IET Research Journals

IET Journals
The Institution of Engineering and Technology

Indian Sign Language Recognition by Feature Extraction Using SURF

ISSN 1751-8644
doi: 0000000000
www.ietdl.org

Subrajeet Mohapatra[1], *Subham Sanghai*[1*], *Aman Aryan*[1]

[1] *Computer Science Engineering Department, Birla Institute of Technology Mesra, Ranchi, Jharkhand 835215, India*
* *E-mail: sanghaisubham@gmail.com*

Abstract: This project aims at identifying alphabets in Indian Sign Language from the corresponding gestures. Gesture recognition and sign language recognition has been a well researched topic for American Sign Language(ASL), but few research works have been published regarding Indian Sign Language(ISL) and all of them require use of technologies like Accelorometer, Kinect Sensors, gyroscopes, etc. Here instead of using high-end technology like gloves or kinect, we aim to solve this problem using state of the art computer vision and machine learning algorithms. A data-set of around 767 segmented images was collected for 20 letters of the Indian Sign Language giving each letter around 35 images. SURF was firstly used to detect key points and describe them because the SURF features were invariant to image scale and rotation and were robust to changes in the viewpoint and illumination. SURF method is used to extract all the features of certain types of signs, then formed theirs code book from all the features by using K means clustering and finally classified using KNN, SVM and other supervised models to train the model.

1 Introduction

Sign language is a language that uses manual communication to convey meaning. An example can implicate simultaneously fusing hand geometry, movement, or orientation of the hands, arms or body, and facial expressions to convey a speaker's ideas. Sign languages often share significant similarities with their respective spoken language.

Wherever communities of deaf people exist, sign languages have developed, and are at the cores of local deaf cultures. Although signing is used primarily by the deaf and hard of hearing, it is also used by hearing individuals, such as people who can hear but cannot physically speak, or have trouble with spoken language due to some other disability.

For a native signer, sign perception influences how the mind makes sense of their visual language experience. For example, a hand shape may vary based on the other signs made before or after it, but these variations are arranged in perceptual categories during its development. The mind detects hand shape contrasts but groups similar hand shapes together in one category.Different hand shapes are stored in other categories. The mind ignores some of the similarities between different perceptual categories, at the same time preserving the visual information within each perceptual category of hand shape variation.

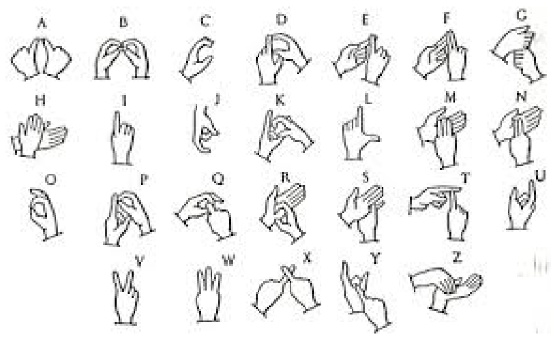

Fig. 1: Indian Sign Language.

Indian Sign Language (ISL) is the predominant sign language in South Asia, used by at least several hundred thousand deaf signers. As with many sign languages, it is difficult to estimate numbers with any certainty, as the Census of India does not list sign languages and most studies have focused on the north and on urban areas.

In this project we have developed an intelligent sign language recognition system for the Indian Sign Language by extrapolating the methods used by various researchers on American, Chinese and Australian Sign Languages.

2 Literature Review

Over years , numerous methods have been proposed. One of those papers [4] proposed a sign language recognition technique using 2D image sampling by constructing the training data from a sign language demonstration video at a certain sampling rate. The learning process was implemented using Convolutional Neural Network. This network consisted of three convolution layers and two full-connect layers, like the network commonly used in the MNIST problem. As a result, high accuracy was obtained using only 2D images from a low-cost camera with much less data size than previous studies. The overall technique of the paper is as given in the figures:

