Neural Computing and Applications (2021) 33:4945–4963
https://doi.org/10.1007/s00521-020-05279-7

ORIGINAL ARTICLE

# A new framework for sign language alphabet hand posture recognition using geometrical features through artificial neural network (part 1)

Hoshang Kolivand[1,2] · Saba Joudaki[3] · Mohd Shahrizal Sunar[1] · David Tully[4]

Received: 8 January 2020 / Accepted: 5 August 2020 / Published online: 19 August 2020


**Abstract**
Hand pose tracking is essential in sign languages. An automatic recognition of performed hand signs facilitates a number of applications, especially for people with speech impairment to communication with normal people. This framework which is called ASLNN proposes a new hand posture recognition technique for the American sign language alphabet based on the neural network which works on the geometrical feature extraction of hands. A user's hand is captured by a three-dimensional depth-based sensor camera; consequently, the hand is segmented according to the depth analysis features. The proposed system is called depth-based geometrical sign language recognition as named DGSLR. The DGSLR adopted in easier hand segmentation approach, which is further used in segmentation applications. The proposed geometrical feature extraction framework improves the accuracy of recognition due to unchangeable features against hand orientation compared to discrete cosine transform and moment invariant. The findings of the iterations demonstrate the combination of the extracted features resulted to improved accuracy rates. Then, an artificial neural network is used to drive desired outcomes. ASLNN is proficient to hand posture recognition and provides accuracy up to 96.78% which will be discussed on the additional paper of this authors in this journal.

**Keywords** Sign language alphabet · Hand posture recognition · Depth-based geometrical sign language · Geometrical features sign language

## 1 Introduction

Hand posture recognition is applicable to many different domains. Even though various wearable instruments like gloves have been used recently, vision-based approaches are capable of capturing the actual hand postures without the need of a physical device. The recent mentioned methods make it possible for a more natural relationship between computers or other employed devices and users. Vision-based methods are increasingly becoming popular due to the considerable range of application fields, as reported in many recent surveys [1]. It should be noted that hand posture recognition through images and videos needs a complex process to obtain a high success rate. Inter-occlusions between the fingers of a hand are one of the existing issues which usually occurs in several poses. Colour bases recognition is problematic due to the low levels of contrast which reduces feature detection. This is prominent when multiple parts of a hand is very similar colours. For these causes, it is very challenging to

This work is prepared in two separated papers due to the complexity of prepared materials and needs of much discussion on the obtained results. Both papers are submitted at the same time in Neural Computing and Applications.

✉ Hoshang Kolivand
h.kolivand@ljmu.ac.uk

[1] MaGIC-X (Media and Games Innovation Centre of Excellence), Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia

[2] Department of Computer Science, Liverpool John Moores University, Liverpool L3 3AF, UK

[3] Department of Computer Engineering, Khorramabad Branch, Islamic Azad University, Khorramabad, Iran

[4] Scenegraph Studios, 4St Pauls Square, Liverpool L3 9SJ, UK

recognize sophisticated hand postures through the two-dimensional (2D) representation provided by image or video.

Hand posture is an attractive case for researchers because it is capable of replacing the computer mouse as an interface. This leads to allow a more natural interaction between the computer and human users. Hand posture is employed in the mobile devices, tablets, and also the newer wearable devices like the Google glasses. In healthcare applications, hand posture recognition is used in order to handle analysis, as well as medical operations. Data and surgical devices, sign language can be interpreted as another crucial application of the Kinect that actually allow disable people (deaf and dump) to communicate with computers.

Depth data include an extremely useful three-dimensional information, which can be used for posture recognition systems. This study has used depth data to obtain the reliable extract of the hand silhouette. This permits to apply many methods derived from depth-based hand postures and exploits an additional amount of useful information included in depth data. The proposed feature extraction methods show how several features based on the hand geometry in depth map represent the hand and finger postures and employed to appropriately recognize of the complex hand postures.

Changes due to different lighting conditions have a negative effect on the recognition tasks due to shadow or undesired effects on the objects (in our case hands) [2–4]. Furthermore, the recognition process is lead to lower accuracy in a cluttered background than a plain background [5]. Compared to the body or skeleton recognizing procedures, the recognition of the hand or another specific part of the body are more sensitive tasks. In these cases, the other objects in the scene can lead to occlusion of the specific body part, in our case, the hand, and consequently wrong detection and classification occurs. These issues have a negative impact on accuracy. In order to make a system that works in both simple and cluttered backgrounds, indoor or outdoor with different lightening conditions, a new approach is necessary to solve these issues.

The proposed research has one new approach in the segmentation step. Then, the geometrical features have been proposed in the feature extraction step which can be used in the robustness and reliable way for clarification process. Based on our findings, a geometrical feature can be very flexible against the rotation and even the change of the angle of the object. We even considered the vibration and hand tremors, so our method can be replaced with other method for measuring these variations.

These approaches can be used in sign language recognition systems for helping speech and hearing-impaired people. Educational purposes, improved user interface for deaf and dumb people, improved inter-signer telecommunications, and finally fast and reliable recognition are very important goals in this area. Suppose sign language could be used for educational tools with a simple depth sensor only for capturing images in speech therapy centres all over the world, so a deaf and deaf person can interact with none disabled people, faster and more accurate than the past.

## 2 Related work

There are 26 signs in the American Sign Language alphabet, but two signs ‘J’ and ‘Z’ have motion. In fact, these two signs belong to the dynamic sign category. This research is going to work on the static signs, but due to hand geometrical feature extraction, these signs can also be considered. This will be discussed in future work. As regards to this fact that the used geometrical features in this study are independent of the rotation or orientation of the hand, the motional signs do not make any problem in the recognition process. The labelling of signs according to their corresponding hand posture is a typical multiclass classification task, and here, there are 26 labels assigned. A number of machine learning patterns can be used in this case; support vector machine (SVM) [6], artificial neural networks (ANN) [7, 8], decision tree (DT), and random forest (RF) [9, 10], and convolutional neural network (CNN) [11]. There is clearly a robust idea that the multiclass tasks can avoid the over-fitting problem and is significantly efficient on large database. Therefore, the SVM classifier was selected as the machine learning algorithm in this research. The main purpose of SVMs is to carry out data correlation via nonlinear mapping. Kernel methods enable to operate in a high dimensional, implicit feature space without ever computing the coordinates of the data in that space, but rather by simply computing the inner products between the images of all pairs of data in the feature space. This operation is often computationally cheaper than the explicit computation of the coordinates [11]. The ANN design was not chosen due to the fact that the ANN design has failed to achieve a good result in this field. SVM not only can perform linear classifications, but also can perfectly conduct the nonlinear classification using a kernel mathematical property, by mapping the inputs into high-dimensional feature spaces [12].

The studies on sign language recognition were not limited in a specific language, so a real-time system was developed to detect the continuous Chinese sign language (CSL) [13]. The system used a 3D tracker and two cybergloves for collecting data. A dynamic programming method and Welch–Baum algorithm were applied for training sentences segmentation into principal units and re-evaluating. The system was tested on 220 words with 80 sentences with a recognition rate of 94.7% was achieved.

An algorithm for a wireless Bluetooth data glove was explained in a Bahasa Isyarat Malaysia (BIM) sign

language recognition system [14]. In this system, an electronic device was employed to translate the signs into speech. All of the signs are detected by the sensors which they are connected to a personal computer wirelessly via Bluetooth. They only used 25 popular words in BIM using hidden Markov models (HMM) techniques.

Paulraj et al. [15] proposed a simple technique for converting sign language into voice signals. A feature extraction method based on discrete cosines transform (DCT) on a binary image and from the video stream was presented. An artificial neural network using back propagation algorithm performed the recognition of postures process. The experimental results showed that the accuracy of the recognition is about 91%.

Assaleh et al. [16] proposed a user independent approach for recognizing Arabic sign language postures. In this method, the signers wear gloves to detect the signs. Two different classification techniques; polynomial networks and *K*-nearest neighbour, estimate feature extraction step. Similarly, Vatavu et al. [17] suggested a conditional random field (CRF) model in a glove-based system using reusable posture for interaction between computer and human.

A method for sign language recognition (SLR) system with a data glove and electromyography (EMG) sensors was proposed [18]. Electromyography signals are captured from hand muscles for a word in continuous SLR.

An Australian sign language recognition system using moment invariant was proposed by [19]. The system developed a database, including 10 images for each sign and used moment invariant method for feature extraction. The classification was done with a neural network. The experimental results showed that the presented approach do correctly classify for interpreting six postures and four postures have not successful recognition and it might misclassify 5–15% occasionally [19].

Traditionally in the "geometric moments analysis", moment invariants are computed based on the information provided by both the shape boundary and its interior region. The moments used to construct the moment invariants are defined in the continuous but for practical implementation they are computed in the discrete form. Therefore, it is adopted in this research. On the other side, the use of cosine rather than sine functions is critical for compression, since it turns out that fewer cosine functions are needed to approximate a typical signal, whereas for differential equations the cosines express a particular choice of boundary conditions. The proposed feature extraction methods show how several features based on hand geometry in depth-based images represent the hand and finger postures and employed to appropriately recognize of the complex hand postures.

In 2013, a system for dynamic hand posture recognition via video stream of eight different signers was proposed. Feature extraction was applied on the videos by skin colour detection algorithms. In this system, 20 different Arabic postures were investigated and 85.67% recognition rate was achieved. The presented system was able to reduce the error rate from 44 to 27.6%, but the recognition between similar postures is still a problem (Abdalla and Hemayed).

Shukla and Dwivedi [20] presented a technique for hand posture recognition by Microsoft Kinect. They used the fact that Kinect permits capturing depth, density, and also 3D scans of the objects. Some image processing methods were employed to find contour of segmented hand. Then, they classified the postures using Bayes classifier and obtained an exact classification rate of 100%, but their system could be calculated for five postures only. In addition, it was not able to recognize two hands in different rotations and orientations.

In hand posture recognition field, Barros proposed a hand posture recognition using multichannel convolutional neural network (MCNN) which allows the hand posture recognition with implicit feature extraction. This architecture used a cubic kernel to enhance the features for classifying and a multichannel flow of information to recognize images even with a small size on two datasets, the Jochen Triesch hand posture dataset (JTD) and a dataset using the video camera of a NAO robot. The multichannel architecture was used to specialize the tuning of the filters based on the Sobel operator, but it was not able to extract an optimal set of features. They obtained a 91% recognition rate in all images in all backgrounds and a score of 92% for the smaller images in the dataset and 94% for the original size images.

