Android based American Sign Language Recognition System with Skin Segmentation and SVM

Sakshi Lahoti, Shaily Kayal, Sakshi Kumbhare, Ishani Suradkar, Vikul Pawar
Department of Computer Science and Engineering
Government College of Engineering, Aurangabad
Maharashtra, India
ssakshilahoti@gmail.com, shailykayal03@gmail.com,sakshikumbhare@gmail.com,
isuradkar@gmail.com, vikul.pawar@geca.ac.in

Abstract – People with hearing impairment use sign language for communication. They use hand gestures to represent numbers, letters, words and sentences, which allows them to communicate among themselves. The problem arises when they need to interact with other people. An automation system that can convert sign language to text will make the interaction easier. Recently, many such systems for sign language recognition have been developed. But most of them were executed using laptop and computers, which are impractical to carry due to their weight and size. This article is based on the design and implementation of an Android application which converts the American Sign Language to text, so that it can be used anywhere and anytime. Image is captured by the smart phone camera and skin segmentation is done using YCbCr systems. Features are extracted from the image using HOG and classified to recognize the sign. The classification is done using Support Vector Machine (SVM).

Keywords: Pattern Recognition, Sign language, Android, Support Vector Machine (SVM), Machine learning.

I. INTRODUCTION

Deaf and mute people use sign language, that is, visual gestures of hand combined with body movement and facial expressions to communicate with each other. Most of the ordinary people are unable to interpret their sign language and this leads to a communication gap between them. This affects their social and working life. A translator is desirable when a person wants to converse with hearing impaired people, but there is a lack of availability of such experienced and educated interpreters. An automation system that can translate sign language into spoken language can aid the hearing impaired in interacting with normal people.

Plenty of research has been carried out to develop a sign language recognition structure using wearable devices such as a hand glove equipped with a flex sensor and an accelerometer sensor [11]. More advanced research for recognition methodology is proposed, which utilizes a webcam and a Kinect in place of wearable devices [12]. But most of the approaches mentioned above have been implemented using computers and laptops, which are nowadays impractical to carry along everywhere. The panacea to all these problems is the smartphone, which will act as an effective way to bridge the existing gap between the deaf-mute and other people. In this paper, the development of a system to recognize and interpret American Sign Language (ASL) based on Android devices has been presented.

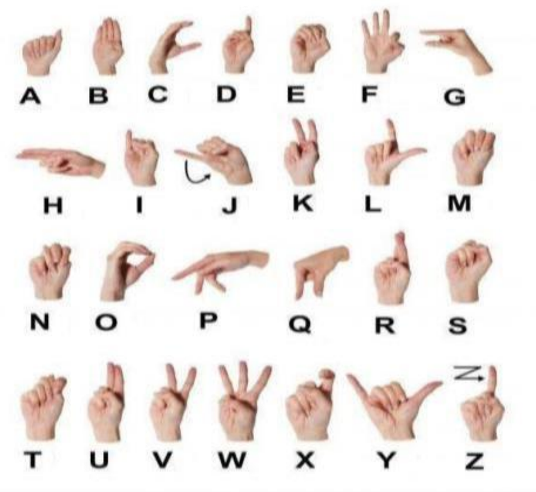


Fig1. American Sign Language as shown in Bahan, Benjamin [1]

II. RELATED WORKS

Over the past years, many researchers have proposed different approach and algorithms for the development of sign language recognition. Brandon Garcia and Sigberto Alarcon Viesca [2] implemented a robust model for the letters a-e and a modest one for letters a-k. The system is trained with the Surrey and Massey datasets on ASL and the Google Net architecture. The system takes the input of the user's sign language video, classifies the gestures for each letter and then tries to come up with the most accurate words. They attained a validation accuracy of nearly 98% with a-e and 74% with a-k.

Hardie Cate et al [3] use machine learning algorithms on high quality data for sign language recognition. They use devices like Myo armband to collect data about the position of the user's hand. They implement the baseline SVM and logistic regression model on high quality data, and use long short term memory architecture and sequential pattern mining method on lower quality data since it requires more sophisticated approach.

J. Rekha et al [4] propose two systems which are used for real time hand gesture recognition. They use K-Nearest neighbor (KNN) and Support Vector Machine (SVM) model are used for classification of a signed letter. This paper also proposes an approach for recognizing finger spelled words using a lexicon based Hidden Markov Model (HMM). This system achieves a recognition rate of 72.3%.

