An Implementation of Sign Language Alphabet Hand Posture Recognition using Geometrical Features through Artificial Neural Network (Part 2)

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

[1 3]MaGIC-X (Media and Games Innovation Centre of Excellence), Universiti Teknologi Malaysia 81310 Skudai, Johor, MALAYSIA

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

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

*Corresponding Author: Saba Joudaki, joudaki@khoiau.ac.ir

Abstract-In the sign language alphabet, several hand signs are in use. Automatic recognition of performed hand signs can facilitate the communication between hearing and none hearing people. This framework proposes hand posture recognition of the American Sign Language alphabet based on a Neural Network (NN) which works on geometrical feature extraction of the hand. The user's hand is captured by a 3D depth-based sensor camera. Consequently, the hand is segmented according to the depth features. The proposed system is called "Depth-based Geometrical Sign Language Recognition" (DGSLR). The DGSLR adopted an easier hand segmentation approach, which is further used in other segmentation applications. The proposed geometrical feature extraction framework improves the accuracy of recognition due to unchangeable features against hand orientation or rotation compared to Discrete Cosine Transform (DCT) and Moment Invariant. As a Support Vector Machine (SVM) is a type of Artificial Neural Network (ANN), it is used to drive desired outcomes. Since there are 26 different signs in the Sign Language alphabet, a multi-class SVM versus a single SVM classifier with 26 classes by an RBF kernel was used to validate each class. The proposed framework is proficient to hand posture recognition and provides an accuracy of up to 96.78 %. The findings of the iterations demonstrated that the combination of the extracted features resulted in a better accuracy rate in the recognition process in the classification step.

Keywords: Sign Language, hand posture, segmentation, geometrical features

1 INTRODUCTION

Sign language is a visual language, which transfers the signs of the hands using not only the movement and orientation of hands, arms, or bodies, but also facial expressions instead of sound patterns. There is no uniform sign language across the world. Each country has its own sign language but in this study, we have considered the American Sign Language which is most popular among existing Sign Languages.

The previous studies on sign language recognition failed to supply a complete or reliable model without restriction. In particular, most of them are depending on users, in other words, they are not able to be applied for independent user systems. It can be conducted by some methods, especially in the feature extraction steps due to the image base system. Furthermore, they can involve mimic of the face or body postures for more details, for example, portray anger and emotion through the hands. Likewise, all of the studies on Artificial Neural Network (ANN) have shown that it has a robust learning capability, and there are varieties of ANN systems used in hand posture recognition systems. On the other side, the Support Vector Machine (SVM) approaches have very effective results on recognition systems (Dominio *et al.*, 2014).

Our novelty in this work is using a new method of geometrical feature extraction which leads to get more accurate classification in our classifier. In fact, a new integration of the extracted features, geometrical features of the hand are presented in Sign Language recognition system. Furthermore, the proposed system uses a new simple approach for segmentation in different backgrounds. The concept regarding Microsoft's Kinect sensor returns to the attainment of 3D data for paving the way in a new solution for quite a few challenging computer vision issues, including human activity analysis, object tracking, indoor 3D mapping, supervision scenarios, and recognition especially hand gesture recognition. Changes due to different lighting conditions have a bad effect on the recognition process. Furthermore, the recognition process is more difficult in a cluttered background than a plain background. This issue has an important impact on accuracy. In order to make a system that works in both simple and cluttered backgrounds, indoor or outdoor with different lighting conditions, a new approach is necessary to solve these problems.

A realistic Sign Language Recognition with error-free recognition is an ambitious goal for many outstanding researchers in computer science especially pattern recognition. The effects of illumination changes on hand recognition as well as occlusion by another object in the scene in the cluttered background have been attempted in this research. In addition, finding some features of the hand which are independent to the hand orientation or direction have been important issues which this research tried to address. The proposed methods cater for the weaknesses in the hand posture recognition system to develop an SLR system. These methods are applied in segmentation and feature extraction phases and can increase the overall accuracy due to the depth-based images and geometrical features of the hand.

This paper used SVM in DGSLR recognition. All algorithms in each part have been explained in detail. The model used in support vector machines, especially in the most basic cases (eg two-class classification), is a model with a linear structure and very similar to what is used, for example, in the multilayer perceptron neural network or MLP. In fact, along with some other differences between the two models, they actually teach a very similar structure in two different ways. In MLP neural network, the parameters of this model are adjusted by error minimization, but in SVM, the risk of incorrect classification is defined as a target function and the parameters are adjusted and optimized accordingly. For some issues, the error rate may be as low as zero, but of all the zero-error models, there is only one that has the lowest operational risk. Therefore, in some cases, the SVM output, in addition to its better performance, will also show more robustness to changes and noise in the data. Because it is basically designed and trained to withstand such uncertainties and to perform well. On the other hand, the use of the term neural network (artificial) or any other similar term to refer to such devices has been merely to create a metaphor that is appropriate and close to nature, and the essence of the theorem is the mathematical relationship behind these systems. From this perspective, many of the systems and models used in the field of machine learning use very similar (and sometimes identical) mathematical structures, and only in the way the problem is expressed, the way the models are set up and described with They are different from each other. For further study, it is recommended that you read the second edition of Simon Haykin's famous book, Neural Networks: A Comprehensive Foundation, published in 1999. In the introduction of this book, it is well explained that SVM is a type of neural network. The third edition of this book, with the new title "Neural Networks and Learning Machines" was published in 2008. Another suitable reference for further studies in this regard is the book "Neural Networks in a Soft-computing Framework" (neural networks in the framework of soft computing), which in the introduction and chapter ten of this book, the topic of support vector machines, and the fact that they are a special form of artificial neural networks has been debated. The book "Pattern Recognition and Machine Learning", written by Christopher M. Bishop (Christopher M. Bishop), is another very important and practical reference in this field, and interested for more information, you can refer to this important and practical reference.

This paper introduces an American Sign Language Alphabet recognition method to hand gesture recognition to help deaf and dumb people. It also presents some geometrical features of the hand for achieving more reliable recognition. Then it explains the literature review in depth-based on hand gestures in sign language recognition systems. In the next part, the research methodology and the procedure of the research are described. Segmenting the signer's hand is performed and the appeared issues are discussed. The Level set method is implemented and reported their results. The feature extraction method is in accordance to hand geometrical features. The Support Vector Machine (SVM) algorithm is implemented to classify the extracted features in the previous step for recognizing the performed gestures. Then, it expounds implementation step. Finally, a comparison discussion between the proposed method by SVM and two classifiers, K Nearest Neighbor (K-NN) and Decision Tree (DT) are employed. The evaluation and testing of the system are applied and then the accuracy rate of the proposed method is shown as charts and tables. Also, errors due to wrong recognition are shown. The paper ends with a conclusion and some suggestions for further research in the future, which may provide ways to easier hand gesture recognition in order to apply in the recognition systems.

The idea behind this work is: users can act on desired signs while the proposed system detects the signs. The detected signs can be converted to sound or text for normal people. The new idea in this research is depth-based segmentation and geometrical features which distinguishes it from other methods. It can be developed by a depth-based camera embedded on a cell phone. The depth-based camera can lead to subtract the background more easily whether simple or clutter. On the other side, geometrical features are independent of the orientation, location, or position of the hand. So, the emotional signs do not make any problem in the recognition process. There is natural variability in the executed signs because of the different positions of the hand in the same signs, and the observations are error-prone, thus applying a method other than the existing exact matching of features is needed without considering the finger's positions. Furthermore, it can be developed on a system in public places such as airports or libraries, or even educational places like universities. It can be used in conferences or other scientific assemblies.

After introducing an American Sign Language Alphabet recognition system, some related works were explained in the literature review in section 2. In section 3, the research methodology and depth-based geometrical features procedure are described. The used dataset, Segmentation method, proposed feature extraction methods and finally classification step have been defined. Experimental results and discussion are in section 4. The paper ends with a conclusion and some suggestions for further research in the future, which may provide ways to easier hand gesture recognition in order to apply in the recognition systems.

2 RELATED WORKS

There are several challenges which we will try to solve. Complex background and lighting conditions are more important than the rest factors. The distance between the user and Kinect Camera during the capturing images can be considered as a limitation of this research. However, some ordinary cameras can solve this issue, but they have no depth-based application. The process is very sensitive to hand movements due to the illumination changes. This may lead to the occlusion of some parts of the hand by other parts. Two letters ‘J’ and ‘Z’ are motional signs and it is much better to remove them from the hand posture recognition field. These two signs are very similar to ‘I’ and ‘G’, they have similar features together. It caused to confuse the conditions in the classifier process.

Limitations and constraints in the existing vision-based methods have been caused to obtain the unsatisfying results in the previous research. Object recognition in the cluttered scene, or with long sleeve clothes of the signer, or the necessity of motionless head or face are some of these restrictions. Likewise, steady hand movements, stable pose and location of the body, determined primary location for hands, and restricted vocabulary is other discussed limitations in this field.

Lee et.al (2013) explained a computer vision based method for posture recognition of a hand posture and its application on an iOS iPhone. The proposed algorithm used YCbCr images (Lee et al., 2013) to set skin regions. They eliminated noise caused by slanted hand posture. Then ANN was used for sign recognition and applied to another device like iPhone. The accuracy rate of recognition was 89%in the motion hand posture and it was 94.6 percent for static hand posture, but the skin detection was affected by the illumination conditions of the environment. This issue caused a low accuracy in some states or orientations.

The feature extraction step is one of the crucial steps in every recognition system. There is a diverse huge collection of feature extraction methods that each of them has some advantages and disadvantages, such as Scale-invariant Feature Transform (SIFT) (Dardas and Georganas, 2011, Gurjal and Kunnur, 2012), Wavelet Moments (Chen *et al.*, 2012), Histogram of Oriented Gradients (HOG) (Mihalache and Apostol, 2013, Nölker and Ritter, 1998, Nölker and Ritter, 1999), and Gabor Filters (GF) (Amin and Yan, 2007, Pugeault and Bowden, 2011). These techniques are very robust in the recognizing process but for a small number of simple hand postures (Dong *et al.*, 2015). For example, Dardas and Georganas (2011) obtained an accuracy rate of 96.23% for recognizing six signs using SIFT based and an SVM classifier. Pugeault and Bowden (2011) implemented the recognition of 24 static ASLalphabet signs using the Gabor Filter (GF) method. The mean accuracy of 75% was reported. Moreover, the proposed method had a high confusion rate of 17% between similar signs such as "r" and "u". In short, these methods are usually not able to obtain desirable accuracy in complex classifying or variations of a lot of ASL signs.

In addition, Dominio *et al*. (2014) presented multiple depth-based descriptors. The descriptors included some features of the hand such as distance and elevation, the hand's contour curvature, and properties of the palm region to be extracted. The achieved accuracy was 93.8% by SVMclassifier in an experimental set of 12 static and digit signs of ASL alphabet. Liang *et al*. (2014) improved the per-pixel based hand parsing method by distance-adaptive feature selection scheme and super-pixel partition-based Markov Random Fields (MRF). The improved algorithm was led to increase from 72% to 89% of accuracy in per-pixel classification. The above methods recognize only a small number of simple postures (less than 15) including ASL digits and custom signs which are a small portion of ASL alphabet signs.

Changes due to different lighting conditions have a negative effect on the recognition tasks due to the shadow or undesired effects on the objects (Chai et al., 2013, Kishore and Kumar, 2012a, Zhu et al., 2010). Furthermore, the recognition process is more difficult with a cluttered background than a plain background (Prasad *et al.*, 2015). Compared to 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 (Chai et al., 2013, Sharma et al., 2013). On the other word, the selected extracted features of the hand in these previous hand recognition systems is dependent on the position or direction of the signer's hand (Oikonomidis *et al.*, 2011, Yeo *et al.*, 2013). Then, the recognition process is performed correctly just for a specific user and it does not work properly for generic users. 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 fingertips as a feature (Liang et al., 2014). The main weakness of the use of hand fingertips in the extracted features is that they can be occluded by other fingers. There is a natural variability in the executed signs because of the different positions of the hand in the same signs. Furthermore, if the observations are error-prone, then a method other than the existing exact matching of features is needed without considering the finger's positions.