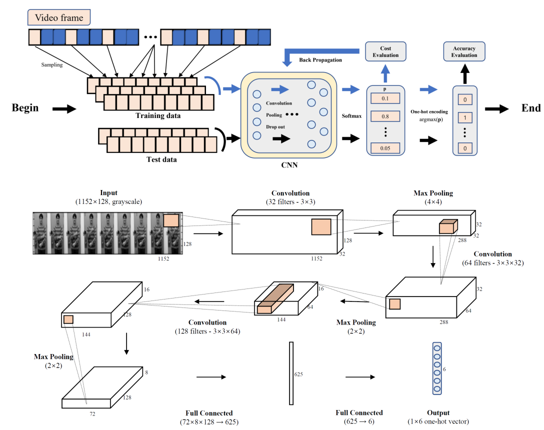

Fig. 2: Convolutional Neural Network

Again Using Kinect , HOG and SVM algorithms with the Kinect Software Libraries were used to recognize sign language by recognizing the hand position, hand shape and hand action features. Histogram of oriented gradients (HOG) is a feature descriptor used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of an image. This method is similar to that of edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts, but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy.In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting).In order to realize this method a special 3D design language dataset containing 72 words is collected with Kinect and experiments are conducted.It is shown in the experimental results that the use of HOG and SVM algorithms significantly increase the recognition accuracy of the Kinect and is insensitive to background and other factors.

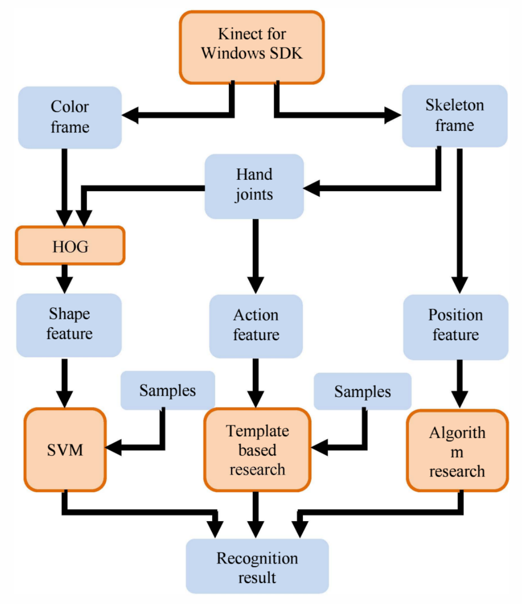

Fig. 3: Overall Methodology

An approach that gives a technique for improving Sign Language Recognition system.Use of sensors which are incorporated on a glove to detect the gestures and are converted to speech with the help of a Bluetooth module and an Android Smart phone.The gloves track three kinds of movements âĂŞ

• Finger bends using flex sensors: The flex sensors calculate the resistance for bend in each finger. The lesser the radius of bend, higher the resistance.
• Angular movement using gyroscope: The gyroscope calculates the angular movement in space by checking rate of change of angle along each axis. We require three measurements âĂŞ
◦ GyroZeroVoltage: Voltage reading when the gyroscope is still.
◦ Voltage received from the gyroscope.
◦ GyroSensitivity: Voltage sensitivity of the gyroscope.
• Orientation using accelerometer: The accelerometer calculates the orientation of the hands in space by determining the axes readings. Again, the axes reads are detected in voltage.

The proposed gesture recognition system converts Indian Sign Language to speech with the help of variety of sensors like flex sensor, gyroscope and accelerometer in order to successfully determine the position and orientation of the hand gesture.This system also aims at integrating the results of the sensor with a smart phone that map the sensor reading to a corresponding sign which is stored in a database.The output is the form of speech which can be easily understood by others.

Another approach included recognizing manual signs and finger spellings using Leap motion sensor. The sensor comes with the associated Application Programming Interface that provides an easy access to capture the 3D position of fingertips with a sampling rate of 120 fps.Raw data captured through the API of the device are then preprocessed and relevant features are extracted.BLSTM-NN is a sequence modeling classifier that has been popularly used in gesture and handwriting recognition problems. The classifier is able to process the input sequences in both directions, i.e. forward as well as backward with the help of two hidden layers. Both the layers are connected to a common output layer. The framework facilitates a signer to communicate using modalities in real-time, i.e. manual and finger-spelling. The recognition process has been done in two stages. Firstly, SVM classifier has been used to distinguish input gestures into two classes corresponding to manual and finger-spelling. In the second stage, two BLSTM-NN classifiers have been trained for recognition of distinguished gestures using sequence classification and sequence transcription based approaches. A dataset of 2240 gestures is prepared using the proposed framework.

3 Materials and Methods

It has been observed from the above literature study that Extensive work has been done on American sign languagerecognition but Indian sign language differs significantly from American sign language and this field has not been explored.Again from the various studies it has been observed that there are innumerable hearing and speech impaired Indians whose primary mode of communication is sign language.For them communicating with non-signers is a daily struggle and they are often at a disadvantage when it comes to finding jobs , accessing healtcare etc.

So our main focus in this project is, implementing techniques in machine learning and image processing , where we hope to obtain a high level accuracy in distinguishing in between letters in the English alphabets of the Indian Sign Language.

