Persian sign language detection based on normalized depth image information

Shahab Rajabi[1], Amir Mousavinia[2]

Abstract

This paper provides a system for Persian sign language recognition based on depth information in videos. In this method Microsoft Kinect Console in used for recording and extracting the depth images. At first, the shape of the hands were extracted by using deep based on threshold methods and in the next step, the wavelet transform and a new circular descriptor introduced in this system, extract the essential features. Neural networks are then used for an initial screening on the letters. Based on the extracted features by the circular descriptor, the recognition of the desired letter on the suggested alphabets of the output of neural networks is done using the support vector machines and based on the extracted features. The position of the user relative to the camera in the proposed system is between 80 to 130 centimeters and the proposed system can perform the recognition with an accuracy of 96.7 percent, in an acceptable time.

Keywords: Static Persian Alphabets, Kinect, Wavelet Transform, Circular Descriptor, Neural Network

1. Introduction

The investigation of sign language is a proliferating area of research to improve the well being of people with hearing impairments, as a result many studies have been done about hand gesture and especially sign language alphabet recognition. Some of these studies have used a specific Kernel. The main characteristic of these methods, is to project a special set of vectors to a higher order space. The probability of a better classification of patterns increases in this space (Moghadam, Nahvi, and Hassanzadeh (2011)). Modelling of hand and its geometry, form, and boundaries have been used in other studies for the recognition of the position

Shahab Rajabi, Department of Electrical Engineering, Faculty of Electrical Engineering, Khaje Nasir Toosi University of [1] Technology sh.rajabi@ee.kntu.ac.ir

[2] Amir Mousavinia, Computer Engineering Department, Faculty of Computer, Khaje Nasir Tusi University of Technology moosavie@kntu.ac.ir

of the hand relative to the camera, and it has yielded promising results. This method uses seven steps including convolution and sub-sampling, for gesture recognition (Wang et al. (2014) ، Huang et al. (2015)). Sign language recognition based on depth information Due to the improved human-computer interaction, user comfort and the availability of sensors at affordable prices, such as Kinect, are also considered by many researchers is usually more accurate and covers a wider range of words in comparison with the color or two-dimensional version (Agris, et al. (2008) ، Shah, Rathod, and Agravat (2014)). Verma et al have proposed a method using Kinect sensors that starts to record videos and keeps recording unless both of the user's hands go down then all of the recorded frames will be used by the system, for translation (Verma, Aggarwal, and Chandra (2013)). Yang extracted the depth information of hand movement by Microsoft Kinect and used a hierarchical random conditional domain for the recognition of some movements of hand. This method considers a hierarchical random conditional space for the recognition of sign candidates by using the movement of hand, then utilizes an enhanced mapping to distinguish between hand gestures and categorized signs (Yang (2014).

In another Kinect-based method, a codding algorithm is used to model different signs. In this method, based on the differences between signs, a specific number of videos in each category of them, is trained. Then each video is described by a set of similarities between frames and selected samples .finally the best samples are formulated for a framework and a video categorizer of symbols is generated simultaneously, for the recognition of signs (Sun et al. (2013)). Bengio have used the Spare-Auto encoder (SAE) algorithm and depth information provided by Kinect, for feature training and then used PCA algorithm in SVM category, to maximize the accuracy of the results (Li et al. (2015)).

In another method, have used a convolution neural network instead of constructing the complex features, that extracts the features automatically and they were able to reach the accuracy of 91.7 percent, for Italian sign language detection.

The discussed methods mainly model the time information and are not able to adapt to a lot of changes in sign words. To address this issue, since long short-term memory (LSTM) can model time sequence information well, an end-to-end system for SLR based on LSTM is proposed. Based on moving path, this method considers 4 skeletal joints as input without any prior knowledge and explicit features (Huang et al. (2015)). This architecture method is comprised of 7 layers, including the input layer. The first layer has a 12 dimensional vector, which includes four 3D coordination vectors that represent the skeletal joints. The next layer is a LSTM layer with a dimension of 512 and then there are two fully connected layers. The first and the second fully connected layers include 512 and 100 neurons, respectively which are connected to 100 classes that follow a soft max function and adds to a layer. The last layer is the output layer of selected class and then recognition is performed (Liu et al. (2016)). Almeida has also proposed a method for Brazilian sign language which at first extracts seven features from RGB-D images. Each feature is related to one or two structural element in BSL and then the relationship among these extracted features and the structural elements, is calculated based on movement patterns and the position. In the end, recognition is done based on SVM categorizer and the accuracy of 80% is reported (Almeida et al. (2014)). Particle filters also, have an important role in human gesture recognition. In a study, Lim has proposed a serial filter based on covariance matrix, for language recognition. After performing the pre-analysis for hand recognition, this filter is used to mark the hand in a sequence of frames that tracks both hands at the same time. After the generation of covariance matrix of features in whole of tracked area by reducing dimension, ASL recognition has been done with 87.33% rate (Lim, Tan, and Tan (2016)).

Zhao has labeled the behavioral model with a graph and performs the categorization by matching the database graph with the input image (Zhao and Martinez (2016)). To prevent any complications in matching and the challenges of recognition of signs in video interconnected sequences, a Kernel graph is considered to increase speed and accuracy and a specific approach in the modelling of sub categorical signs is used by them, respectively. This EDP framework works by combining spatiotemporal clustering and Dynamic Time Wrapping techniques. Considering the fact that sign language both includes space feature and time feature vector, dynamic time wrapping is used to measure the distance of 2 close signs. This distance is then used as a temporal feature vector when spatial feature vectors are being clustered by means of Minimum Entropy Clustering. This process is done in a recursive manner to cluster all the intermediate movements without the use of explicit or implicit modelling, dynamically (Elakkiya and Selvamani (2017)). The best system for Persian sign language recognition so far is proposed in (Karami, Zanj, and Kiani (2011)), by using wavelet transform and neural networks and the necessary images for sign alphabet are provided by a digital camera. The extra parts of the color images are deleted, their sizes are changed and they are transformed to black and white. Then the discrete wavelet transform is applied on the black and white images and some of the features are extracted. At last, the extracted features are used to train a multi-layered neural network perceptron algorithm. In this system no gloves or marking systems were used and this system only needs the bare hand for recognition and is comprised of two parts: feature extraction phase and categorization phase. In the proposed system, wavelet transform is used for feature extraction since it is simple and efficient in flexible parts of human body. After many simulations in this study, a neural network with 92 nodes, one hidden layer with 21 neurons and 5 linear output neurons were used.416 images from 32 Persian sign language (13 images per letter) were used. +1 and -1 were used as the bipolar output of the 32 signs. In this method, the multi-layered neural network is trained by neural network toolbox in Matlab and variable learning rate back propagation algorithm is used. This method works well in noise patterns in training and increases the accuracy of trained neural networks in unfamiliar samples. The efficiency of the multi-layered neural network is tested by the mean squared error of the outputs of the network and the desired output.

