IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 12, December 2019

Classification of Sign Language Gestures using Machine Learning

Abhiruchi Bhattacharya[1], Vidya Zope[2], Kasturi Kumbhar[3], Padmaja Borwankar[4], Ariscia Mendes[5]

B.E., Computer Engineering Department, VES Institute of Technology, Mumbai, India[1,3,4,5]

Assistant Professor, Computer Engineering Department, VES Institute of Technology, Mumbai, India[2]

Abstract: The low awareness of sign language among the general public presents a hurdle in communication with the deaf and dumb communities and their integration within society. A sign language translator application can help reduce this communication gap. This paper presents a system developed to detect fingerspelling gestures from video and give their English equivalent letter using machine learning, with a focus towards developing a potential solution for everyday use. A classifier is trained on a dataset of 24 fingerspelling gestures using the bag of visual words approach, wherein features detected in the images are clustered to form a codebook, then each image is expressed as a histogram denoting frequency of observed codewords in that image. The SURF and BRISK algorithms are explored for automatic feature detection. Four classification algorithms are evaluated, namely K-nearest neighbours, logistic regression, Naive Bayes and Support Vector Machine. The best performing model has been used to classify sign language gestures from video frames.

Keywords: Bag of visual words, codebook, descriptors, histogram of codewords, image processing, feature detection, machine learning, sign language, speeded up robust features

I. INTRODUCTION

The majority of hearing people are unaware of the semantics and rules of sign language, rendering communication very difficult between an average person and a speech-impaired or hearing-impaired individual in most cases. The deaf community in India especially in regions of low economic prosperity[1] still faces major challenges in overcoming stigmas against sign language. Currently, there is still little to no formal recognition of Indian sign language by the government agencies[2]. There is a high potential for developing an application for sign language translation and is essentially useful to bridge the communication gap between the hearing majority and the deaf & dumb communities. While dictionary-like web and mobile applications exist in order to translate English letters and words into signs, relatively very few free or publicly available applications have been released to convert sign language gestures into English presently.

With advancements in machine learning, deep learning and natural language processing technologies, we have solutions like Google Translate that enables smooth communication among people from myriad linguistic backgrounds. Sign language presents several different challenges compared to spoken languages. The problem of sign language classification lies in the domain of computer vision, rather than language processing. This paper describes the development of an application that aims to detect and differentiate fingerspelling gestures from video frames using image processing and machine learning techniques, and discusses results and future prospects.

II. LITERATURE SURVEY

Research on sign language recognition for American Sign Language (ASL) has been prevalent and several standard datasets are available. Similar research for Indian Sign Language (ISL) is in its initial stages. This could be due to the low prevalence of ISL in society, its rudimentary nature and lack of recognition. Because of lack of standardisation, many differences also exist between sign language practiced in different regions [1]. Consequently, no particular standard dataset exists for ISL as of yet, hence research is mostly done on existing datasets for other sign languages, or by creating or accumulating datasets from various sources.

Work in the field of sign language recognition generally adopts two kinds of approaches, i.e. a vision-based approach or a sensor-based approach [2]. The high-level pipeline for hand gesture recognition using a vision-based approach involves three main steps:

[1] "The fight to make Indian Sign Language official - Times of India." 8 Oct. 2018, https://timesofindia.indiatimes.com/home/sunday-times/the-fight-to-make-indian-sign-language-official/articleshow/66101577.cms. Accessed 8 Jan. 2020.

[2] "The quest to formalise Indian Sign Language is gaining momentum" https://www.businessinsider.in/the-quest-to-formalise-indian-sign-language-is-gaining-momentum/articleshow/66534786.cms. Accessed 8 Jan. 2020.

1) Image Pre-processing: This step involves clean-up and filter operations such as blurring, thresholding, image morphing, skin masking, etc. in order to isolate the hand gesture from the image.
2) Feature extraction: Here, features are extracted from the cleaned images. A numeric representation of the image dataset is formed. Feature extraction is required because raw pixel data can be discordant and sensitive to noise, inversion, rotation or illumination, thereby rendering the learning task difficult. Features may be either manually engineered as in [4] or extracted automatically through feature detection algorithms as in [5].
3) Classification: The features extracted from the image dataset act as the training data for a learning model.
Ewald et al. [3] provide a methodology for a complete solution for sign language translation, in the form of an Android application. A smartphone is used to capture an image of a hand gesture, which is sent to a PHP script that invokes the MATLAB code required to perform the classification. The ouput of classification, which is the letter represented in the image, is pushed back to the Android application. Features extracted from two methods, namely Histogram of Centroid Distances (HOCD) and Gabor filters, are concatenated to create the final feature vectors used for training a K-Nearest neighbours (KNN) classifier. Ewald et al. explore the problem of skin segmentation and attempt to resolve it using a likelihood model and report that the extent of errors and noise make it impractical to implement and use images with plain black backgrounds.