3.1 Image Data Collection

A dataset of around 767 segmented images were collected for 20 different Indian Alphabets giving each letter around 35 images.

3.2 Image Pre-processing

Image processing is often viewed as arbitrarily manipulating an image to achieve an aesthetic standard or to support a preferred reality. The human visual system does not perceive the world in the same manner as digital detectors , with display devices imposing additional noise and bandwidth restrictions. Using pre-processing steps all the images were converted to a form that would allow a general algorithm to solve it and increase the accuracy of the applied algorithm.The following pre-processing steps were applied to the segmented image collected.

3.2.1 Re-sizing: For faster computation , the original image was first re-sized into a 100X100 image using re-size function in python.

3.2.2 Skin Masking: Apart from the main object of interest, everything was made black. The RGB model consisting of Red, Blue and Green Pixel values was converted to HSV image consisting of Hue, Saturation and Intensity where Hue defines the dominant color present. The colorfulness was measured using saturation component. The intensity component was used to measure the brightness.

The RGB color space doesn't separate luminance and chrominance hence R,G and B components were found to be highly correlated. So, HSV was found to be more suitable color space for color based skin segmentation which made making the background black accurate.

To detect colors in images, the first thing we did is defined the upper and lower limits for our pixel values

Once we have defined our upper and lower limits, then make a call to the cv2.inRange method which returns a mask, specifying which pixels fall into your specified upper and lower range.

After that we detected many small false-positive skin regions in the image. To remove these small regions ,we created an elliptical structuring kernel . Then, we used this kernel to perform two iterations of erosions and dilations, respectively. These erosions and dilations helped to remove the small false-positive skin regions in the image. This smoothing step, while not critical to the skin detection process, produces a much cleaner mask.

3.2.3 Removing Arm: Arm does not play any significant role in detecting what a particular sign means. It is totally dependent on the orientation of the hands. So we cropped down 5% of the image for removing the arm so that the learning algorithm can focus only detecting patterns from hand movement.

3.2.4 Contour Detection: Contours are curves joining all the continuous points (along the boundary) having same color or intensity. It is a useful tool for shape analysis and object detection and recognition. In opencv contour image is like finding white object from black background. The largest contour was detected to find the main object we have to focus on.

The perimeter of the largest contour was found and whitened.

3.2.5 Centering Re-sizing: The main object found after contouring was centred using dimensions of the largest contour obtained from the previous step. The image obtained was again re-sized into a 30X30 image for faster training.

4 Feature Extraction Using SURF

The Speeded Up Robust Feature (SURF) technique is used to extract descriptors from the segmented hand gesture images. SURF is a novel feature extraction method which is robust against rotation, scaling, occlusion and variation in viewpoint. For orientation assignment , SURF uses wavelet responses in horizontal and vertical direction for a neighbourhood of size 6s. Adequate Gaussian weights are also applied to it. Then they are plotted in a space. The dominant orientation is established by calculating the sum of all responses within a sliding orientation window of angle 60 degrees. Wavelet Response can be found out using integral images very easily at any scale. For many applications, rotation invariance is not required , so no need of finding this orientation , which speeds up the process. Before extracting the features or key pointâĂŹs we applied some image pre processing steps as we applied previously including:

- Re-Sizing
- Making Background Black
- Making skin White.
- Removing Arm
- Contour Detection

4.1 Canny Edge Detection

Canny Edge technique is employed to identify and detect the presence of sharp discontinuities in an image, thereby detecting the edges of the figure in focus. To decide which edges are really edges and which are not we need two threshold values minVal and maxVal. Any edges with intensity gradient more than maxVal are sure to be edges and those below minVal are sure to be non-edges, so discarded. Those which lie between the two thresholds are classified edges or non-edges based on their connectivity .If they are connected to âĂIJsure-edgeâĂİ pixels , they are considered to be part of edges. Otherwise , they are also discarded.Here, for canny edge detection in case of Hand Gesture Recognition we select a minVal of a maxVal of 60 .

After Canny edge detection we resize the frame and move on to the most important part which is Feature Extraction using SURF.