The multi-layered neural networks in this method are tested by 224 images (7 images per sign). The experimental results show that this recognition system is as accurate as 94.06 percent. In most of the previously proposed methods, there were limitations for the users such as putting on gloves, or a complete laboratory space with a simple background, or use of color space in the recognition of different hand gestures in some methods. It is obvious that in the presence of noise, the accuracy of these algorithms decrease significantly (Neiva and Zanchettin (2018) ، Wang, Liu, and Chan (2015)). Also some of the algorithms require a significant amount of time and calculation, and are not suitable for applications and real-time programs (Albrecht, Haber, and Seidel (2003), Caillette, Galata, and Howard (2008). In addition, in a large number of hand-detect algorithms, as well as all algorithms that are effective in identifying the Persian sign alphabet, are highly dependent on the user's angle of the camera, and so most of them use a specific dataset with simple background. (Francke, Solar, and Verschae (2007) ،Krotosky and Trivedi (2006))

We propose a system in this paper without any limitations such as gloves for the hearing impaired, with desired accuracy, and efficient and suitable calculations for immediate implementation, to recognize Persian sign language in a typical and real environment of a video.

2. The proposed system

Deep images are more suitable for tracking or recognition of hand and fingers in videos, and after separating the hand from the background, it is possible to reach a more desirable result than color images, by selecting the suitable features and extracting them well, based on a powerful categorizer for the recognition of human hand gestures. In this paper the Kinect device which was introduced by Microsoft for its Xbox Console was used. The block diagram for the proposed system can be seen in figure 1.

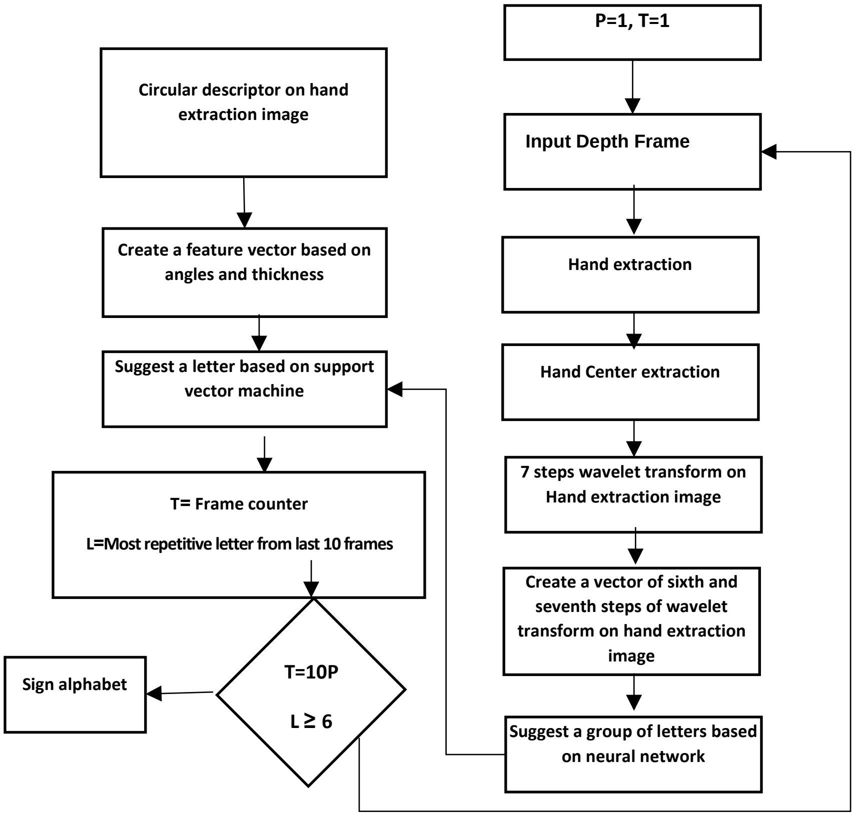

P=1, T=1
Input Depth Frame
Hand extraction
Hand Center extraction
7 steps wavelet transform on Hand extraction image
Create a vector of sixth and seventh steps of wavelet transform on hand extraction image
Suggest a group of letters based on neural network
Circular descriptor on hand extraction image
Create a feature vector based on angles and thickness
Suggest a letter based on support vector machine
T= Frame counter
L=Most repetitive letter from last 10 frames
T=10P
L ≥ 6
Sign alphabet

Figure 1. The block diagram for the proposed system

2.1 Extracting human hand from deep images

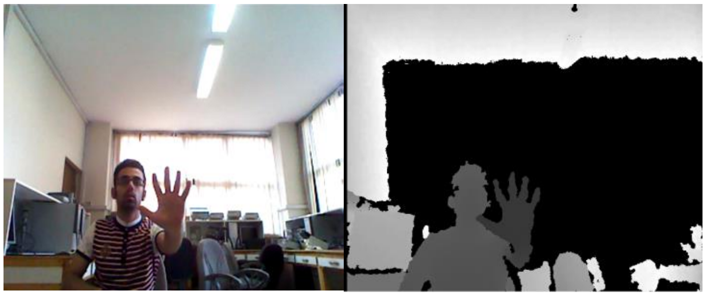
In this system the deaf person puts their hand in front of themselves in each frame. Also in a typical condition in this system, each deaf person moves their hand a little and takes it a little further from the camera. To extract the human hand in this situation, calculating a fixed threshold value based on depth, is the best way to separate the hand palm. This threshold due to low differences in thickness of human hands is almost a constant and suitable quantity which is calculated based on the diameter of a normal human hand that is standardized, and is defined as equation 1:

$$D(n) = D(n) + \alpha \quad (1)$$

In this equation, n, is the number of pixels in an image with the minimum non-zero depth. D (n) is the depth of pixel n, and α is a constant that is normalized based on the standardization of the hand. The hand pixels are extracted based on equation 2 and they eliminate the background:

$$I(q) = \begin{cases} D(q), & 0 < D(q) \leq D(n) + d \\ 0, & otherware \end{cases} \tag{2}$$

In this equation, I is the extracted picture of the hand, q is the pixel counter in image and D (q) is the depth amount of pixel number q. In the proposed system, d is set at 55 mm for a reference hand, which will be defined below. In figure 2, a sample of a Kinect color and depth images for hand extraction, can be seen. It can be seen that the extraction of hand has been done.



