# A Mobile Application of American Sign Language Translation via Image Processing Algorithms

Cheok Ming Jin, Zaid Omar
Faculty of Electrical Engineering
Universiti Teknologi Malaysia
81310 Skudai, Malaysia

Mohamed Hisham Jaward
School of Engineering
Monash University Malaysia
47500 Bandar Sunway, Malaysia

***Abstract*— Due to the relative lack of pervasive sign language usage within our society, deaf and other verbally-challenged people tend to face difficulty in communicating on a daily basis. Our study thus aims to provide research into a sign language translator applied on the smartphone platform, due to its portability and ease of use. In this paper, a novel framework comprising established image processing techniques is proposed to recognise images of several sign language gestures. More specifically, we initially implement Canny edge detection and seeded region growing to segment the hand gesture from its background. Feature points are then extracted with Speeded Up Robust Features (SURF) algorithm, whose features are derived through Bag of Features (BoF). Support Vector Machine (SVM) is subsequently applied to classify our gesture image dataset; where the trained dataset is used to recognize future sign language gesture inputs. The proposed framework has been successfully implemented on smartphone platforms, and experimental results show that it is able to recognize and translate 16 different American Sign Language gestures with an overall accuracy of 97.13%.**

***Keywords*— Computer Vision, Gesture Recognition, Image Processing, Machine Learning, Sign Language***

## I. INTRODUCTION

Sign language is a form of hand gestures involving visual motions and signs, which are used as a system of communication notably by the deaf and verbally-challenged community. It is seldom however used by normal hearing people and few are able to understand sign language. This poses a genuine communication barrier between the deaf community and the rest of the society, as a problem yet to be fully solved.

Modern research into sign language recognition can be categorised into contact-based and vision-based approaches. Contact-based approaches involves physical interaction between user and sensing devices [1]. It typically uses instrumented glove which utilizes electromyography, inertial measurement or electromagnetic to collect finger flexion, position, orientation or angle data of the sign performed [2]. Vision-based approaches uses data collected from images or video frames captured using camera as input of the system. It can be further divided into 3D model-based and appearance-based approaches. 3D model-based approaches generally attempt to infer the pose of palm and joint angle hand in 3D spatial into 2D projection [3]. Whereas appearance-based uses features extracted from visual appearance of the images and recognition is done by comparing these features [2].

Vision-based approach is often preferred over contact-based approaches as it often does not involve the wearing of instrumental gloves or other hardware besides camera to perform the recognition process. In some vision-based research however, coloured gloves are used to alleviate the hand segmentation process. The major challenges faced by visual based approach is that the accuracy is often affected by noises, lighting condition, variation of viewpoint and the presence of complex background.

Sign language recognition in general involves a few phases of process namely the segmentation, feature extraction and classification [4]. The main objective of the segmentation phase is to remove the background and noises, leaving only the Region of Interest (ROI), which is the only useful information in the image. In the feature extraction phase, the distinctive features of the ROI will be extracted. These features can be the curvatures, edges, shapes, corners, moments, textures, colours or others. In the context of sign language recognition, these features are essentially analogous to the identity of each sign language gesture. Next, the features extracted will undergo classification whereby the features of each gesture will be grouped accordingly, and this will be used as a database to match new sign language gesture inputs to which of the groups classified earlier do they belong.

In previous work, Pansare *et al.* [5] performed Indian Sign Language (ISL) recognition by first applying median and Gaussian filters to remove noises before using morphological operations. Sobel edge detection method is utilised to detect the edges of ROI, which are used to determine the centroid and area of the ISL. Euclidian distance is calculated for classification of 26 ISL with 100 samples each and achieved average accuracy of 90%. Rekha *et al.* [6] proposed recognition of ISL by first segmenting hands region using skin colour detection in YCbCr colour space. ISL features are then extracted using Principle Curvature Based Region (PCBR) detector, Wavelet Packet Decomposition (WPD-2) and complexity defects method. With 23 ISL and 40 training samples each, multi class SVM is used in classification of ISL and able to recognize with 91.3% accuracy. Dardas and Georganas [7] proposed a framework of recognizing hand gesture in real time using Bag-of Feature (BoF) and Support Vector Machine (SVM). Skin colour segmentation method is used to segment the face and hand region from the background in HSV colour space, Viola and Jones algorithm is then used