Kiseľák et al (2020) introduced a new method as "scaled polynomial constant unit activation function – SPOCU" for a medical image in some cancer detection. Such a novel activation function relates to complex patterns through the phenomenon of percolation, and thus, it can overcome already introduced activation functions, e.g., SELU and ReLU. Discrimination between mammary cancer and mastopathy tissues plays a crucial role in clinical practice. In this case, a more precise activation function in the classifier is necessary which can detect the tissue and its complexity. But in our case, using such an activation function only leads to increasing computational time.

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. As Hinton (2008) mentioned the SVM causes to get a good generalization on a big dataset. Since a big data set requires a complicated model and the full Bayesian framework is very costly in computation. In contrast, the SVM is faster and still has a good generalization solution. Furthermore, due to a very big set of non-linear task-independent features, SVM has a clever way to prevent Over-Fitting problem.

3 DEPTH-BASED GEOMETRICAL FEATURES IN HAND RECOGNITION

3.1 Dataset

Two separate datasets are employed in this research. The first one is the chosen dataset by the research which is called DGSLR. The other one 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 the signs. Each letter was repeated for five times.

After the preparing step and teaching the signs to the signers, the images were captured by the Kinect Explorer – WPF application at 30 frames per second. In this coloured image capturing application, the hand is detected by a distinct colour due to the depth feature.

The capturing process was performed in both plain and cluttered backgrounds in different variations of illumination. As Figure 1 illustrates, the other objects in the cluttered background do not have any interference in the detection procedure. The farther objects are removed and the closer objects are shown in the different depth with the user in the foreground. Thus, the hand is still shown as different colours in the RGB mode (Figure 1 (left)) and brighter view in the depth mode (Figure 1 (right)). The hand is also recognizable in two modes.

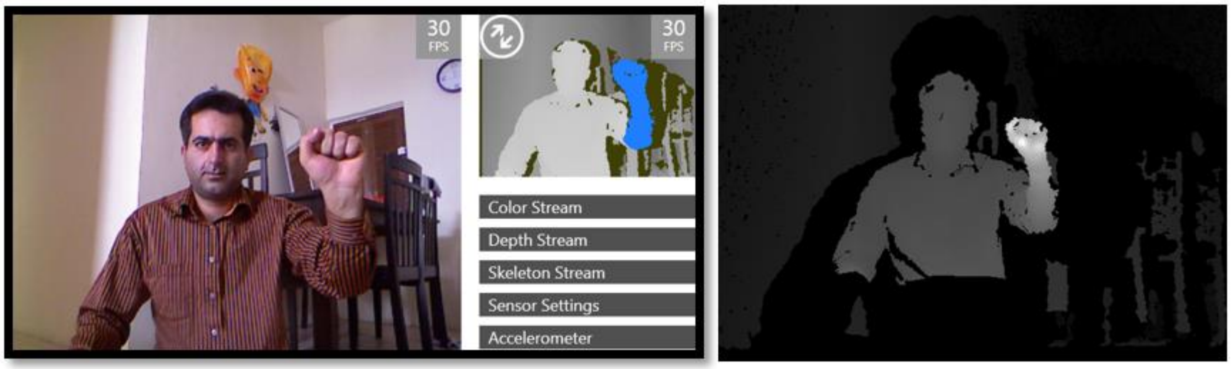


Figure 1 Cluttered background in RGB and Depth mode

In order to validate the data, a huge standard dataset from the Centre for Vision, Speech and Signal Processing, University of Surrey (Pugeault and Bowden, 2011), was used. The images have been captured from 9 people in different backgrounds similar to the research dataset. The images gathered by Kinect and are only depth-based. In addition, there are more than 400 repetitions on each sign in different postures and directions. The users changed their hand direction and also the distance to the Kinect sensor.

Posture or gesture recognition methods can be divided into two types: one is to use Kinect (for example in our work), Leap motion and other depth cameras to obtain image depth information, such as position. The other one is to split the gesture from the background by traditional methods and then extract the apparent image characteristics of the posture by neural networks to perform posture recognition.

In this case, according to the type of neural network (MLP) and learning paradigms (Backpropagation), and also the desired task which is “Pattern Recognition”, the “Fermi function” can be used. This study uses a non-linear SVM, and since there are different kernel functions in the non-linear SVM structure, choosing a kernel based on the prior knowledge of invariances as suggested by Cawley and Talbot (2007) is an excellent idea. The Gaussian Radial Basis function (GRBF) kernel is one of the most common kernel which is used in this research.

3.2 Segmentation of the Hand

Hand extraction is a crucial step in hand recognition systems because all of the following processing steps are performed on the segmented regions only. The proposed scheme for segmenting the hand is based on the depth data. A scenario used in this research is to have users facing the Kinect camera with their hands held in front of themselves. In this case, the hand seems brighter than the other objects because of the depth capability in the image. It caused to place the body or other objects in the scene in the deeper layer and the hand seems by different colour due to changing light conditions compared to the rest of the body. The distance between the user and the Kinect was 150 cm. In addition, the lighting conditions were changeable during the signing process.

3.3 Morphological Object Dilation

There are some noisy points in the obtained depth images in this study. Then, a post-processing procedure has been to improve the obtained depth images. These noisy points can be due to hand movements or shaking during the signing. Furthermore, the Kinect sensitivity to the illumination conditions can also have an effect on the images. A filtering operation can perform on the image to address this issue, but according to the review on the filtering methods, they are commonly time-consuming procedures (Chiang *et al.*, 2013, Pal *et al.*, 2014). On the other hand, in our depth images, no need to rectify the edge and only some morphological operations are applied for smoothing the binary depth-based image and remove the noisy points on the hand surface as the demonstrated example in Figure 2.

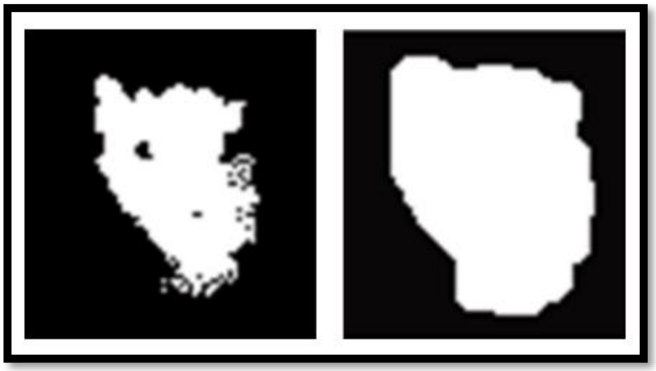


Figure 2 The binary image before and after morphological operations

The first step, all the images were resized to a 128-by-128 pixel matrix. A unified dataset of images, all of equal size allows for modifications in later stages if needed. These points of the image should be distinguishable from the rest black points like background points. Since the number of these type of images was little in this study, the mentioned issue was resolved by a series of morphological functions in Matlab as following definition.

The dilation of A by B is implicated $A \oplus B$ where defined as:

$$A \oplus B = \left\{ z \middle| (\hat{B})_z \cap A \neq \phi \right\} \tag{1}$$

Where $\hat{B}$ is the reflection of the structuring element *B*. In fact, it is the set of pixel locations Z, where the reflected structuring element overlaps with foreground pixels in *A* when translated to *Z*. In the grayscale dilation, the structuring element has a height. The grayscale dilation of A(x,y) by B(x,y) is as:

$$(A \oplus B)(X, Y) = \max \left\{ A(x - x', y - y' + B(x', y') \middle| (x', y') \in D_B \right\} \tag{2}$$

where D_B is the domain of the structuring element B and A(x,y) is assumed to be $-\infty$ outside the domain of the image. To create a structuring element with non-zero height values, the syntax strel (sdom, height) is used, where height shows the height values and sdom corresponds to the structuring element domain. The grayscale dilation is commonly performed with a flat structuring element (B(x,y) = 0). Grayscale dilation using such a structuring element is equivalent to a local-maximum operator:

$$(A \oplus B)(X, Y) = \max \left\{ A(x - x', y - y') \middle| (x', y') \in D_B \right\} \tag{3}$$

3.4 Feature Extraction

After hand segmentation and post processing based on depth hand images, selected feature vectors are expected to represent the position of fingers and palm. Consequently, fingers should be roughly characterized by a robust approach.

3.4.1 Hand Geometry

The hand area (*HA*) and hand perimeter (*HP*) are the first feature descriptors which were calculated by morphological operators. In order to compute the perimeter of the hand, the distance between each adjacent pair of pixels around the hand contour is calculated. The discontinuous areas in the hand region may lead to unexpected results. All noisy points should be removed to gain better results in the hand area and perimeter here. Two mentioned parameters, *HA* and *HP* are for all the fingers are closed and when they are open. This is the minimum and maximum value, respectively, so the other signs are within this range.

3.4.2 Convex Hull of the Hand

The convex hull of the hand is calculated in order to gain the desired geometry information. It should be noted that the forearm or arm of the hand were removed from the initial images as it did not contain any important information. In the hand image of the research, a convex hull is a *n-by-m* matrix that determines the smallest convex polygon containing the hand region. The parameter *n* is the number of the pixels and *m* represents the vertexes. Each row of the matrix demonstrates the coordinates of one vertex of the circumscribed polygon of the hand. In the next section, the concept of convex polygon will be introduced. Consequently, 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 Figure 3 the polygon around the points is a convex hull and the six points which are on the boundary are called "hull points".

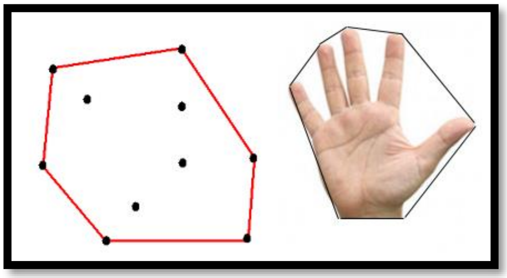


Figure 3 Convex hull of (left) a points set, (right) segmented hand

The convexity defects of the hand have some geometry properties which can be used as features of the proposed system in this study. The area of the convexity defects, *CDA*, was computed by a similar algorithm of the convex hull. Likewise, the number of convexity defects represents the number of open or closed fingers. The empty spaces between the opened fingers are also convexity defects, so the number of these spaces can be represented for some specific signs in the classification step. This is much more useful for designing a reliable recognition system.

3.4.3 Ratio Feature

Another extracted feature is the ratio between the hand area, *HA*, and the area of the convex polygon, *CHA*, enclosing it. As mentioned above it is called convex hull. So it is named convex hull area ratio that is:

$$\Re_{CHA} = \frac{Handarea(HA)}{ConvexHullarea(CHA)} \qquad (4)$$

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

$$\Re_{CHP} = \frac{Handperimeter(HP)}{ConvexHullperimeter(CHP)} \qquad (5)$$

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

$$\Re_{CDA} = \frac{Handarea(HA)}{Convexitydefectarea(CDA)} \tag{6}$$

3.4.4 Distance Feature

The height and width of the signer's hand are other measurable features which are considered in this research. The height and width values can represent the hand postures. Although the similar signs have similar values of height and width, they can be classified in the same class for more clarity in the classifier. For example, as represented in Figure 4, for three signs 'A', 'S', and 'T' the value of the height and width are close together. This similarity also occurs between 'R' and 'U'.

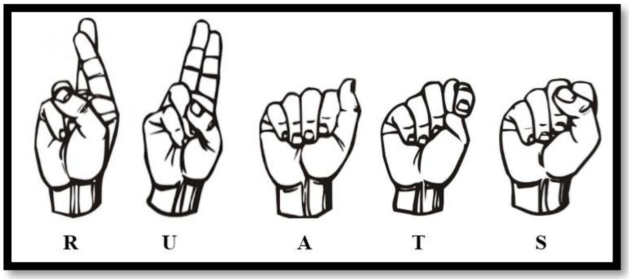


Figure 4 Similar signs with close geometrical values

For computing the height and width of the hand, the edge of the hand should be detected. Then, the longest diameter of the hand in vertical and horizontal directions is computed based on the Eigenvalue and the Eigenvector concepts. As the last step, the calculation of the distance feature was performed by the Euclidean distance between the ending points of these diameters on the hand boundary.