Mohamed Hassan et al [5] propose a system which is tested in two different datasets. These datasets are collected by using data gloves. One data set is collected by using DG5-VHand data glove and other by using Polhemus G4 tracker. For classification of the images, they use the Hidden Markov Model (HMM) and the Modified KNN approach. The word recognition rate obtained of this system using HMM is 83%.

In the research related to Android devices, Setiawardhana et al [6] have researched on developing android application for sign language translation. They use the KNN model for detecting the hand signals and the Viola – James model for object detection. Mahesh M et al [7] proposed a method using Oriented Fast and Rotated BRIEF ((Binary Robust Independent Element Features) for comparison. The skin segmentation is done using HSI model. The accuracy obtained using this method is around 70%.

III. SYSTEM PROPOSED

ASL is one form of hand gesture recognition operated through signs. Due to its ease of understanding and simplicity it is recognized as world's second language. We have implemented ASL recognition through an Android app.

Dataset:

We have created a database of American Sign Language using 36 symbols containing alphabets A to Y, numbers from 0-9 and spacebar. Z is not involved in this database as Z is not a steady gesture and is represented by movements of hands and hence, it requires video capturing and frame partitioning. For each symbol, 500 images are used for training. All the training images are of size 200X200 pixels with black background so that skin segmentation and edge detection becomes easy and reduce the storage space and load time.

The steps for sign language recognition process are as follows:

1. Hand gesture capture and skin segmentation
2. Feature Extraction
3. Classification using SVM

A. Hand gesture capture and skin segmentation –

The gesture of the hand is captured using smartphone camera. The camera is started and video taken is divided into smaller frames so that the hand movement is properly detected. Most of the gesture capturing software use external systems like depth sensors et al to detect the motion of hand. But these systems can sometimes turn out to be slower and consume more space. To overcome this drawback, we made our camera more efficient by developing a rectangular region where in the user has to place his/her hand to carry out recognition process. The benefits of this technique are i) user can easily handle the system anywhere without getting muddled ii) skin segmentation becomes easier due to trimming of the video frame.

The next step of the process is carrying out skin segmentation. The colour image in the bounding box is transformed into a skin-likelihood image. This image is then modified to a gray-scaled image, where the gray values of pixels represent its likelihood belonging to skin. Adaptive thresholding is used for segmenting skin of different colors to acquire better results.

Skin segmentation is done by YCbCr color systems where Y represents the luma (brightness of an image in a video) component, Cb is the blue-difference and Cr is the red difference. The formula for the conversion of RGB to YCbCr is as follows:

$$\begin{pmatrix} Y \\ Cr \\ Cb \end{pmatrix} = \begin{pmatrix} 0.2290 & 0.5870 & 0.1140 \\ -0.1687 & -0.3313 & 0.5000 \\ -0.5000 & 0.4187 & -0.0813 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + 128$$

For each pixel value that falls within the range of Cb and Cr will be assumed as part of skin and the values outside the range will be considered as non-skin part and will be converted to black pixel (0 values).

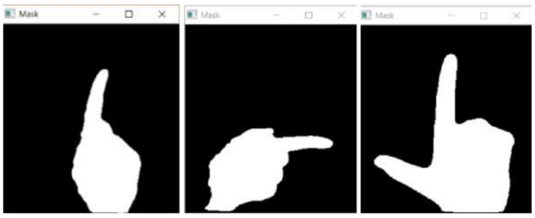


Fig.2 Segmented Image with skin detection

B. Feature Extraction-

Feature extraction is the most important step in sign language recognition. We have used HOG (histogram of oriented gradients) which acts as a human/feature descriptor for this purpose. It counts the occurrences of gradient orientation in the neighbourhood portions in an image. The image is first divided into smaller connected regions called cells. For each pixel in the cell, a histogram of gradient directions is calculated. Finally, the concatenation of these histograms is done to form the global histogram. To get better results, the local histograms are quantized and the gradient magnitude is evaluated. The main benefit of using HOG over SIFT and other descriptors is that, it is unaffected by photometric and geometric transformations. Algorithm of HOG can be summarized as follows:

1) The first step includes calculating the gradient values.
2) The next step involves orientation binning where in the cell histograms are calculated.
3) After cell creations, descriptor blocks are created to account the changes in illumination and contrast.
4) The next step includes block normalization in which a normalization factor is calculated and multiplied to block vector.
5) The final step involves object recognition for which HOG is combined with algorithms like Support Vector Machines (SVMs) which are discussed in the succeeding section.