978-1-5090-0931-2/16/$31.00 ©2016 IEEE

to remove the face region. Shift-Invariant Feature Transform (SIFT) algorithm is employed to extract keypoints from the detected hand region which were first quantized using *K*-means clustering and then mapped into BoF. With six classes of gestures and 100 train images each, the overall accuracy achieved using multi-class SVM classifier is 96.23%. In Arabic Sign language (ArSL) recognition, Tharwat *et al.* [8] first extract SIFT features from sign performed. The authors used Linear Discriminant Analysis (LDA) method to reduce the dimensionality of SIFT features extracted and shown to have reduction in computational time. The performance of classifiers are compared between SVM, *k*-Nearest Neighbour and minimum distance, and it is found that classification using SVM yields the highest accuracy. Using 30 ArSL with 7 train images each, an average accuracy of 99% is obtained. The authors shown that the proposed system are robust against occlusion and rotation.

Recognition of face component and objects using SVM classification on Speeded Up Robust Feature (SURF) keypoints has shown prominent result in [9, 10] respectively. In this paper, our framework proposed the use of SURF features instead of SIFT in American Sign Language (ASL) recognition. Performance comparison between SIFT and SURF done in [11] shows that SURF has a faster processing speed. Similar findings are obtained in [9]. The dimensionality of SURF descriptors extracted can be reduced to improve efficiency by representing descriptors using BoF model [7].

More recent researches on sign language recognition software utilizes additional hardware to improve the performance of sign language recognition. As an example, Chai *et al*. [12] utilized Kinect to obtain colour and depth information which are then used to create a 3D motion trajectories database. Euclidean distance is then calculated between new motion trajectories with those in database for matching. Leap Motion Controller (LMC) are used in [13, 14] to facilitate sign language recognition process. The data generated by LMC namely palm center, fingertip position, hand angle and orientation are able to provide useful features to improve classification result. Nevertheless, LMC has several drawbacks such as finger occlusion blind spot and is unable to detect when fingers are touching with each other.

Despite many works have been done on computer platform, little has been done on smartphone platform. Previous research of sign language translation on smartphone have shown that the limitation in processing power in smartphone is one of its major constraints [15]. However, the benefits of using a smartphone platform, such as its mobility and ease of use, are a great advantage over desktop systems. We therefore propose a framework of established algorithms to show that sign language recognition can be performed with high accuracy in real-time application on smartphone platform, thus aiding sign language users to ease the communication gap.

## II. Methodology

### A. System Overview

In this paper, 16 static ASL alphabets are to be recognized in real-time using Nokia Lumia 1520 smartphone with Windows Phone 8.1 operating system. The algorithm is developed on top of EmguCV library. All images captured are sampled into resolution of 320 x 240 pixels in RGB format. The proposed framework first sample RGB value of skin pixels of the signer for calibration to implement automatic detection of hand gesture.

The focus on this framework is on a smartphone platform, where several known conditions of sign language input are utilized to simplify the overall complexity. Firstly, the hand gestures of signer is always placed in the middle of the image frame in order for the whole hand to fit in. Next, the variation of distance between the hand gesture and the camera are limited to a certain range. Application of an appropriate scale-invariant techniques eliminates the need to normalize size of hand gesture. We utilized these conditions where the sure-foreground and sure-background pixels can be determined, and hence the seeded region growing method is suitable to be used in segmentation stage. Canny edge detection is used to extract the edges of hand gestures for feature extraction phase, which are also utilised as a limiting boundary of seeded region growing to improve segmentation accuracy.

SURF features are extracted from the edge detected image, and are clustered into 16 classes of sign language using *K*-means clustering. The BoF model is applied to create histogram of visual vocabulary from the cluster centroid. A total of 1600 sample images, 100 for each sign languages are used as the training images. Another 800 images of sign language are collected as test images. A multi class SVM is trained and is used to classify the sign language test images. The flowchart of sign language recognition is as shown in Figure 1. Finally, the output of the recognized sign language is displayed as both text and audible speech through text-to-speech library – Microsoft Speech SDK in actual implementation of the system on smartphone.