The first step, 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. Matlab software also includes several algorithms for calculating the object's boundary, but edges may include the adjacent number of rows which creates a 'thick' edge as shown in Figure 5.

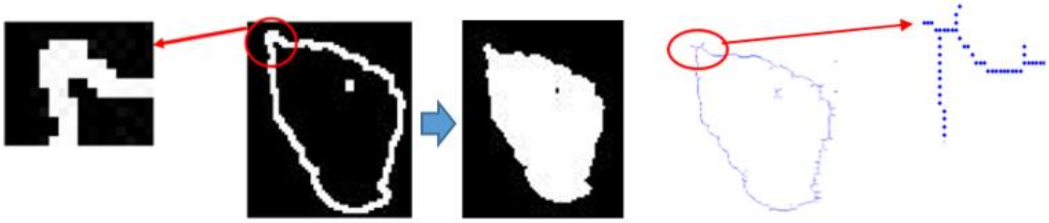


Figure 5 Thick edge includes several points, Image edge detection algorithm, then original image, detected contour, more detailed view are extracted.

In statistics, a covariance is a matrix which its element in the *i, j* position means the covariance between the i^{th} and j^{th} elements of a random vector variable. Each element of this vector is a scalar variable with a finite number of appeared experimental values or by a finite or infinite number of possible values determined by the theory of joint probability distribution of all the random variables.

The covariance between two jointly distributed real-valued random variables *X* and *Y* with finite second moments is (Statistics, 2002):

$$\sigma(X,Y) = E[X - E[X])(Y - E[Y])] \ = E[XY] - E[X]E[Y]$$

where *E[X]* is the expected value of *X*. Since all probabilities p_i adds up to one, $p_1 + p_2 + ... + p_k = 1$, the expected value is shown as the weighted average:

$$E[X] = \frac{x_1 p_1 + x_2 p_2 + ... + x_k p_k}{1} = \frac{x_1 p_1 + x_2 p_2 + ... + x_k p_k}{p_1 + p_2 + ... + p_k} \qquad (7)$$

An eigenvector of a square matrix in linear algebra is a vector that does not change its direction under the linear transformation. If *v* is a non-zero vector, then the *v* is an eigenvector of the square matrix *A* as *Av* is a scalar multiple of *v*.There is a relationship between *n* by *n* square matrices and linear transformations. The linear transformation of *n*-dimensional vectors specified by an *n* by *n* matrix *A* is:

$$Av = w \qquad (8)$$

where,

$$w_i = A_{i,1}v_1 + A_{i,2}v_2 + ... + A_{i,n}v_n = \sum_{j=1}^{n} A_{i,j}v_j \qquad (9)$$

If *w* and *v* be the scalar multiples then:

$$Av = \lambda v \qquad (10)$$

which *v* is an eigenvector of the linear transformation *A* and the factor *λ* is the eigenvalue of it.

The approximate longest diameter in the hand and then the perpendicular line to it should be computed as shown in Figure 6. The coordinate of the points on the hand contour was computed in the boundary detection algorithm. So, the gravity centre point is easily obtained. Then the covariance matrix is computed. The direction and value of the longest diameter will be obtained by calculating the Eigenvalue and the Eigenvector.

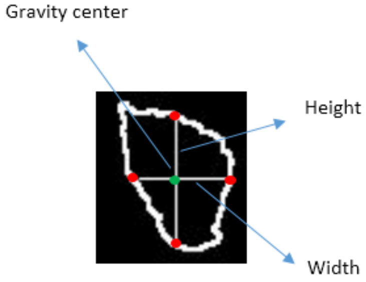


Figure 6 Height and width of the hand

3.5 Feature Vector Structure

All computed features on both DGSLR and standard datasets were saved in two repositories in a CSV (comma separated file) which we utilized Microsoft Excel for easy usage. The first one which belongs to the DGSLR dataset includes three sheets where each sheet corresponds to each user. The rows and columns of this file represent the letters and features respectively. The last column is considered as a label column for labelling each sign within 1 to 26. Considering the leave-one-out approach which will be explained in the next part in classification, one person is kept for testing and the rest is considered in the training phase. The second excel file corresponds to the standard data set that consists of 26 sheets which each of which belong to a specific sign. We took a regular training procedure of 70/30 split, 70% of images is used for training, while 30% is used for testing'.

3.6 Classification

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. In this research, a multi-class one versus one SVM classifier has been used, and in 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 non-linear SVM, and since there are different kernel functions in the non-linear SVM structure, choosing a kernel based on the prior knowledge of invariances as suggested by Cawley and Talbot (2007) is an excellent idea.

The Gaussian Radial Basis function (GRBF) kernel is one of the most common kernel which is used in this research as obtained by Equation 11.

$$k(x_i, x_j) = \exp(-\gamma \left\| x_i - x_j \right\|^2) \quad \text{for } \gamma > 0 \tag{11}$$

The Gaussian radial basis function kernel supports the corresponding feature space in an infinite dimension. The maximum margin in the classifier is well regularized, and it is widely believed that the infinite dimensions do not spoil the results (Jin and Wang, 2012). The GRBF kernel makes a good default kernel in a non-linear model. It may lead to having an efficient-to-compute and high accuracy approach without having the huge and potentially infinite-dimensional feature vector. The optimized run time of the GRBF is one of the other reasons to employ it in the classifier of this research. The GRBF execution time is bounded by *O(nlogn)*, where *n* is the number of training samples.

In this research, there are two datasets of the depth-based image of the sign language alphabet. Firstly, the classification process is applied on the DGSLR dataset, so the training set contains data from three available users. A cross validation method as K-fold cross validation is used by *K* equals to 5 and 10 in the testing step. In the K-Fold validation method, the collected data is partitioned into the K subsets. In these subsets, one of them is used for validating data and *K-1* subsets for the training process. This procedure is repeated *K* times and all the data are used once for training and once for testing. Finally, the average of these *K* procedures is selected as the final estimation.

The two parameters C and φ of the RBF kernel are subdivided with a regular grid which when C is considered, equals to 1,10,100, and 1000, and parameter φ equals to .001, .01, .1, 1. Similar to other classifiers, for each couple of these parameters, the training collection is divided into two categories, $N-1$ users in the training set and the rest for validating. We reiterate the 70/30 split between training and testing. The accuracy is assessed and the testing process is iterated frequently based on changing the iteration number. Finally, the parameter pair which gives the most accuracy is selected and applied to the SVM structure.

In order to measure the classifier accuracy, two statistical parameters called '*Sensitivity*' and '*specificity*' were used. 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. Similarly, it is complementary to the false positive rate.

A perfect predictor approach describes samples as 100% sensitive and 100% specific, but in fact, there is no perfect predictor and theoretically, all of them have a minimum error bound called the Bayes error rate. As concluding the four outcomes can be formulated derived a confusion matrix as follows:

- True positive (TP) = correctly identified
- False positive (FP) = incorrectly identified
- True negative (TN) = correctly rejected
- False negative (FN) = incorrectly rejected

Two equations can be formulated and derived from a confusion matrix as follows (Fawcett, 2006, Powers, 2011):

$$Sensitivity = TruePositiveRate(TPR) = \frac{NumberofTruePositives}{NumberofTruePositives + NumberofFalseNegatives}$$

$$= \frac{\sum TruePositive}{\sum ConditionPositive} \tag{12}$$

$$Specificity = TrueNegativeRate(TNR) = \frac{NumberofTrueNegatives}{NumberofTrueNegatives + NumberofFalsePositives}$$

$$= \frac{\sum TrueNegative}{\sum ConditionNegative} \tag{13}$$

These statistical parameters can be represented in the confusion matrix as shown in Table 1.

Table 1 Statistical parameters in confusion matrix to measure the classifier accuracy

	Predictive Results of Classification	
	Yes	No

Actual Results of Classification	Yes	True Positive (TP)	False Positive (FP)
	No	False Negative (FN)	True Negative (TN)

4. EXPERIMENTAL RESULTS AND DISCUSSION

The experiments were divided into two categories, our own dataset and the standard dataset. The experiments were performed on a gesture dataset in the Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, GU2 7XH, the United Kingdom allowing comparison with state-of-the-art techniques in this study. A number of practical tests were performed to evaluate the proposed methods and computed the accuracy of the system with different parameters. Using a larger data set will definitely lead to more accurate results. The proposed method is independent of the size, angle and rotation of the hand. Therefore, the increase in the dataset size leads to better network learning and finally more accurate results.

Some geometrical features were used as parameters:

- hand area (HA),
- hand perimeter (HP),
- A convex hull is an nXm matrix that determines the smallest convex polygon containing the hand region. So, the area of the convex polygon (CHA), & perimeter of the convex polygon (CHP) are considered as new parameters.
- The area of the convexity defects is computed by a similar algorithm to the convex hull. Likewise, the number of convexity defects represents the number of open or closed fingers. The empty spaces between the opened fingers are also convexity defects, so the number of these spaces can be represented for some specific signs in the classification step. The area of the convexity defect of the hand (CDA) is another parameter that is considered in this paper.
- The longest diameter of the hand in vertical and horizontal directions is another parameter which is computed based on the Eigenvalue and the Eigenvector concepts.

These parameters and the ratio between them are considered in the computational process. The parameters under study depend directly on the type of the signs. Since some of these signs are very similar, the values are very close together. A threshold has been considered for each sign to avoid interfering and overlapping. For example in two signs, i & j, the convexity defects are very close together as you can see in the figure.

A tolerance between ± 0.1 is error-prone in each repeat for the similar sign but by one specified signer because it depends on the size of the signer’s hand. Furthermore, in different signers with the same sign, it increased to ± 0.5 in each iteration.

In other parameters the tolerance was different, so for each parameter, a different tolerant was considered.

4.1 Data Collection

After collecting the desired data from both DGSLR and standard datasets, the signer's hand should be separated from the rest of the body and other objects in the scene. The proposed segmentation approach is represented based on the depth-based image property. After converting the grayscale images to the binary mode, Otsu's thresholding algorithm (Batenburg and Sijbers, 2009) was applied to the images as described in the previous chapter. Some samples of the experimental results are shown in Figure 7.

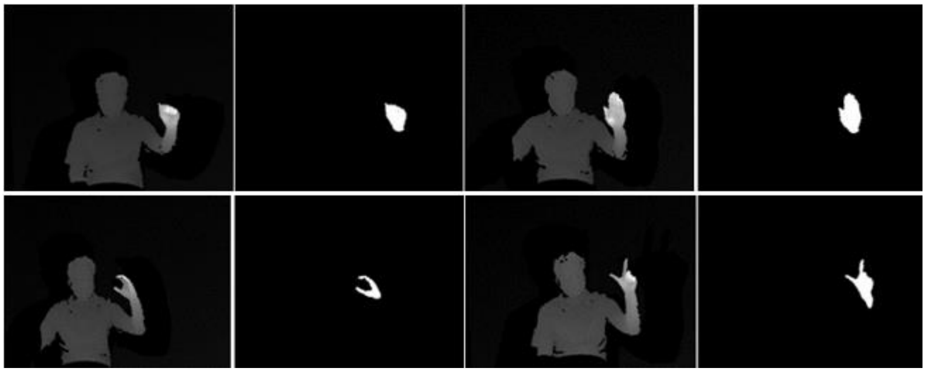


Figure 7 Hand segmentation

It is clearly observed that no need to trace the hand or determine the bounding box around the hand region. In addition, there is no difference between the left or right hand, because the coordinate of the hand location is not important. The hand can be segmented well only based on the illumination intensity. For more clarity, the segmented hands were cropped and zoomed in as shown in Figure 8.