The systems that use gloves or interface hardware are very expensive and difficult for setting up, but they have reliable and simple computing process [3, 21]. In contrast, computer vision-based systems are not cumbersome and expensive for the user, but they are not completely reliable [22, 23]. In addition, they employed complicated calculation procedures in terms of the computational cost.

Changes due to different lighting conditions have a bad effect on the recognition tasks due to the shadow or undesired effects on the objects [2–4]. Furthermore, the recognition process is more difficult in a cluttered background than a plain background [5]. In comparison with the body or skeleton recognizing procedures, the recognition of the hand or another specific part of the body is more sensitive tasks. In these cases, the other objects in the scene can lead to occlusion, and consequently wrong detection procedure. These issues have an important impact on accuracy. In order to make a system that works in both simple and cluttered backgrounds, indoor or outdoor with

different lightening conditions, a new approach is necessary to solve these problems.

Most of the previous researches are dependent to the signer [2, 24]. The selected extracting features of the hand in these previous hand recognition systems depend on the position or direction of the signer's hand [25, 26]. Then, the recognition process is performed correctly just for a specific user and it does not work properly for a generic user. Using features independent of the user's hand shape, orientation, location, position, and direction is highly desirable. On the other hand, most of the previous research used finger tips as a feature [27]. The main weakness of the use of hand fingertips as the extracted features is that they can be occluded by the other fingers. There is natural variability in the executed signs because of the different position of the hand in the same signs. Furthermore, the observations are error-prone, and then a method other than the existing exact matching of features is needed without considering the finger's positions.

## 3 Methods

The aim of this research is to develop a technique to achieve more accurate and faster sign language recognition system for both plain and cluttered backgrounds with different users to help speech and hearing-impaired people in their life.

The methodology of this study is based on the depth-based images, and the geometrical features of the hand are presented. According to these properties, the research is called depth geometrical sign language recognition (DGSLR). Depth data include extremely useful three-dimensional information of the hand pose, which can be used for posture recognition systems. This study has used depth data to obtain the reliable extraction of the hand silhouette. This permits to apply many methods derived from depth-based hand postures and exploits an amount of useful information included in depth data. They used of this fact that Kinect camera permits capturing depth and 3D scans of the objects. Some image processing methods were employed to find the contour of the segmented hand. They classified the postures using Bayes classifier and obtained an exact classification rate of 100%, but their system could be calculated for five postures only. In addition, it was not able to recognize two hands in different rotations and orientations simultaneously.

The methodology presents several phases which cover the research entirely. After investigating several approaches, the challenge survey in the existing sign language recognition systems is followed. A new-fangled device is called Microsoft's Kinect.

Sensors are employed for collecting data. Three novice signers in sign language alphabet sit in front of the Kinect camera and accomplish the signs five times for each sign language alphabet letter. The segmentation process is well averse for separating the hand from the rest of the body. The results are completely compared with the level set segmentation technique. The segmentation process was completed by level set method (LSM). With the advent of level set methods, the changes of the object topological were automatically managed. The segmentation of object using deal with problems taking into consideration the applied images. Level set method is applied in cased with edge detection difficulty using active contour based. Firstly, the contours may be preoccupied if the image background contains clutter items. The second problem is light intensity that it is very difficult to find true boundaries of objects because of various lighting and it may be to blend the objects with the image background. This issue is usually solved if the texture of the object is considered as a feature for the contour to fit the object's boundaries. Furthermore, the LSM minimizes the energy function combined by diverse image features, for example texture, colour, shape, and boundary. The LSM has some advantages: it is parameter free, implicit, and capable of estimating geometric features of the evolving structure. It also has changeable topology. Furthermore, it is a very convenient framework to approach many applications of medical image and computer vision analysis. This research has used LSM in images which are fade due to light intensity in the segmentation step because of its ability in the detection of the image contour. Findings indicate that this approach supplies highly accurate segmentation upon signer images intended for diverse signs underneath various backgrounds. Occlusion means that some fingers or parts of the hand would be disappeared or be hidden by other parts, so the sign cannot be detected accurately. In order to resolve this issue, the LSM was applied and can detect the hand region correctly. The LSM does not consider any illumination condition. It uses defined points in segmenting process. So if the point set of a hand region would be defined, it can recognize that region accurately. In this case, it was applied to the signer's image for recognizing the missed parts of the hand because of the light, cluttered background or even hand angle. As described in the previous section, some parts of the hand may be missed due to illumination directions and the position of the hand. The hand could be segmented by definition of a set of the arbitrary points around the hand region.

Then, a new geometrical feature extraction method is presented for improving finger spelling recognition accuracy rate as image. The achievements of this phase cover the second research objective. The third objective is obtained in the next phase. In the last step, the support

vector machine method is used to classify the performed gestures, and the recognition rate is tested. The iteration process was well performed in 1 and 10 pass for all images in the dataset. Besides, the *k*-fold cross-validation was applied in the training set in the SVM method. In this research, the *K*-fold cross-validation is used in the training phase for obtaining the highest accuracy of results against the over-fitting problem due to the large size of the dataset. One important advantage of this method is that all observations are used for both validating and training processes, and each data are employed for testing exactly once. The validation step in this research is implemented by 5-fold and 10-fold cross-validation separately, and the results are compared. One of the advantages of this method over repeated random sub-sampling is that all observations are used for both training and validation, and each observation is used for validation exactly once. 10-Fold cross-validation is commonly used, but in general *K* remains an unfixed parameter.

Then, the estimation probability of the test instances was obtained. The last process of the training section was prediction procedure conducted by the confusion matrix.

### 3.1 Data collection

The selected dataset by the research which is called DGSLR dataset includes three users, one man, and two women. Two separate datasets are employed in this research. The first is the selected dataset by the research of DGSLR dataset, and the other is a standard dataset. In the DGSLR dataset, three novice users of sign language, one man and two women, were employed in this study. They were asked to sit down in front of the Kinect camera and perform signs. Each letter was repeated five times. All the images were captured in depth-based mode. The American sign language is performed by one hand against the British sign language which uses two hands for signing. The gestures were captured in both plain and cluttered background in depth-based applications by Kinect. The capturing process was performed in both plain and cluttered backgrounds in different variations of illumination. Objects in the cluttered background do not make any interference in the detection procedure. The farther objects are removed, and the closer objects are shown in the different level with the user in the foreground. Thus, the hand is still shown as different colour band in the RGB mode and brighter colour band in the depth mode. The hand is also recognizable in two modes. In order to validate data, a huge standard dataset from the Centre for vision, speech, and signal processing and University of Surrey is used. The images have been captured from nine people in different background similarly the research dataset.

Figure 1 shows some sample of the plain and cluttered environments for ordinary image.

The standard dataset including depth images by Centre for vision, speech and signal processing, University of Surrey, Guildford, UK is employed to validate the proposed approach [28]. The images have been captured from nine people in two very different environments and lit background similarly to the research dataset. The images gathered by the Kinect and only the depth base. In addition, there are too many repetitions, about 400, in each sign in different postures and directions. The users changed their hand direction and also distance to the Kinect sensor. The standard dataset includes more than 400 repetitions in each sign, which 70% of them are used for training and 30% for validating. There is no more standard dataset in this field which uses images by depth feature.

In the DGSLR dataset, three novice users of sign language, one man and two women, were employed in this study. They were asked to sit down in front of the Kinect camera and perform the signs. Each letter was repeated for five times. A sample of these signs is shown in Fig. 2.

Figure 3 shows some samples of this dataset for two signs ‘A’ and ‘B’. The image on the right bottom corner is the original image on this dataset, and the other are segmented images of this dataset by this research.

### 3.2 Depth base segmentation

The hand can be detected by a simple algorithm based on the conversion of the grayscale image to the binary image. In fact, hand segmentation is performed by a simple threshold of the depth data. This is our own innovative approach to extract the hand region. Considering this threshold could help us to detect the hand pixel only. This criterion leads to more accurate segmentation process in the next step of the process.

The hand can be separated from the other objects on the scene after applying this algorithm. In relation to this, the algorithm of the hand segmentation from a depth-based image is given in Algorithm 1.

**Algorithm 1: Segmenting the signer’s hand**

***Step 1****: Capture the signer’s image of Depth Basic Windows Presentation Foundation (WPF) application of the Kinect camera*

***Step 2****: Save the all images in the arbitrary repository*

***Step 3****: Read the depth image*

***Step 4:*** *Compute the binary image*

***Step 5****: Applying Otsu’s method to set the threshold on the image*

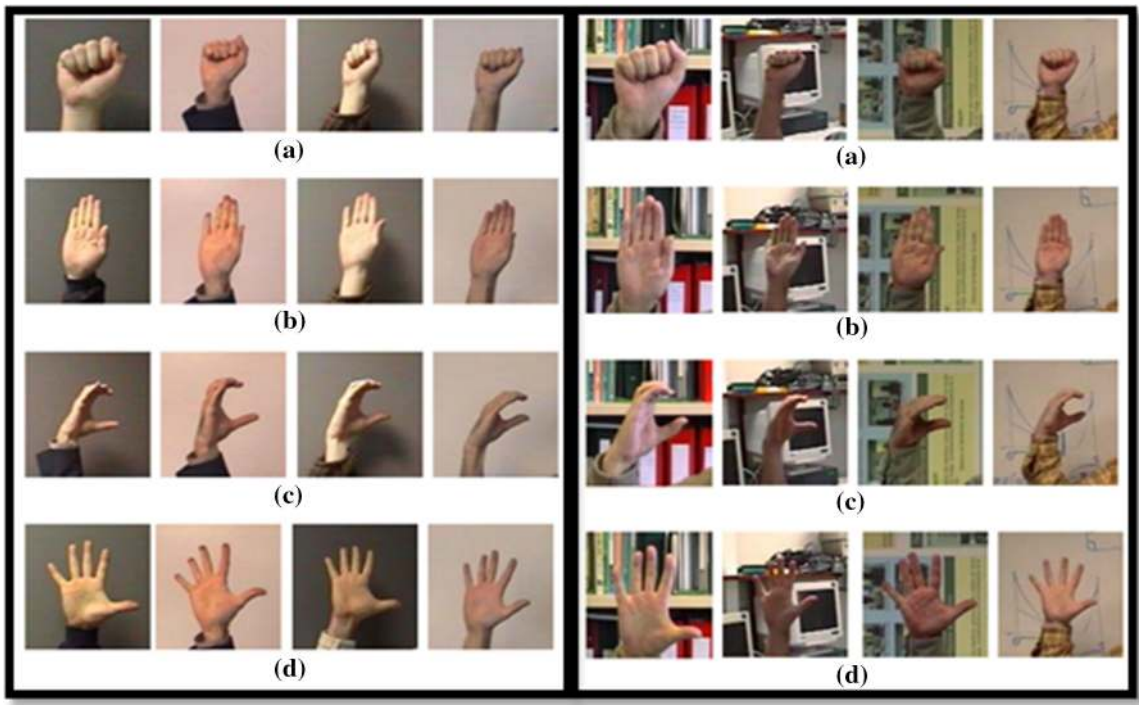


**Fig. 1** (Left) Plain background: **a** "A" letter, **b** "B" letter, **c** "C" letter and **d** "five" number. (Right) Clutter background: **a** "A" letter, **b** "B" letter, **c** "C" letter and **d** "five" number