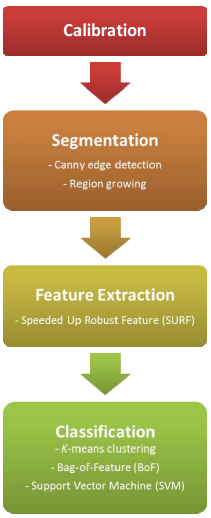



Fig. 1. Flowchart of Sign Language Recognition Framework

### B. Calibration

In order to improve the robustness of the system towards different users and variation in lighting condition, a one-time calibration of skin pixels colour is performed. A red rectangular box is displayed and signer is prompted to capture an image of the hand with the hand larger than the rectangular box, as shown in Figure 2. The maximum and minimum values of all R, G and B channels are then recorded. In online testing application, when the camera detects skin pixels range in the centre of image frame, it will automatically captures the frame. This method however is prone to misclassify if background objects similar to skin colour is present.

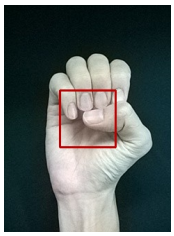


Fig. 2. Rectangular box for calibration

### C. Segmentation

The image is first converted from RGB colour space into single channel grayscale image. Canny edge detection which is a technique to identify and detect the presence of sharp discontinuities in an image, is applied to the image to detect the edges. Next, seeded region growing method is used to segregate the hand region from the background. The region growing method is first used with an initial seed point located on the sure-backgrounds to subtract the background from the image. The above process is iterated three times with three different seed points location namely on the top left corner, top right corner and bottom left corner, shown as red dots in Figure 3. The results of this background subtraction are as shown in Figure 4(b). Next, region growing method is applied with the seed point on the centre of the image, as shown as green dot in Figure 3. It can be observed that the green seed point will always falls on the hand gestures, while red seed points falls on the background.

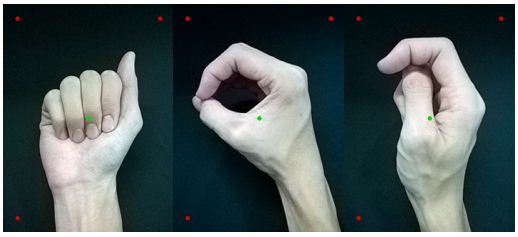
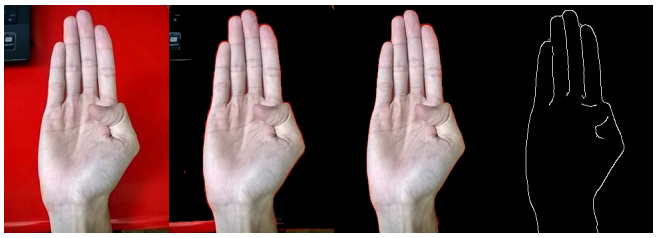


Fig. 3. Seed points used in region growing

In the region growing process whereby the initial seed point is placed in the centre of image, the previous edge detected image is used as a mask which serves to limit the region growing boundary. The mask contains both edges of the hand and the background, whereby the edges detected are the only non-zero (white pixels) and the remaining pixels are all zero (black pixels). A region growing condition is set to grow to the neighbouring four pixels only if the pixels is a non-zero, and hence it limits the region to only within the boundary of the signer's hand. The overall result of region growing is as shown in Figure 4(c). Lastly, Canny edge detection is again applied to detect the edges in the segmented hand image as shown in Figure 4(d). The edges detected will be used in feature extraction phase.



(a) (b) (c) (d)

Fig. 4. Segmentation result **(a)** Original image, **(b)** Background subtracted image, **(c)** Region grew image and **(d)** Edge detected image

### D. Feature Extraction

The SURF technique which is developed based on SIFT, is employed to extract descriptors from the segmented hand gesture images. SIFT is a novel feature extraction method which is robust against rotation, scaling, occlusion and variation in viewpoint introduced by Lowe in [16]. SURF is proposed by Bay *et al.* in [11] as a feature extraction alternative to the existing method which is more computation efficient. As opposed to Difference of Gaussian (DoG) used in SIFT, SURF approximates the Laplacian of Gaussian (LoG) with a box filter. The convolution of box filter calculated using integral images is faster and can be done in parallel with differing scales, thus it is much faster compared to SIFT [11].