(b) **(a)**

Figure 2. a) Kinect RGB Image b) Kinect Depth Image

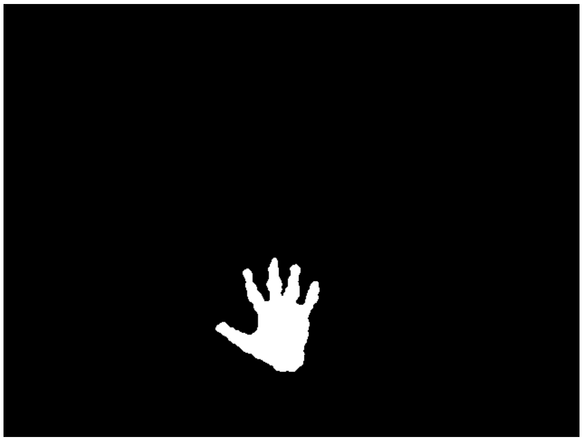


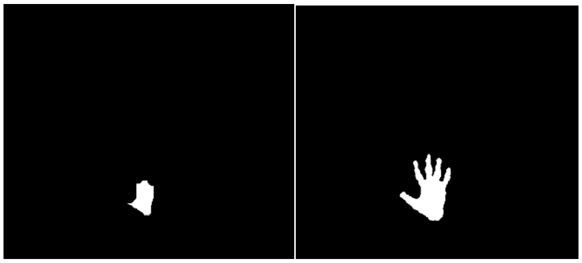
Figure 3. Hand extraction

2.2 Extraction of the center and the radius of hand

These days, morphology algorithms have found many applications in signal processing such as, identification of fractions, points of strains, and also image restoration. In this system extraction of the center and the radius of hand, based on the type of image the algorithm delivers, and the elimination of the fingers, it is possible to provide a good approximation of the palm of hand. Erosion of Image A by a structure element B follows equation 3.

$$A \ominus B = \{ z \in E | B_z \subseteq A\} \tag{3}$$

In equation 3, E is an euclidean space and A is a binary image. B_z is in fact the inverse of B .Figure 4 is an image of a human hand related to erosion of figure 3 that structure element is disk with radius r. As we can see, the palm of the hand remains and the fingers are eliminated.



(b) (a)

Figure 4. Erosion process: a) Human hand extraction b) Extraction palm of hand by erosion of primitive image.

To be able to calculate the center of the hand from the output images of the last step, first the palms should be labeled from the highest amount of interconnectedness to the lowest. This labeling indicates the weight of that interconnected palm. Equation 4 shows the weight allocation basis:

(4)

$$W_1=nX$$

$$W_2 =(n-1)X$$

.

.

.

$$W_n=X$$

In this equation, n is the number of interconnected mass in the image and X is the lowest weight of a palm in an image which belongs to the Wn. The summation of all the weights in the image can be seen in equation 5:

$$\sum_{V=1}^{n} W_V = 1 \tag{5}$$

This equation can be simplified as follows:

$$\sum_{L=0}^{n-1} (n-L)X = 1 \tag{6}$$

$$X = \frac{1}{\sum_{L=0}^{n-1}(n-L)}$$

With the use of equation 6 is assigned a special weight to any pixels of every mass (v) then the center of palm O, is calculated by equation 7:

$$O = \frac{\sum_{V=1}^{n}\sum_{K=1}^{CV} W_V q_{k_v}}{\sum_{V=1}^{n} W_V C_V} \tag{7}$$

In equation 7, Cv shows the number of pixels of the palm v, and q_{k_v}shows the the coordination of the Kth pixel from Vth cluster.

After the calculation of the center, we now turn to the calculation of the radius of the palm, r. We start r from the least possible non-zero amount and consider a cost function like the one in equation 8 for it.

$$J(r) = \sum_{\theta=0}^{2\,\pi-Res(\theta)} |(r - rp_\theta)(Cos(\theta) + Sin(\theta)|^2 \tag{8}$$

In this equation, the more Res (θ) is decreased, the better the accuracy of r will be and θ will increase in J(r) equation. rp_θ is in fact the length of radius in the direction of θ and its magnitude is calculated to the point of reaching the first non-zero amount in that palm, as the length increases in the direction of θ, and a constant magnitude is reached for all needed angles. In equation 8, after differentiating relative to r, the cost function will be equaled to zero and based on equation 9, the optimized magnitude of r is calculated. Figure 5 indicates the extraction of the center of hand.

$$J'(r) = 0 \tag{9}$$

$$\sum_{\theta=0}^{2\,\pi-Res(\theta)} (-rp_\theta)(Cos(\theta) + Sin(\theta))^2(r - rp_\theta) = 0$$

$$r = -\frac{\sum_{\theta=0}^{2\,\pi-Res(\theta)}(rp_\theta(Cos(\theta) + Sin(\theta))^2}{\sum_{\theta=0}^{2\,\pi-Res(\theta)} rp_\theta(Cos(\theta) + Sin(\theta))^2}$$

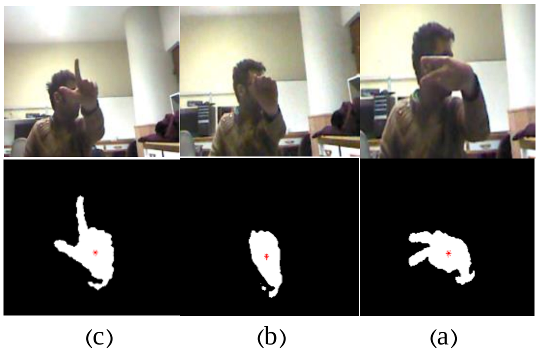


Figure 5. Hand extraction and hand center extraction in 3 differences signs

2.3 Using Haar wavelet for feature extraction

In the proposed system, haar wavelet transform is used for encoding the Persian alphabet images in different families. In this section bank filter method is used to apply wavelet transform in an image and h0 and h1 as vector bases in the haar wavelet transform filter bank, are illustrated in figure 6.

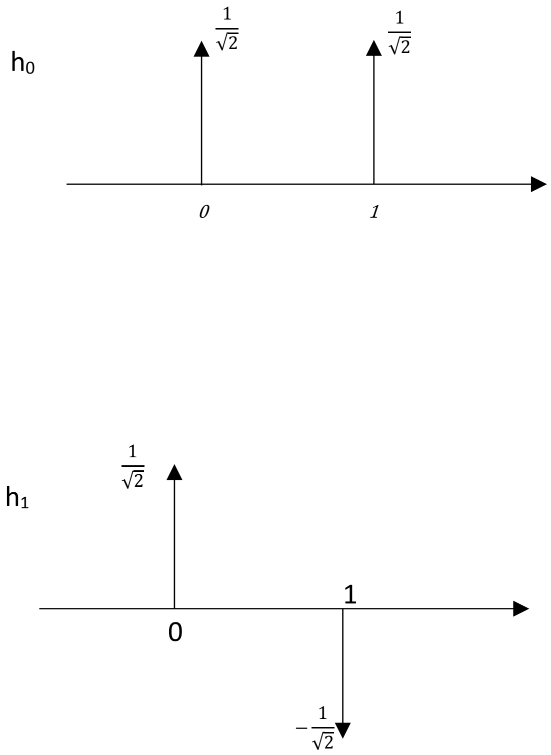


Figure 6. Vector base coefficients Haar Wavelet in filter bank

After applying one step of a wavelet transform, the approximation of an image and the horizontal, vertical and diametrical details of the image, for an F image with m*n dimension and the equation (10) will be at hand.