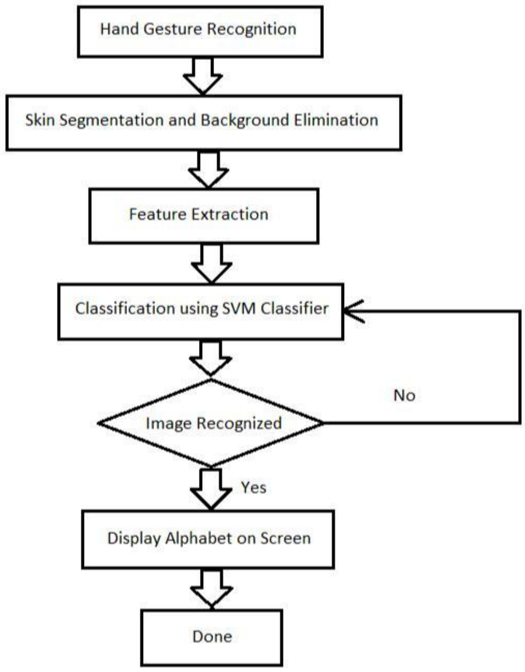


Fig.3 Data Flow Diagram

C. Classification using SVM-

Support vector machines (SVMs) are supervised learning model which are used for

classification of objects. They construct a hyperlane or set of hyperlanes in high dimensional model to make the classification and regression model. Given a training dataset, the job of SVM is to classify the new test image in one particular category where it is supposed to belong to, making it a non- probabilistic linear model. In SVM, each data is represented as a point in plane, the training set categories are separated by a gap which is made clear and wide enough as possible for correct classification. The new test samples are assigned these categories based on the placement of their point on the sides of the gap. SVM defines a kernel function which is most suited to the problem. To achieve a better overall fit and make the separations more precise SVM has a concept of "soft margin function", denoted by C, which controls the influence of each individual support vector [10]. This process involves trading error penalty for stability and precision. To handle complicated relations, the dot product in SVM can be replaced by a non-linear kernel function (like Gaussian radial function).

We use gamma which is a free parameter in the radial function and is denoted by:

$$K(x_i, x_j) = \exp(-\gamma||x_i - x_j||^2), \gamma > 0$$

Where x_i and x_j are support vectors.

A large change in gamma represents low variance model and a high bias. In the proposed system, we have used the value of C and Gamma as 2.67 and 5.383 respectively.

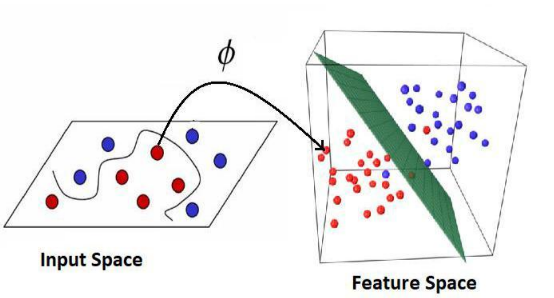
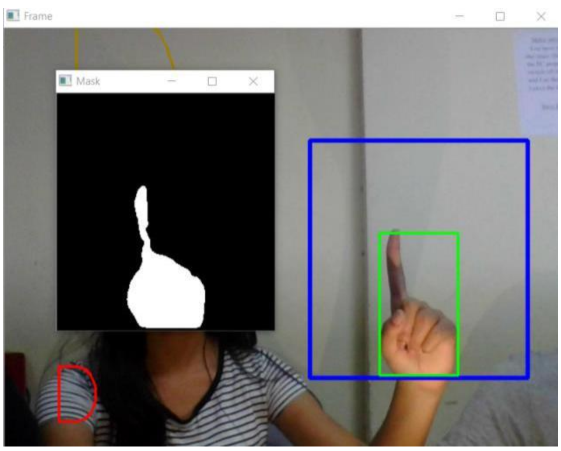
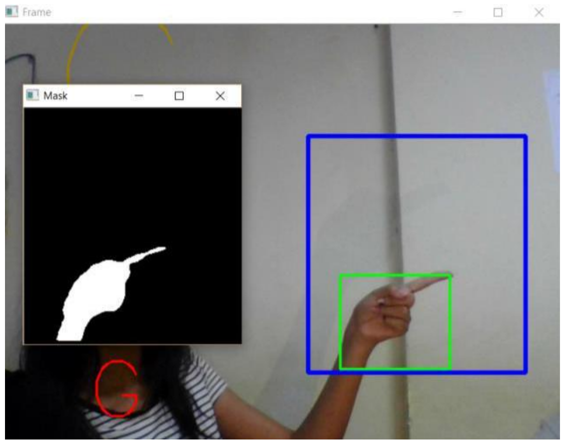