**Fig. 2** Five repeats of each letter in DGSLR dataset. **a** Sign 'A' and **b** sign 'F'

After segmenting the hand, two major issues in some of the signs were appeared. Firstly, the forearm had been segmented with the hand, while the forearm region did not include any useful information for achieving the aim of this study. The other problem was occlusion between fingers. This is due to the hand position in some signs. In fact, the missing parts of the hand or fingers occur in some performed signs. This issue is also due to the high sensitivity of the Kinect to the hand shaking because of the illumination conditions. Consequently, two separate solutions were proposed for above issues in the two following steps.

In this way, the hand seems brighter than the other parts, and it leads to easier recognition. The distance between the user and the Kinect was 150 cm. In addition, the lighting conditions were adjustable during the signing process. However, the images were captured indoor and the illumination conditions were adjusted manually. Figure 4 shows different situations of the user's hand in different lighting environments. As it can be seen, the hand is recognizable in the all positions.

The presented hand detector is applied on the input depth images. This detector facilitates the segmentation problem with some assumptions. Firstly, it assumes the hand is the closest object to the Kinect camera. Secondly, the in-plane hand rotation is limited as far as it does not miss the meaning of sign. For example, for doing some of

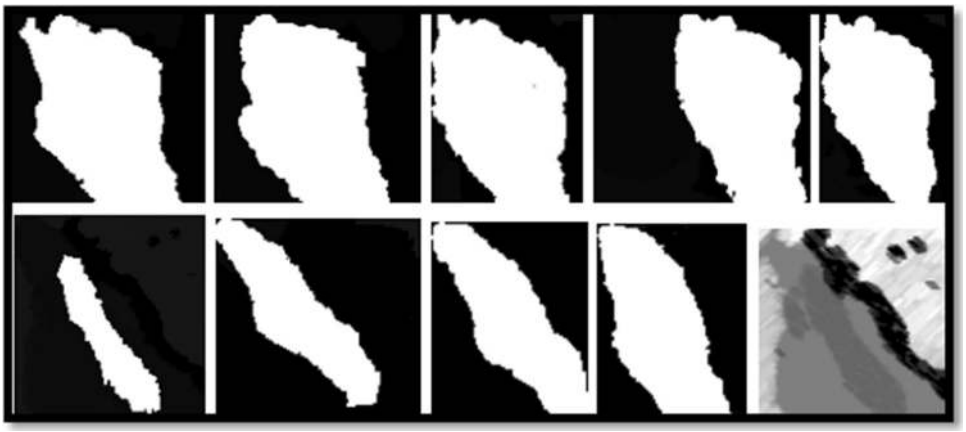

**Fig. 3** Standard dataset in 'A' and 'B' signs

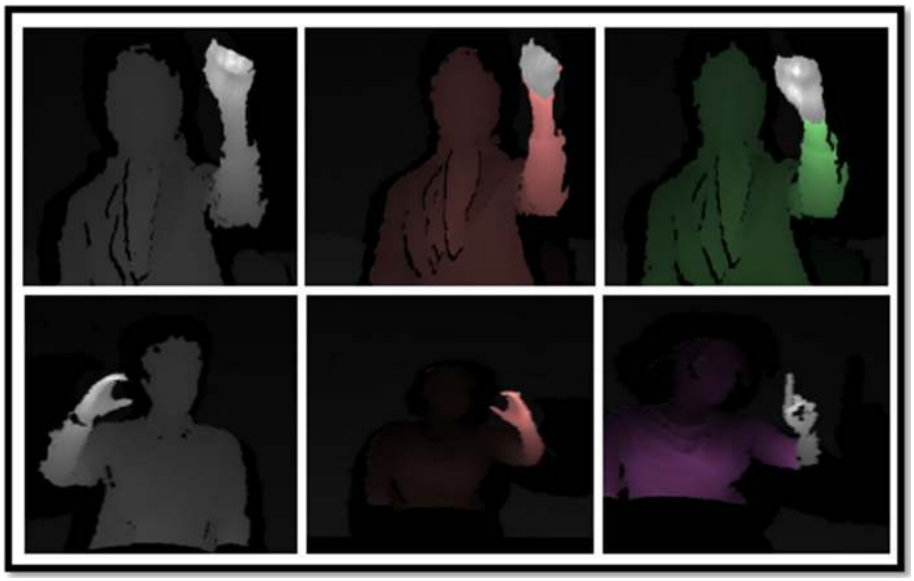

**Fig. 4** Different lighting conditions, hand brighter than the other parts of body

the signs like "B", the hand palm should be the opposite of the sensor. So, if the hand rotates more than a certain level, the meaning of the sign may be changed. In another example, if the hand rotates more than 90° while signing the 'D' letter, it seems like the 'G' sign which can be within (− 45°, 45°). Thirdly, the hand must be kept in such a way that the colour of the wrist, palm, and fingers seem brighter than the forearm. It assumes that the signers keep their wrist in the correct position farther than the forearm, so the depth values between the pixels within hand region exclude forearm region are confined within (100, 255). According to mentioned assumptions, the definition of a threshold in depth range pixels can be applied to separate the hand. Nevertheless, for robustness concerns, we apply a basic threshold based selection method. It is done by a definition of the depth threshold image to achieve the hand. Here, the hand consists of the pixels which satisfy this criteria:

$$\mathrm{d}h < \mathrm{d}T \tag{1}$$

where d$h$ is the depth value of the hand pixels, and d$T$ is the defined threshold depth value in the image. This ensures that only the hand region is extracted from the image. The first objective is met in this step. The sampling result of this algorithm for two different signers is shown in Fig. 5.

### 3.3 Forearm extraction

It is cumbersome for signers that were asked to keep their hand always in the correct position. As mentioned before, they have to keep their hand in front of their body. It causes the hand be the closest object to the sensor than the other objects. This may lead to the object be brighter than the other objects in the scene. Some signers in this research were novice in sign language and did not know how to sign proficiently, so it seems that it is necessary to find a way of removing the forearm. The first stage, the hand boundary is computed for obtaining more accurate points on the hand's edge. Then, the centre of the palm is computed by

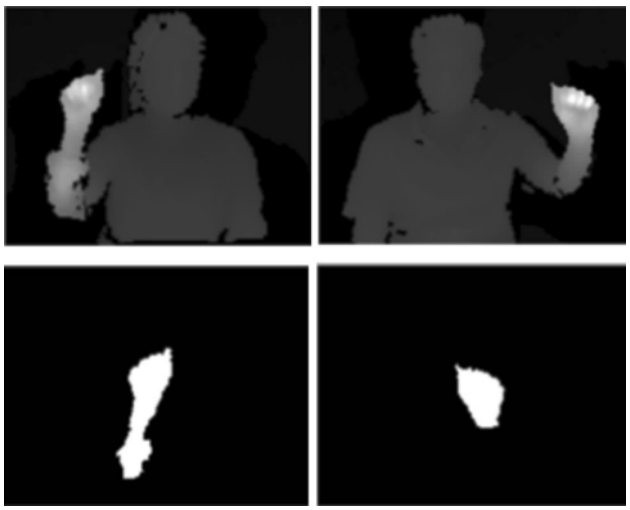

**Fig. 5** Segmented hands

morphological operations. The inscribed circle considering the centre point and edge point as point range is drawn. Based on Algorithm 2, forearm can be separated from the wrist of the hand.

**Algorithm 2 Separating the forearm of the hand**

***Step 1***: *Compute the hand contour*
***Step 2***: *Find the center of the hand palm*
***Step 3***: *Draw an inscribed circle in the hand palm with the determined center in the previous step*
***Step 4***: *Draw the longest diameter of the hand (D1); Length*
***Step 5***: *Draw the perpendicular line to the D1 line and tangent to the inscribed circle (D2); Width*
***Step 6***: *Crop the wrist by D2*

## 3.4 Level set segmentation

Occlusion means that some fingers or parts of the hand would be covered (not in view to the camera) or hide by other parts of the scene, so the sign cannot be detected accurately. In order to resolve this issue, the level set method (LSM) was applied and fortunately was observed that the LSM can detect the hand region correctly. This research intends to introduce a system for all users in every range of age. Some of these users may be children. They move their hands more than adults, and it is hard to hand detection process. Furthermore, it may lead to occlusion or changing in view. Since the above method (LSM) does not consider any illumination condition and uses only some defined points in the segmenting process, it can recognize the hand region properly. This is not obvious for other proposed methods.

The LSM does not consider any illumination condition. It uses only some defined points in segmenting process. So if the point set of a hand region would be defined, it can recognize that region properly. Algorithm 3 demonstrates the segmentation procedure.

**Algorithm 3 Occlusion problem by LSM**

***Step 1***: *Set value of the alpha parameter*
***Step 2***: *Set the image in a 2D double matrix*
***Step 3***: *Create a signed distance map from the mask (SDF)*
***Step 4***: *Loop is begun from 1 to an arbitrary iteration number*
***Step 5***: *Find interior and exterior mean*
***Step 6***: *Force of image information*
***Step 7***: *Force from curvature penalty*
***Step 8***: *Gradient descent to minimize energy*
***Step 9***: *Maintain the Courant–Friedrichs–Lewy condition*
***Step 10***: *Evolve the curve*
***Step 11***: *Keep SDF smooth*
***Step 12***: *Make mask from SDF*
***Step 13***: *Get mask from LS*

Occlusion between fingers is a common phenomenon during the sign performing in the sign language recognition system. The employed signers in this research were novice in sign language, so some mistakes happened due to the incorrect position of their hand. On the other side, they may shake the hand, so some parts of the hand do not appear in the images. The LSM method was used in this study for solving this problem due to the ability to correctly detect features in complex scenes. Both problems, occlusion and missed regions, are obviously visible in Fig. 6. The upper row is the result of the bad lighting in a specific position and the lower row shows occlusion. As it can be seen in Fig. 6, the level set can detect the signs correctly.