(10)

$$A_1 = \frac{1}{2}(F(2\,m\,,2\,n) + F(2\,m - 1\,,2n) + F(2\,m\,,2n - 1) + F(2\,m - 1\,,2\,n - 1))$$

$$H_1 = \frac{1}{2}(F(2\,m\,,2\,n) + F(2\,m - 1\,,2n) - F(2\,m\,,2n - 1) - F(2\,m - 1\,,2\,n - 1))$$

$$V_1 = \frac{1}{2}(F(2\,m\,,2\,n) - F(2\,m - 1\,,2n) + F(2\,m\,,2n - 1) - F(2\,m - 1\,,2\,n - 1))$$

$$D_1 = \frac{1}{2}(F(2\,m\,,2\,n) - F(2\,m - 1\,,2n) - F(2\,m\,,2n - 1) + F(2\,m - 1\,,2\,n - 1))$$

An overall scheme of wavelet transform based on filter bank is illustrated in figure 7.

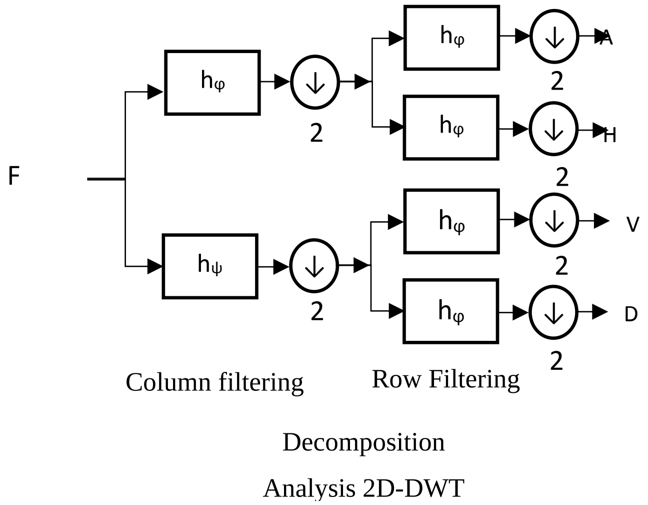


Figure7. Wavelet Transformation Schema

In the proposed system, first the size of each input frame given by the abovementioned process, will be normalized to 200*300, to prepare it for applying wavelet transform. We call this process pre-analysis from which a sample can be seen in figure 8.

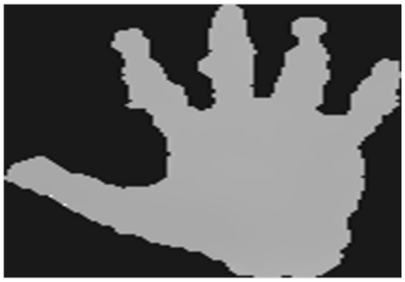


Figure 8. The image of palm based on Depth value

After the pre-analysis for each frame, applying the wavelet transform will begin. Based on different experiments, haar wavelet transform has been chosen due to its simplicity and efficiency, and after 7 steps of wavelet transform which is done on the approximation of the previous wavelet, the feature vector in (11) will follow. The following goals are desired from this vector:

1) Assigning a code to each image and decreasing the feature space

2) A desirable, sufficient, and efficient feature vector to differentiate various letters

$$F = [A(6)\ \ H(6)\ \ V(6)\ \ D(6)\ \ H(7)\ \ D(7)] \tag{11}$$

A, H, V, and D components are in fact image approximation, horizontal details, verticals details, and diametrical details coefficients, respectively. The numbers in the parentheses indicate the level of the applied wavelet transform. The coefficients will be defined as arranged in columns, and a vector is generated from F with 92 dimensions, which is the code for the input hand gesture. The generated images from the coding can be seen in figure 8.

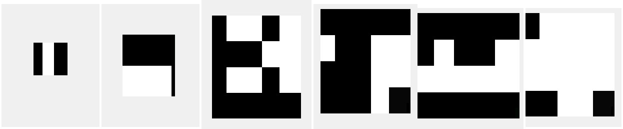


Figure 9. Figure 8 encoding by applying wavelet transform: A) the sixth level approximation; b) the horizontal details of the sixth level; c) the vertical details of the sixth level; d) the diameter of the surface of the sixth; e) the horizontal details of the seventh surface; c) the seventh level vertical detail

In fact, in this step, we have tried to extract the features of images from lower resolutions of the original pictures and in the end with the help of a categorizing algorithm, each code is assigned to a family of letters, which will be discussed below.

2.4 Using neural networks to classify the features of wavelet transform

In recent years, use of neural network based categorization to solve complex nonlinear problems, has grown extensively. In this system, like (Van den Bergh and Van Gool (2011)) neural network was used to categorize the features extracted from the wavelet transform. In the designed neural network, the tangent sigmoid transform function has been used as a middle layer nonlinear function to train the hidden layer, which is indicated in equation 12 (Vogl et al (1988)).

$$Tansig(n) = \frac{2}{1+e^{-2\,n}} - 1 \tag{12}$$

In this system, an output neuron with a value of 1 is considered for each class and other neurons will naturally be set to 0. Also in the output section for selecting classes, the maximum linear norm 1 function encoders the out with 0 and 1, while in our proposed method the output is encoded with a positive or negative weight. In figure 10, the sample system is comprised of 3 layers of input, output and hidden layer.

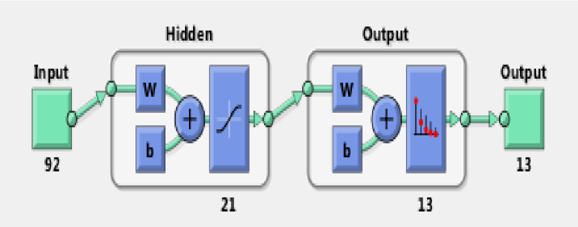


Figure 10. Neural Network diagram block

In the proposed system, 12 letters and their hand gesture have been investigated. Altogether 13 neurons are considered for the output. 20 frames are considered for each letter and each frame is applied as an input to system, with the slightest movement or turn of the hand and altogether 260 frames were considered to train this system. The training equation of each layer is like (13).

$$net = w^t F = w_1 f_1 + w_2 f_2 + \cdots + w_{92} f_{92} \tag{13}$$

In this equation w and F, are the weight vector of each neuron after training and the feature vector derived from equation (11), respectively. After training the network and reaching an approximate function for each letter, the test data can be given to the network as input. The overall output equation for using this network is as follows:

$$y = f(net) = f\left(\sum_{i=1}^{92} w_i f_i\right) \tag{14}$$

This equation suggests the letter, after calculating the least square root for each of these outputs for different letters (Hagan et al (2014)). This neural network is used to recognize the 13 proposed families, and because of the problems due to the varieties in hands and differences of distances, it is not used lonely to identify alphabets. In table 1, 13 suggested groups that occur when a neural network detects a single letter:

Table 1. 13 suggested groups that occur when a neural network detects a single letter

Letter	**Suggested group of letter**
اَ	اَ ، آ ، ع ، غ
آ	، ه اَ ، آ ، ی
ع	ع ، غ ، ن
گ	ع ، ه ، گ ، ن ، ل
غ	اَ، ع ، ه ، غ ، ن
ه	ه ، ن

ل	آ ، ل ، گ ، ی
م	م ، ع
ن	ه ، م ، ن ، ی ، گ
أُ	گ ، ل ، أُ
ط	ل ، ط ، ی ، م
ی	ل ، ی ، گ