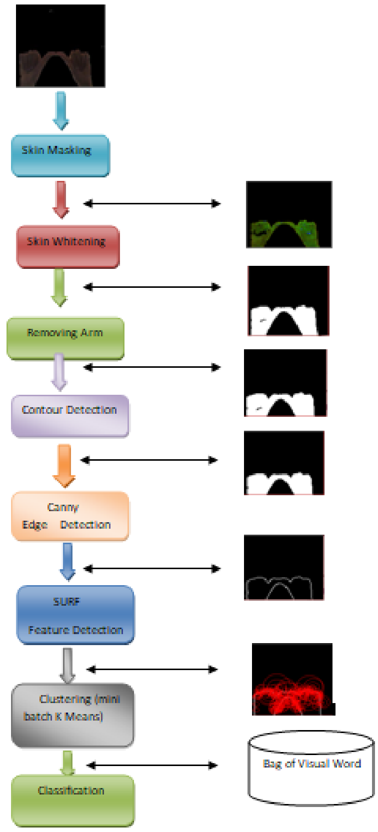

Fig. 4: Overall Methodology

4.2 Feature Extraction

SURF.detect() finds the key-point in the images .Each Key point is a special structure of which has many attributes like its(x,y) coordinates ,size of the meaningful neighborhood, angle which specifies its orientation , response that specifies strength of keypoints etc. After finding the keypoints the descriptors computed from the key-points using SURF.compute() or detect and compute the descriptors from key points directly using surf.detectandCompute() as we have used .

The SURF descriptors extracted from each image are different in number with the same dimension (64). However, a supervised Machine Learning model requires uniform dimensions of feature vector as its input.So we applied Bag of Features (BoF).

4.3 Bag of Visual Words

Bag of Features (BoF) was therefore implemented to represent the features in histogram of visual vocabulary rather than the features as proposed. SURF descriptors are 128-dimensional vectors so we simply make a matrix with every SURF descriptor in our training set as its own row, and 128 columns for each of the dimensions of the SURF Features. The descriptors extracted were first quantized into k clusters using K-means clustering. Given a set of descriptors, where K-means clustering categorizes numbers of descriptors into K numbers of cluster center. Next we went through each individual image and assigned all of its SURF descriptors to the bin they belong in. All the SURF descriptors were converted from a 128-dimensional SURF vector to a bin label. Finally we made a histogram for each image by summing the number of features .

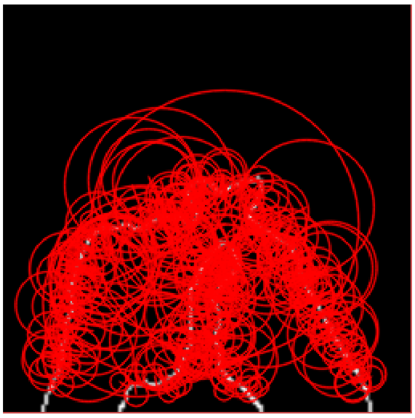

Fig. 5: Key Point Extraction

The clustered features then formed the visual vocabulary where combination of feature or visual word corresponded to an individual sign language gesture. With the visual vocabulary, each image is represented by the frequency of occurrence of all clustered features. BoF represented each image as a histogram of features, in this case the histogram of 20 classes of sign languages gestures.

K-means clustering technique categorized m numbers of descriptors into x number of cluster centre. The clustered features formed the basis for histogram i.e. Each image is represented by frequency of occurrence of all clustered features.

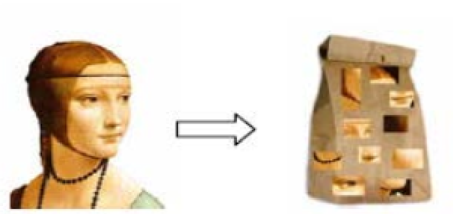

Fig. 6: Local Feature Extraction

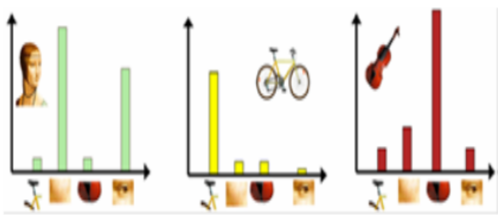

Fig. 7: Conversion of Local Feature into Histogram of frequency of feature descriptor

5 Training The Model

After applying Bag of Visual Word and feature extraction , next and the final step was training the model on the image which was trained using the following models with a cross validation value of 6:

- **Gaussian NaÃŕve Bayes:**Using NaÃŕve Bayes classifier we got the best cross validation score [0.73584906 0.79591837 0.86666667 0.88636364 0.92682927 0.92105263] for k=150.

- **Logistic Regression:** From this classifier we achieved score of [0.98113208 0.97959184 0.97777778 0.95454545 1. 0.94736842] for k=250. This was because Logistic regression gives best results for classification problem.

