 0.5 and the batch size was 32 to fit the memory. For the software requirements, The VGG blocks from the TFlearn, Python 2.7 version, Anaconda 2. Pillow library and Opencv2 library were used. The PC specifications, Intel Core i7 860 @ 3.2 GHz, Ubuntu 16.04 LTS, 8 GB RAM, NVIDIA GTX 780, and NVIDIA CUDA 8, were installed as a backend.

Table 1. The dataset includes 28 classes, comprised of 26 ASL alphabets, in addition to two classes for space and delete.

Alphabets	Class	number of samples
A	0	2164
B	1	2168
C	2	2063
D	3	2108
E	4	2317
F	5	2209
G	6	2120
H	7	2162
I	8	2283
J	9	2246
K	10	2215
L	11	2147
M	12	2172
N	13	2147
O	14	2445
P	15	2166
Q	16	2182
R	17	2157
S	18	2182
T	19	2284
U	20	2193
V	21	2224
W	22	2201
X	23	2208
Y	24	2207
Z	25	2244
Delete	26	2115
Space	27	2285
28 classes		61614

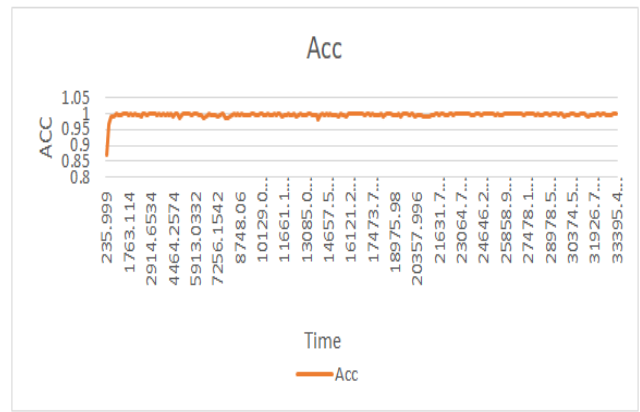
**4.2. Results and Discussion**

A real-time ASL fingerspelling recognition with a CNN algorithm using real colouring images from a PC camera was introduced. In this paper, deaf signs are translated into text statements to help creating a writing system that can be used as an input system for a computer using any computer camera. This system showed good results by taking advantage of a deep learning technique. This section discusses all the results that were obtained from the experiment. A multi-class recognition system was built using VGG\_Net. CNNs were used as the recognition system, in which each ASL sign was represented as an individual category. The classifier result would be one of 28 classes starting from 0 to 27. As a first step, the system succeeded in recognising 10 ASL letters (A, B, C, D, E, F, G, H, I, J). The system was trained with just 10 labels around 20256 training data for each class to produce less than 2000 images for each class to produce less than 2000 images (Table 2).

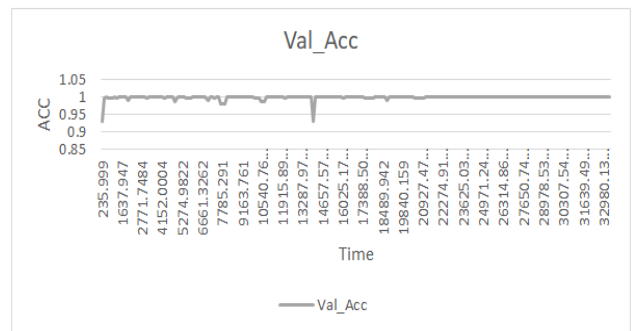
Table 2. Number of images for each sample that was used to train the model.

Alphabets	Class	number of samples
A	0	1983
B	1	2097
C	2	1932
D	3	2021
E	4	2185
F	5	2045
G	6	2001
H	7	2022
I	8	2134
J	9	2080
10 classes		20256

An accuracy of 0.9967% was obtained for the training, an accuracy of 0.99% from the testing data that were not used in the training phase, and an accuracy of 100% for the validation. The use of VGG\_Net showed a better improvement in performance than in previous efforts.