2.5 Adaptive plane model based on three dimensional regression

In this paper a three dimensional backward polynomial model is proposed for each pixel plane of hand. As it is mentioned above, what is needed as input in this section, is the minimization of the space between the points and the approximate plane it is evident that this will never be zero and we are looking for the least amount. In equation (15) a three dimensional plane is defined:

$$ax + by + z + d = 0 \tag{15}$$

The distance between a three dimensional point (xp,yp,zp) relative to a plane characterized by equation (15), is calculated based on equation (16):

$$D = \frac{ax_p + by_p + z_p + d}{\sqrt{a^2 + b^2 + 1}} \tag{16}$$

To reach the best plane characteristics including a, b, and d components, the plane optimization problem should be solved with its best adaptation relative to the extracted points from the hand. If we consider R as the number of extracted points from the hand, D_T is total squared of the distance from each point to the plane, equals (17):

(17)

$$D_T = \sum_{i=1}^{R} |D_i|^2 = \sum_{i=1}^{R} \left| \frac{ax_i + bx_i + z_i + d}{\sqrt{a^2 + b^2 + 1}} \right|^2$$

Differentiation relative to a, b, and d components will yield the following equations:

(18)

$$\frac{dD_T}{db} = 0$$

$$\sum_{i=1}^{R} ({y_i}^2)b^3 + [y_i(ax_i + z_i + d - 1)]b^2 + (a^2{y_i}^2 + {y_i}^2 - ax_i - z_i - d)b + (a^3x_iy_i + a^2y_iz_i + a^2y_id + ax_iy_i + z_iy_i + dy_i) = 0$$

(19)

$$\frac{dD_T}{da} = 0$$

$$\sum_{i=1}^{R} (x_i)^2a^2 + [x_i(by_i + z_i + d - 1)]a^3 + (b^2{x_i}^2 + {x_i}^2 - by_i - z_i - d)a + (b^3y_ix_i + b^2x_iz_i + b^2x_id + bx_iy_i + z_ix_i + dx_i) = 0$$

(20)

$$\frac{dD_T}{dd} = 0$$

$$\sum_{i=1}^{R} ax_i + by_i + z_i + d = 0$$

$$d = \frac{\sum_{i=1}^{R} ax_i + by_i + z_i}{R}$$

If we combine the abovementioned equations, a, b, and d components will follow which can be optimally calculated in MatLab by the graphical environment cftool. Figure 11 shows two samples of approximated planes by a polynomial plane.

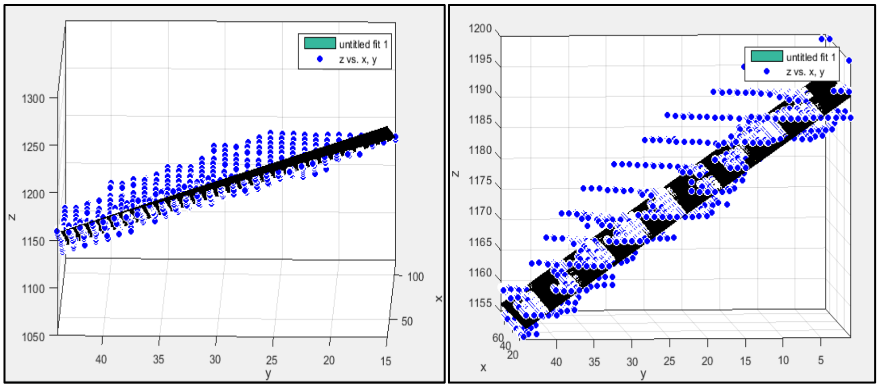


Figure 11. 3D estimates of flat mass perpendicular to Kinect by polynomials

By using the angles of the approximated planes relative to the normal vectors of the coordinate planes, and by applying the rotation matrices in (21) on depth pixels, it is possible to eliminate the unintentional and minor movements of the user's hand relative to the axis perpendicular to the camera, for a better recognition of the sign letter. Then, the angle of this normal vector to the normal vector of the camera z (0, 0, 1) and the x (1, 0, 0) and the y (0, 1, 0) will be calculated. The rotation matrices based on these angles are calculated for each page (Craig, J, J. (2009)).

(21)

$$R_z = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

$$R_y = \begin{bmatrix} cos\,(\beta) & 0 & sin\,(\beta) \\ 0 & 1 & 0 \\ -sin\,(\beta) & 0 & cos\,(\beta) \end{bmatrix}$$

This part is important because it can provide the hand angle as an input for a feature vector, in addition to preparing the hand plane for the next step descriptor. After applying the mentioned operations on figure 11 and eliminating the rotations, the results are magnified and shown in figure 12.

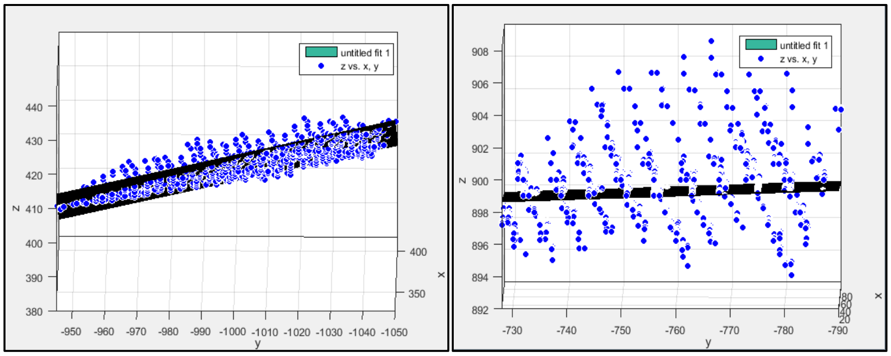


Figure 12. Eliminating the rotations by applying rotation matrices on 3D points of Figure 11

2.6 Normalization coefficients

This is the most important characteristic for the stability of the proposed system because it makes it possible that the equations in later steps for the recognition of human hands stay unchanged, although people and consequently the depth will change. The normalization coefficient for a person’s hand and also a standardization depth coefficient is given is (22) and (23):

$$N_r = \frac{R_{New}}{R_{Refrence}} \tag{22}$$

$$N_D = \frac{D_{New}}{D_{Refrence}} \tag{23}$$

In (22) and (23) R_{New} is the hand radius of the user, and $R_{Refrence}$ is the reference hand radius in the radial normalization coefficient N_r, and in depth normalization coefficient N_D, D_{New} is the length () of the user’s hand and $D_{Refrence}$ is the length of the reference hand. Both these parameters are used simultaneously because they eliminate for the effect of length and distance of the hand on the proposed algorithm in later equations.