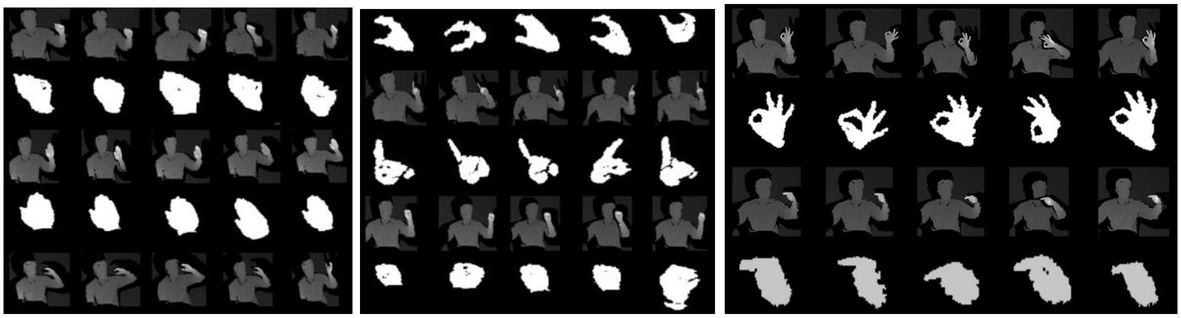


Figure 8 Hand segmentation

In order to separate the wrist from the forearm, the hand contour was computed, and an inscribed circle with a palm centre was drawn. The longest diameter of the hand was calculated based on the Eigenvector and Eigenvalue of the hand image. Then, the perpendicular line to the longest diameter and also tangent to the inscribed circle was plotted as represented in the previous section in detail. The green star represents the tangent point between the inscribed circle and the perpendicular line (hand width) in the lowest point of the circle. Figure 9 shows some experimental results a none expert user.

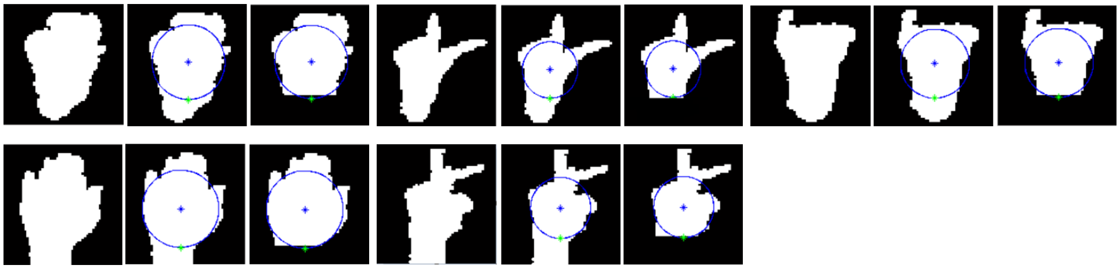


Figure 9 Removed forearm

In order to recognize the hand position, the Level Set method (LSM) was employed due to the low computational cost and high speed (Gonzalez *et al.*, 2009). In this case, it was applied to the signer's image for recognizing the missed parts in hand. 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 the definition of a set of arbitrary points around the hand region. Some examples of the experimental results of the Level Set Method have been highlighted in Figure 10.

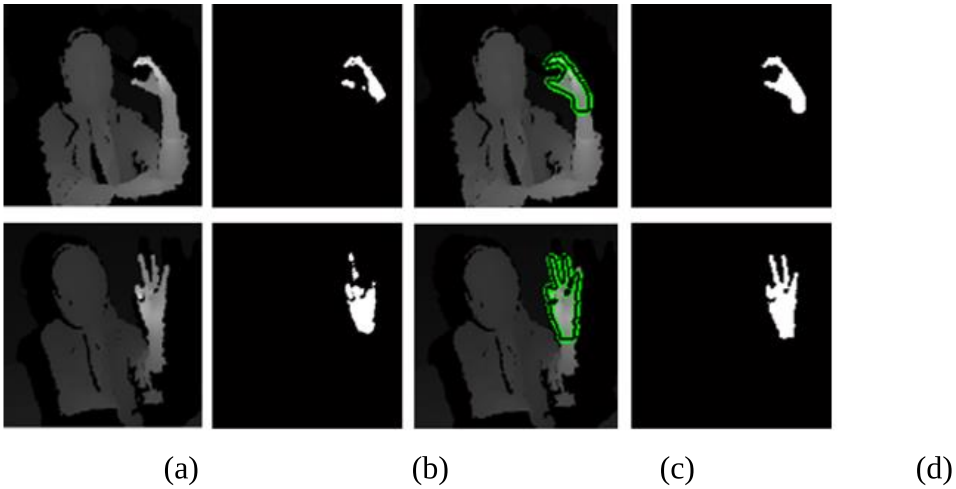


Figure 10 Comparision between the Kinect and Level Set segmentation, (a) depth image, (b) Kinect segmentation, (c) LSM execution, (d) LSM segmentation

4.2 Feature Extraction

The next step includes extracting features from the segmented hand. These features will be used in the classification step for evaluating the conducted gestures.

4.2.1 Hand Geometry Features

The geometrical properties of the hand are reliable features for hand gesture recognition systems because properties like area or perimeter are constant against rotation or changing the location of the hand. The signer may move a bit or the signer's hand may be shaken and change its position or one signer might use the right hand for some signs and the left for other signs. Table 2 shows HA and HP for the three signers in DGSLR dataset.

Table 2 The area and perimeter of the hand for three different signers

	HA	**HP**	**HA**	**HP**	**HA**	**HP**
a1	2283	230.9949	1369	170.0244	1854	170.235
a2	1765	175.4386	1482	180.5097	956	154.2384
a3	2037	200.0244	1356	171.3245	1803	165.2301
a4	1768	196.8528	1407	161.5391	1456	164.0213
a5	1582	185.3381	1334	162.9533	2016	190.5064
b1	2744	237.3381	2119	224.4092	1758	160.5489
b2	2673	242.4092	2043	218.7523	2013	215.324
b3	2355	230.9949	2026	207.5807	2254	254.159
b4	2175	209.9655	1990	198.9949	2189	220.407
b5	1793	193.6812	2139	210.9533	2105	218.754
c1	1270	230.9949	1025	256.3289	1985	198.564
c2	1032	256.2082	1105	273.3209	2246	247.196
c3	2034	267.1787	956	293.1289	950	194.3245
c4	1373	246.7939	1074	303.5635	785	194.2301
c5	1238	234.9856	881	249.4214	1125	215.321
d1	1039	224.8944	1062	176.2254	1048	231.6523
d2	1254	212.2082	1142	195.4327	1001	256.3214
d3	1099	213.7229	1213	191.7817	1057	205.678
d4	1076	209.4802	1254	201.3564	1023	182.36895
d5	1096	215.8234	1349	200.1665	985	174.523

4.2.2 Convex Hull of the Hand

The convex hull of the 2D depth-based hand shape was computed by the interpolation and computational geometry of mathematic functions. It can be one of the constructions of the existing descriptors for the hand posture (Pedersoli *et al.*, 2012). All the binary segmented hand images were resized and then the convex hull function was applied to them. Some instances of the results were represented in Figure 11.

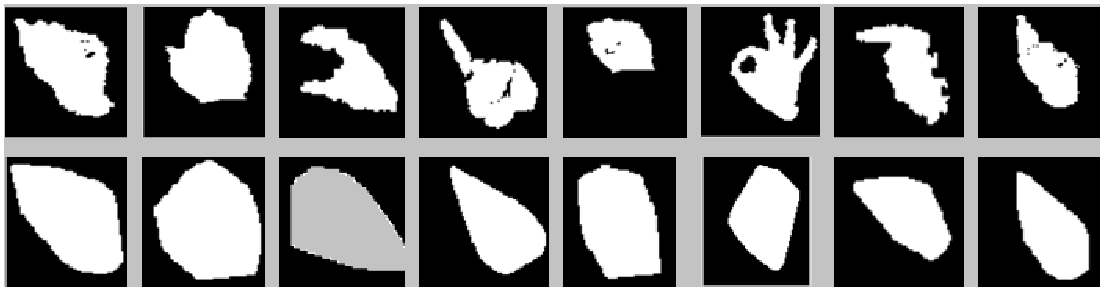


Figure 11 Convex hull of the hand shape

As it can be observed in the above figure, the similarity between the sign may lead to the similar convex hull polygon around them like the first and fifth signs which represent the ‘A’ and ‘E’ signs. This similarity is also observed in Figure 11 between the fourth and the last sign which are ‘D’ and ‘R’ signs. meanwhile, a little difference in the geometrical features such as area and perimeter of the convex polygon is acceptable for this classification process.

The convex hull area and perimeter have been shown with *CHA* and *CHP* abbreviations, respectively which the results them for DGSLR dataset is presented in Table 3

Table 3 The area and perimeter of the convex hull for three different signers

	CHA	CHP	CHA	CHP	CHA	CHP
a1	2584	208.267	1950	184.2614	1516	159.4386
a2	1860	166.9533	1258	125.6041	1660	166.6102
a3	2205	183.8823	1900	180.2398	1432	154.6897
a4	1961	177.8823	1540	158.02364	1525	155.5391
a5	1768	164.267	2231	212.9876	1454	154.8112
b1	3013	219.5807	1856	175.234	2323	208.7107
b2	2944	219.5807	2219	201.9875	2217	204.3675
b3	2613	210.7523	2679	254.7745	2212	197.9239
b4	2330	190.1665	2215	223.128	2119	190.5097
b5	1977	176.5097	2265	214.5879	2306	200.0244
c1	1772	172.468	2054	165.328	1764	195.3245
c2	1705	168.9533	2542	214.7107	1810	181.3797
c3	2555	213.6812	1781	142.03214	1695	219.6224
c4	1941	179.196	1745	112.2131	2158	196.2082
c5	1736	169.9828	1986	120.3564	1590	164.5097
d1	1494	168.0244	1821	135.21	1342	160.8112
d2	1698	179.2376	1854	120	1654	175.8641
d3	1588	175.1371	1985	165.432	1643	174.9533
d4	1490	166.6102	1421	165.3223	1721	179.9239
d5	1558	172.0244	1052	103.5024	1694	176.6102

4.2.3 Convexity Defects of the Hand

An applicable way of estimating the shape of a specific object is to calculate its convex hull and then its convexity defects. As mentioned the convexity defects are some parts of an object which are contained in the convex hull of the object but it does not belong to the object. There are several ways to compute the convexity defects of an object (Keskin *et al.*, 2012). Some experimental results of the applied procedure to obtain the convexity defects of the hand have been illustrated in Figure 12.

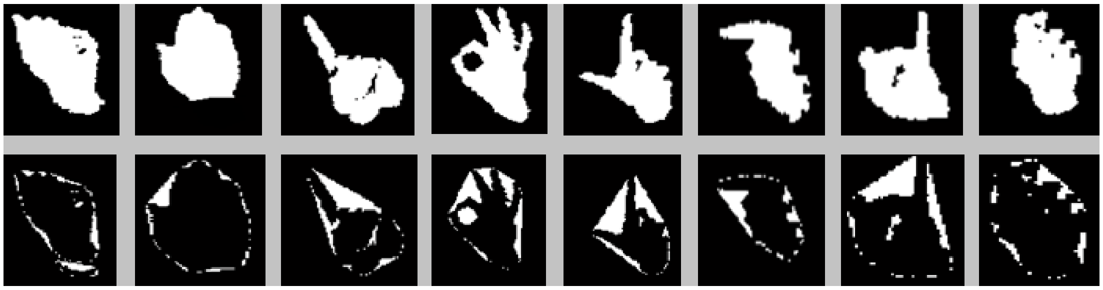


Figure 12 Convexity defects of the hand shape

Too many informative data can be extracted from convexity defects of the hand, as shown in Figure 12 Some signs like ‘F’, the fourth sign in the figure from the left can represent the number of open fingers by counting the convexity defect spaces between the fingers. Each space between two fingers consists of one point which belongs to the hand and has a maximum distance to the convex hull. The number of these points is also helpful to understand the shape of the hand in the hand posture. Then the area computation procedure is followed similarly to the convex hull process. The results of this procedure are as shown in Table 4.