(a)



(b)

Figure 2. (a) Training accuracy with 10 classes; (b) Validation accuracy with 10 classes

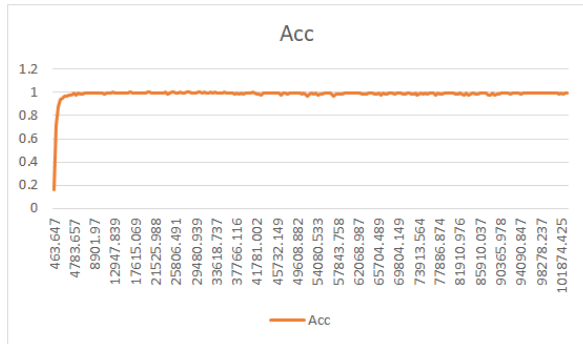
The highest accuracy of about 99.9994% was obtained for the letter I, as it is shown in Table 3. However, the lowest accuracy of about 98.8082% was obtained for the letter D (Table 3).

Table 3. Testing accuracy for each label using the 10-classe model.

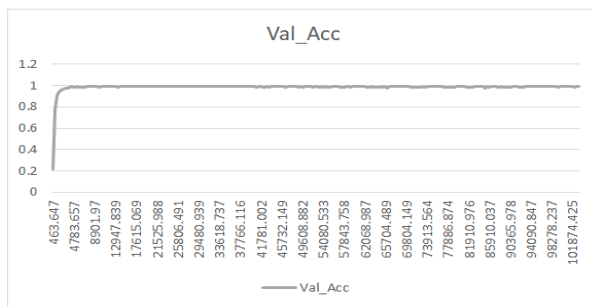
ASL letters	Accuracy
A	99.3443%
B	99.9003%
C	99.8087%
D	98.8082%
E	99.8294%
F	99.8212%
G	99.9069%
H	99.8539%
I	99.9994%
J	99.8561%



In the second step, all the ASL signs with 28 labels were used. All the 61.614 images were used, which was around at least 2200 images for each class. Around 0.30 of the training datasets was used, so there were around 43,120 images for the training and 18,480 images for the validation. An accuracy of 98.53% was obtained for the training set, and 98.84% for the validation set. In addition, the system displayed a high accuracy when new test data, which had not been used in the training, were introduced, whereby the lowest accuracy obtained was 98.6506% (Figure 3).



(a)



(b)

Figure 3. (A) Training accuracy with 28 classes; (B) Validation accuracy with 28 classes.

The highest accuracy of about 99.9567% was obtained for the letter I. The second highest accuracy of 99.9533% was for Delete, and 99.9069% for the letter G, as shown in Table 4.1. However, the lowest accuracy of 97.3182% that was obtained was for the letter M, and the second lowest accuracy of 97.3793% was for the letter N, as shown in Table 4. The reason is that the letters M and N have the same view and can only be distinguished by the position of the thumb. In addition, although the letters J and Z involve motion, the system was still able to successfully recognize them by taking three different viewpoints for each movement.

Table 4. Tested accuracy of each label when using the 28-class model.

ASL letters	Accuracy
A	98.5443%
B	99.1803%
C	99.8087%
D	98.8296%
E	99.8294%
F	99.8212%
G	99.9069%
H	97.8539%
I	99.9567%
J	99.7361%
K	99.7325%
L	99.8161%
M	97.3182%
N	97.3793%
O	99.7169%
P	99.6354%
Q	99.4118%
R	99.5423%
S	99.3665%
T	99.0484%
U	99.1445%
V	99.1119%
W	99.5514%
X	99.3292%
Y	99.5526%
Z	99.7799%
Delete	99.9533%
Space	99.8703%

### 5. Conclusion

In this paper, a real-time ASL fingerspelling recognition with CCNs networks was built using real colouring images to help creating a writing system for use as the input to a PC using a normal webcam. This paper discussed all the researches that have been done in this field. However, the model that was built in this work is the first fingerspelling recognition system to classify a total of 26 alphabets, including J and Z, and two classes for space and delete. New datasets were built to contain a wider variety of features for example different lightings, different skin tones, different backgrounds, and a wide variety of hand gestures. This work utilized 28 classes, comprised of 26 classes for American Sign Language alphabets from A to Z and a class for ‘space’ and another for ‘delete’. The system is a re-training VGG system that achieved a maximal accuracy of about 98.53% for training and 98.84% for the validation set. In addition, the system showed a high accuracy with the introduction of new test data that had not been used in the training, with the lowest accuracy achieved, which is 98.6506%. Attempts should be made to recognize the dynamic ASL gestures in future works. In addition, the dataset should be improved by involving more volunteers to cover all the different skin tones and increase the number of images in each class.