2.7 Circular descriptor

In this section we introduce a new descriptor that can make a significant difference in different hand gestures. After calculating the center of the hand and eliminating the unwanted rotation of the user, some equations will follow that the adaptive radius to calculate the number of open and closed fingers, are based on them. The equation for this radius is set for a range of 80 to 130 centimeters from Kinect, based on many studies and experiments which are compared and matched with a sample size of more than 50 hands. Equation (24) shows this radius:

(24)

$$R = \begin{cases} \alpha \times N_D \times N_r & , N_D < 1 \\ \dfrac{\alpha \times N_r}{N_D} & , N_D \geq 1 \end{cases}$$

α is a constant that is calculated in different experiments and in two conditions that differ in the distance between the hand and the camera and N_r. The circle descriptor provides interesting features such as angles between upright fingers and also finger thickness, to identify if two or more fingers are close to each other or if that is a single finger. By averaging the designated points on each finger, in figure 13, one criterion is chosen. This criterion for each finger as a representative of that finger can then determine the angle of that finger relative to the center of the hand.

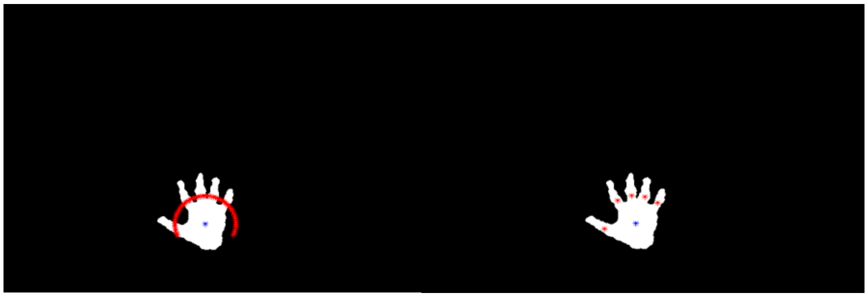


Figure 13. Display the circular descriptor, Palm and representative of each finger

The angles of each finger is compared with the center of the hand by equation 25.

(25)

$$\theta = tan^{-1}(\frac{y_{PALM} - y_{FINGER}}{x_{PALM} - x_{FINGER}})$$

These angles will each be considered a feature in the input feature vector. The thickness of each finger is then determined based on the number of white points on each connected part of the descriptive circle and by considering the normalization coefficients. So the resulting feature vector from this descriptor for each frame containing a hand gesture, can be characterized by 10 features that can be seen in X= [T1 T2 T3 T4 T5 θ1 θ2 θ3 θ4 θ5]. If there is no white part, 0 is considered for its angle and thickness. So the feature vector in this section has 10 features including angles of each connected part with the color intensity of 1 and also their angles.

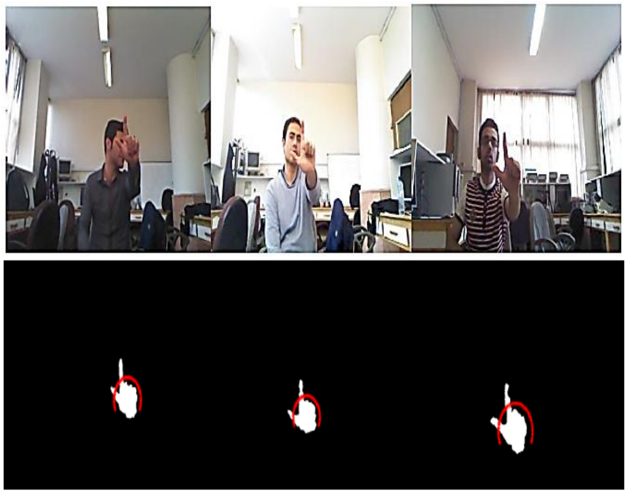


Figure 14. Picture of same sign by 3 different people and apply descriptor circular

In this section we consider the radial and deep reference values for our normalization coefficients, which, according to numerous experiments, this value is obtained for a reference radius of 36 pixels and for a reference depth value of 86 centimeters. The important point in the descriptor's radius is the magnitude of α. Due to the variability of the error of Kinect in determining the depth, the descriptor's radius cannot always be calculated based on one normalization coefficient. Equation (26) shows the magnitude of α in the acceptable ranges and different hands, in the proposed method, which is 80 to 130 centimeters.

$$\alpha = \begin{cases} 65, & (N_D < 1 \; OR \; N_D > 1.11) AND (N_r \leq 0.85) \\ 50, & Otherwise \end{cases} \tag{26}$$

This range has been calculated after many experiments about the relation of Kinect error and the descriptor's radius. After determining α and the reference values, by any radius and minimum distance in each frame, the normalization coefficients are obtained. Then, 2/3 of a circle whose points are selected with accuracy π/50 for thickness of the fingers and angles will be plotted. The anatomical feature vector for these three images contain the calculated thickness and angles of the centers of each connected white parts relative to the palm is [116.1662 166.3964 0 0 0 3.8972 5.8458 0 0 0], [114.1633 182.5638 0 0 0 4.0521 4.0521 0 0 0] and [119.7736 161.2114 0 0 0 4.9567 4.1306 0 0 0], respectively. In figure 15 the result from the proposed descriptor can be seen for different hand position from left to right, 101, 127, and 84 centimeters, respectively.

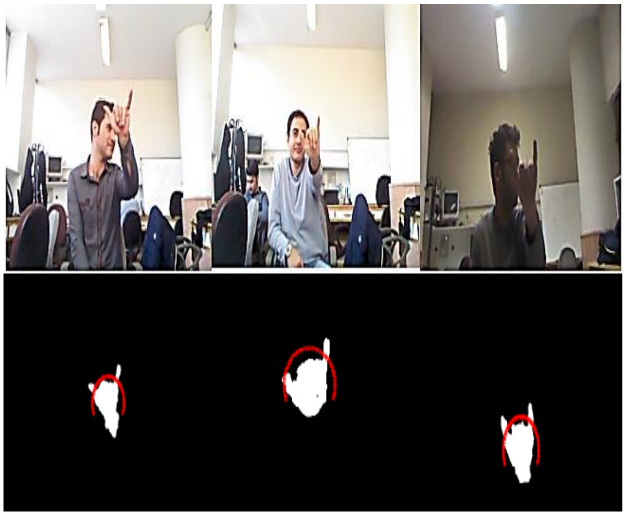


Figure 15. Picture of same sign by 3 different people and apply descriptor circular with 101, 127 and 84 centimeters respectively from left to right.

2.8 Classification based on the anatomical features of the hands in a family of neural networks by Support Vector Machine

One of the methods that is widely used for classification is the support vector machine. The basis of this method is a linear classification of the data. Support vector machine is in fact a binary classifier that separates two classes by a linear boundary. In this method by using all the bands and an optimization algorithm, the samples that make up the class boundaries are derived. These samples are called support vectors. The Kernel function is a linear separator which allows one to make linear separators in a feature space, although they are in a nonlinear space.