- **K- Nearest Neighbors:** We started simple by using K-Nearest Neighbors to train our model. We tried different values of k and the best score was [0.94339623 0.97959184 0.95555556 0.93181818 0.95121951 0.94736842]. This result motivated us to go for an advanced algorithm like SVM.

- **Support Vector Machine:** Multiclass SVM using different kernels like polynomial, rbf and linear along with different values for maximum margin (C) was tried on a flattened out vector of the images. The best result was given by polynomial kernel with C=0.1 was [1. 0.95918367 1. 0.97727273 1. 0.92105263]. This was a significant improvement on the previous result .

The results obtained were a significant improvement over all the above models implemented without applying Feature Extraction first.

6 Convolutional Neural Networks

Convolutional Neural Networks , like other neural networks are made of neurons with learnable weights and biases. Each neuron receives several inputs , takes a weighted sum over them , pass it through an activation function and responds with an output.CNNâĂŹs have a wide applications in image and video recognition , recommender systems and natural language processing. Here, the input vector is a multi-channelled image with the convolution layer is the main building block of the network which comprises of a set of independent filter initialized randomly which are learned by the network subsequently.

The architecture we used involved the following:

- Zero Padding
- Convolving
- ReLU Activation
- Max Pooling
- Flattening
- Fully Connected Neural Network
- Softmax Activation

The complete architecture can be shown as follows:

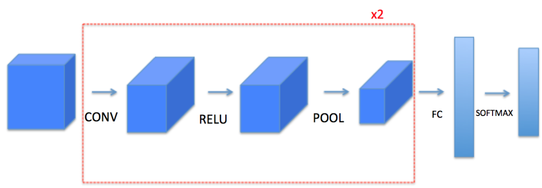

Fig. 8: ConvNet Architecture.

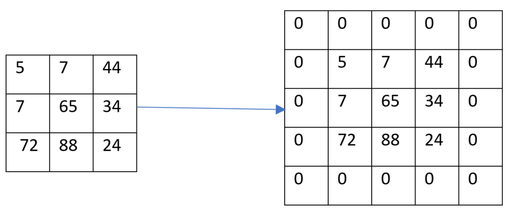

Fig. 9: Padding

6.0.1 Zero Padding: Image padding introduces new pixels around the edges of an image. The border provides space for annotations or acts as a boundary when using advanced filtering techniques. Zero-padding allows space for wrap-around to occur without contaminating actual output pixels.

In our case the input image of size 30X30 was padded with a single of zeros on all sides to give images of size 32X32.

6.0.2 Convolving: The Conv layer is the core building block of a Convolutional Network that does most of the computational heavy lifting.The CONV layerâĂŹs parameters consist of a set of learnable filters. Convolutional filtering is used to modify the spatial frequency characterstics of an image by the use of general purpose filter effect for images which are actually matrices comprised of integers. It works by determining the value of the central pixel by adding the weighted values of all its neighbors together. The output is a new modified filter image.

We used tensorflow for implementing the convolutional step. The built-in function

tf.nn.conv2d(X,W1,strides=[1,s,s1],padding = 'SAME')

where,

X=Input Image of size (613 x 30 x 30 x 1)

W1=Set of weight matrices which act as kernels each of size (4 x 4 x 1 x 8)

third input=[1,f,f,1] represents the strides for each dimension of input

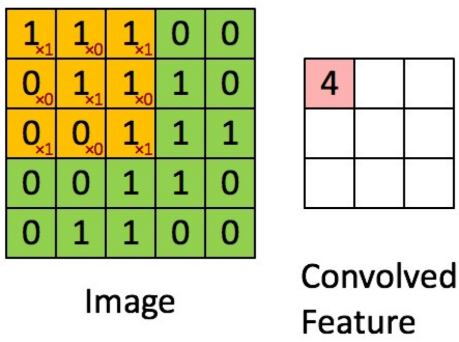

Fig. 10: Convolving

6.0.3 ReLU Activation: The Convolution layer is followed by ReLU activation function which can be described as:

$$f = \begin{cases} (x < 0) & f(x) = 0 \\ (x \geq 0) & f(x) = x \end{cases}$$

Fig. 11: ReLU Equation

The graph for the same can be depicted in in the below figure as follows:

In tensorflow the function **tf.nn.relu(Z1)** computes the element-wise ReLU of Z1 .