Rokade et al. [4] adopt a manual method for extracting feature vectors from images. For skin segmentation, a probabilistic model similar to that initially considered in [3] has been used in which RGB colour histograms are constructed from manually annotated skin and background images. These histograms are then normalised and used as probabilistic models to predict the skin region in the training and testing images. Two kinds of feature sets are chosen for feature extraction, namely the central moments of the normalised Fourier descriptors and the first seven Hu moments of an image. These features are calculated after applying a distance transformation on the image. Best performance is achieved on a neural network model with both feature sets.

Ansari et al. [5] use the Microsoft Kinect console as their input device instead of the usual camera, giving them the advantage of being able to obtain depth information from the Kinect's infrared sensor in addition to RGB colour information. The cluster of points closest to the camera is assumed to be the hand region. A combination of three feature extraction methods are considered, namely Viewpoint Feature Histogram (VFH), Scale Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF). Each are used for classification individually, and the class predicted by the majority is given as the final output. Having depth information can greatly improve the performance of a practical sign language translator. However, common cameras lack such depth information, so such an approach may not be suitable for implementation on commonly used gadgets like laptops, mobile phones etc.

The SIFT algorithm [6] returns the points of interest (or 'keypoints') found in an image and their descriptors in the form of 128-dimensional feature vectors. SURF [7] is an optimization of the SIFT algorithm, and uses integral images for faster computation. SURF returns keypoints and associated 64-dimensional feature vectors. SIFT and SURF commonly find application in computer vision settings such as object tracking, pose estimation and image matching [8]. While SIFT and SURF are patent-protected patch descriptors, the Binary Robust Invariant Scalable Keypoints (BRISK) [10] algorithm is a binary descriptor that encodes keypoint information as a binary string, which makes computation more efficient [11].

III. PROPOSED METHOD

In the presently developed system, first a machine learning classifier is trained to differentiate between images of static fingerspelling gestures. The trained classifier is then used to detect fingerspelling signs from video and give their corresponding English letter. The computer system used for development has the following specifications: Intel Core i5-7200U CPU @ 2.5 GHz and 8.00 GB RAM. We use OpenCV library for image and video processing and scikit-learn in Python for training and testing classifiers. Fig 1 shows the block diagram of the system.

A. Classification on Image Data

We use the dataset and code provided in [12] as the basis for our machine learning model. The dataset consists of a total of 4972 images of 24 static single-handed fingerspelling gestures on a plain background.

1) Image Pre-processing: Steps for image pre-processing include
- Resizing the image to 256x256 size.
- Thresholding the image using HSV (Hue, Saturation and Value) colour space, to isolate the skin area. Table I lists the HSV ranges used for skin segmentation. All figures are according to the HSV ranges used in OpenCV :

$H \in [0,180)$
$S \in [0,256)$
$V \in [0,256)$

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 12, December 2019

- Median blur with kernel size=5, in order to reduce noise.
- Canny edge detection on the masked hand region.