In a general sketch of a support vector machine with two classes, it is assumed that each x_k, which is the feature vector of kth sample, is transformed with a φ linear or nonlinear transformation depending on the user, to a higher order space to differentiate the classes better. This is shown in equation (27) by y (Duda, Hart, and Stork (1973).

$$y = \varphi(x_k) \tag{27}$$

It is assumed for each pattern or sample n=1,2,…,k that z_k=-1 or z_k=+1 labeling, is based on whether the k pattern is in ω_1 or ω_2 class. A linear differentiator in y space, which is also used in our proposed method, is calculated as in (28) by the dot product of the weight vector a, of each of the features, in the transformed pattern by φ transformation.

$$g(y) = a^t y \tag{28}$$

In which $a_0 = w_0$ and $y_0 = 1$. So one of the separating planes of the two classes will be (29) :

$$z_k g(y_k) \geq 1 \tag{29}$$

Now the objective is to find the separating plane that can have the maximum margin from the two classes of ω_1 and ω_2 , which makes it possible to make the best classification by the support vector machine. The distance from each transformed y vector relative to the separating plane is $\frac{|g(y)|}{\|a\|}$ and assuming a positive marginal gain b, we will finally desire a weight vector that can maximize the value of b, based on equation (30).

$$\frac{z_k g(y_k)}{\|a\|} \geq b \tag{30}$$

In this method, to separate the existing classes in the algorithm, a SVM based on a linear Kernel on the basis on a soft margin, was used. In this system, the features of each class are made up of 30 examples of educational data by extracting anatomical features. All classes of a same family selected by the neural networks are tested and, by statistical mode, the class that was most successful in the competition with other classes, can be chosen as the output class.

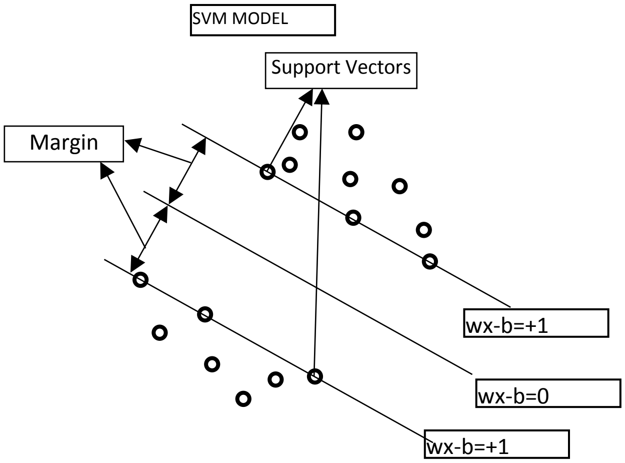


Figure 16. Support Vector Machine model for two types of data

To prevent noise applied to sequential frames of a video, apply the Median filter to a 10x1 window of frames. This algorithm is performed one time after each 2 Consecutive frames due to very low differences between them and also improve speed of algorithm, After that, the output from the neural network and the support vector machines are saved and the operation is repeated for the next two frames. This process is repeated 10 times and statistical mode is calculated. If in more than 6 frames the output of support vectors point to a letter, the letter will be displayed. The objective of this process is the elimination of noise from playing a video, from the output and making the algorithm more reliable. An example of a separation done by this classifier in shown in figure 16.

3. The Experimental results

In this section, the results of the experiments done on each part of the algorithm is explained. After illustrating the results from the circular descriptor it is time for presenting the output from the proposed algorithm, in each frame. In figure 17 three letters of A, T, and AA, can be seen in the graphic environment of MATLAB.

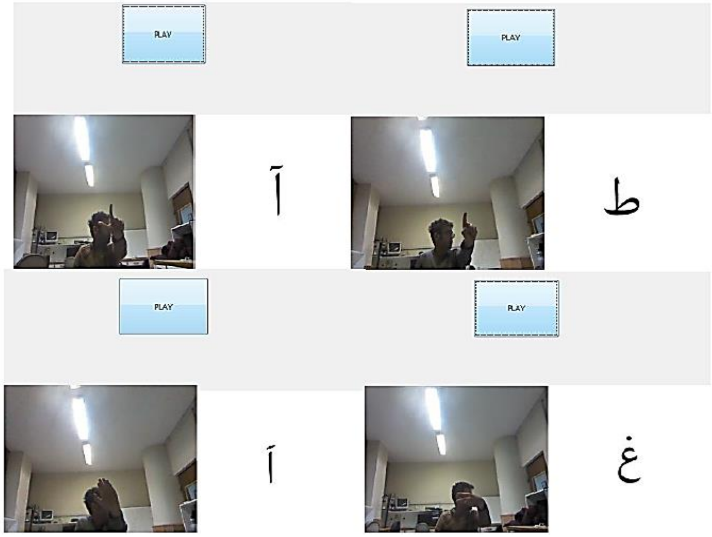


Figure 17. Display four letters of the deaf alphabet under the proposed system

We aim to compare the proposed method with the best method so far, for Persian sign language recognition ((Karimi, Zanj, and Kiani, 2011). In this system we considered 30 samples for each letter, which means altogether 360 frames were used to train the neural network. In the proposed method, color images were used for the recognition of sign language. As a result, our criterion for this comparison, is the success rate of the proposed method in the recognition of letter. This is presented in (31).

$$Rate = \frac{Number\ of\ correctly\ classified\ signs}{Total\ number\ of\ signs} \times 100 \tag{31}$$

In this system, 423 frames were made, as shown in figure 18.

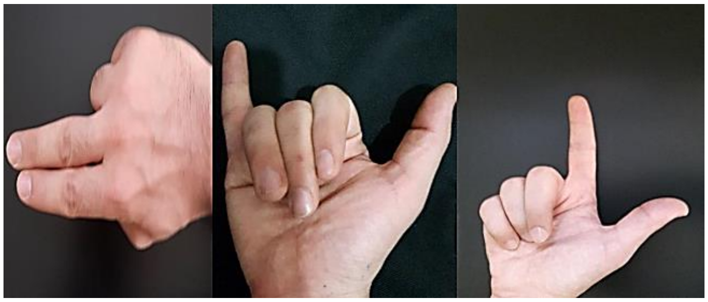


Figure 18. Dataset for (Karimi, Zanj, and Kiani, 2011)

In the generation of the data set in this paper a black and simple background was used and all the pictures were taken in a distance of about 50 CMs. The camera was a Galaxy A8, 16 mega pixels. Two people worked to provide 276 pictures to train the designed neural network and it's available in DBBI [3]database. In table 2, the accuracy for each letter is shown in comparison with the proposed method.

Table2. Compare the success rate of signs in the proposed method and (Karimi, Zanj, and Kiani, 2011)

Sign	Accuracy (Karami, Zanj, and Kiani 2011)	Accuracy of proposed system
اَ	81.81%	96%
آ	81.81%	98.71%
ع	90.9%	89.28%
گ	95.45%	96%
غ	100%	100%
ه	90.9%	95.85%
ل	100%	94%
م	90.9%	100%
ن	90.9%	91.5%
اُ	81.81%	100%
ط	100%	100%
ی	90.9%	100%

As it is obvious, the rate of recognition in the proposed method is significantly improved and the overall recognition rate for these letters is 96.7%. In (Karami, Zanj, and Kiani 2011), It calculates the training data entered into its neural network as a testing data and it Can increase its accuracy considerably.

In this paper, we reached 91.28% for our generated data set. This is true while we applied a simple and minimally changed background, to that system. In addition to this, the speed of the proposed method was 0.25s for each frame, working with MATLAB in a computer with 4G RAM, and a 2.3 GHz CPU, which is an appropriate time for being online.