Table 4 The area of the convexity defect for three different signers

[redacted]	[redacted]	[redacted]	[redacted]
a1	301	96	147
a2	95	302	178
a3	168	97	99
a4	261	84	110
a5	186	215	120
b1	269	98	204
b2	271	206	174
b3	258	425	186
b4	155	26	129
b5	184	160	167
c1	502	69	666
c2	673	296	705
c3	521	831	646
c4	568	960	1084
c5	451	861	709
d1	455	773	280
d2	444	853	271
d3	489	928	430
d4	414	398	428
d5	462	67	345

4.2.4 Hand Ratio

There is another good feature which is considered in this study. It is the ratio between the hand shape area and perimeter and the convex hull enclosing it. This ratio is also computed for convexity defects areas. Equations 14 to 16 show these mathematical relationships. Table 5 shows some instances results of these equations.

$$\Re_{CHA} = \frac{Handarea(HA)}{ConvexHullarea(CHA)} \tag{14}$$

$$\Re_{CHP} = \frac{Handperimeter(HP)}{ConvexHullperimeter(CHP)} \tag{15}$$

$$\Re_{CDA} = \frac{Handarea(HA)}{Convexitydefectarea(CDA)} \tag{16}$$

Table 5 The area and perimeter ratio for three different signers

[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
	RCHA	RCHP	RCDA	RCHA	RCHP	RCDA	RCHA	RCHP	RCDA
a1	0.8835	1.10912	7.58471	0.95076	0.92387	19.3125	0.90303	1.06639	9.31292
a2	0.9489	1.05082	18.5789	0.75993	1.22797	3.16556	0.89277	1.08342	8.32584
a3	0.9238	1.08778	12.125	0.94894	0.91672	18.5876	0.94692	1.10753	13.6969
a4	0.9015	1.10664	6.77394	0.94545	1.03795	17.3333	0.92262	1.03857	12.7909
a5	0.8947	1.12827	8.50537	0.90363	0.89444	9.37674	0.91746	1.05259	11.1166
b1	0.9281	1.04499	12.9230	0.94719	0.91619	17.9387	0.91218	1.07521	10.3872
b2	0.9107	1.08086	10.2007	0.90716	1.06602	9.77184	0.92151	1.07038	11.7413
b3	0.9079	1.10396	9.86346	0.84135	0.99758	5.30352	0.91591	1.04879	10.8924

b4	0.9012	1.09604	9.12790	0.98826	0.98780	84.1923	0.93912	1.04453	15.4263
b5	0.9334	1.10411	14.0322	0.92935	1.01941	13.1562	0.92758	1.05463	12.8083
c1	0.9069	1.09728	9.74456	0.96640	1.20103	28.7681	0.58106	1.31232	1.53903
c2	0.9061	1.12362	9.65853	0.88355	1.15129	7.58783	0.61049	1.50689	1.56737
c3	0.7167	1.33934	2.52988	0.53340	1.36817	1.14320	0.56401	1.33469	1.47987
c4	0.6052	1.51644	1.53343	0.44985	1.73090	0.81770	0.49768	1.54714	0.99077
c5	0.7960	1.25036	3.90403	0.56646	1.78902	1.30662	0.55408	1.51615	1.24259
d1	0.7073	1.37722	2.41725	0.57550	1.71327	1.35575	0.79135	1.09585	3.79285
d2	0.7131	1.38240	2.74501	0.53991	2.13601	1.17350	0.69044	1.11127	4.21402
d3	0.6041	1.44011	1.52631	0.53249	1.24327	1.13900	0.73828	1.09618	2.82093
d4	0.6954	1.33846	2.28351	0.71991	1.10311	2.57035	0.72864	1.11911	2.92990
d5	0.7385	1.18394	2.82432	0.93631	1.68617	14.7014	0.79634	1.13338	3.91014

4.2.5 Distance Features

The approximate longest diameter of the hand can be calculated via the Eigen value and Eigen vector concepts. In addition, the approximate width is also computable by drawing a line perpendicular to this line. Figure 13 presents some selected results of the above procedure for signer 'A' in both states with and without hand contour. The results have been zoomed till 200% for more clarity.

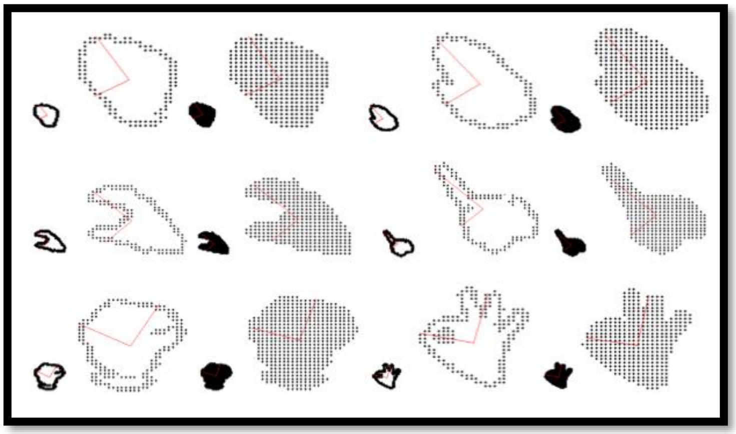


Figure 13 Eigen vectors of the hand

As can be observed in the results in Figure 13, the vectors are drawn from hand contour to hand centre. If they are continued to the opposite points, the approximate length and width of the hand can be computed easily.

After trying all signs on the dataset, it was observed that this procedure cannot lead to a good result in some signs, as shown in Figure 14, so there is a complementary idea which explained in the following subsection to solve this issue.

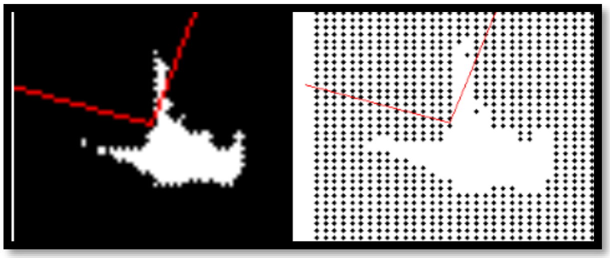


Figure 14 Bad results of the Length and width calculation

4.3 Classification

4.3.1 Discussion on DGSLR Dataset

In the last commands on the SVM, the average accuracy for trained and test sets was calculated. The experimental results were computed for extracted features lonely and also the combination of them. In addition, the program was repeated in 1 and 10 iterations for 5-Fold and 10-Fold cross validation. Tables 6 and 7 show the accuracy rate in 5-Fold cross validation in the training phase and final accuracy of the testing phase for one iteration in the DGSLR dataset. The DGSLR dataset consists of three users with 390 depth-based images. As it can be seen in Tables 6 and 7 the accuracy rate is increased considerably when the features are combined. For instance, when the convexity defect is a feature lonely, the trained accuracy rate is 23.88%. This rate in the validation phase equals to 21.88%. While, the combination of this feature with other features affects highly the recognition rate, as it reaches 80.64% in the training phase and 80.81% in the testing phase.

Table 6 Accuracy of single extracted features from the DGSLR dataset

HA+HP (Hand shape)	65.46%	63.56%
CHA+CHP (Convex Hull)	75.43%	69.45%
CDA (Convexity Defect)	23.88%	21.81%
R_{CHA}+R_{CHP} (Convex Hull Ratio)	79.37%	76.65%
R_{CDA} (Convexity Defect Ratio)	29.40%	26.67%
D (Euclidian Distance)	45.63%	44.72%

Table 7 Accuracy of combination of extracted features from the DGSLR dataset

HA+HP+CHA	75.52%	75.83%
HA+HP+CHA+CHP	79.41%	79.45%
HA+HP+CHA+CHP+CDA	80.64%	80.81%
HA+HP+CHA+CHP+CDA+R_{CHA}	86.33%	86.67%
HA+HP+CHA+CHP+CDA+$R_{CHA+RCHP}$	89.36%	89.65%
HA+HP+CHA+CHP+CDA+R_{CHA}+R_{CHP}+R_{CDA}	90.50%	90.83%
R_{CHA}+R_{CHP}+R_{CDA}	79.32%	79.67%

D+HA+HP+CHA+CHP	89.65%	91.32%
D+HA+HP+CDA	87.56%	86.24%
D+ R_{CHA}+R_{CHP}+R_{CDA}	89.78%	89.58%

The following figures represent the results as line charts for more clarity. It can be seen that the training and testing phase have very close results in both single and specially combined features. Refer to Figure 15, the convex hull feature has the most impact on the accuracy. The accuracy rate in two points which is related to the convex hull, (*CHA*+*CHP*) and (R_{CHA}+R_{CHP}), is close to 80%. This value for the distance feature is approximately 50%, and it shows that the distance feature is an important feature in this case.

The overall accuracy rate for a single feature vector in the training phase is 53.195%. This magnitude in the testing phase is 50.48%. Likewise, the recognition accuracy in training and testing phases are 84.807% and 85.005% respectively. Figure 15 shows the overall results in the DGSLR dataset.

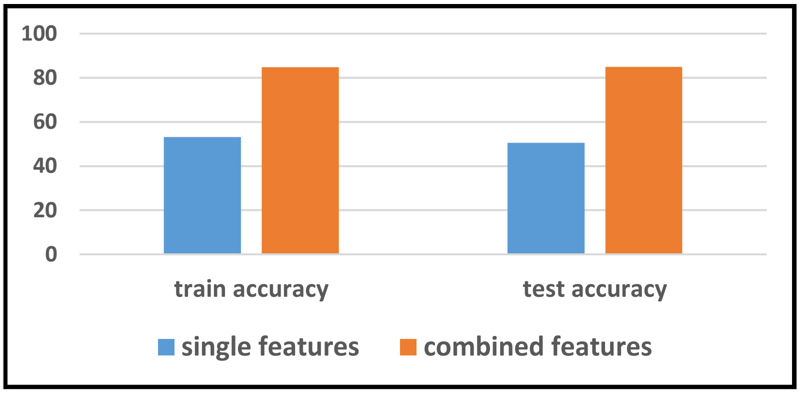
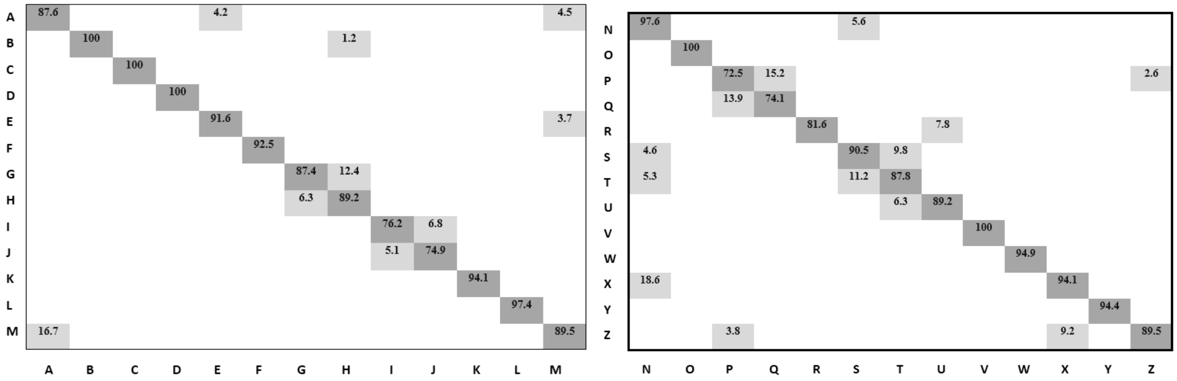


Figure 15 Accuracy rate in a single and combined feature vector

Figures 16 shows the confusion matrix of 26 signs in the DGSLR dataset by three users. As it is observed those signs which are similar have some recognition error and cannot be detected 100% in all cases. For example, sign ‘M’ has been predicted correctly with an 89.5% rate and predicted as ‘A’ sign in 16.7% prediction rate.