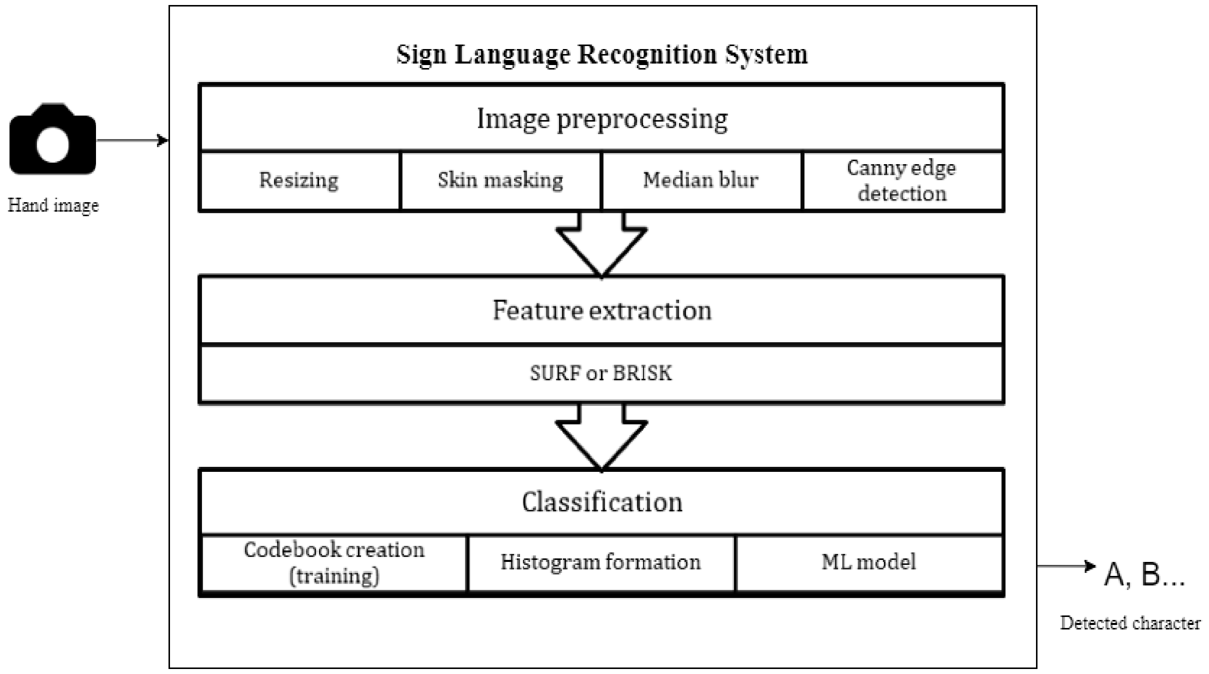

Fig. 1 System Block Diagram

Table I HSV ranges used for skin segmentation

Limits	**Hue**	**Saturation**	**Value**
Lower	0	40	30
Upper	43	255	254

1) Feature extraction: The ideal scenario for sign language gesture recognition would be to extract features invariant to changes in illumination, translation and rotation, since a practically implementable system would need to be able to extract the similarity between images of the same hand gesture irrespective of the lighting conditions, skin colour and minor variations in hand position. The SIFT algorithm is a good candidate for such an extraction algorithm, since its features are scale and rotation invariant and capture gradient information. Due to the comparatively slower calculations in SIFT [9], we consider the SURF and BRISK algorithms. After pre-processing, the image is given as input to the feature extraction algorithm and the keypoints and descriptors are returned. The descriptors returned for each hand image constitute our initial training data.

2) Classification: The classification module uses the bag of visual words approach. The descriptors extracted from an image can be likened to "words" in a document. Descriptors from all training images are first clustered by a mini-batch k-means clustering model (with k=150 and batch size=100) to learn a codebook or vocabulary of 150 visual words. This clustering model is then used to create a histogram of codewords present in each image to be classified, which becomes the training example for that image. The final classification of the images is done on the basis of these histograms. We evaluate the performance of four classification algorithms on histograms generated from both SURF and BRISK features:

- KNN classifier
- Logistic regression
- Naive Bayes classifier
- Support Vector Machine (SVM)

B. Classification on Video

For detecting gestures from video, we treat each frame of the video stream as an image to be classified. Each frame is pre-processed in a similar manner as the training images. After feature extraction and clustering, the histogram for the

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 12, December 2019

image is given as input to the trained classifier, which returns a letter as the class to which the frame belongs. Ten frames are analysed at a time and the letter found in the majority of the frames is given as the output.