## 3.5 Post-processing

Depth maps can be affected by some noisy points, named 'Hole', because of the sensor sensitivity to the motion or illumination. These noisy points can affect the quality of the depth map and likewise on the accuracy in the next step. These noisy points are black points on the some parts of the image which have to be white. This issue occurs using existing various techniques like stereo correspondence, time of flight, or even structured light. Regardless of the methods, it is necessary to find holes in the depth image

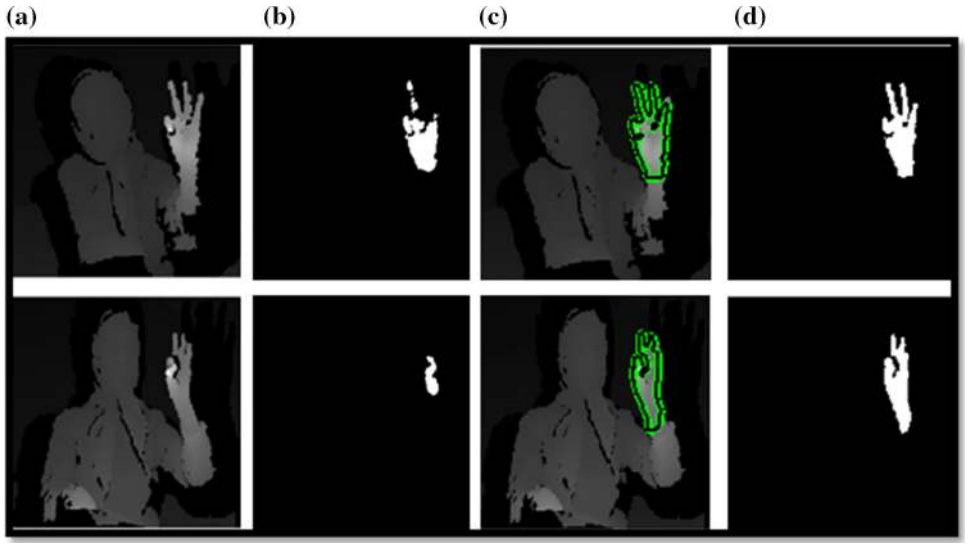


**Fig. 6** Kinect and LSM: **a** depth image, **b** Kinect segmentation, **c** LSM execution and **d** LSM segmentation

to avoid missing depth data. An example of this issue is presented in Fig. 7.

Many researchers have paid close attention to develop depth images in order to obtain high quality scenes in the human body recognition systems. One suggested option in the previous works is to use bilateral filtering in order to apply the hole-filling algorithms. For example, Chen et al. [30] proposed a combination of the colour data with bilateral filtering in the depth images to rectify edges. In brief, any type of edge rectification for 3D based camera images is very similar to the used methods in the hole-filling algorithms. Some edge improvement criteria between colour and motion-based images or the colour and depth-based images have been proposed in [30–32].

After segmenting the hand, these holes remain to be seen on the segmented hand. This is because of the hand movements and lighting conditions during the signing. Another reason to occur this problem is occlusion. 3D cameras need to multiple views of an object to acquire depth. When an object is used in just one view, or one camera is used for capturing, occlusion may occur. This ambiguity occurs because Kinect sensor uses a single viewpoint for obtaining depth data. As presented in Fig. 8, it is obviously clear that the 'P' point of the object has ambiguity in the view point place, because it seems as shadow in this point while in the camera view it is object not shadow. If there is one more camera in the viewpoint, this issue will be addressed.

Of course, post-processing procedure should be employed to improve these acquired depth images, but according to the review on the filtering algorithms in this study, the filtering process is a time-consuming process and in the some cases it causes to higher execution time. Furthermore, in our depth images, no need to rectify the edge and only some morphological object dilation operations are applied for smoothing the binary depth-based image and remove the noisy points on the hand surface.

### 3.6 Feature extraction

After segmenting the hand and removing its noise, it is time to extract the features. This study is going to extract the geometrical features of the hand due to having an unchangeable property against the changed direction or

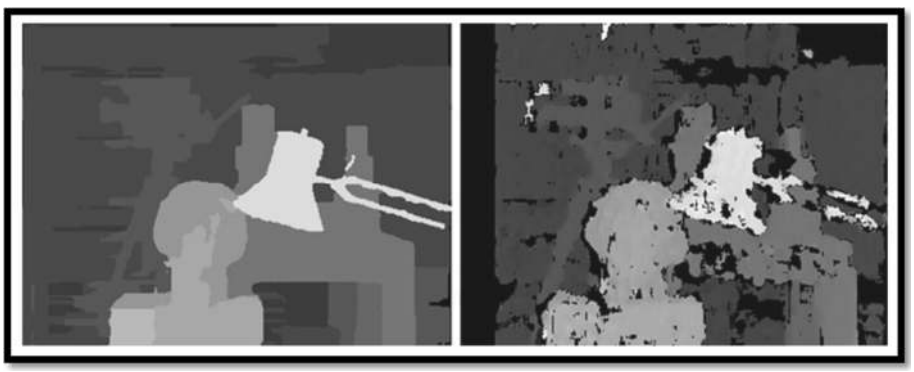

**Fig. 7** Noisy points in depth image [29]

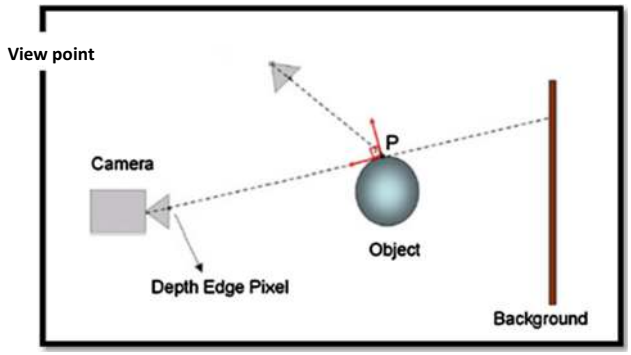


**Fig. 8** Ambiguity in the 'P' point

rotation. The second objective of the research is achieved at the end of this phase. The presented features are invariant against rotation or direction changes. They are also robust to misalignment or hand orientation variations. Nevertheless, unlike some of the hand features like fingerprints, hand geometrical features are expected to be more prone to deformations, because of free finger position.

### 3.6.1 Hand geometrical feature

Hand shape detection in sign language identification or verification systems is a well-known recognition manner [33–35]. Many hypotheses in this field are roughly based upon the hand geometries like finger lengths or widths and palm area. Likewise, another considerable hand recognition technique takes into account palm prints [36, 37] along with the contours of finger and palm. Anyway, the fusion of above approaches has a natural trend for hand recognition systems. This research is only concerned with shape-based features. However, in comparison with iris-based or fingerprint techniques, hand geometry has lower performance in accuracy, but it has been a significant upward trend. This can be due to its low cost, low data storage requirement, and unobtrusiveness [38].

### 3.6.2 Hand convex hull

The convex hull of the hand images was obtained in achieving the area and perimeter of the hand as the next feature descriptor. Before doing this process, the images have to be binary. The procedure for obtaining this feature is given by Algorithm 4.

**Algorithm 4 Convex hull of the hand**

***Step 1**: Read the depth-based segmented hand*

***Step 2**: Convert the image to a binary image*

**Algorithm 4 Convex hull of the hand**

***Step 3**: Generate the convex hull image from the binary image as follows:*

*Three arguments are necessary. The logical 2D binary image of the hand.*

*Method = objects: Compute the convex hull of each connected component of the*

*binary image individually.*

*Conn: Connectivity. It can have the following scalar values:*

***4**: Two-dimensional, four-connected neighborhood.*

***8**: Two-dimensional, eight-connected neighborhood.*

***Step 4**: Save the convex hull shape of the hand*

***Step 5**: Display a logical, convex hull image, containing the binary mask of the convex hull of all foreground objects in the image*

The area and perimeter of the convex hull shape were calculated as the next feature extraction item as shown in Algorithm 5. It is obviously clear that the area or perimeter if the hand convex hull is larger than the area and perimeter of a hand. They will be used for computing some ratios as described in the following sections.

**Algorithm 5 Convex hull area (CHA) and perimeter (CHP)**

***Step 1**: Read the convex hull shape of the hand*

***Step 2**: Measure a set of properties by morphological operators for each connected component (object) in the image. The image can have any dimension. In this case, there is just one object that it is the user's hand. Properties can be a list of strings, a cell array containing strings, the single string 'all', or the string 'basic'. If properties are the string 'all', it computes all the shape measurements on N-D inputs. Some of these measurements are: 'Area', 'Perimeter', 'BoundingBox', 'Centroid', 'FilledArea', 'FilledImage', 'Image', 'PixelIdxList', 'PixelList', and 'SubarrayIdx'. For executing in a shorter time, only two factors 'area' and 'perimeter' were used in this step*

***Step 3**: Save the area and perimeter of the convex hull*

For a nonempty points set in a certain plane, the convex hull is the smallest convex polygon which includes all these points in the set. For instance, in Fig. 9 the polygon around the points is a convex hull and the six points which are on the boundary are called "hull points".

### 3.6.3 Hand convexity defect

The convexity defects of the hand images were obtained as the difference between the convex hull and hand space. A

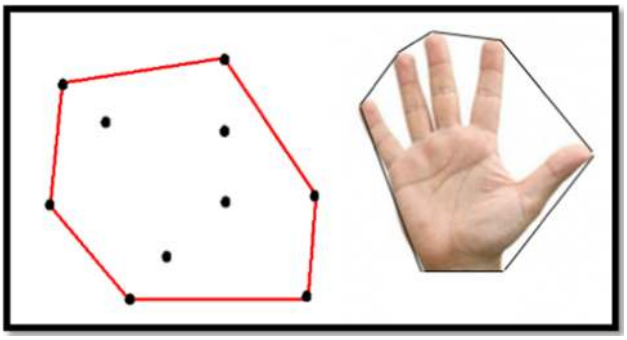

**Fig. 9** Convex hull of (left) a points set, (right) segmented hand

similar procedure to the hand convex hull is performed to achieve the convexity defects (Algorithm 6).