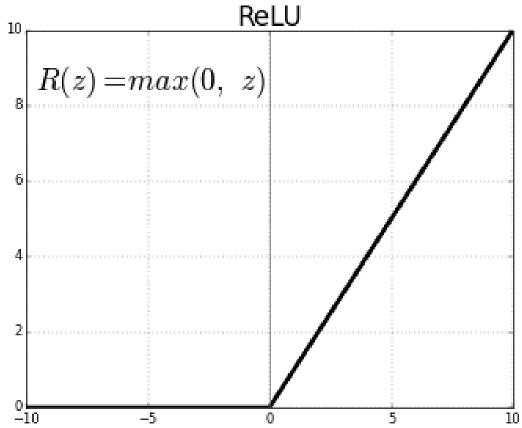

Fig. 12: ReLU

6.0.4 Max Pooling: Max pooling is a sample-based discretization process. The objective is to down-sample an input representation (image, hidden-layer output matrix, etc.), reducing its dimensionality and allowing for assumptions to be made about features contained in the sub-regions binned. The pooling (POOL) layer reduces the height and width of the input. It helps reduce computation, as well as helps make feature detectors more invariant to its position in the input. It does not use padding and only uses hyperparameters instead of parameters so no backpropagation for learning is required.In Max-pooling we slide an (f,f) window over the input and stores the max value of the window in the output.

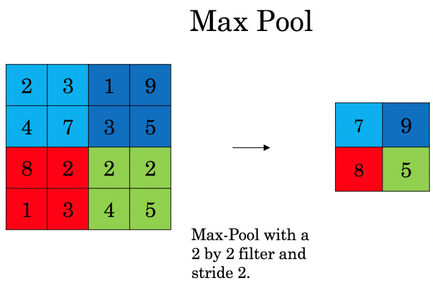

Fig. 13: Max Pooling

In tensorflow

(tf.nn.max_pool(A, ksize = [1,f,f,1],strides=[1,s,s1], padding = 'SAME')

helps us achieve the same which uses an input A and a window filter size f x f and a stride of size (s,s) to carry out max pooling over each window .

The convolution , RELU and max pooling was applied again on the consecutive inputs ,but this time the convolution was done on a new weight matrix(W2 of size (2 x 2 x 8 x 16)) whereas the window size of max pooling was of 4x4 instead of 8x8 with a stride of 4x4 .

6.0.5 Flattening: The output after applying Convolution,RELU and Max Pooling twice was flattened into a single vector to train the fully connected neural network.

6.0.6 Fully Connected Neural Network: A fully connected neural network with twenty output layers and softmax activation was trained. The output layer dimension was taken as twenty because we had one-hot encoded the letters.The fully connected neural network was trained using backpropagation and cross-entropy loss function.

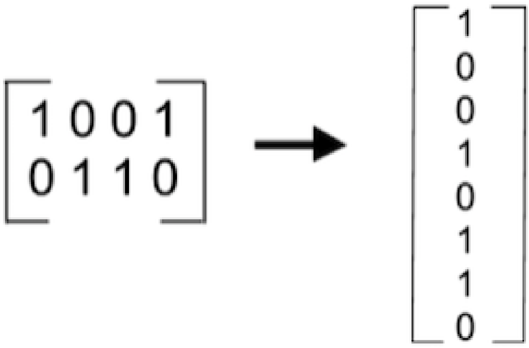

Fig. 14: Flattening

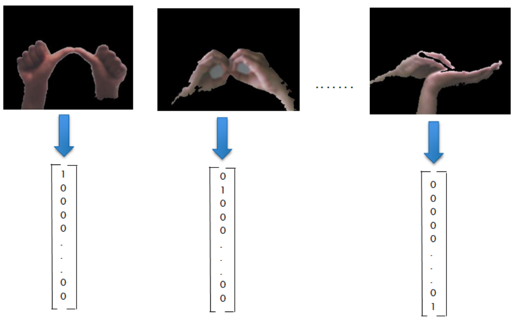

Fig. 15: One-Hot Encoding

SNo	Optimizer	Learning Rate	Epochs	Train Accuracy
1	Momentum	0.5	300	33.7
2	Momentum	0.5	265	94.12
3	Momentum	0.5	265	72.9
4	**Momentum**	**0.3**	**265**	**92.6**
5	Momentum	0.2	100	90.0
6	AdaGrad	0.5	300	91.2
7	AdaGrad	0.5	500	97.553
8	AdaGrad	0.3	500	98.2
9	AdaGrad	0.3	700	99.5
10	Gradient Descent	0.5	270	79.1

We tried out different optimizers and accuracies for each of them were reported.

7 Results

7.1 Witout Using SURF

Starting off with training our models without using Feature Extraction,the best accuracy achieved was using CNN. This section provides snippets of simulation results of some of the proposed techniques. In case of Convolution Neural Nets we had tried out various Optimizers and epochs which gave us different training and testing accuracies. The plots for the loss functions in the epochs and results for various optimizers are shown in Table 1.