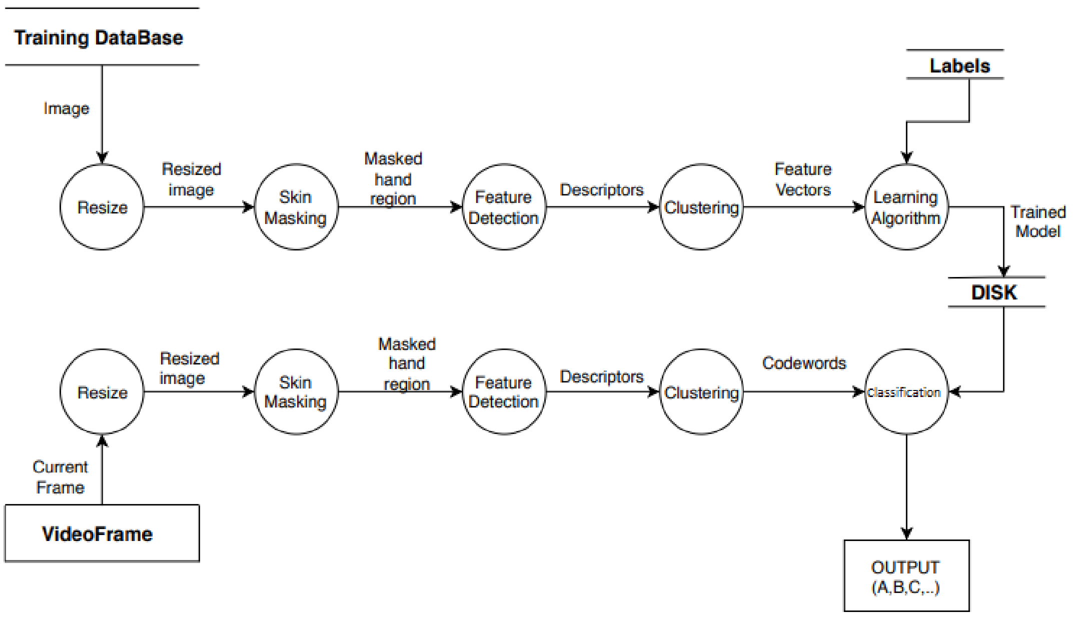

Fig. 2 Data flow diagram depicting training on image data (upper sequence) and classification on video (lower sequence).

IV. RESULTS

The results of the two phases of development are discussed below.

A. Classification on image data

The accuracy scores of the chosen models with SURF and BRISK features are given in Table II and Table III respectively. The scores represent the weighted averages of precision, recall and F1 scores over all 24 classes. A 60:40 split for training and testing set was used to calculate the metrics. The best performance was achieved using a linear kernel SVM with SURF features. SVM performed the best with BRISK features as well. Figure 3 indicates the confusion matrix for the SVM model trained on SURF features. The most misclassifications can be seen for gestures that appear similar to each other such as ‘M’,’N’, ‘S’, etc.

Table II Performance of models using SURF features (weighted averages)

Model	**Accuracy (%)**	**Precision**	**Recall**	**F1 score**
SVM	91.35	0.914	0.913	0.913
KNN	86.97	0.875	0.866	0.865
Logistic Regression	84.31	0.846	0.843	0.843
Naive Bayes	84.11	0.844	0.841	0.840

Table III Performance of models using BRISK features, weighted averages

Model	**Accuracy (%)**	**Precision**	**Recall**	**F1 score**
SVM	91.15	0.912	0.911	0.911
KNN	87.38	0.886	0.873	0.874
Logistic Regression	84.32	0.847	0.843	0.843
Naive Bayes	84.62	0.8477	0.8461	0.844