**Algorithm 6 Convexity defect of the hand**

***Step 1***: *Read the depth-based segmented hand*

***Step 2***: *Convert the image to a binary image*

***Step 3***: *Generate the convex hull image from the binary image as presented by Algorithm 3.4*

***Step 4***: *Save the convex hull shape of the hand*

***Step 5***: *Compute a logical difference (abs) between the hand convex hull image, containing the binary mask of the convex hull and hand*

***Step 6***: *Save the convexity defects as images*

The area of the hand convexity defect (CDA) was calculated by differences between the hand and its convex hull. If the convexity defects of the hand are considered as a single object, its perimeter is also computable. Furthermore, a number of the signs have too small a convexity defects and the perimeter of these signs is too large. Then, this research ignored of the convexity defect perimeter as another feature.

### 3.6.4 Hand geometry feature ratio

After getting the convex hull and convexity defects geometrical feature, another optimal feature is the ratio between them and the hand shape where it can be calculated by the division of the hand area and perimeter to the convex hull area and perimeter, respectively. Furthermore, the ratio between the hand area and convexity defect area is also considered as a optimal feature as represented in the following equations and discussed further.

$$\Re_{CHA} = \frac{Handarea(HA)}{ConvexHullarea(CHA)} \tag{2}$$

The ratio between the perimeter of the hand shape (HP) and the convex hull perimeter (CHP) is another useful parameter. Those gestures with closed fingers typically related to perimeter less than when some fingers are open. Likewise, the rate of hand perimeter to the convex hull is close to 1. The following Equation shows this relationship.

$$\Re_{CHP} = \frac{Handperimeter(HP)}{ConvexHullperimeter(CHP)} \tag{3}$$

Similarly to the convex hull, the rate of hand geometry area (HA) to the convexity defect area can be considered as an informative feature for reliable recognition system. This rate has been calculated by:

$$\Re_{CDA} = \frac{Handarea(HA)}{Convexitydefectarea(CDA)} \tag{4}$$

### 3.6.5 Hand distance feature

This study is going to compute the height and width of the hand as the next feature descriptor. It is achieved by computing the eigenvalue and the eigenvector concepts considering the hand boundary. Algorithm 7 shows the process.

**Algorithm 7 Hand height and width**

***Step 1***: *Read the image*

***Step 2***: *Get the hand contour*

***Step 3***: *Calculate the gravity point as X, Y*

***Step 4***: *Calculate the Covariance matrix*

***Step 5***: *Calculate the Eigenvalue and Eigenvectors of the matrix*

***Step 6***: *Length of diameter is obtained by the Eigenvector and Eigenvalue*

All the common hand boundary detection methods detect the edge as a thick layer around the object. This layer may consist of two or more pixels in each row and column in a specific point. To get a more accurate result, it is better to improve these methods as a thinner layer in the object boundary. It is presented in Algorithm 8.

**Algorithm 8 Hand contour**

***Step 1***: *Read the image*

***Step 2***: *Resize the image to 128 by 128 matrix*

***Step 3***: *Convert image to the binary image*

***Step 4***: *Apply the morphological operation which removes the pixels on the hand boundaries until does not occur to break apart of the hand. The remaining hand shape is the skeleton of the hand.*

**Algorithm 8 Hand contour**

***Step 5***: *Assign size of rows (r) and columns (c) to two desired variables*
***Step 6***: *Assign a counter (m) with Initial value 1*
***Step 7***: *Loop x from 1 to row size (r)*
***Step 8***: *Loop y from 1 to column size (c)*
***Step 9***: *If image value equal to 1 begin*
***Step 10***: *Assign the coordinate of this point to a matrix*
***Step 11***: *Plus m to 1*
***Step 12***: *End of Loop*
***Step 13***: *End of Loop*
***Step 14***: *Plot the matrix*

Firstly, the hand boundary should be calculated. There are some predefined functions which can be applied on the images for detecting the edges of the objects. The Matlab software also includes several algorithms for calculating the object's boundary, but the detected edges in these algorithms are very thick and include a number row of pixels as shown in Fig. 10.

As mentioned above, this research was implemented by Matlab software. We can capture the video stream and convert it to some frame which each frame represents a picture, but it is a time-consuming process. To use just image, it would be enough to directly use related built-in library of Matlab software. It is applicable in image processing, but for using video processing, OpenCV library can be helpful. It has C++, Python, Java, and MATLAB interfaces and supports Windows, Linux, Android, and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. Image processing is a multidisciplinary field, with contributions from different branches of science including mathematics, physics, optical, and electrical engineering. Moreover, it overlaps with other areas such as pattern recognition, machine learning, artificial intelligence, and human vision research. Different steps involved in image processing include importing the image with an optical scanner or from a digital camera, analysing and manipulating the image (data compression, image enhancement and filtering), and generating the desired output image. Matlab supports a specific toolbox for image processing in advanced.

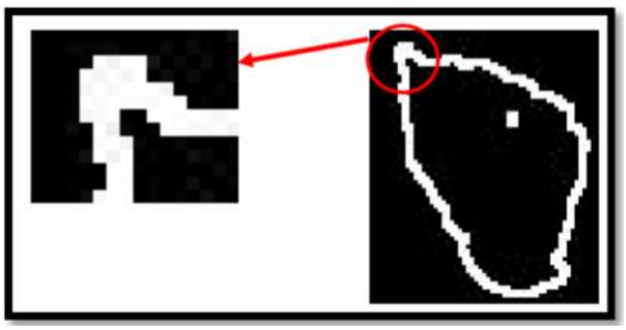

**Fig. 10** Thick edge includes several points

Since some of the performed signs are too similar to each other and their geometrical features are also close together, it is very important to gain an accurate edge of the hand to achieve the higher accuracy height and width of the hand. So, the proposed algorithm has been applied on the edge detector algorithm to reach this goal. Figure 11 shows the instance of the mentioned applied method.

### 3.6.6 Feature vector

This is the time of the feature vector making process. There are nine features, hand area (HA), hand perimeter (HP), convex hull area (CHA), convex hull perimeter (CHP), convexity defect area (CDA), convex hull area ratio ($R_{CHA}$), convex hull perimeter ratio ($R_{CHP}$), convexity defect area ratio ($R_{CDA}$), and distance ($D$), which are saved in an CSV file for easier access. In addition, a label is assigned to each letter. Then, there are 26 labels totally in each repeat. These labels are also considered as a feature. Then, there are ten features in the feature vector as seen in Algorithm 9.

**Algorithm 9 Feature vector construction**

***Step 1***: *Read the images one by one and save to the desired 'dir'*
***Step 2***: *Begin Loop*
***Step 3***: *Read binary image*
***Step 4***: *Call the feature calculating functions one by one*
***Step 5***: *Save the output of the functions on a feature vector*
***Step 6***: *End of Loop*

## 3.7 Classification

Likewise, all of the studies on artificial neural network (ANN) have shown that it has a robust learning capability. There are varieties of ANN systems used in hand posture recognition systems. The ANN is one of the most popular modelling methods, and MLP is one of the most commonly applied ANN methods. The ANN Toolbox in MATLAB was used to design the MLP network [39]. Alternatively, support vector machine (SVM) approaches have effective results on recognition systems [40].

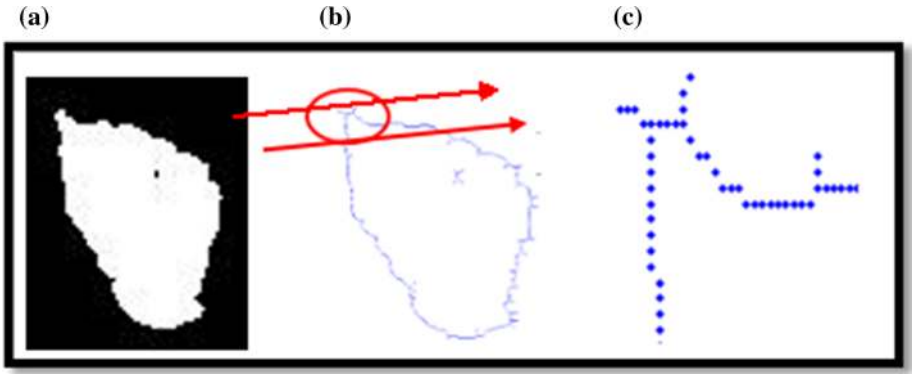


**Fig. 11** Image edge detection algorithm: **a** original image, **b** detected contour and **c** more detailed view

The last step of the proposed recognition system includes an appropriate machine learning method to classify the extracted features in the previous step in order to recognize hand gestures. There are many approaches based on neural networks, fuzzy logic, decision tree, support vector machine, and other methods for gesture classification which have been mentioned in detail in the literature review section. In this research, two own datasets of depth-based images of a sign language alphabet is presented. Firstly, the artificial neural network classification process is applied on the DGSLR dataset, so the training set contains data from three available users. A method of *K*-fold cross-validation is used with *K* equal to 5 and 10 in the testing step. In the *K*-Fold validation method, the collected data are partitioned into *K* subsets. In these subsets, one is used for validating the data and $K - 1$ subsets for the training process. This procedure is repeated *K* times. All data are used exactly once for training and once for testing. Finally, the average of these *K* procedures is selected as the final estimation. While it can be possible to use other methods for combining the results, the 10-fold cross-validation is commonly applied on this data [41]. One important advantage of this method is that all observations are used for both the validating and training processes, and each data set is employed for testing exactly once. The validation step in this research is implemented by 5-fold and 10-fold cross-validation separately, and the results are compared together. The two parameters of *C* and $\varphi$ of the RBF kernel are subdivided with a regular grid which *C* is considered which equals to 1, 10, 100, and 1000, and parameter φ equals to 0.001, 0.01, 0.1, 1. Similarly to other classifiers, for each of these parameters, the training collection is divided into two categories, 70% of the data is established for training and the 30% for testing sets. The performance is assessed according to testing iteration process. The iteration process was done between 10 and 1000 times, and the best result was obtained in the 700th rank. The experimental results were computed for extracted features lonely and also the combination of them. In addition, the program was repeated for 5-fold and 10-fold cross-validation. Finally, the parameter combination which gives the highest accuracy is selected. In this research, based on the above explanation, a multiclass classifier versus a SVM classifier has been used, and accordance with a set of $n(n - 1)/2$ binary SVM classifiers used to test each gesture against each other. Each output is selected as a vote for a certain gesture and as mentioned before the gesture with the maximum votes is the recognition process result. This study uses a nonlinear SVM because there are different kernel functions in the nonlinear SVM structure. Computing the kernels needs minimal effort and basic processing, but calculating the feature vector is processor intensive. Choosing a kernel based on the prior knowledge of invariances as suggested by Cawley and Talbot [42] is an excellent idea which aims to reduce the heavy usage of calculating the feature vector.

