American Sign Language

Zainiya Manjiyani
Computer Science Department, California State University Sacramento
6000 J Street, Sacramento, CA 95819
zainiyamanjiyani@csus.edu

***Abstract*— People who have disability in speaking and hearing face lot of trouble in communicating with people around them. The only thing which distinguish them from able people is communication. If we solve that problem by interpreting sign language into the language understood by normal people we can reduce the communication gap. Formally the problem is to determine if image contains sign representation of American Sign Language(ASL) character and if yes which character it represent. The work presented here uses different classification models to interpret ASL and compare the accuracy of results.**

***Keywords*— American Sign Language, artificial intelligence, classification, convolution neural network, deep learning, machine learning, sign language recognition**

I. Introduction

There are so many people who are not able to speak and hear. They use sign languages for communication and most of us are not well versed with sign languages. This creates communication barrier between disable people and able people. To establish smooth communication between disable people and able people we need some kind of interpreter in-between that can convert sign language into the text or may be speech. Traditionally, this kind of interpretation used to be done by human but with the advancement in machine learning and artificial intelligence we can develop a software solution for it.

There are various methods available for sign language recognition. We can classify all the methods into two broad categories: vision based methods, wearable technology based methods. Vision based methods either takes images or video streams as an input, apply some preprocessing and then use machine learning and neural network models to provide output. Vision based algorithms also use probabilistic approaches such as Hidden Markov Model(HMM) or classification algorithms such as Support Vector Machine(SVM), K nearest neighbor, etc. Wearable technology based methods takes input from different type of sensors such as accelerometer, gyroscope, motion sensors, etc and apply mathematical transformation on data to prepare it for further processing. Wearable technology based methods also use machine learning to some extent. Some other techniques also uses colors to identify the sign language where user wear glove that has different color on each finger tip.

So machine learning is a crucial part of both vision based methods and wearable technology based methods. If we consider american sign language to text conversion as a vision based classification problem then there are different classification methods available. We can use any of these: Convolution Neural Network(CNN), K nearest neighbor(KNN), Support Vector Machine(SVM), Naive Bayes Classification, Gaussian Classification, etc. Accuracy of your solution varies based on the input and parameters of your machine learning model.

This paper shows the accuracy received for different classification methods when applied to ASL dataset for predicting alphabetic character from image. This article also shows how different parameters affects the model accuracy and how we tune the models. Following are the algorithms that have been applied on ASL dataset and accuracy has been calculated:

- Convolution Neural Network(CNN)
- Fully connected Neural Network
- Long Short Term Memory networks
- Ensemble methods
  - AdaBoost classifier
  - Bagging classifier
  - Random forest classifier
  - Voting classifier
- Naive Bayes
  - Gaussian Naive Bayes
  - Bernoulli Naive Bayes
  - Multinomial Naive Bayes
- K-Nearest Neighbor
- Semi-supervised learning
  - Label propagation
  - Label spreading
- Support Vector Machine(SVM)
  - Linear SVC
  - Nu SVC
  - SVC
- Decision tree
  - Decision tree classifier

○ Extra tree classifier

● Logistic Regression

Further sections in this article gives the details about each of the algorithm models, how data is prepared to give as an input to these models, how I tuned the models to make it more accurate and what are the results that I have achieved.

II. Algorithm/model design

A. Convolution Neural Network

CNN is a type of feed forward neural networks and widely used for processing visual images. It is inspired from biological image analyzing done by human brain. It consist of multilayer perceptrons that minimize the processing [3]. Following figure depicts the convolution process. Small convolution window traverse on top of the input image and calculate the pixel values for each pixel on output.

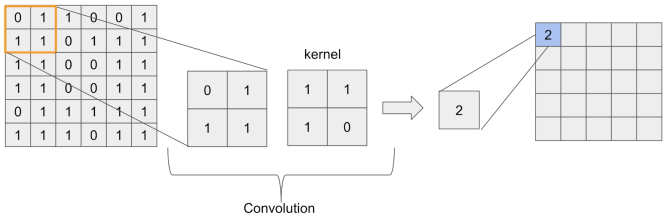

Fig. 1 Convolution process

General architecture of CNN consist of convolution layers, each convolution layer is followed by pooling layers which restricts over-fitting. After the series of convolution and pooling layer there comes layer which flattens the input to one dimension and give it finally to simple deep learning network. Following figure shows the architecture of CNN model that has been applied to ASL dataset.