Fig.4 Hyperlane separating the input data into two spaces.

In the above figure hyperlane is shown as a green rectangle. The two classes are differentiated using blue and red colours. Phi denotes the phase shift that occurs during the classification.

The below figures will demonstrate how SVM is successful in recognizing the alphabets segmented in Fig.2 displayed in the rectangular box. The character is printed in red colour at the bottom of the screen.





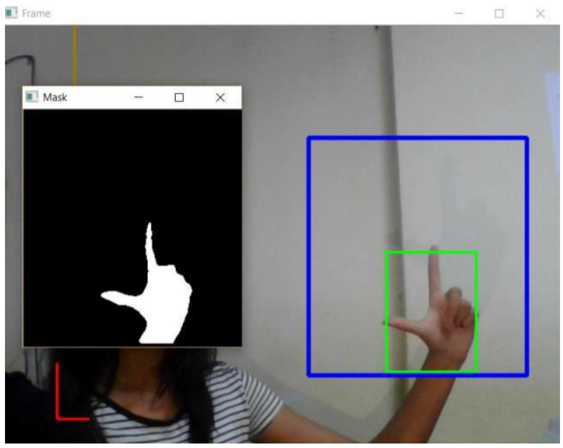


Fig.5 Recognized alphabets of the above segmented images in Fig.2

After performing the training and classification process, the final dataset will be uploaded to a cloud platform. This technique will be beneficial because it will conserve the phone's memory and load. Also, the dataset will be easily accessible to any future modifications or additions without having an effect on the application.

IV. RESULT AND DISCUSSION

The input of the proposed system is hand gesture image captured through the smart phone camera. The hand gestures are then processed and skin segmentation is done by using YCbCr system. HOG process is used to create a histogram of the hand gesture. The output obtained after the HOG pre-processing is classified on the basis of training dataset using SVM. The output is in the form of printed symbol on screen.

The next step is testing the accuracy of the system. We have done it by two approaches. The first approach includes testing images in the database against others of the same alphabet. First 100 images of an alphabet are tested using the next 400 images of the same alphabet's dataset. The accuracy of first approach i.e. database testing is 98%. Second approach of the testing is testing in real time domain. In this approach the hand gestures captured by smart phone camera are tested using the images in dataset. The accuracy achieved by this system turns out to be 89.54%. The system can be made more accurate by proper illumination. The accuracy varies with complex background.

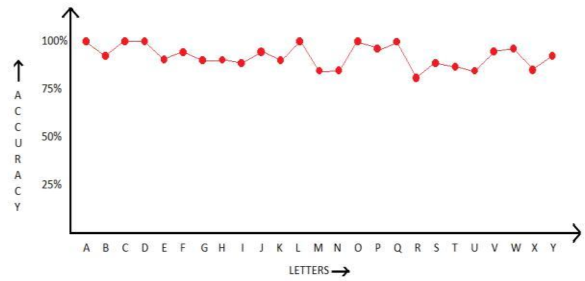


Fig.6 Graph showing the accuracy of recognized letters.