In machine learning, the radial basis function (RBF) kernel is a popular kernel function used in various kernelized learning algorithms. In particular, it is commonly used in support vector machine classification. It has two important advantages: simple design and high tolerance to input noise. It should be noted that the other kernels could be used in SVM.

The Gaussian radial basis function (GRBF) kernel is one of the most common kernel used in this type of research as obtained by Eq. (5).

$$k(x_i, x_j) = \exp\left(-\gamma \left\| x_i - x_j \right\|^2\right) \quad \text{for } \gamma > 0 \tag{5}$$

This study focuses on the classification by SVM because of its clarity and simplicity in the classification. Furthermore, its usability to resolve the various problems is one of another reason to use it, as some approaches like decision trees are not simplicity used in the various problems. SVM causes to get the good generalization on a big dataset. Since a big data set requires to a complicated model and the full Bayesian framework is very costly in computation. In contrast, the SVM is faster and still has an optimal generalization solution. Furthermore, due to very big set of nonlinear task-independent features, SVM has a clever way

to prevent the over-fitting problem. Here, the applied algorithms in this step will be explained. Firstly, a general schema of the SVM Algorithm is presented and then discussed its details. The classification task was computed by the SVM implementation provided by the LIBSVM library. We utilized classic grid search selected random feature vectors sub-samplings to split the datasets in the training set and test sets. In contrast to the classic grid search, this study used a leave-one-out method. Figure 12 demonstrates the leave-one-out method used for the dataset.

LIBSVM is a popular open source machine learning library. LIBSVM implements the sequential minimal optimization (SMO) algorithm for SVMs with kernel functions, regression, and supporting classification [43]. SMO is used to solve the quadratic programming (QP) issue arising during the SVM training phase.

**Algorithm 10 SVM Implementation**

***Note:*** *Algorithm repeats for 1 and 10 iterations*
*Algorithm repeats for 5-fold and 10-fold cross validation*

***Step 1****: Assign the size of column and row to arbitrary variables in TrainData set*

***Step 2****: Assign labels 1, 2, 3, …, 26 to them*

***Step 3:*** *Compute zscore of TrainData set*

***Step 4****: Do Normalization*

***Step 5****: Save Size and Maximum label*

***Step 6****: Call K-Fold cross validation Function*

***Step 7****: Compute zscore of TestData set*

***Step 8****: Do Normalization*

**Algorithm 10 SVM Implementation**

***Step 9****: Predict the class with the highest probability*

***Step 10****: Compute the mean accuracy of TrainData set and TestData set*

The extracted features have to be divided for training and testing procedures. In the desired DGSLR dataset in this study, there are five repeats for each signs, so four of them are considered in train set and one for validating. In the standard dataset, there are 400 repeats in each signs on average, so 280 signs are used in the training phase and 120 signs for the testing phase for each letter (sign). Algorithm 11 shows this division.

**Algorithm 11 Feature dividing**

***Step 1****: Loop of number of signs*

***Step 2****: Read features from feature vector according to their numbers*

***Step 3****: Assign 70% of them to the TrainData set and 30% to the TestData set*

***Step 4****: End of Loop*

***Step 5****: Loop of number of signs*

***Step 6****: Loop of number of iteration in each sign*

***Step 7****: Read features and put in the TrainData set*

***Step 8****: End of Loop*

***Step 9****: End of Loop*

***Step 10****: Loop of number of iteration in each sign*

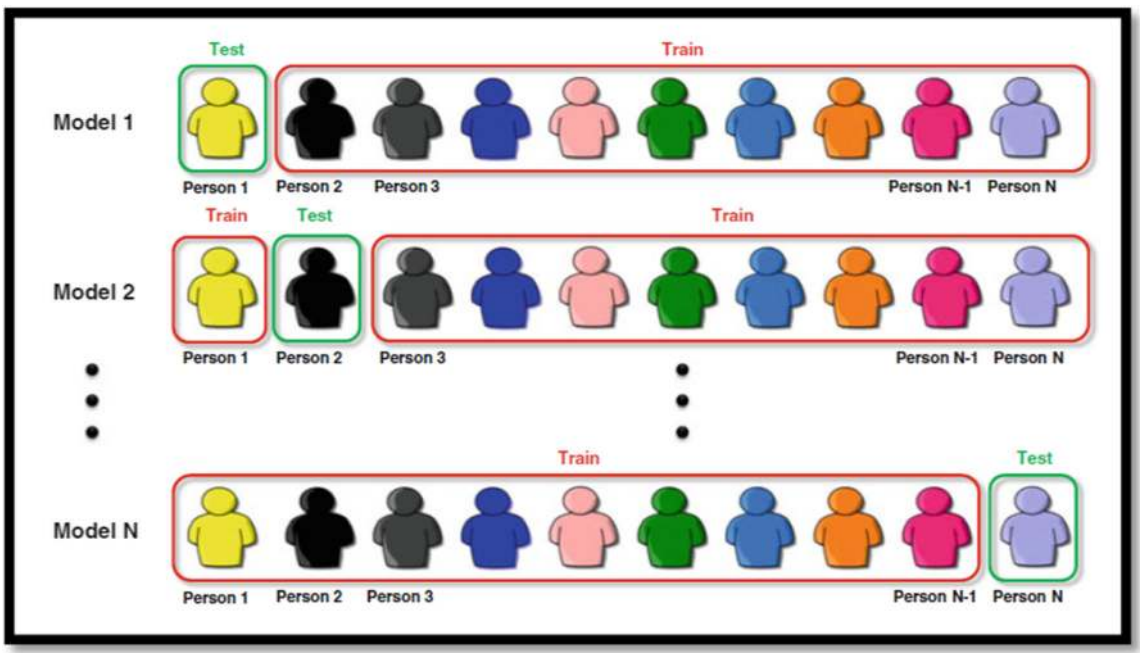


**Fig. 12** Exemplification of leave-one person-out method in SVM [40]

**Algorithm 11 Feature dividing**

***Step 11***: *Loop of number of signs*

***Step 12***: *Assign the features to the FM structure*

***Step 13***: *End of Loop*

***Step 14***: *End of Loop*

***Step 15***: *Assign 70 percentage of the FM size to Train Set*

***Step 16***: *Assign row number 1 to Train of FM into the FM_TrainData*

***Step 17***: *Assign row number Train + 1 to end of FM into the FM_TestData*

#### 3.7.1 Multiclass SVM

The proposed feature extraction techniques in the previous section provided different feature descriptors relevant to measurable properties of the hand. So, there are multi-features and they have to feed into a multi-classifier method. Multiclass SVM assigns labels to samples, where the labels are from a finite set of several elements. The applied approach is to reduce the single multiclass issue into multiple binary classification issues [44]. Some methods for this aim have been discussed in [44, 45]. A binary classifier can distinguish between two categories: the first category is one of the labels versus the rest, and the second type is one versus one (every pair of classes). Classification of the first type is conducted by a winner-takes-all method. For the one-versus-one approach, classification is conducted by a max-wins voting method, that each classifier assigns the sample to one of the two classes, then the vote is increased one by one vote. At last the class with the most votes determines the sample classification. Directed acyclic graph SVM [46] and error-correcting output codes [47] are multiclass SVM samples. Singer and Crammer proposed a multiclass SVM method which distributes the multiclass problem into a one optimization problem, rather than using one of the multiple binary classification problems [48].

The proposed approach is used to perform the recognition process in multiple feature descriptors. The proposed solution is to combine the different features into a vector; '*F*'. It is a combination of the hand geometrical features, convex hull and convexity defects, distance, and ratio between them as $F$ = [FH, FCH, FCD, FD, FR] which it can be fed to the multiclass SVM classifier [40].

The multiclass SVM classifier was applied on the standard dataset and the obtained results are as follows. The employed standard dataset includes more than 10,000 set of depth-based images of nine users in approximately 400 repetitions on each sign, roughly equal to 10,400 images for each user. Within this work, just one user has been considered. Seen in Table 1, the most value of the recognition accuracy rate is related to the convex hull with 58.99% in the training phase and 59.65% in testing phase. The second valued characteristic is related to the ratio between convex hull and hand. These results are similar to the DGSLR dataset results. Table 1 shows the extracted features combination where the highest value of accuracy rate belongs to a combination of three features: distance, hand convexity, and hand convex hull.

As Table 2 shows, the recognition rate of the classifier is more than 90% on DGSLR dataset and 96% on the standard dataset which has a significant differences compared to the previous research.

### 3.8 Evaluation and validation

Obtaining the accuracy rate of the recognition process is the end point of this. It can be interpreted by confusion matrixes and statistical charts. Furthermore, two statistical parameters, sensitivity and specificity, represented the classifier performance considering the confusion matrix. The sensitivity parameter or true positive rate measures the proportion of actual positive samples which are correctly identified. It is also complementary to the false negative rate. The specificity parameter or true negative rate measures the proportion of negative samples that are correctly identified. In fact, sensitivity determines the avoiding of false negatives, while specificity performs for false positives. A perfect predictor in the classifier would be defined as 100% sensitive and 100% specific, but theoretically any predictor has a minimum error rate which known is called the Bayes error rate [57, 58]. At the end of this step, it is hopefully to get the higher accuracy recognition rate than the previous works and benchmark.