The cost vs epochs graphs obtained for the trained networks shown above are as follows:

The accuracy of the various models tried without Feature Extraction is been shown as follows:

7.2 Using SURF

SURF was firstly used to detect key points and describe them because the SURF features were invariant to image scale and rotation

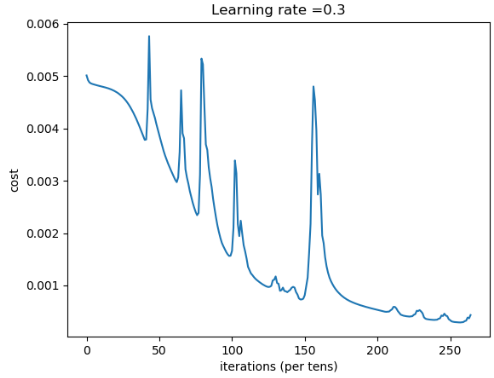

Fig. 16: Momentum

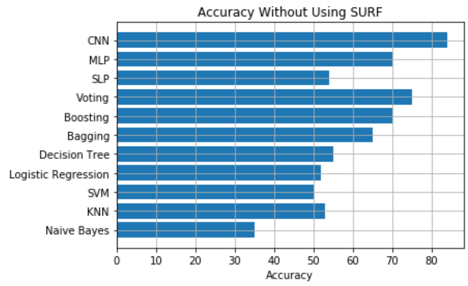

Fig. 17: Accuracy Without Using SURF

and were robust to changes in the viewpoint and illumination. SURF method used to extract all the features of certain types of signs, then formed theirs code book from all the features by using K means clustering and finally classified using KNN, SVM and other supervised models to train the model. The following figure shows the accuracy vs k value used in k means clustering in BOW algorithm

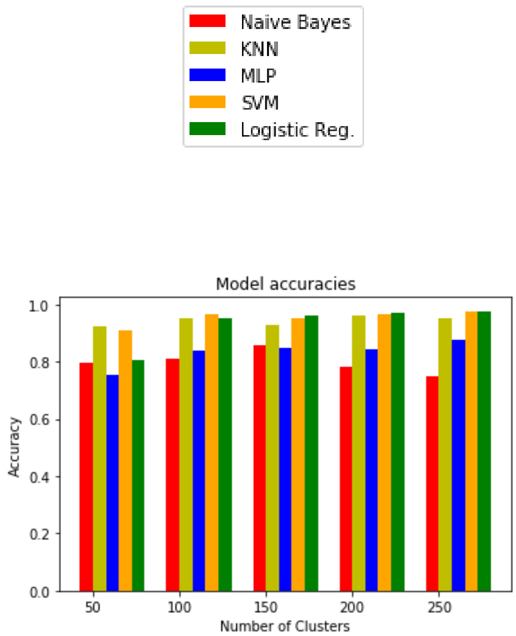

Fig. 18: Accuracy vs No. of Clusters used

As we can see the best result obtained for almost all the models was when k was set to 250 as shown in the following figure.

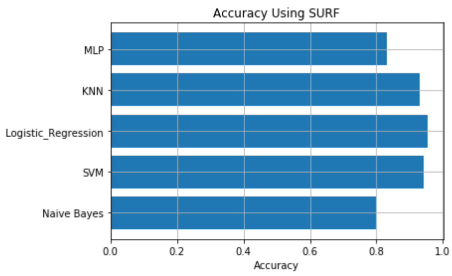

Fig. 19: Accuracy Using SURF

8 Conclusion and Future Work

In this project, attempts were made to achieve state of the art results for the Indian Sign Language like the ones that have been achieved for American Sign Language. The best accuracy was achieved by SVM after Feature Extraction using SURF which were invariant to to scaling,rotation .Codebooks were formed after feature extraction using SURF and K means .The accuracy reported here cannot be reported as a perfect representation of actual results because we are limited by data but can give us a direction as to which methods can be used when data is abundant.

The results can further be improved by collecting more data from various schools for the specially abled. CNN architectures like VGG16, Le-Net5, etc. can be tried along with various optimizers and learning rates to achieve higher accuracies. We have right now seen methods developed only on American Sign Language and tried to extrapolate it. Various methods developed on European, German, Mexican and Ukranian Sign languages can also be applied to ISL. As future work it is also planned to add to the system a learning process for dynamic signs.