To detect descriptors, SURF uses an integer approximation of the determinant of Hessian blob detector, which can be computed with three integer operations using a pre-computed integral image. Its feature descriptor is based on the sum of Haar wavelet response around the point of interest. Square-shaped filters are used as an approximation of Gaussian smoothening. Integral image is the sum of intensity value, $I$ for all points in the image with a location less than or equal to $(x, y)$ as shown in (1):

$$S(x,y) = \sum_{i=1}^{x}\sum_{j=1}^{y} I(i,j) \tag{1}$$

SURF employs hessian blob detector to obtain interest points. The determinant of Hessian matrix describes the extent of the response and is an expression of local change around the area. The Hessian matrix with point $x$ and scale $\sigma$ is defined as in (2):

$$H(x,\sigma) = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \tag{2}$$

where $L_{xx}(x,\sigma)$ is the convolution of the image with the second-order derivative of the Gaussian as described by Bay *et*

*al.* [11]. The box filter is an approximation of Gaussian with $\sigma = 1.2$ and is the lowest level in blob-response maps.

The scale space is realized as an image pyramid to make the system scale-invariant. With the use of integral image and box filter, the scale space can be realized by up-scaling. By filtering the image with an increasing masks, the filter increase from 9 × 9 which corresponds to scale, $\sigma = 1.2$, to 15 × 15, 21 × 21 and so on. Finally, non-maximum suppression is applied in a 3 × 3 × 3 neighbourhood to localize interest point in the image. The Hessian threshold is set to 500, number of octaves is 4, and descriptors of 128 dimension is calculated. The descriptors extracted are as shown in Figure 5 and are circled in blue.

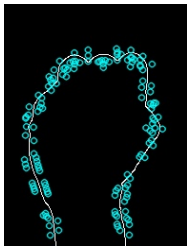


Fig. 5. SURF descriptors extracted

### *E. Classification*

The SURF descriptor extracted will each have a different dimension. However, a multiclass SVM requires uniform dimensions of feature vector as its input. BoF is therefore implemented to represent the features in histogram of visual vocabulary rather than the features as proposed in [7]. The descriptors extracted are first quantized into 16 clusters using *K*-means clustering. Given a set of descriptors $X_i$, where $(i = 0, 1, 2 \dots n)$, *K*-means clustering categorizes $n$ numbers of descriptors into *k* numbers of cluster centre. The objective is to minimize the sum of squared Euclidean distances between data points $X_i$ and their nearest cluster $\mu_k$ centre in (3):

$$sum\ of\ squares = \sum_{j=1}^{k} \sum_{i=1}^{n} ||X_i - \mu_k||^2 \quad (3)$$

In the assignment step, sum of squares is computed for each data point and the data points are assigned to the cluster which conveys the nearest value. This is followed by the update step which involves calculating the new mean to be the centroid of each cluster and each cluster centre are assigned to the new centroids. The assignment and update step are iterated until convergence is achieved. This results in separation of *n* data points into *k* clusters with each point possessing the nearest sum of squares to its respective centroid. The convergence of *K*-means clustering is defined as reaching a maximum of 20 iterations, or a squared error of less than 0.001 in (3). The clustered features then form the visual vocabulary where each feature corresponds to an individual sign language gesture. With the visual vocabulary, each image is represented by the frequency of occurrence of all clustered features. BoF represents each images as a histogram of features, in this case the histogram of 16 classes of sign languages gestures.