	A	B	C	D	E	F	G	H	I	J	K	L	M
A	87.6				4.2								4.5
B		100						1.2					
C			100										
D				100									
E					91.6								3.7
F						92.5							
G							87.4	12.4					
H							6.3	89.2					
I									76.2	6.8			
J									5.1	74.9			
K											94.1		
L												97.4	
M	16.7												89.5

	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
N	97.6					5.6							
O		100											
P			72.5	15.2									2.6
Q			13.9	74.1									
R					81.6			7.8					
S	4.6					90.5	9.8						
T	5.3					11.2	87.8						
U							6.3	89.2					
V									100				
W										94.9			
X	18.6										94.1		
Y												94.4	
Z			3.8								9.2		89.5

Figure 16 (left) Confusion matrix, signs A-M, DGSLR dataset with three users,(right)Confusion matrix, signs N-Z, DGSLR dataset with three users

In sign ‘T’, the sign has been predicted correctly in the 87.8% of cases but it has been detected as ‘N’ and ’S’ in 5.3% and 11.2% of tested cases respectively. Likewise, some signs like ‘B’ and ‘V’ were predicted correctly in all cases. Consequently, the overall recognition rate equals 90.250% which is an acceptable rate considering the previous works in this field study.

Totally for one and ten iterations in 5-Fold and 10-Fold cross validation in this case study of multiclass RBF SVM, the average accuracy rate in the training and testing phases were according to the presented charts as Figures 17.

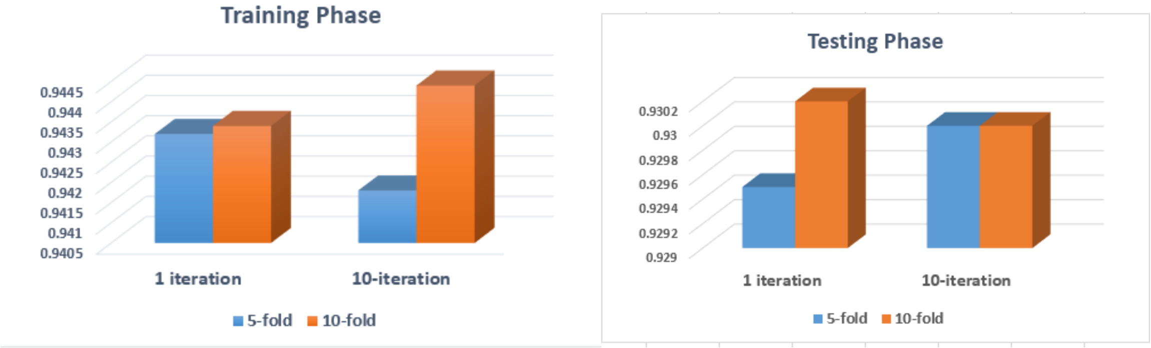


Figure 17 (left)Training phase accuracy rate,(right) Testing phase accuracy rate

4.3.2 Discussion on Standard Dataset

The multi-class SVM classifier was also applied on the standard dataset and the obtained results are as follows. The employed standard dataset includes a huge set of depth-based images of nine users in approximately 400 repetitions on each sign, so includes about 10400 images for each user. Here, just one user has been considered. As can be seen in Table 8, 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 the testing phase. The second most value is related to the ratio between convex hull and hand. It is similar to DGSLR dataset results. Table 9 shows the extracted features combination where the highest value of accuracy rate belongs to a combination of distance, hand and the convex hull of the hand.

Table 8 Accuracy of single extracted features from the standard dataset

HA+HP (Hand shape)	35.22%	32.02%
CHA+CHP (Convex Hull)	58.99%	59.65%
CDA (Convexity Defect)	33.12%	29.96%
R_{CHA}+R_{CHP} (Convex Hull Ratio)	49.57%	44.17%
R_{CDA} (Convexity Defect Ratio)	23.29%	20.93%
D (Euclidian Distance)	45.18%	46.48%

Table 9 Accuracy of combination of extracted features from the standard dataset

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+R_{CHA}	89.43%	90.24%
HA+HP+CHA+CHP+CDA+$R_{CHA+RCHP}$	89.85%	89.31%
HA+HP+CHA+CHP+CDA+$R_{CHA+}R_{CHP+}R_{CDA}$	92.50%	93.43%
$R_{CHA+}R_{CHP+}R_{CDA}$	89.39%	89.77%
D+HA+HP+CHA+CHP	93.64%	96.85%
D+HA+HP+CDA	87.32%	89.14%
D+ $R_{CHA+}R_{CHP+}R_{CDA}$	89.71%	91.54%

The following charts represent the results of the recognition accuracy rate in single and combined features to represent the recognition trend on the standard dataset. Similar to the DGSLR results, Figure 17(left) compared with Figure 17(right) has a higher accuracy rate. Furthermore, the trend of the combined features is increasingly upward.

A recognition accuracy comparison between the proposed method and previous works which used the Kinect sensor has been presented in Table 10. According to the table, Random Occupancy Pattern and Eigen joints demonstrated a high accuracy rate between the other examined classifiers in the recognition process. Some applicable classifiers based on histograms have also illustrated the positive results on recognition. Moreover, the graph based classifiers have an accuracy rate of more than 70%. Whereas neural network based classifiers are widely used in most of the recognition processes, but compared with the other classifiers, they have a low accuracy rate. The hidden Markov Model has shown a high accuracy recognition in Sign Language applications as also discussed in the literature. The recognition accuracy rate of this research is based on SVM and examined on DGSLR and standard datasets. As Table 10 shows, the recognition rate of the classifier is more than 90% on DGSLR dataset and 96% on the standard dataset which is a good result compared to the previous research.

Table 10 Recognition accuracy comparison

Recurrent neural network (Han *et al.*, 2013)	0.425
Dynamic temporal warping (Hossny *et al.*, 2012)	0.540
Hidden Markov Model (Caon *et al.*, 2011)	0.900
Action graph on bag of 3D points (Anand *et al.*, 2013)	0.847
Histogram of 3D joints (Rafibakhsh *et al.*, 2012)	0.789
Random occupancy pattern (Luber *et al.*, 2011)	0.862

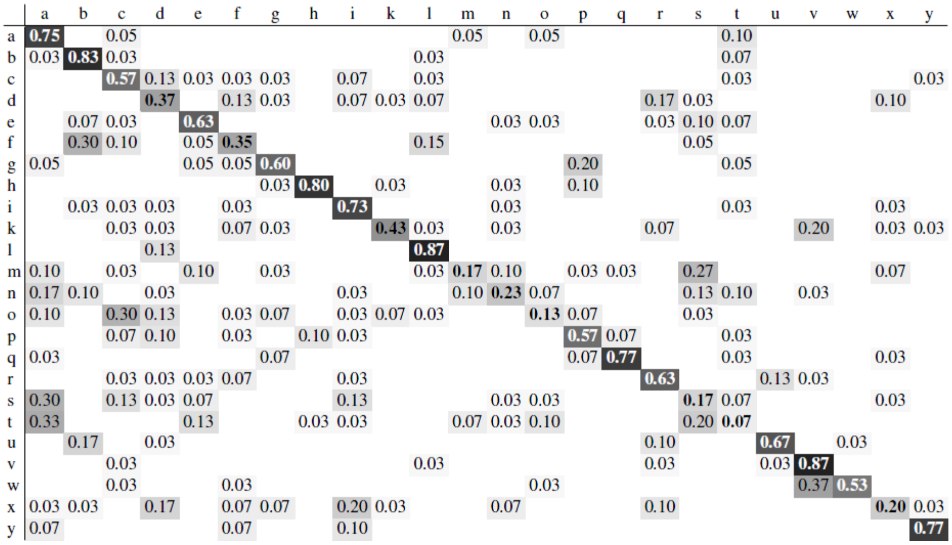
Eigen joints (Machado and Ferreira, 2013)	0.823
Sequence of most informative joints (Maimone and Fuchs, 2012)	0.471
Proposed method on DGSLR dataset	0.903
Proposed method on standard dataset	0.968

4.3.3 Discussion and Comparison on Benchmark

The experimental results of this research are according to previous principal research that used the mentioned standard dataset. Here there is a quick look at this research and then some comparisons between this study and the main research are conducted in obtained practical results as tabular form.

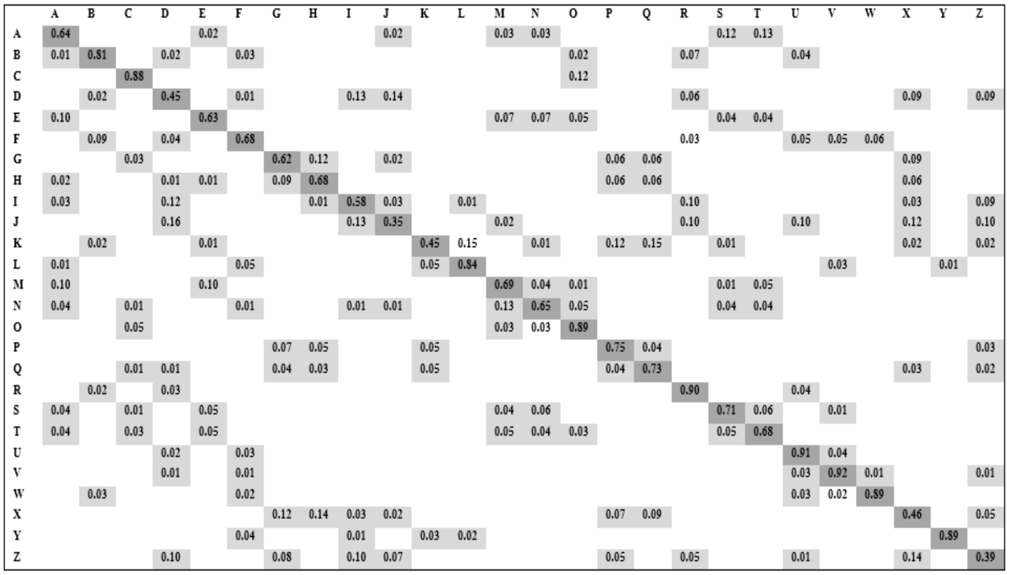
In the principal research which this study built on it, the depth-based detection of the user's hand has been performed using the OpenNI+NITE (Middleware, OpenNI) framework on a Kinect. This library provides functions for detecting hands in 3D space by the depth image made by the Kinect sensor. Then, the hand is segmented from the depth-based image assuming that the hand is a continuous region. For the feature extraction step, the hand shape features used were based on Gabor filtering of the depth images and intensity. The learning and classification process is well established and utilized via a multi-class random forest, discussed in detail earlier. The random forest has good accuracy in learning (Daugman, 1985) and can handle large feature space and large datasets. It has shown some good results in fast training. The flow work of this research is presented as follows.