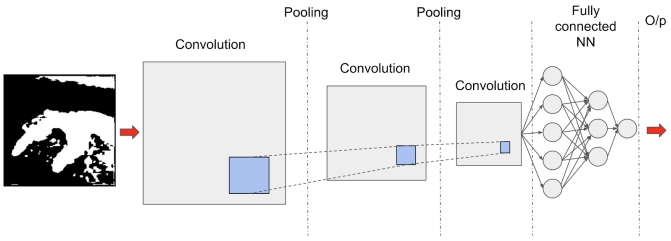

Fig. 2 CNN model

B. Fully-connected Neural Network

This is simplest type of neural network used for deep learning. It is a feed forward network in which neurons in each layer are connected to all the neurons in the next layer. During learning process it accepts one input at a time along with weights and apply processing on weights. Weights are adjusted each time and once all the inputs are analyzed the process repeats. The network trains by adjusting the weights to predict the correct class label of input samples [4].

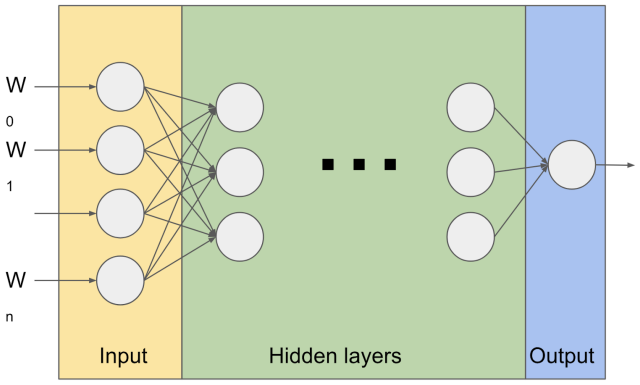

Fig. 3 Fully connected neural network

C. Long Short-Term Memory Networks

LSTM network is a special type of Recurrent Neural Network(RNN). The downside of RNN is long term dependency problem and LSTM is designed to overcome it. LSTM network is generally well suited for classification of time series data. In each LSTM cell it memorize the context and transfer some context to next cell. Operations on LSTM memory is done using gates.

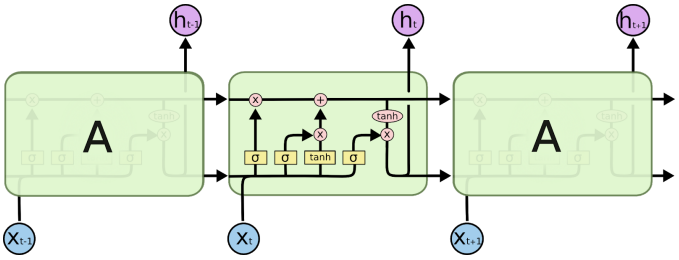

Fig. 4 LSTM network[2]

D. Ensemble Methods

Ensemble methods provides a way to combine predictions of multiple baseline algorithms and provide robust solution. There are two general ways to combine predictions: averaging and boosting. In averaging final prediction is the average of all the predictions done by different algorithms. Examples of averaging are bagging classifier, random forest classifier, etc. In boosting baseline predictions algorithms run sequentially and each algorithm tries to minimize the bias between current and previous prediction. In this way combination of several weak algorithms can be used to provide powerful result. Example of boosting is AdaBoost

E. Naive Bayes

This consist of supervised learning methods which uses Bayes theorem to find conditional independence between each pair of feature and class value. Different naive Bayes classifier makes different assumptions for calculating this independence between pair of feature and class label. There are three naive Bayes method used here: Gaussian Naive Bayes, Bernoulli Naive Bayes and Multinomial Naive Bayes.