SVM is a supervised machine learning technique and is used in the classification stage. SVM generally finds the optimal hyperplane to separate data points belonging to different classes. Support vectors are data points which lie nearest to the decision hyperplane. SVM maximizes the margin around the separating hyperplane. Support planes are formed to separate the support vectors of two different classes to obtain the decision boundary. Two planes are found which best represent the data. Let $w$ be the weight vector for $(w = [w_1, w_2, \dots, w_n])$, $x$ is the feature vector for $(x = [x_1, x_2, \dots, x_n])$ and $b_0$ is the bias. The weight vector decides the orientation of decision boundary, whereas bias point decides its location. The plane crossing upper point is shown in (4), plane crossing lower point is as shown in (5).

$$w_0^T x + b_0 > 1 \quad (4)$$

$$w_0^T x + b_0 < -1 \quad (5)$$

The Margin, $M$ is twice the distance between support vector and plane, hence margin is defined as in ().

$$M = \frac{2}{||w||} \quad (6)$$

In order to maximize the margin, $M$, weight vector, $w$ in must to be minimum, which in turn minimizes the training error, $\xi_i$ in (7). This formulation is the soft-margin SVM, whereby the $C$ parameter controls trade-off between training errors and margin [17].

$$min\ \frac{1}{2}||w||^2 + C \Sigma_i \xi_i \quad (7)$$

$$s.t\ \forall_i y_i(w_0^T x_i + b) + \xi_i \geq 1,\ \xi_i \geq 0$$

In this paper, 100 images from each 16 ASL alphabet will undergo the segmentation and feature extraction processes as described above, to train the SVM database. We utilise the one-versus-all multi class SVM to classify all 16 different classes of gestures. Radial Basis Function (RBF) kernel is used for non-separable data points by mapping into higher dimensional space. In order to optimise the soft margin parameter, *C* and the kernel hyper parameter, γ, a 10-fold cross validation is performed, and the parameters which returns minimal classification error are selected.

## III. Experimental Results

### *A. Segmentation Result*

The proposed framework is able to segment the hand gestures in the presence of moderately complex background and different illumination condition. Figure 6 below shows the result of segmentation and Canny edge detected image with different background and illumination condition.

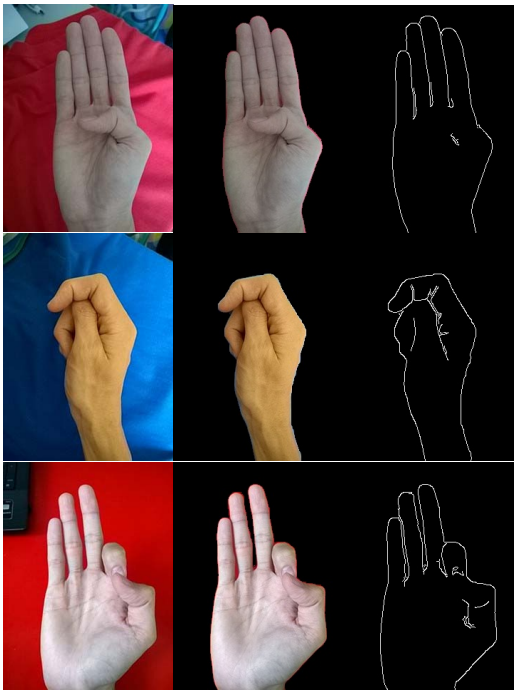



Fig. 6. Sign Languages **(a)** Original image, **(b)** Segmented image and **(c)** Edge detected image

### *B. Recognition Rate*

The performance of sign language recognition framework is evaluated for each of the 16 gestures comprising the sign language alphabet: A, B, C, D, E, F, H, I, L, N, O, R, S, T, U and Y. A total of 1600 image samples with 100 images from each ASL alphabets are taken to train the SVM database. The evaluation of accuracy are performed offline with a total of 800 test images. All images are collected from a single signer and each signs with slight variation in scale and rotation. We conducted experiment to evaluate the performance between our proposed framework with that in [7, 8]. Figure 7 shows the recognition rate of the overall framework, obtained using SIFT and SURF feature extraction technique respectively. The average accuracy obtained using SURF is 97.13%, while the accuracy obtained using SIFT is 92.25%.