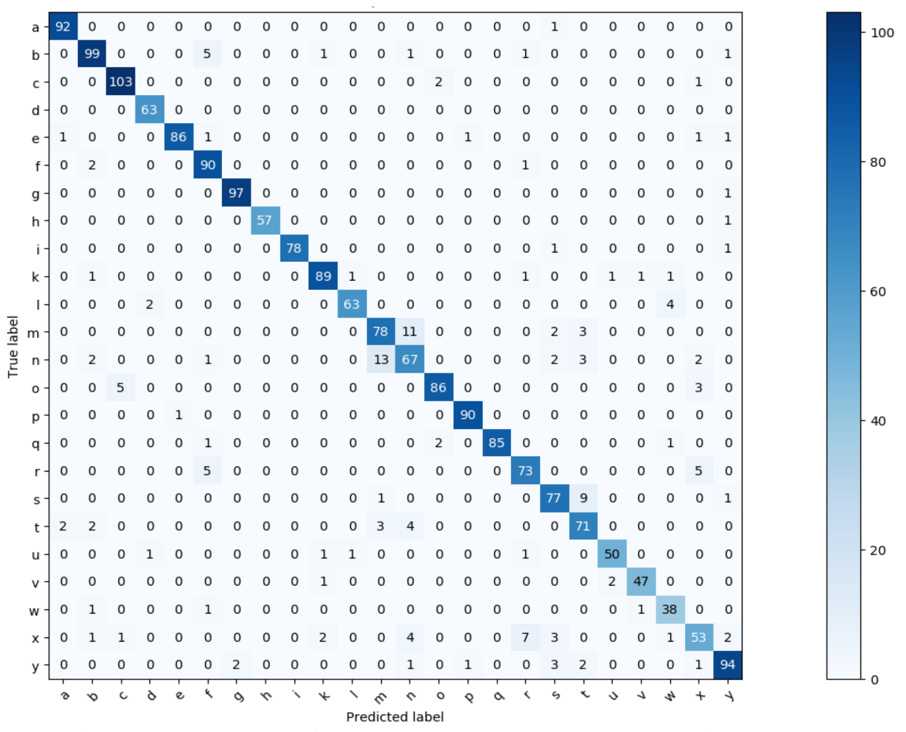

Fig. 3 Confusion matrix for SVM with SURF features. Numbers indicate number of training examples classified under the respective predicted label and having the corresponding true label.

Fig. 4 indicates the learning curves with training and cross-validation accuracies of all four classifiers for a range of iterations. For evaluating the score at each iteration, ten-fold cross validation was used with a 60:40 split of training and cross-validation sets. The learning curves may indicate overfitting as convergence is not fully observed for the training and cross-validation scores. This may be due to the relatively small dataset used for training.

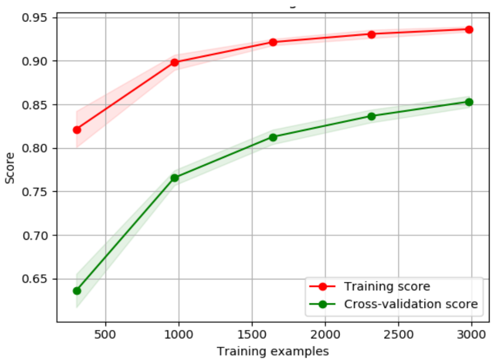

Fig. 4a KNN learning curve

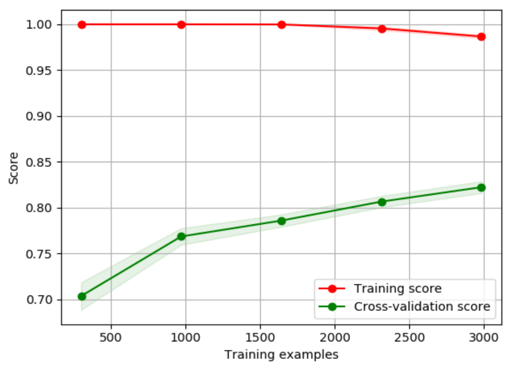

Fig. 4b Logistic regression learning curve

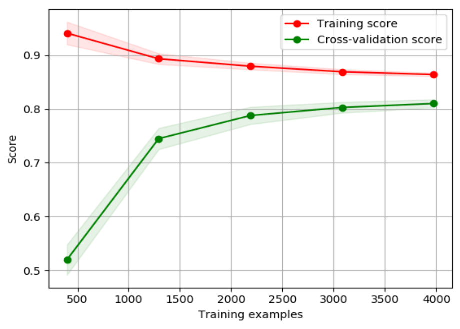

Fig. 4c Naive Bayes learning curve

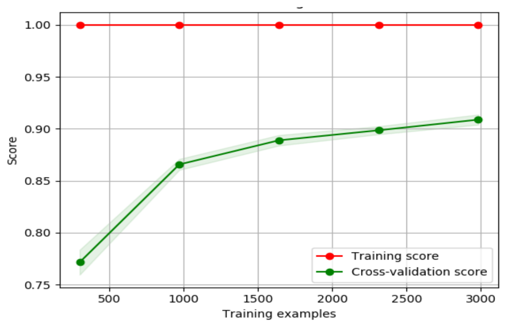

Fig. 4d SVM learning curve

Fig. 4 Learning curves for four classifiers: training (upper curve, red) vs. cross-validation (lower curve, green) accuracy vs. number of examples