Figures 18 and 19 show the confusion matrix for the detection of all signs in the mentioned research and this research, using a combined feature vector.



	a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y
a	0.75		0.05									0.05		0.05					0.10					
b	0.03	0.83	0.03								0.03								0.07					
c			0.57	0.13	0.03	0.03	0.03		0.07		0.03								0.03					0.03
d				0.37		0.13	0.03		0.07	0.03	0.07						0.17	0.03					0.10	
e		0.07	0.03		0.63								0.03	0.03			0.03	0.10	0.07					
f		0.30	0.10		0.05	0.35					0.15							0.05						
g	0.05				0.05	0.05	0.60								0.20				0.05					
h							0.03	0.80		0.03			0.03		0.10									
i		0.03	0.03	0.03		0.03			0.73				0.03						0.03				0.03	
k			0.03	0.03		0.07	0.03			0.43	0.03		0.03				0.07				0.20		0.03	0.03
l				0.13							0.87													
m	0.10		0.03		0.10		0.03				0.03	0.17	0.10		0.03	0.03		0.27					0.07	
n	0.17	0.10		0.03					0.03			0.10	0.23	0.07				0.13	0.10		0.03			
o	0.10		0.30	0.13		0.03	0.07		0.03	0.07	0.03			0.13	0.07			0.03						
p			0.07	0.10		0.03		0.10	0.03						0.57	0.07			0.03					
q	0.03						0.07								0.07	0.77			0.03				0.03	
r			0.03	0.03	0.03	0.07			0.03								0.63			0.13	0.03			
s	0.30		0.13	0.03	0.07				0.13				0.03	0.03				0.17	0.07				0.03	
t	0.33				0.13			0.03	0.03			0.07	0.03	0.10				0.20	0.07					
u		0.17		0.03													0.10			0.67		0.03		
v			0.03								0.03						0.03			0.03	0.87			
w			0.03			0.03								0.03							0.37	0.53		
x	0.03	0.03		0.17		0.07	0.07		0.20	0.03			0.07				0.10						0.20	0.03
y	0.07					0.07			0.10															0.77

Figure 18 Confusion matrix of all signs in the dataset in benchmark research



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	0.64				0.02					0.02			0.03	0.03					0.12	0.13						
B	0.01	0.81		0.02		0.03									0.02			0.07			0.04					
C			0.88												0.12											
D		0.02		0.45		0.01			0.13	0.14								0.06						0.09		0.09
E	0.10				0.63								0.07	0.07	0.05				0.04	0.04						
F		0.09		0.04		0.68												0.03			0.05	0.05	0.06			
G			0.03				0.62	0.12		0.02						0.06	0.06							0.09		
H	0.02			0.01	0.01		0.09	0.68								0.06	0.06							0.06		
I	0.03			0.12				0.01	0.58	0.03		0.01						0.10						0.03		0.09
J				0.16					0.13	0.35			0.02					0.10			0.10			0.12		0.10
K		0.02			0.01						0.45	0.15		0.01		0.12	0.15		0.01					0.02		0.02
L	0.01					0.05					0.05	0.84										0.03			0.01	
M	0.10				0.10								0.69	0.04	0.01				0.01	0.05						
N	0.04		0.01			0.01			0.01	0.01			0.13	0.65	0.05				0.04	0.04						
O			0.05										0.03	0.03	0.89											
P							0.07	0.05			0.05					0.75	0.04									0.03
Q			0.01	0.01			0.04	0.03			0.05					0.04	0.73							0.03		0.02
R		0.02		0.03														0.90			0.04					
S	0.04		0.01		0.05								0.04	0.06					0.71	0.06		0.01				
T	0.04		0.03		0.05								0.05	0.04	0.03				0.05	0.68						
U				0.02		0.03															0.91	0.04				
V				0.01		0.01															0.03	0.92	0.01			0.01
W		0.03				0.02															0.03	0.02	0.89			
X							0.12	0.14	0.03	0.02						0.07	0.09							0.46		0.05
Y						0.04			0.01		0.03	0.02													0.89	
Z				0.10			0.08		0.10	0.07						0.05		0.05			0.01			0.14		0.39

Figure 19 Confusion matrix of all signs in the standard dataset in the proposed research

Considering both above confusion matrixes, it can be found that some signs like ‘A’, ‘B’, ‘M’, ‘N’, ‘S’, and ‘T’ which have similar posture, the recognition rates are close together. For example, the recognition rate for ‘A’ sign equals to 0.64 (64%) and for ‘E’ is 0.63. These rates in the benchmark results are 0.75 and 0.63. Similar signs can be wrongly detected. This wrong detection occurs in ‘Y’ and ‘L’ or ‘F’ and ‘W’. In addition, the prediction error has similar results, for example, the ‘A’ sign is detected wrongly as ‘M’ with 0.05 rate of prediction in the principal study. This rate in this proposed research is 0.03. The ‘O’ sign is predicted as ‘C’ with 0.3 rates, while this rate in this research is 0.05. On the other hand, some recognition rates have been improved while some of them, vice versa. But with an overall look at both figures it can be realized that most of the rates have been improved in the proposed research. One another considerable note is about two signs ‘J’ and “Z’. These signs are motional and have movement while signing. Since this research is related to the study of images, so having a look at the figures, can be found these two signs have a low recognition rate. The benchmark research removed these signs from its field, and this research got an average of the different poses of the sign. It means that, while the signer doing the sign, the images were captured one by one, and then calculate the average of geometrical features of them.

Consequently, in the benchmark research, the best results were obtained for two signs ‘L’ and ‘V’ with 0.87 prediction rate and the lowest rates for ‘O’ sign with 0.13 and ‘S’ and ’m’ with 0.17. These rates in the proposed research are 0.35 and 0.39 for ‘J’ and ‘Z’ respectively. It means that two motional signs have the lowest recognition rates between all the performed signs. The overall recognition rate in the benchmark research is 52.95% while this rate equals to 66.07% in the proposed research. As mentioned before this rate is 90.25 in the DGSLR dataset in this research with three users, and 96.85 on the standard dataset.

4.3.4 Discussion and Comparison based on Different Classifiers

The results of two common classifiers, K-Nearest Neighbours (K-NN) and Decision Tree (DT) have been represented and compared to SVM. For obtaining the better result, the signs were divided into some categories which five signs in each category. The categories are as A to E, F to J, …, U to Z, by labeling 1 to 5, 6 to 10,…, 21 to 26 respectively. Firstly, the results of SVM with 5 and 10-fold cross validation are presented as shown in Table 11, 12.

Table 12 shows the accuracy rate of recognition methods for training and test phases in each class of SVM in 5-fold cross validation. The average of accuracy in each class is also presented. Eventually, the final accuracy in train and test is illustrated.

Table 11 SVM by 5-Fold cross validation in training phase

Train accuracy rate in each class	0.9429 0.9143 0.8857 0.9143 1.0000	0.9706 0.7941 0.9706	0.8235 0.9706	1.0000 0.8182 0.9697	0.9091 0.9394	1.0000 0.9697 1.0000	0.9697 1.0000	1.0000 0.9063 1.0000 0.9074	0.9375 0.9063
Test accuracy rate in each class	0.8194 0.9167 0.8333 0.7778 0.9861	0.8333 0.8056 0.9722	0.9722 0.7639	0.8333 0.7917 0.9722	0.8889 0.7500	0.8333 0.8611 1.0000	0.9722 0.7917	0.8333 0.8472 0.9861 0.8903	0.9444 0.8056
Mean accuracy of Train	0.9429	0.9216		0.9394		0.9899		0.9479	
Mean accuracy of Test	0.8889	0.8796		0.8472		0.9306		0.9444	
Total accuracy	0.94834 , 0.89814								

Table 12 shows the accuracy rate of recognition methods for training and testing phases in each class of SVM in 10-fold cross validation. The average accuracy in each class is also presented. Eventually, the final accuracy in train and test is presented.

Table 12 SVM by 10-Fold cross validation in training phase

Train accuracy rate in each class	1.0000 0.8824 0.8824 1.0000 1.0000	0.9412 0.9412 1.0000	0.9412 0.8824	1.0000 0.8750 1.0000	0.8750 1.0000	1.0000 0.8333 1.0000	0.8889 0.8889	1.0000 0.9444 1.0000 1.0000	0.9444 0.9444
Test accuracy rate in each class	0.8333 0.9722 0.8333 0.7778 1.0000	0.8333 0.8472 1.0000	0.9583 0.8056	0.8194 0.8472 0.9722	0.8750 0.8056	0.8333 0.8472 1.0000	0.9167 0.7917	0.8333 0.8611 1.0000 0.9812	0.9722 0.7917
Mean accuracy of Train	1.0000	0.9412		1.0000		0.9444		0.9630	
Mean accuracy of Test	0.9444	0.9444		0.8750		0.8889		0.9306	
Total accuracy	0.96972 , 0.91666								

Comparing two Tables 11 and 12 shows that the accuracy rate of SVM in 10-Fold cross validation is higher than the 5-Fold.

Table 13 represents the accuracy rate of recognition by K-NN classifier which k equals to 10. Two last iterations in each class have been presented as a sample. For example, in the third class of 11 to 15 labels, related to ‘K’ to ‘O’ signs, the sign ‘k’ with label 11 as input and the classifier predicts it as ‘O’ with a label of 15. In the next iteration, it is predicted as ‘N’ with the label of 14. Whereas, in this class, two signs ‘L’ and ‘M’ with 12 and 13 labels, are predicted correctly in accordance to input. The total accuracy rate is roughly 85% which is less than the SVM classifier.

Table 13 K-NN accuracy recognition, K=10

	input	predict	input	predict	input	predict	input	predict	input	predict
Sample	1	1	6	7	11	15	16	16	21	21
	2	2	7	7	12	12	17	18	22	22
	3	3	8	8	13	13	18	20	23	21
	4	4	9	9	14	13	19	20	24	24
	5	5	10	10	15	15	20	19	25	25
	1	1	6	8	11	14	16	17	26	26
	2	2	7	8	12	12	17	18	21	24
	3	4	8	8	13	13	18	20	22	22
	4	4	9	9	14	14	19	19	23	21
	5	5	10	10	15	13	20	20	24	24
									25	25
									26	26
Test accuracy rate in each class	1.0000 0.7833 0.9333 0.7833 0.7000		0.9000 0.7833 1.0000	0.8167 0.9333	0.9167 0.9000 0.7167	1.0000 0.7667	0.8667 0.8167 0.6833	0.7667 0.8000	0.7778 0.8611 1.0000 0.9028	0.9306 0.8333
Mean accuracy of Test	0.8400		0.8867		0.8600		0.7867		0.8843	
Total accuracy	0.85154									

Table 14 presents the accuracy rate of recognition by K-NN classifier which k equals to 20. Two last iterations in each class have been presented as a sample. For example, in the second class of 6 to 10 labels, where are related to ‘F’ to ‘J’ signs, sign ‘F’ with label 6 is the input and the classifier predicts it as ‘H’ with the label of 8. In the next iteration, it is predicted the same. Whereas in this class, the sign ‘G’ with label 7, is predicted correctly in accordance to input in the second iteration, meanwhile it is predicted as ‘J’ in the first iteration. The total accuracy rate is more than 84% which is less than the SVM classifier.

Table 14 K-NN accuracy recognition, K=20

	input	predict	input	predict	input	predict	input	predict	input	predict
Sample	1	1	6	8	11	14	16	16	21	21
	2	2	7	10	12	12	17	18	22	22
	3	3	8	7	13	13	18	20	23	21
	4	3	9	9	14	14	19	20	24	24
	5	5	10	10	15	15	20	19	25	25
	1	1	6	8	11	14	16	17	26	26
	2	2	7	7	12	12	17	18	21	24
	3	4	8	8	13	13	18	20	22	22
	4	3	9	9	14	13	19	19	23	21
	5	5	10	10	15	13	20	19	24	24
									25	25
									26	26

Test accuracy rate in each class	1.0000 0.8667 0.8333 0.6833 0.7833		0.8833 0.8000 0.9833	0.7833 0.8167	0.9167 0.8667 0.7667	1.0000 0.7500	0.8833 0.8333 0.6333	0.7500 0.7667	0.7917 0.8889 1.0000 0.9167	0.9583 0.8333
Mean accuracy of Test	0.8333		0.8533		0.8600		0.7733		0.8981	
Total accuracy	0.8436									

Table 15 presents the DT results as the next classifier. The total accuracy rate of recognition is about 81%. Which is less than the K-NN and SVM, but because of its simple structure it is widely used in the classification goals.

Table 15 DT accuracy recognition

	input	predict	input	predict	input	predict	input	predict	input	predict
Sample	1	1	6	8	11	13	16	16	21	21
	2	2	7	6	12	12	17	18	22	22
	3	3	8	7	13	13	18	18	23	21
	4	3	9	9	14	13	19	19	24	25
	5	4	10	10	15	14	20	19	25	25
	1	1	6	8	11	11	16	16	26	26
	2	2	7	6	12	12	17	18	21	25
	3	3	8	7	13	13	18	19	22	22
	4	3	9	6	14	15	19	19	23	21
	5	4	10	10	15	13	20	19	24	25
									25	25
									26	25
Test accuracy rate in each class	1.0000 0.8333 0.7833 0.7333 0.8500		0.7500 0.6833 0.9167	0.6667 0.8167	0.9333 0.7833 0.7333	1.0000 0.7833	0.8333 0.7333 0.8000	0.7667 0.6667	0.8194 0.8750 0.8750 0.8750	0.7917 0.7917
Mean accuracy of Test	0.8400		0.7667		0.8467		0.7600		0.8380	
total	0.81028									

Figure 20 shows the results of the recognition rates for three classifiers. It is clear that the SVM classifier has the most accuracy rate compared with K-NN and DT classifiers. It is surprising that the K-NN with K=10 has a higher accuracy rate than the K-NN with K=20, which is an unexpected result. In the end, the DT classifier has the least recognition rate between two other classifiers.