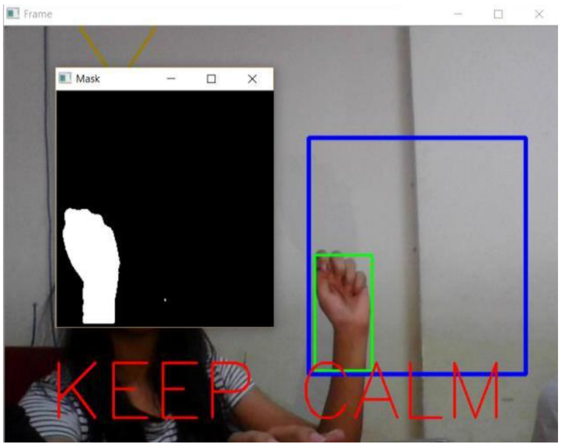


Fig.7 Demonstration of recognition of identified words.

The letters presented in Fig.7 are recognized one after the other. As soon as the first word (KEEP) ends, we displayed the space gesture and started with the next word (CALM). After the recognition of the two words (KEEP CALM), screenshot is captured while displaying the succeeding character in the rectangular box.

The existing systems include the programs developed to run on computers and laptops. These are impractical to carry around because of their weight and size.

Another approach uses hand gloves which consist of different trackers and sensors embedded in it to locate the position of the hand. The system then creates an image based on the data the sensors send, and provide the output. But, the sensors and trackers are sensitive to humidity, hence they cannot be carried in rain. The glove also restricts movements of hand for any other purposes since it is bulky.

With the development of mobiles and handheld devices, it has become much more easier to create such systems. The system we have proposed is based on mobile phones which are portable and available to everyone. There are some applications on Android based on this concept but the accuracy reached is not as anticipated. Also, these applications tend to exhaust the space and RAM of the mobile. We are meticulous about these drawbacks and, are developing an application which is lightweight and error-free.

V. CONCLUSION

This paper proposes an android based sign language recognition system for American Sign Language (ASL). The system translates the sign language into text from the images captured from the smart phone camera. The image database on which training takes place is stored on cloud. The accuracy obtained by this system is 89.54%. In this system, skin segmentation is done using YCbCr system, and SVM model is used for classification of the signs. In future, this system can be expanded to other sign languages. Also, it can be further developed to recognise gestures of words used in day-to-day life.

ACKNOWLEDGEMENT

Authors are grateful to the project research guide Prof. V.J. Pawar for inspiring us on the context and motivating us to implement our research in a proper and courteous way. We would also like to thank our Head of Department Prof. Vivek.P.Kshirsagar for conducting timely reviews and surveillance which helped in directing our ways out. Our parents for providing the moral support and developing patience in our minds. The last but not the least our almighty God for letting us into the world and providing us an opportunity to work under our magnanimous research guide.

REFERENCES

[1] Bahan, Benjamin (1996). Non-Manual Realization of Agreement in American Sign Language. Boston University.

[2] Brandon Garcia, Sigberto Alarcon Viesca "Real time American Sign Language Recognition with Convolutional Neural Networks".
[3] Hardie Cate, Fahim Dalvi, Zeshan Hussain December 11, 2015 "Sign Language Recognition using Temporal Classification"
[4] Rekha, J. Bhattacharya, and S. Majumder, "Shape, texture and local movement hand gesture features for Indian sign language recognition," in 3rd International Conference on Trends in Information Sciences & Computing (TISC2011), 2011.
[5] Mohamed Hassan, Khaled Assaleh, Tamer Shanableh "User-dependent Sign Language Recognition Using Motion Detection" International Conference on Computational Science and Computational Intelligence, 2016
[6] Setiawardhana, Rizky Yuniar Hakkun, Achmad Baharuddin "Sign Language Learning based on Android For Deaf and Speech Impaired People" International Electronics Symposium (IES), 2015
[7] Mahesh M, Arvind Jayaprakash, Geetha M "Sign language translator for Mobile Platforms" IEEE 2017
[8] S.Reshna, M. Jayaraju "Spotting and Recognition of Hand Gesture for Indian Sign Language Recognition System with Skin Segmentation and SVM" IEEE WiSPNET 2017
[9] Rudy Hartanto, Annisa Kartikasari "Android Based Real-Time Static Indonesian Sign Language Recognition System Prototype" ICITEE 2016
[10] www.rodrigo-rivera.com
[11] Munib, Q.. "Android sign language(ASL) recognition based on Hough transform and neural networks", Expert Systems With Applications, 2000701
[12] Davi Hirafuji Neiva, Cleber Zanchettin. "Gesture Recognition: a Review Focusing on Sign Language in a Mobile Context", Expert Systems with Application, 2018