3. Conclusions

The objective of the current study was to provide a novel and more efficient method for sign language recognition in videos. We tried to propose an efficient method considering both the accuracy and performance time to make it suitable for commercial use. While there have been many advances in artificial intelligence, although there are more than half a million deaf persons in Iran, not paying enough attention

[3] Digital black background images

to these methods, have resulted in inability of many people with hearing impairments to communicate with others. The proposed method does not require limitations such as hand gloves or background specifications, and people can communicate via this system, with minimum requirements.

References

1) Wang, C., Liu, Z., Chan S. (2015). Superpixel-Based Hand Gesture Recognition with Kinect Depth Camera. IEEE TRANSACTIONS ON MULTIMEDIA,17, pp. 29 – 39.
Wang, Liu, and Chan (2015)

2) Almeida, S.G.M., Guimar˜aes, F.G., Ram´ırez, J.A., (2014). Feature extraction in brazilian sign language recognition based on phonological structure and using rgb-d sensors. Expert Systems with Applications, 41, pp.7259–7271.
Almeida et al. (2014)

3) Neiva D. H., Zanchettin C. (2018). Gesture Recognition: a Review Focusing on Sign Language in a Mobile Context. Expert Systems with Applications.
Neiva and Zanchettin (2018)

4) Lim, M.K., Tan, W.C. A., Tan, C.S. (2016). A feature covariance matrix with serial particle filter for isolated sign language recognition. Expert Systems with Applications, 54, pp.208-218.
Lim, Tan, and Tan (2016)

5) Zhao, R., Martinez, M. A., (2016). Labeled Graph Kernel for Behavior Analysis. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 38, pp.1640 – 1650.
Zhao and Martinez (2016)

6) Sun, C., Zhang, T., Bao, B., Xu, C., Mei, T. (2013). Discriminative Exemplar Coding for Sign Language Recognition with Kinect. IEEE TRANSACTIONS ON CYBERNETICS, 43, pp.1418 – 1428.
Sun et al. (2013)

7) Elakkiya, R., Selvamani, K. (2017). Enhanced dynamic programming approach for subunit modelling to handle segmentation and recognition ambiguities in sign language. Journal of Parallel and Distributed Computing.
Elakkiya and Selvamani (2017)

8) Wang, L. C., Wang, R., Kong, D., Yin, B. (2014). Similarity Assessment Model for Chinese Sign Language Videos. IEEE Transactions on Multimedia, 16, pp.751 – 761.
Wang et al. (2014)

9) Li, S.Z., Yu, B., Wu, W., Su, S.Z., Ji, R.R., (2015). Feature learning based on SAE–PCA network for human gesture recognition in rgbd images. Neurocomputing, 151, pp.565–573.
Li et al. (2015)

10) Huang, J., Zhou, W., Li, H., Li, W., (2015). Sign language recognition using 3d 1430 convolutional neural networks, in: Multimedia and Expo (ICME), 2015 IEEE International Conference on IEEE, pp.1–6.
Huang et al. (2015)

11) Liu, T., Zhou, W., Li, H., (2016). Sign language recognition with long short-term memory, in: Image Processing (ICIP), 2016 IEEE International Conference on IEEE, pp.2871–2875.

12) Van den Bergh, M., Van Gool, L. (2011). Combining RGB and ToF cameras for real-time 3D hand gesture interaction. IEEE Workshop on Applications of Computer Vision (WACV).
Van den Bergh and Van Gool (2011)

13) Hagan, M. T., Demuth, H. B., Beale, M. H., De Jesús O. (2014). Neural network design.
Hagan et al. (2014)

14) Vogl, T. P., Mangis, J.K., Rigler, A.K., Zink, W.T., Alkon, D.L. (1988). Accelerating the convergence of the backpropagation method. Biological Cybernetics, 59, pp.257–263
Vogl et al. (1988)

15) Markelj, P., Tomaževič, D., Likar, B., Pernuša, F. (2012). A review of 3D/2D registration methods for image-guided interventions. Medical image analysis, 16(3), pp.642-661.
markelj et al. (2012)

16) Pizzoli, M., Forster, C., Scaramuzza, D. (2014). REMODE: Probabilistic, monocular dense reconstruction in real time. IEEE International Conference on Robotics and Automation (ICRA).
Pizzoli, Forster, and Scaramuzza, (2014)

17) Duda, R. O., Hart, P. E., Stork, D. G. (1973). Pattern classification, Journal of Classification, 24, pp.305–307.
Duda, Hart, and Stork, (1973)

18) Karami, A., Zanj B., Kiani A. (2011). Persian sign language (PSL) recognition using wavelet transform and neural networks. Expert Systems with Applications, 38(3), pp.2661-2667.
Karimi, Zanj, and Kiani, (2011)

19) Verma, H. V., Aggarwal, E., Chandra, S. (2013). Gesture recognition using kinect for sign language translation. 2013 IEEE Second International Conference on. IEEE Image Information Processing (ICIIP).
Verma, Aggarwal, and Chandra (2013)

20) Yang, H.D. (2014). Sign language recognition with the kinect sensor based on conditional random fields. Sensors 15(1), pp.135-147.
Yang (2014)

21) Moghadam, M., Nahvi. M., Hassanzadeh, P.R. (2011). Static Persian Sign Language Recognition Using Kernel-Based Feature Extraction. 7th Iranian. IEEE Machine Vision and Image Processing (MVIP).
Moghadam, Nahvi, and Hassanzadeh (2011)

23) Agris, U, V., Zieren, J., Canzler, U., Bauer, B., Kraiss, K. (2008). Recent developments in visual sign language recognition. Universal Access in the Information Society, 6(4), pp.323-362.
Agris, et al. (2008)

24) Shah, N, K., Rathod, R, K., Agravat, J, S. (2014). A survey on Human Computer Interaction Mechanism Using Finger Tracking. International Journal of Computer Trends and Technology (IJCTT). 7(3), pp.174-177.
Shah, Rathod, and Agravat (2014)

25) Albrecht, I., Haber, J., Seidel, H. (2003). Construction and animation of anatomically based human hand models. *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation SCA '03*, pp.98-109
Albrecht, Haber, and Seidel (2003)

26) Caillette, F., Galata, A., Howard, T. (2008). Real-time 3-d human body tracking using learnt models of behavior. CVIU, 109(2), pp.112-125.

27) Francke, H., Solar J, R., Verschae, R. (2007). Real time hand gesture detection and recognition using boosted classifiers and active learning. PSIVT 2007: Advances in Image and Video Technology, 4872, pp.533-547.
Francke, Solar, and Verschae (2007)

28) Krotosky, S., Trivedi, M. (2006). Registration of Multimodal Stereo Images Using Disparity Voting from Correspondence Windows. IEEE International Conference on Video and Signal Based Surveillance. pp. 91-91.
Krotosky and Trivedi (2006)

29) Craig, J, J. (2009). Introduction to Robotics: Mechanics and Control (3rd Edition)