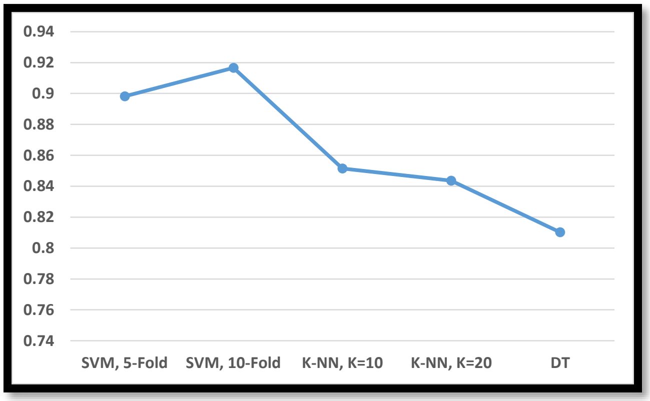


Figure 20 Comparison between recognition accuracy rate of SVM, K-NN, and DT

Lastly, a comparison between the research and its benchmark is represented in Table 16. We utilized Matlab and the LIBSVM library for the development of the algorithms. The benchmarking process used the OpenNI and NITE libraries. The segmentation, feature extraction, and classification phases have been implemented differently, but both types of research used multiclass classification due to the number of the sign language alphabets.

Table 16 Method Comparison

Devices	typical laptop, Intel Core i5 2430M processor at 2.40GHz Kinect camera	typical laptop, Intel Core i5 M430 processor at 2.27GHz Kinect camera
Software	Matlab+LIBSVM+SDK	OpenNI+NITE framework
Hand Segmentation	Programing in Matlab	OpenNI+NITE predefine functions
Feature Extraction	Hand geometrical features	Hand shape features based on Gabor filtering
Classification	Multiclass SVM	Multiclass random forest
Dataset	Centre for Vision, Speech and Signal Processing, University of Surrey	Centre for Vision, Speech and Signal Processing, University of Surrey

5. CONCLUSION

We aimed to examine the accuracy of the proposed hand recognition technique on both DGSLR and standard datasets which contain a number of samples of the American Sign Language alphabet. The effectiveness of the proposed techniques was first evaluated on the DGSLR dataset by three users and acceptable recognition rates were obtained. Later, the evaluation approaches were carried out on the standard dataset and achieved considerable results which were very promising. Besides experimental results, different tabular analyses and discussions of the charts are also reported. Finally, a comparison discussion between the benchmark research and the proposed research with their final results are investigated. Furthermore, two classifiers, K-NN and DT are employed and the obtained results are compared to as an SVM classifier.

Since there are 26 different signs in the Sign Language alphabet, a single multi-class versus a single SVM classifier with 26 classes by an RBF kernel was used to validate each class. The accuracy and accuracy of the proposed method were evaluated and the procedure was repeated by changing each parameter (C, σ) for the validation. The selected pair gave the best average accuracy from the group. Then, the SVM was trained on the selected training set with these optimal parameters. This method is also used to perform the recognition process by utilizing multiple feature descriptors which is multiple feature descriptors which is a combination of the extracted features. Experiments were conducted on the selected and standard datasets.

The combination of the extracted features reveals the superiority of the proposed method over the existing work on this subject. The selected dataset was used by three different users, which two users were novices in Sign Language. Each sign was repeated five times for getting improved accuracy. The standard dataset has more than 400 repetitions in each sign. The process was well done in 1 and 10 passes for all data in the dataset in 5 and 10-Fold cross validation. The confusion matrix is used in the proposed machine learning process which permits visualization of the algorithm efficiency.

The significant finding of this research is the realization of the significant improvements in Sign Language recognition accuracy. Combined features give better results than a single feature. The distance feature has a major contribution on the recognition rate. Evaluations on the selected dataset report the recognition rate of 90.25% while this magnitude on the complete standard dataset using the proposed approaches, reports an identification rate of 96.85%, the best overall identification rate reported so far on the considered dataset. According to the confusion matrix visualization obtained from benchmarking and the proposed research, in specific cases, alternative techniques and combinations of machine learning algorithms provide higher accuracy of Sign Language recognition. This work is to create a generalized Sign Recognition process. Our research and proposed machine learning process for the creation of a generalized Sign Language Recognition system capable of being used in cluttered, varied lit environments, has given improved results from previous research utilizing the chosen dataset.

In this research, geometric features along with some new features such as hand key-points for estimating and tracking have been employed. This has been done to detect multi-frame videos of our gestures by Deep Neural Network. Feature Learning and deep neural network are too time-consuming and overfitting, therefore, there are rooms to take them into account for future work. However, this has been done to detect multi-frame videos of our gestures by deep neural network.

Conflict of Interest

We confirm that there is no conflict of interest for this paper. All work is original.

References:

Amin, M. A. and Yan, H. (2007). Sign language finger alphabet recognition from Gabor-PCA representation of hand gestures. *Machine Learning and Cybernetics, 2007 International Conference on*, 2007. IEEE, 2218-2223.

Anand, A., Koppula, H. S., Joachims, T. and Saxena, A. (2013). Contextually guided semantic labeling and search for three-dimensional point clouds. *Int J Robot Res*, 32(1)**,** 19-34.

Batenburg, K. J. and Sijbers, J. (2009). Adaptive thresholding of tomograms by projection distance minimization. *Pattern Recognition*, 42, 2297-2305.

Caon, M., Yue, Y., Tscherrig, J., Mugellini, E. and 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., 7-12.

Cawley, G. C. and Talbot, N. L. (2007). Preventing over-fitting during model selection via Bayesian regularisation of the hyper-parameters. *The Journal of Machine Learning Research,* 8**,** 841-861.

Chai, X., Li, G., Lin, Y., Xu, Z., Tang, Y., Chen, X. and Zhou, M. (2013). Sign Language Recognition and Translation with Kinect.

Chen, L., Lin, H. and Li, S. (2012). Depth image enhancement for kinect using region growing and

Dardas, N. H. and Georganas, N. D. (2011). Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques. *Instrumentation and Measurement, IEEE Transactions on,* 60**,** 3592-3607.

Daugman, J. (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by twodimensional visual cortical filters. *Journal of the Optical Society of America A,* 2(7)**,** 1160-1169.

Dominio, F., Marin, G., Piazza, M. and Zanuttigh, P. (2014). Feature Descriptors for Depth-Based Hand Gesture Recognition. *Computer Vision and Machine Learning with RGB-D Sensors.* Springer International Publishing Switzerland: Springer. 215-237.

Dominio, F., Marin, G., Piazza, M. and Zanuttigh, P. (2014). Feature Descriptors for Depth-Based Hand Gesture Recognition. *Computer Vision and Machine Learning with RGB-D Sensors.* Springer International Publishing Switzerland: Springer. 215-237.

Dong, C., Leu, M. and Yin, Z. (2015). American Sign Language Alphabet Recognition Using Microsoft Kinect. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015. 44-52.

Gonzalez, R. C., Woods, R. E. and S. L. Eddins, S. L. (2009). Digital Image Processing Using MATLAB. Gatesmark Publishing.

Han, J., Shao, L., Xu, D. and Shotton, J. (2013). Enhanced computer vision with microsoft kinect sensor: a review.

Hossny, M., Filippidis, D., Abdelrahman, W., Zhou, H., Fielding, M., Mullins, J., Wei, L., Creighton, D., Puri, V. and Nahavandi, S. (2012). Low cost multimodal facial recognition via kinect sensors. *Proceedings of the land warfare conference (LWC): potent land force for a joint maritime strategy*, 2012 Commonwealth of Australia. 77-86.

Jin, C. and Wang, L. (Year). Dimensionality dependent PAC-Bayes margin bound. *Advances in Neural Information Processing Systems*, 2012. 1034-1042.

Kiseľák, J., Lu, Y., Švihra, J., Szépe, P., & Stehlík, M. (2020). “SPOCU”: scaled polynomial constant unit activation function. *Neural Computing and Applications*, 1-17.

Keskin, C., Furkan, K., Kara, Y. and Akarun, L. (Year). Hand pose estimation and hand shape classification using multi-layered randomized decision forests. *Proceedings of the European conference on computer vision (ECCV)*, 2012. 852-863.

Kishore, P. and Kumar, P. R. (2012a). Segment, Track, Extract, Recognize and Convert Sign Language Videos to Voice/Text. *International Journal,* 3.

Lee, H.-C., Shih, C.-Y. and Lin, T.-M. (2013). Computer-Vision Based Hand Gesture Recognition and Its Application in Iphone. *Advances in Intelligent Systems and Applications, Springer,* 2.

Liang, H., Yuan, J. and Thalmann, D. (2014). Parsing the hand in depth images. *Multimedia, IEEE Transactions on,* 16**,** 1241-1253.

Luber, M., Spinello, L. and Arras, K. O. (2011). People tracking in RGBD-D data with on-line boosted target models. *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 2011 of Conference.: IEEE, 3844-3849.

Machado, J. and Ferreira, A. (2013). Retrieval of objects captured with low-cost depth-sensing cameras. *SHREC2013.Springer*.

Maimone, A. and Fuchs, H. (2012). Reducing interference between multiple structured light depth sensors using motion. *Virtual reality workshops (VR)*, 2012 of Conference.: IEEE, 51-54.

Mihalache, C. R. and Apostol, B. (Year). Hand pose estimation using HOG features from RGB-D data. *System Theory, Control and Computing (ICSTCC), 2013 17th International Conference*, 2013. IEEE, 356-361.

Nölker, C. and Ritter, H. (1998). Detection of fingertips in human hand movement sequences. *Gesture and Sign Language in Human-Computer Interaction.* Springer. 209-218.

Nölker, C. and Ritter, H. (1999). GREFIT: Visual recognition of hand postures. *Gesture-Based Communication in Human-Computer Interaction.* Springer. 61-72.

Oikonomidis, I., Kyriazis, N. and Argyros, A. A. (Year). Efficient model-based 3d tracking of hand articulations using kinect. *Proceedings of the 22nd British machine vision conference (BMVC),* 2011.

Pedersoli, F., Adami, N., Benini, S. and Leonardi, R. (Year). Xkin-extendable hand pose and gesture recognition library for kinect. *Proceedings of ACMconference on multimedia 2012-open source competition*, October 2012 Nara, Japan.

Prasad, M. V. D., Raghava, P. C., Rahul, R. and V, K. P. V. (2015). 4-Camera Model for Sign Language Recognition Using Elliptical Fourier Descriptors and ANN. SPACES-2015, Dept of ECE, K L UNIVERSITY.

Pugeault, N. and Bowden, R. (2011). Spelling it out: real-time asl fingerspelling recognition. *Proceedings of the 1st IEEE workshop on consumer depth cameras for computer vision*, 2011. 1114-1119

Rafibakhsh, N., Gong, J., Siddiqui, M. K., Gordon, C. and Lee, H. F. (2012). Analysis of xbox kinect sensor data for use on construction sites: depth accuracy and sensor interference assessment. *Constitution research congress*, 2012 of Conference., 848-857.

Sharma, R., Nemani, Y., Kumar, S., Kane, L. and Khanna, P. (Year). Recognition of Single Handed Sign Language Gestures using Contour Tracing Descriptor. *Proceedings of the World Congress on Engineering*, 2013.

Yeo, H.-S., Lee, B.-G. and Lim, H. (2013). Hand tracking and gesture recognition system for human-computer interaction using low-cost hardware. *Multimedia Tools and Applications,* 74**,** 2687-2715.

Zhu, Q.-S., Xie, Y.-Q. and Wang, L. (2010). Video Object Segmentation by Fusion of Spatio-Temporal Information Based on Gaussian Mixture Model. *Bulletin of advanced technology research,* 5.