#### 3.8.1 Over-fitting

In statistic science, over-fitting happens when a statistical model illustrated a noise or random error rather than the basic relationship. Generally, when a mathematical model is strongly complex, the over-fitting occurs, for example, having a lot of parameters in accordance with the number of observations. This over-fitted model leads to the poor prediction. It can lead to overstate in minor fluctuations detection or vice versa in the data analysis. Over-fitting depends not only on the number of data and parameters, but also the adaptation between the model structure and the data shape, as well as the value of model error compared to the expected error or noise [59]. As shown in Fig. 13, the noisy data in both linear and polynomial functions are fitted. It can be seen that the data points on the polynomial function have an excellent fit, but the linear function misses

**Table 1** Performance of combination of extracted features from the standard dataset

| Type of features | Train accuracy (%) | Test accuracy (%) |
|---|---|---|
| HA + HP + CHA | 85.50 | 85.78 |
| HA + HP + CHA + CHP | 85.71 | 86.69 |
| HA + HP + CHA + CHP + CDA | 85.43 | 86.88 |
| HA + HP + CHA + CHP + CDA + RCHA | 89.43 | 90.24 |
| HA + HP + CHA + CHP + CDA + RCHA + RCHP | 89.85 | 89.31 |
| HA + HP + CHA + CHP + CDA + RCHA + RCHP + RCDA | 92.50 | 93.43 |
| RCHA + RCHP + RCDA | 89.39 | 89.77 |
| D + HA + HP + CHA + CHP | 93.64 | 96.85 |
| D + HA + HP + CDA | 87.32 | 89.14 |
| D + RCHA + RCHP + RCDA | 89.71 | 91.54 |

**Table 2** Recognition accuracy comparison

| Methods | Accuracy |
|---|---|
| Recurrent neural network [49] | 0.425 |
| Dynamic temporal warping [50] | 0.540 |
| Hidden Markov model [51] | 0.900 |
| Action graph on bag of 3D points [52] | 0.847 |
| Histogram of 3D joints [53] | 0.789 |
| Random occupancy pattern [54] | 0.862 |
| Eigen joints [55] | 0.823 |
| Sequence of most informative joints [56] | 0.471 |
| Proposed method on DGSLR dataset | 0.903 |
| Proposed method on standard dataset | 0.968 |

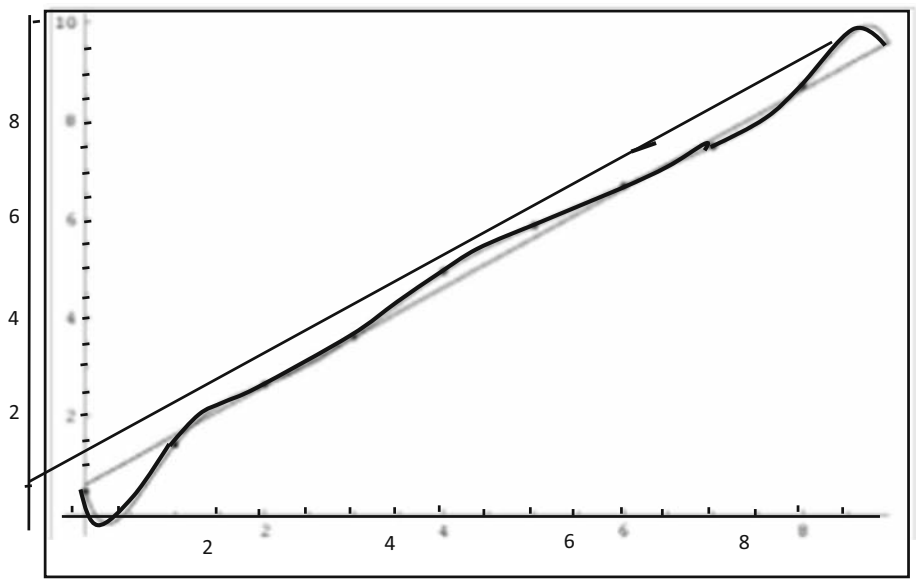


**Fig. 13** Over-fitting problem on two different functions

some of the data points. Although the linear version can be better, and if the regression curves were applied to extrapolate the data, the over-fitting problem would do worse.

Over-fitting is an important concept in machine learning. A learning algorithm is trained using some training data. Then, it is expected for the learner to be able to predict the correct data for other examples which not

presented during the training process. So, in too long learning process or in the infrequent training examples, the learner has to select the specific random features of the training data. These data may have no causal relation to the target function. In this type of over-fitting, the performance on the training data examples increases, while the performance on unknown data examples becomes worse. In Fig. 14, testing error has been shown in red and training error in blue. When the testing error increases in positive slope, the training error continuously decreases in a negative slope. In this case, it can be said the over-fitting has occurred.

In summary, over-fitting usually occurs when the model is too complex regarding to the training data size. It can be said:

- If the data are in two-dimensional space, the model is a line, and there are 10,000 points in the training set, it is *under*-fit.
- If the data are in two-dimensional space, the model is 100-degree polynomial, there are 10 points in the training set, it is over-fit.

### 3.8.2 LIBSVM

LIBSVM is a famous open source machine learning library, which developed at the National Taiwan University. It has been written in C++ language with the C application programming interface. LIBSVM implements the sequential minimal optimization (SMO) algorithm for support vector machines (SVMs) with kernel functions, regression, and supporting classification [43]. SMO is a useful algorithm that released in order to solve the quadratic programming (QP) issue arise during the SVM training phase. SMO was proposed in 1998 and widely used for training SVMs by implementing the LIBSVM tool [43, 60]. The broadcasting of the SMO algorithm led to a great revolution in the SVM world, because the previous methods for SVM training phase had too much complexity. Besides, they need expensive third-party QP solvers [61]. QP is a special type of mathematical optimization problem. It defines as minimizing or maximizing way of a quadratic function of variables under the linear restrictions on these variables. The other open source machine learning toolkits are also used the SVM learning code in the LIBSVM library. Some of these toolkits are KNIME, GATE, Orange, and scikit-learn [62] which can be implemented in several existing programming languages such as MATLAB, Java, and R.

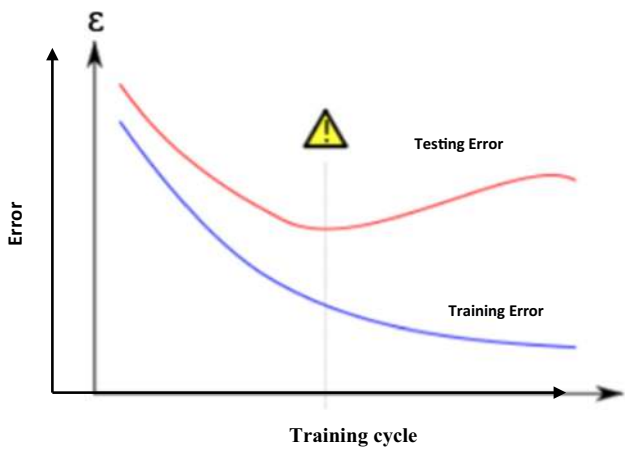


**Fig. 14** Over-fitting/overtraining in supervised learning

As mentioned before, in this study the multiclass SVM has been used, so LIBSVM opens a way to the goal of the research. It implements the one-versus-one technique for multiclass classification [63]. The following solution is assumed to solve the multiclass problem. The $K$ parameter is the number of classes, so $k(k-1)/2$ classifiers are considered and each classifier trains data from two classes. Equation (6) is applied in two-class classifier which can be used for training data process from $i$th and $j$th class.

$$\min_{w^{ij},b^{ij},\xi^{ij}} \frac{1}{2}\left(w^{ij}\right)^T w^{ij} + c\sum_t \left(\xi^{ij}\right)_t \tag{6}$$

$$\text{Subject to } \left(w^{ij}\right)^T \phi(x_t) + b^{ij} \geq 1 - \xi_t^{ij}, \text{ if } x_t \text{ is in } i\text{th class}, \ \xi_t^{ij} \geq 0$$
$$\left(w^{ij}\right)^T \phi(x_t) + b^{ij} \geq -1 + \xi_t^{ij}, \text{ if } x_t \text{ is in } j\text{th class}, \ \xi_t^{ij} \geq 0$$

The voting strategy is performed by each binary classification, which the votes can be distributed for all data points $x$. Each point is designated to one of the two classes. Finally, a class with the maximum number of votes is selected for determining the sample classification. There are many other approaches for multiclass SVM classification as a comprehensive comparison has been discussed in detail in [45].

## 4 Conclusion

In this paper, we have discussed in detail about the proposed framework called ASLNN. The presented methodology intends to reach the considered aim to enhance sign language recognition systems in American sign language. ASLNN consists of four phases. The first phase is data collecting and introduces standard dataset for validating data in the final phase. Phase two is assigned to the segmentation process and discusses appeared issues and their proposed solutions. Before going to phase three, there is a short way to post-processing procedure in order to prepare the samples for next phase. Phase three is the feature extraction step which some geometrical features of the extracted hand in depth are computed and new Algorithms

for extracting the features are proposed. The last phase is concerned with multiclass classification and testing in the SVM classifier. This phase incorporates testing, evaluating, and validation of the possible results according to the standard dataset. Discussion about the result and more details on validation and evaluation of the proposed method will be presented in another paper submitted to this journal.

The following suggestions are open research directions discussed by this work. First, the proposed sign language recognition framework relies on empirically distance between the user and camera which should be set manually to segment the hand. The camera is sensitive to ambient light which this factor affects distance calculation. It would naturally be desirable to have an automatic system which adjusts itself as a function for setting the resolution and distance. Furthermore, the combination of the extracted features can be further investigated and improved, or even more geometrical features can be extracted. In addition, depth-based grayscale images were only considered in this research which it can be extended to a combination of colour and depth to further enhance the recognition. It would also be an interesting idea to complement the proposed approach with common acts in sign language. It requires recognizing the dynamic signs which are named hand gestures.