A. Classification on video

The practical accuracy of the system differed from the performance of the model on image data. Out of 24 fingerspelling gestures, the system could accurately recognise up to 10 letters (B, C, D, H, L, M, N, V, W, Y) depending upon the lighting conditions and the background. One of the main reasons for the difference in performance is due to the fact that the accuracy of the prediction highly depends upon the hand being properly masked from the background. Since the skin segmentation method uses a fixed range of HSV for skin pixel values, light-coloured background objects are sometimes picked up as skin areas. Hand features are also observed to fall out of colour range in harsh lighting (visible in Fig. 5e). The system performs best in moderate lighting, with minimal objects in the background.

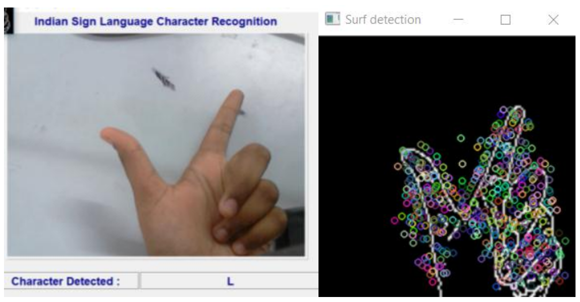

Fig. 5a L

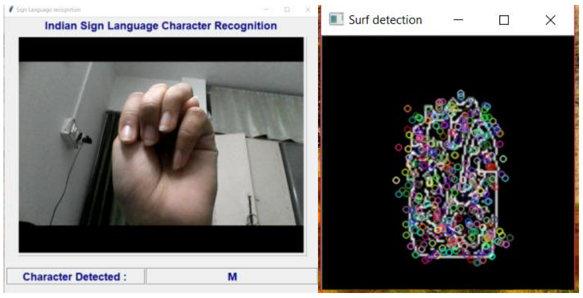

Fig. 5b M

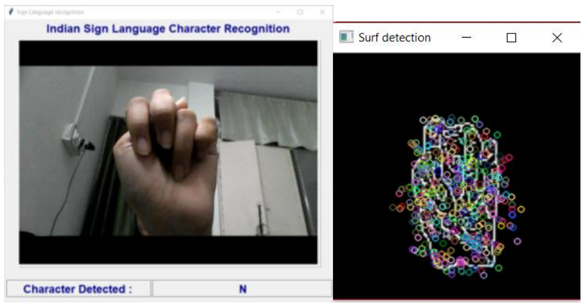

Fig. 5c N

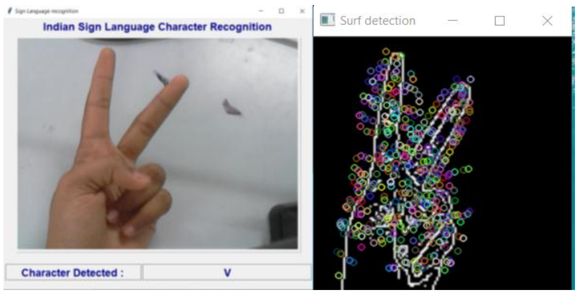

Fig. 5d V

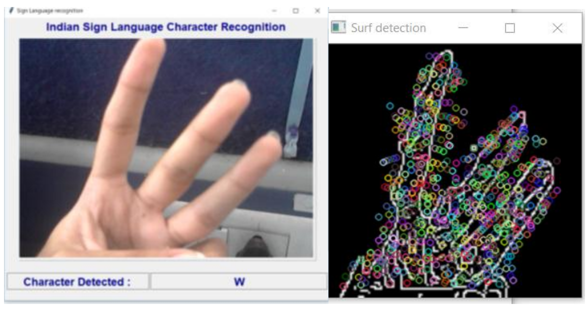

Fig. 5e W

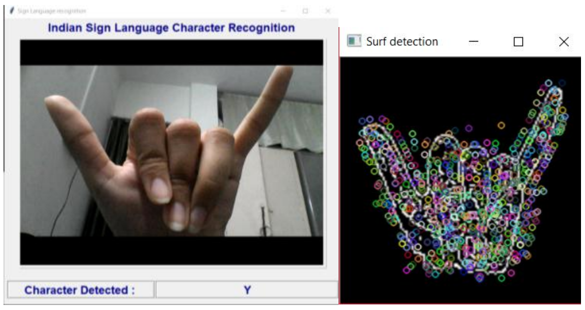

Fig. 5f Y

Fig. 5 Sample screenshots of detected letters on video along with pre-processing and SURF detection