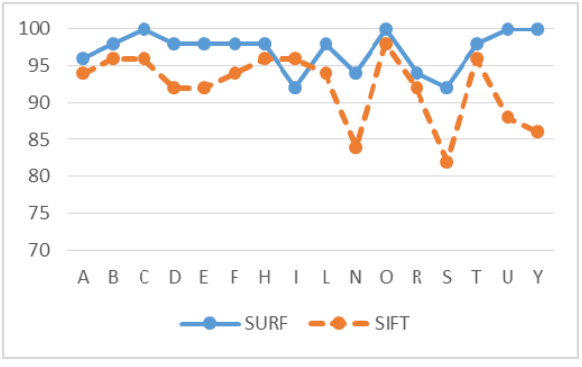



Fig. 7. Recognition rate comparison between SURF and SIFT

The database is trained in an Intel Core i5-2410M CPU 2.30GHz computer with 8 GB RAM. In order to train a database with 1600 images, SIFT used an average of 6 minutes 40 seconds, while SURF only used average of 4 minutes 28 seconds. The online sign language recognition is implemented on Nokia Lumia 1520 with 2.2GHz processor and 2GB RAM. The average time taken for a test sign language input using the proposed framework is 0.62 second.

From our experiment, it is found out that the accuracy of each sign languages is heavily dependent on the similarity in appearance of the gesture as compared to other sign languages in the database. Evidently, sign languages which are visually distinctive are able to be recognized with higher accuracy, for instance alphabet 'C' and 'O'. In contrast, sign language which are visually similar are prone to be misclassified, such as alphabet 'N' and 'S' False prediction is described as the result of recognition of gestures that are misclassified into other gestures and their respective occurrence in percentage. Table 1 below shows the recognition rate of 16 gestures using our proposed framework.

**Table 1: Recognition Rate of 16 ASL gesture**

| Input | Accuracy (%) | False prediction ( % ) | | |
|---|---|---|---|---|
| A | 96 | I(2) | S(2) | |
| B | 98 | F(2) | | |
| C | 100 | | | |
| D | 98 | H(2) | | |
| E | 98 | B(2) | | |
| F | 98 | B(2) | | |
| H | 98 | B(2) | | |
| I | 92 | E(4) | F(2) | S(2) |
| L | 98 | N(2) | | |
| N | 94 | A(4) | I(2) | |
| O | 100 | | | |
| R | 94 | E(2) | L(2) | T(2) |
| S | 92 | A(4) | I(2) | L(2) |
| T | 98 | R(2) | | |
| U | 100 | | | |
| Y | 100 | | | |
| Average | 97.13 | | | |

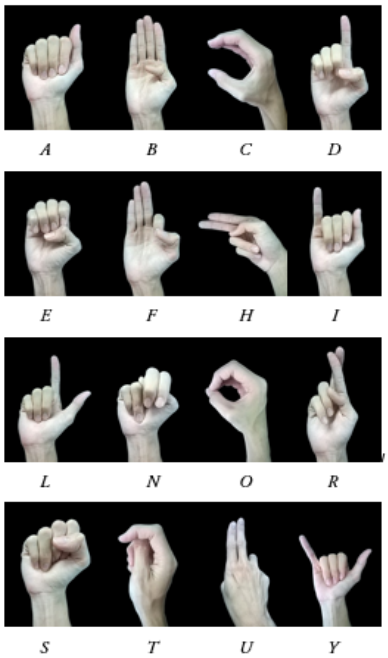
## IV. Conclusion and Future Work

Through this paper, we have proposed a novel implementation of sign language recognition on a smartphone platform. Canny edge detection and region growing technique are used to segment the hand gesture from the background captured by the camera. SURF descriptors are then extracted from the segmented hand gesture, which are then clustered into 16 classes via *K*-means clustering. The BoF model is employed to create a visual vocabulary from the cluster centroids and lastly, SVM is applied to classify the sign languages. The recognized sign language may then be translated into text and speech outputs on the smartphone app.