## References

- [1]. Garcia, B., & Viesca, S. A. (2016). Real-time American sign language recognition with convolutional neural networks. *Convolutional Neural Networks for Visual Recognition*, 2, 225-232.
- [2]. Van den Bergh, M., & Van Gool, L. (2011, January). Combining RGB and ToF cameras for real-time 3D hand gesture interaction. In *2011 IEEE workshop on applications of computer vision (WACV)* (pp. 66-72). IEEE.
- [3]. Liwicki, S., & Everingham, M. (2009, June). Automatic recognition of fingerspelled words in british sign language. In *2009 IEEE computer society conference on computer vision and pattern recognition workshops* (pp. 50-57). IEEE.
- [4]. Zafrulla, Z., Brashear, H., Starner, T., Hamilton, H., & Presti, P. (2011, November). American sign language recognition with the kinect. In *Proceedings of the 13th international conference on multimodal interfaces* (pp. 279-286).
- [5]. Pugeault, N., & Bowden, R. (2011, November). Spelling it out: Real-time ASL fingerspelling recognition. In *2011 IEEE International conference on computer vision workshops (ICCV workshops)* (pp. 1114-1119). IEEE.
- [6]. Kuznetsova, A., Leal-Taixé, L., & Rosenhahn, B. (2013). Real-time sign language recognition using a consumer depth camera. In *Proceedings of the IEEE International Conference on Computer Vision Workshops* (pp. 83-90).
- [7]. Dong, C., Leu, M. C., & Yin, Z. (2015). American sign language alphabet recognition using microsoft kinect. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 44-52).
- [8]. Singha, J., & Das, K. (2013). Hand gesture recognition based on Karhunen-Loeve transform. *arXiv preprint arXiv:1306.2599*.
- [9]. Sharma, R., Nemani, Y., Kumar, S., Kane, L., & Khanna, P. (2013, July). Recognition of single handed sign language gestures using contour tracing descriptor. In *Proceedings of the World Congress on Engineering* (Vol. 2, pp. 3-5).
- [10]. Starner, T., Weaver, J., & Pentland, A. (1998). Real-time american sign language recognition using desk and wearable computer based video. *IEEE Transactions on pattern analysis and machine intelligence*, 20(12), 1371-1375.
- [11]. Suk, H. I., Sin, B. K., & Lee, S. W. (2010). Hand gesture recognition based on dynamic Bayesian network framework. *Pattern recognition*, 43(9), 3059-3072.
- [12]. Admasu, Y. F., & Raimond, K. (2010, November). Ethiopian sign language recognition using Artificial Neural Network. In *2010 10th International Conference on Intelligent Systems Design and Applications* (pp. 995-1000). IEEE.
- [13]. Pigou, L., Dieleman, S., Kindermans, P. J., & Schrauwen, B. (2014, September). Sign language recognition using convolutional neural networks. In *European Conference on Computer Vision* (pp. 572-578). Springer, Cham.
- [14]. Escalera, S., Baró, X., Gonzalez, J., Bautista, M. A., Madadi, M., Reyes, M., ... & Guyon, I. (2014, September). Chalearn looking at people challenge 2014: Dataset and results. In *European Conference on Computer Vision* (pp. 459-473). Springer, Cham.
- [15]. Kang, B., Tripathi, S., & Nguyen, T. Q. (2015, November). Real-time sign language fingerspelling recognition using convolutional neural networks from depth map. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)* (pp. 136-140). IEEE.
- [16]. Tang, A., Lu, K., Wang, Y., Huang, J., & Li, H. (2015). A real-time hand posture recognition system using deep neural networks. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(2), 1-23.
- [17]. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.