V. CONCLUSION

The application software for sign language recognition was successfully developed using machine learning techniques. The software was trained for 24 static fingerspelling gestures. The system can satisfactorily recognise certain fingerspelling signs with reasonable accuracy. However, it could not distinguish all characters because of small dataset and errors in skin segmentation. It can be concluded from the results discussed above that for classification using video, the performance depends upon lighting and skin masking accuracy. Even with well-trained models, these become crucial factors in determining the overall accuracy of the system.

VI. FUTURE SCOPE

There is good scope for further development for better performance of this application with ML approach. The problem of noisy skin segmentation and overfitting may be mitigated by using a large and varied dataset, with signs in different backgrounds and lighting conditions. Other pre-processing methods and colour spaces such as YIQ, YUV etc. may be considered for more robust skin masking. Alternatively, neural networks can be explored for classification, due to their noise immunity and better learning capability. Additionally, the application can be further developed to enhance its usefulness by incorporating two handed signs and basic words into the dataset. For incorporation of dynamic signs, different video processing techniques will be needed.

ACKNOWLEDGEMENT

We are grateful to the Computer Engineering Department, Vivekanand Education Society's Institute of Technology for providing us this opportunity. We would like to acknowledge **Mrs. Vidya Zope** for her insightful guidance and support during the development of this project.

REFERENCES

[1]. R. Johnson and J. Johnson, "Distinction between West Bengal Sign Language and Indian Sign Language Based on Statistical Assessment", Sign Language Studies, vol. 16, no. 4, pp. 473-499, 2016. Available: https://www.jstor.org/stable/26191231.
[2]. M. Cheok, Z. Omar and M. Jaward, "A review of hand gesture and sign language recognition techniques", International Journal of Machine Learning and Cybernetics, vol. 10, no. 1, pp. 131-153, 2017. Available: 10.1007/s13042-017-0705-5.
[3]. H. Ewald, I. Patil and S. Ranmuthu, "ASL Fingerspelling Interpretation", University of Stanford, Reports, 2016.
[4]. Y. Rokade and P. Jadav, "Indian Sign Language Recognition System", International Journal of Engineering and Technology, vol. 9, no. 3, pp. 189-196, 2017. Available: 10.21817/ijet/2017/v9i3/170903s030
[5]. Z. Ansari and G. Harit, "Nearest neighbour classification of Indian sign language gestures using kinect camera", Sadhana, vol. 41, no. 2, pp. 161-182, 2016. Available: 10.1007/s12046-015-0405-3.
[6]. D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004. Available: 10.1023/b:visi.0000029664.99615.94.
[7]. H. Bay, T. Tuytelaars and L. Van Gool, "SURF: Speeded Up Robust Features", Computer Vision – ECCV 2006, pp. 404-417, 2006. Available: 10.1007/11744023_32.
[8]. M. Schaeferling, U. Hornung and G. Kiefer, "Object Recognition and Pose Estimation on Embedded Hardware: SURF-Based System Designs Accelerated by FPGA Logic", International Journal of Reconfigurable Computing, vol. 2012, pp. 1-16, 2012. Available: 10.1155/2012/368351.
[9]. D. Mistry and A. Banerjee, "Comparison of Feature Detection and Matching Approaches: SIFT and SURF", GRD Journals- Global Research and Development Journal for Engineering, vol. 2, no. 4, pp. 7-13, 2017.
[10]. S. Leutenegger, M. Chli and R. Siegwart, "BRISK: Binary Robust invariant scalable keypoints", 2011 International Conference on Computer Vision, 2011. Available: 10.1109/iccv.2011.6126542.
[11]. G. Levi, "Tutorial on Binary Descriptors – part 1", Gil's CV Blog, 2013. [Online]. Available: https://gilscvblog.com/2013/08/26/tutorial-on-binary-descriptors-part-1. [Accessed: 10- Jan- 2020].
[12]. Rishabh Gupta, "Indian Sign Language Recognition Dataset" (2018), Available: https://github.com/imRishabhGupta/Indian-Sign-Language-Recognition. [Accessed: 20- Dec- 2019]