F. K-Nearest Neighbor

K-nearest neighbour is a supervised learning algorithm which is used for both classification as well as regression, but it is widely used for classification. To predict the class label we find predefined number of sample closest to that point and based on the label of those samples we predict label for new point. Number of samples is determined by parameter "k" that is why this method is called K-nearest neighbor algorithm.

G. Support Vector Machine

SVM is a supervised learning methods which can be used for both classification and regression but it is widely used for classification problems. Training data are represented as a points in n-dimensional space where n is a number of features. The idea here is to find a hyperplane in n dimensional space that can distinctly classify the data points. "Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes"[5]. Support vector closest data points from the hyperplane and they affect the geometry of the hyperplane. I have used SVC, Nu SVC and Linear SVC methods.

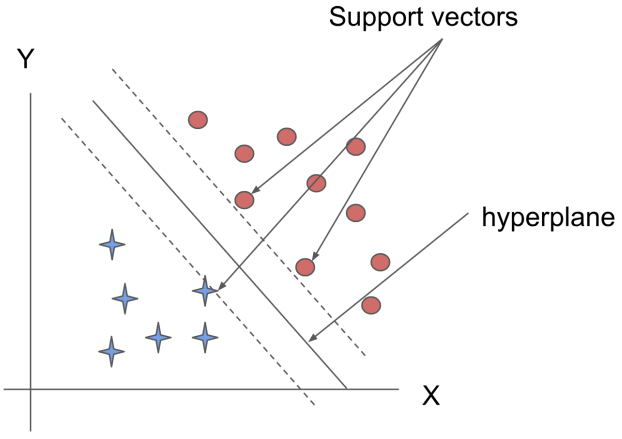

Fig. 5 SVM hyperplane classification

H. Logistic Regression

Despite its name this method is used as a linear model for classification rather than regression. The core of logistic regression is logistic function which is an equation, mostly linear. Input values and associated weights (or coefficients) are given as an input to the equation to predict output value.

I. Other Methods

Other than methods and models described above I have also used semi-supervised learning and decision tree based methods. Semi-supervised learning methods are used when there are some class labels missing in the training dataset. There are two semi-supervised learning methods used in this work: Label propagation and Label spreading. Decision tree methods are supervised learning methods used for classification as well as regression. During training it infers decision rules from training data and make decision tree using it. The deeper the tree the more complex the decision tree rules are and fitter the tree. At the time of prediction it parse the data using decision tree. I have used decision tree classifier and extra tree classifier in this project.

III. Methodology

This section describes operations that are being done on the input to produce the output. I have used American Sign Language image dataset that consist of 3000 unlabeled images. Some of them were taken in illuminated room and some of them were taken when there was a less light. For this project I have focused on less illuminated images. I have used 50 images for training and 10 images for testing for CNN, LSTM and neural network models and for all the other models I have used 100 images for training and 20 images for testing.

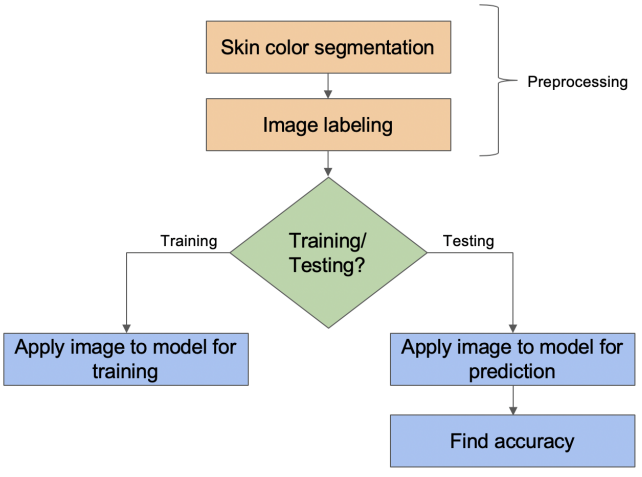

Fig. 6 Process flow

Initially images are in RGB format and when read into the program it creates three 200 X 200 pixel images each representing R, G and B respectively. I have applied skin color segmentation based on specific color range in YCrCb color format. I have taken lower bound on YCrCb image as (0, 133, 77) and upper bound as (255, 173, 127). The result of a image segmentation is black and white binary image as shown in Fig. 7. The image is first converted to YCrCb image using OpenCV and then pixel value that comes in (0, 133, 77) to (255, 173, 127) range in YCrCb are then taken as value 255 which represents white color and other pixel values are set 0 as black.