## Compliance with ethical standards

**Conflict of interest** All authors declare that they have no conflict of interest.



## References

1. Garg P, Aggarwal N, Sofat S (2009) Vision based hand gesture recognition. World Acad Sci Eng Technol 49:972–977
2. Chai X, Li G, Lin Y, Xu Z, Tang Y, Chen X, Zhou M (2013). Sign language recognition and translation with Kinect
3. Kishore P, Kumar PR (2012) Segment, track, extract, recognize and convert sign language videos to voice/text. Int J. https://doi.org/10.14569/IJACSA.2012.030608
4. Zhu Q-S, Xie Y-Q, Wang L (2010) Video object segmentation by fusion of spatio-temporal information based on Gaussian mixture model. Bull Adv Technol Res 5:38–43
5. Prasad MVD, Raghava PC, Rahul R (2015) 4-Camera model for sign language recognition using elliptical fourier descriptors and ANN. SPACES-2015, Department of ECE, K L University
6. Elias I, Rubio JDJ, Cruz DR, Ochoa G, Novoa JF, Martinez DI, Juarez CF (2020) Hessian with Mini-batches for electrical demand prediction. Appl Sci 10(6):2036
7. De Jesús Rubio J (2009) SOFMLS: online self-organizing fuzzy modified least-squares network. IEEE Trans Fuzzy Syst 17(6):1296–1309
8. Aquino G, Rubio JDJ, Pacheco J, Gutierrez GJ, Ochoa G, Balcazar R, Zacarias A (2020) Novel nonlinear hypothesis for the delta parallel robot modeling. IEEE Access 8:46324–46334
9. Dong C, Leu M, 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
10. Kuznetsova A, Leal-Taixé L, Rosenhahn B (2013) Real-time sign language recognition using a consumer depth camera. In: 2013 IEEE international conference on computer vision workshops (ICCVW). IEEE, pp 83–90
11. Shamshirband S, Rabczuk T, Chau KW (2019) A survey of deep learning techniques: application in wind and solar energy resources. IEEE Access 7(1):164650–164666
12. Rautaray SS, Agrawal A (2015) Vision based hand gesture recognition for human computer interaction: a survey. Artif Intell Rev 43:1–54
13. Chunli W, Wen G, Jiyong M (2002) A real-time large vocabulary recognition system for Chinese sign language. In: Gesture and sign language in human–computer interaction. Springer, pp 86–95
14. Swee TT, Salleh S-H, Ariff A, Ting C-M, Seng SK, Huat LS (2007) Malay sign language gesture recognition system. In: International conference on intelligent and advanced systems 2007, ICIAS 2007. IEEE, pp 982–985
15. Paulraj MP, Yaacob S, Desa H, Majid W (2009) Gesture recognition system for Kod Tangan Bahasa Melayu (KTBM) using neural network. In: 5th international colloquium on signal processing and its applications, 2009, CSPA 2009. IEEE, pp 19–22
16. Assaleh K, Shanableh T, Fanaswala M, Amin F, Bajaj H (2010) Continuous arabic sign language recognition in user dependent mode. JILSA 2:19–27
17. Vatavu A, Danescu R, Nedevschi S (2012) Real-time dynamic environment perception in driving scenarios using difference fronts. In: 2012 IEEE intelligent vehicles symposium (IV). IEEE, pp 717–722
18. AL-Ahdal ME, Tahir NM (2012) Review in sign language recognition systems. Symposium on computers & informatics (ISCI), 2012 of conference. IEEE, pp 52–57
19. Premaratne P, Yang S, Zou Z, Vial P (2013) Australian sign language recognition using moment invariants. In: Intelligent computing theories and technology. Springer, pp 509–514
20. Shukla J, Dwivedi A (2014) A method for hand gesture recognition. In: 2014 of fourth international conference on communication systems and network technologies. IEEE computer society
21. Fang G, Gao W (2007) Large vocabulary continuous sign language recognition based on transition-movement models. IEEE Trans Syst Man Cybern 37:1–9
22. Kishore PVV, Kumar PR (2012) A video based indian sign language recognition system (INSLR) using wavelet transform and fuzzy logic. IACSIT Int J Eng Technol 4:537–542
23. Starner T, Pentland A (2013) Real-time American sign language recognition from video using hidden Markov models. Technical report number, vol 375. Technical report, MIT media laboratory perceptual computing section

24. Sharma R, Nemani Y, Kumar S, Kane L, Khanna P (2013) Recognition of single handed sign language gestures using contour tracing descriptor. In: Proceedings of the world congress on engineering
25. Oikonomidis I, Kyriazis N, Argyros AA (2011) Efficient model-based 3d tracking of hand articulations using kinect. In: Proceedings of the 22nd British machine vision conference (BMVC)
26. Yeo H-S, Lee B-G, Lim H (2013) Hand tracking and gesture recognition system for human–computer interaction using low-cost hardware. Multimed Tools Appl 74:2687–2715
27. Liang H, Yuan J, Thalmann D (2014) Parsing the hand in depth images. IEEE Trans Multimed 16:1241–1253
28. Pugeault N, Bowden R (2011). Spelling it out: real-time ASL fingerspelling recognition. In: Proceedings of the 1st IEEE workshop on consumer depth cameras for computer vision, pp 1114–1119
29. Kadambi A, Bhandari A, Raskar R (2014) 3D depth cameras in vision: benefits and limitations of the hardware – with an emphasis on the first- and second-generation kinect models. In: Shao L, Han J, Kohli P, Zhang Z (eds) Computer vision and machine learning with RGB-D sensors, Advances in computer vision and pattern recognition. Springer International Publishing Switzerland, pp 3–26
30. Chen L, Lin H, Li S (2012) Depth image enhancement for Kinect using region growing and bilateral filter. In: Proceedings of the 21st international conference on pattern recognition (ICPR2012). IEEE, pp 3070–3073
31. Kauff P, Atzpadin N, Fehn C, Müller M, Schreer O, Smolic A, Tanger R (2007) Depth map creation and image-based rendering for advanced 3DTV services providing interoperability. Signal Process Image Commun 22:217–234
32. Yoon K-J, Kweon I (2006) Adaptive support-weight approach for correspondence search. IEEE Trans Pattern Anal Mach Intell 28(4):650–656
33. Gonzalez S, Travieso C, Alonso J, Ferrer M (2003) Automatic biometric identification system by hand geometry. In: 2003 of conference 37th annual international Carnahan conference on security technology. IEEE, pp 281–284
34. Sanchez RR, Sanchez AC, Gonzalez MA (2000) Biometric identification through hand geometry measurements. IEEE Trans Pattern Anal Mach Intell 22(10):1168–1171
35. Xiong W, Toh KA, Yau WY, Jiang X (2005) Model-guided deformable hand shape recognition without positioning aids. Pattern Recogn 38:1651–1664
36. Duta N, Jain A, Mardia K (2001) Matching of palmprint. Pattern Recognit Lett 23(4):477–485
37. Wu X, Zhang D, Wang K (2006) Fusion of phase and orientation information for palmprint authentication. Pattern Anal Appl 9(2):103–111
38. Yo RKE, Konukoglu E, Sankur B, Darbon J (2006) Shape-based hand recognition. IEEE Trans Image Process 15(7):1803–1815
39. Bahman N, Sina F, Shahaboddin S, Kwok W, Timon R (2018) Application of ANNs, ANFIS and RSM to estimating and optimizing the parameters that affect the yield and cost of biodiesel production. Eng Appl Comput Fluid Mech 12(1):611–624
40. Nanni L, Lumini A, Dominio F, Donadeo M, Zanuttigh P (2014) Ensemble to improve gesture recognition. Int J Autom Ident Technol
41. McLachlan GJ, Do KA, Ambroise C (2004) Analyzing microarray gene expression data. Wiley
42. Cawley GC, Talbot NL (2007) Preventing over-fitting during model selection via Bayesian regularisation of the hyper-parameters. J Mach Learn Res 8:841–861
43. Chang CC, Lin CJ (2011) LIBSVM: A library for support vector machines. ACM Trans Intell Syst Technol 10(1145/1961189):1961199
44. Duan KB, Keerthi SS (2005) Which is the best multiclass SVM method? An empirical study. In: International workshop on multiple classifier systems
45. Hsu CW, Lin CJ (2002) A comparison of methods for multiclass support vector machines. IEEE Trans Neural Netw 13:1045–1052
46. Platt J, Cristianini N, Shawe TJ (2000) Large margin DAGs for multiclass classification. Advances in neural information processing systems. MIT Press, New York
47. Dietterich TG, Bakiri GB (1995) Solving multiclass learning problems via error-correcting output codes. J Artif Intell Res 22:263–286
48. Crammer K, Singer Y (2001) On the algorithmic implementation of multiclass kernel-based vector machines. J Mach Learn Res 2:265–292
49. Han H, Wu XL, Qiao JF (2013) Nonlinear systems modeling based on self-organizing fuzzy-neural-network with adaptive computation algorithm. IEEE Trans Cybern 44(4):554–564
50. Hossny M, Filippidis D, Abdelrahman W, Zhou H, Fielding M, Mullins J et al (2012) Low cost multimodal facial recognition via kinect sensors. In: LWC 2012: Potent land force for a joint maritime strategy: proceedings of the 2012 land warfare conference. Commonwealth of Australia, pp 77–86
51. Caon M, Yue Y, Tscherrig J, Mugellini E, Abou Khaled O (2011) Context-aware 3D gesture interaction based on multiple kinects. AMBIENT 2011, the first international conference on ambient computing, applications, services and technologies, 2011 of conference, pp 7–12
52. Anand A, Koppula HS, Joachims T, Saxena A (2013) Contextually guided semantic labeling and search for three-dimensional point clouds. Int J Robot Res 32(1):19–34
53. Rafibakhsh N, Gong J, Siddiqui MK, Gordon C, Lee HF (2012) Analysis of xbox kinect sensor data for use on construction sites: depth accuracy and sensor interference assessment. Constitution research congress, 2012 of conference, pp 848–857
54. Luber M, Spinello L, Arras KO (2011) People tracking in RGBD-D data with on-line boosted target models. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), 2011 of conference. IEEE, pp 3844–3849
55. Machado J, Ferreira A (2013) Retrieval of objects captured with low-cost depth-sensing cameras. SHREC2013. Springer
56. Maimone A, Fuchs H (2012) Reducing interference between multiple structured light depth sensors using motion. Virtual reality workshops (VR), 2012 of conference. IEEE, pp 51–54
57. Fawcett T (2006) An introduction to ROC analysis. Pattern Recogn Lett 27:861–874
58. Powers DM (2011) Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation
59. Tetko IV, Livingstone DJ, Luik AI (1995) Neural network studies. 1. Comparison of overfitting and overtraining. J Chem Inf Comput Sci 35(5):826–833
60. Luca Z (2006) Parallel software for training large scale support vector machines on multiprocessor systems. J Mach Learn Res 7:1467–1492
61. Rifkin R (2002) Everything old is new again: a fresh look at historical approaches in machine learning. Ph.D
62. Janez D, Tomaž C, Aleš E (2013) Orange: data mining toolbox in Python. JMLR 14(1):2349–2353
63. Knerr S, Personnaz L, Dreyfus G (1990) Single-layer learning revisited: a stepwise procedure for building and training a neural network. In: Architectures and applications. Springer, Berlin