References

[1] Shihab Shahriar Hazari*, Asaduzzaman, Lamia Alam and Nasim Al Goni, âĂIJDesigning a Sign Language Translation System using Kinect Motion Sensor DeviceâĂİ; International Conference on Electrical, Computer and Communication Engineering (ECCE), Bangladesh

[2] Ashish S. Nikam, Aarti G. Ambekar âĂIJSign Language Recognition Using Image Based Hand Gesture Recognition TechniquesâĂİ; 2016 Online International Conference on Green Engineering and Technologies (IC-GET)
[3] Yuqian Chen, Wenhui Zhang, âĂIJResearch and Implementation of Sign Language Recognition Method Based on KinectâĂİ; 2016 2nd IEEE International Conference on Computer and Communications: Beijing, China.Yuqian Chen, Wenhui Zhang, âĂIJResearch and Implementation of Sign Language Recognition Method Based on KinectâĂİ; 2016 2nd IEEE International Conference on Computer and Communications: Beijing, China.

[4] Yangho Ji, Sunmok Kim, and Ki-Baek Lee, âĂIJSign Language Learning System with Image Sampling and Convolutional Neural NetworkâĂİ; 2017 First IEEE International Conference on Robotic Computing.

[5] Pham Quoc Thang, Nguyen Thanh Thuy, Hoang Thi Lam, âĂIJThe SVM, SimpSVM and RVM on Sign Language Recognition ProblemâĂİ; Seventh International Conference on Information Science and Technology, Vietnam, 2017.

[6] Pradeep Kumar, Rajkumar Saini, Santosh Kumar Behera, Debi Prasad Dogra and Partha Pritam Roy, "Real-Time Recognition of Sign Language Gestures and Air-Writing using Leap Motion"; 2017 Fifteenth IAPR International Conference on Machine Vision Applications, Nagoya University, Nagoya, Japan, 2017.

[7] https://en.wikipedia.org/wiki/Indo-Pakistani$_{sign_{L}anguage}$

[8] Pradeep Kumar, Rajkumar Saini, Santosh Kumar Behera, Debi Prasad Dogra and Partha Pritam Roy, "Real-Time Recognition of Sign Language Gestures and Air-Writing using Leap Motion"; 2017 Fifteenth IAPR International Conference on Machine Vision Applications, Nagoya University, Nagoya, Japan, 2017.

[9] S Yarisha Heera, Madhuri K Murthy, Sravanti VS and Sanket Salvi"; "Talking Hands - An Indian Sign Language to Speech Translation Gloves"; International Conference on Innovative Mechanisms for Industry Applications, 2017.

[10] Brunna Carolinne Rocha Silva, Geovanne Pereira Furriel, Wesley Calixto Pacheco and Junio Santos Bulhoes; "Methodology and Comparison of Devices for Recognition of Sign Language Characters"; IEEE, 2017

[11] Michael Grif, Yuliya Manueva; "Analyses of computer Russian sign language translation system with implemented semantic analyses unit"; 2017 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON),2017.

[12] Maksym Davydov and Olga Lozynska; "Information system for translation into ukrainian sign language on mobile devices"; 2017 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), 2017.

[13] F. Mandita, T. Anwar, Hermawan, G. Kusnanto, W. E. S. Yudha, A. Hermanto and Supangat; "An overview of searchings algorithm for Indonesian to German sign language statistical machine translation"; 2017 6th ICT International Student Project Conference (ICT-ISPC), 2017.

[14] Javier Jimenez, Anabel Martin, Victor Uc and Arturo Espinosa; "Mexican Sign Language Alphanumerical Gestures Recognition using 3D Haar-like Features" IEEE Latin America Transactions, 2017.

[15] Javier Jimenez, Anabel Martin, Victor Uc and Arturo Espinosa; "Mexican Sign Language Alphanumerical Gestures Recognition using 3D Haar-like Features" IEEE Latin America Transactions, 2017.

[16] Jian Hou, Jianxin Kang, and Naiming Qi; "On Vocabulary Size in Bag-of-Visual-Words Representation" IInternational Conference on Innovative Mechanisms for Industry Applications, 2015.

[17] https://ianlondon.github.io/blog/visual-bag-of-words/

[18] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool; SURF: Speeded Up Robust Features" Katholieke Universiteit Leuven.Seventh International Conference on Information Science and Technology, Vietnam, 2013.

1.
[19] Ashwin S pol,Dr S L Nalbalwar and Prof. N.S. Jadhav; Sign Language Recognition Using Scale Invariant Feature Transform and SVM.International Journal of Scientific & Engineering Research , Volume 4 ,Issue 6, June-2013.