Our proposed framework recognizes 16 classes of ASL alphabet at an average accuracy of 97.13%. Our experiment has shown that SURF is more computational efficient and has slightly better accuracy than SIFT. However, it must be noted that the test images used were captured under generally similar illumination and background conditions. The results may decrease somewhat if the background and illumination condition is changed. From our analysis of results, we found that sign language alphabets which share great similarity in appearance are prone to be misclassified and hence derive a lower accuracy. Whereas, visually distinctive sign language gestures tend to have high accuracy. Overall, this paper proposed a framework to develop sign language translator on portable platform that may lead to a practical solution in helping the verbally-disabled to overcome the communication barrier in society. As this is a first application of sign language recognition on a mobile platform, the scope is limited to basic sign alphabets. Future improvements to our framework may be to expand the application to a wider vocabulary. Further, more advanced algorithms may replace the current in place so as to improve the accuracy and processing speed of the system.

## Appendix

American Sign Language used




## Acknowledgment

The authors would like to express their appreciation to the sponsors of this research, the Ministry of Higher Education Malaysia and Universiti Teknologi Malaysia under the Research University Tier 1 Grant (vote number 09H75).

## References

[1] S.S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, *43*(1), pp.1-54, 2015.

[2] N.A. Ibraheem and R.Z. Khan, "Vision based gesture recognition using neural networks approaches: A review," *International Journal of human Computer Interaction (IJHCI)*, *3*(1), pp.1-14, 2012.

[3] G.R.S. Murthy and R.S. Jadon, "A review of vision based hand gestures recognition," *International Journal of Information Technology and Knowledge Management*, *2*(2), pp.405-410, 2009.

[4] R.Z. Khan and N.A. Ibraheem, "Survey on gesture recognition for hand image postures," Computer and Information Science 5, no. 3: p110, 2012.

[5] J.R. Pansare, S.H. Gawande, and M. Ingle, "Real-Time Static Hand Gesture Recognition for American Sign Language (ASL) in Complex Background," *Journal of Signal and Information Processing*, *3*(3), p.364, 2012.

[6] J. Rekha, J. Bhattacharya and S. Majumder, "Shape, texture and local movement hand gesture features for indian sign language recognition," In *Trendz in Information Sciences and Computing (TISC), 2011 3rd International Conference on* (pp. 30-35). IEEE, 2011.

[7] N.H. Dardas and N.D. Georganas, "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques," Instrumentation and Measurement, IEEE Transactions on 60, no. 11: 3592-3607, 2011.

[8] A. Tharwat, T. Gaber, A.E. Hassanien, M.K. Shahin and B. Refaat, "Sift-based arabic sign language recognition system," In *Afro-european conference for industrial advancement* (pp. 359-370). Springer International Publishing, 2015.

[9] D. Kim and R. Dahyot, "Face components detection using SURF descriptors and SVMs," In Machine Vision and Image Processing Conference, 2008. IMVIP'08. International, pp. 51-56. IEEE, 2008.

[10] K. Singh, and S. Chander, "Content Based Image Retrival Using SURF, SVM and Color Histogram – A Review,". International Journal of Emerging Technology and Advanced Engineering, Vol. 4, 2250-2459, 2014.

[11] H. Bay, T. Tuytelaars and L. Van Gool, "Surf: Speeded up robust features," In Computer vision–ECCV 2006, pp. 404-417. Springer Berlin Heidelberg, 2006.

[12] X. Chai, G. Li, Y. Lin, Z. Xu, Y. Tang, X. Chen, M. Zhou, "Sign language recognition and translation with kinect," In IEEE Conf. on AFGR. 2013.

[13] M. Mohandes, S. Aliyu, and M. Deriche, "Arabic sign language recognition using the leap motion controller," In *Industrial Electronics (ISIE), 2014 IEEE 23rd International Symposium on* (pp. 960-965). IEEE, 2014.

[14] C.H. Chuan, E. Regina, and C. Guardino, "American Sign Language recognition using leap motion sensor," In *Machine Learning and Applications (ICMLA), 2014 13th International Conference on* (pp. 541-544). IEEE, 2014.

[15] T.J. Joshi, S. Kumar, N.Z. Tarapore, and V. Mohile, "Static Hand Gesture Recognition using an Android Device," International Journal of Computer Applications 120, no. 21, 2015.

[16] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, *60*(2), pp.91-110, 2004.

[17] A. Ben-Hur and J. Weston, "A user's guide to support vector machines," *Data mining techniques for the life sciences*, pp.223-239, 2010.