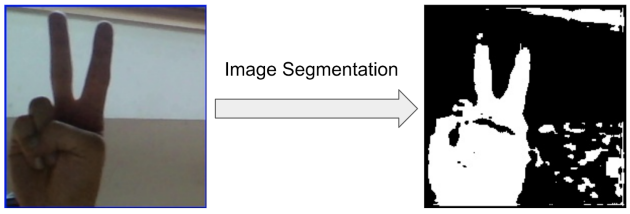

Fig. 7 Skin color segmentation

After skin color segmentation each image is labeled with specific alphabetic character it represents. The labeled images are then used either for training or for testing. Images are applied to any of the models described in section II. After testing accuracy of model is found out along with F1 score, precision and R2 score.

IV. Results

This section describes the accuracy that we have achieved using each model and conclusions that we can draw.

A. CNN

The following table shows the result of CNN when model is tuned using relu activation and adam optimizer function.

TABLE I
Results of Cnn

No	CNN L1		CNN L2		Number of hidden layers	Accuracy
	Kernel	Filter	Kernel	Filter		
1	(3,3)	200	(1,2)	167	1	75.38%
2	(3,3)	200	(1,2)	500	4	71.15%
3	(10,10)	200	(5,5)	300	3	74.62%

In the above experiment for result 1 number of neurons in hidden layer is 92 and acquired precision 0.88, recall 0.75 and f1 score of 0.74. For result 2 number of neurons in hidden layers H1, H2, H3, H4 are 97, 254, 134, 62 respectively and obtained precision of 0.79, recall 0.71 and f1 score 0.69. For result 3 number of neurons in hidden layers H1, H2, H3 are 254, 134, 62 respectively and obtained precision of 0.81, recall 0.75 and f1 score 0.71. From these results we can conclude that increase in number of hidden layers reduces the accuracy. Confusion matrix for CNN is shown in Fig. 8.

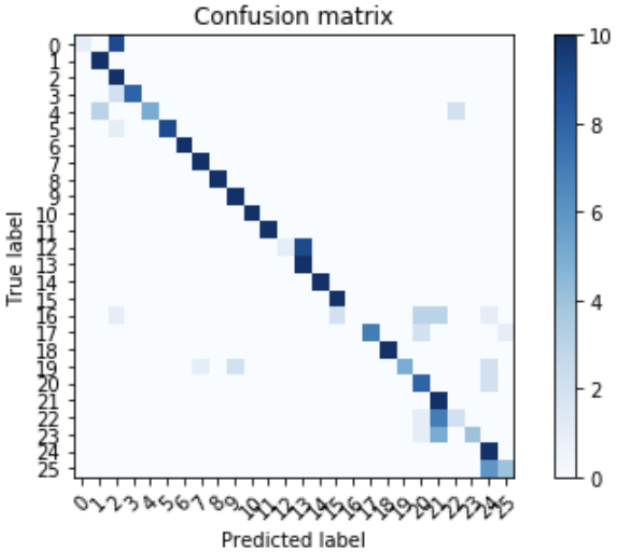

Fig. 8 CNN Confusion matrix

B. LSTM

The following table shows the result of LSTM when model is tuned using sigmoid activation and rmsprop optimizer function.

TABLE II
Results of Cnn

No	LSTM	Number of hidden layers	Accuracy
1	128	1	68.07%
2	128	2	50%
3	128	3	26%
4	435	2	48.07%

In the above experiment for result 1 number of neurons in hidden layer is 30 and acquired precision 0.71, recall 0.67 and f1 score of 0.64. From these results we can conclude that increase in number of hidden layers reduces the accuracy. Confusion matrix for LSTM is shown in Fig. 9.

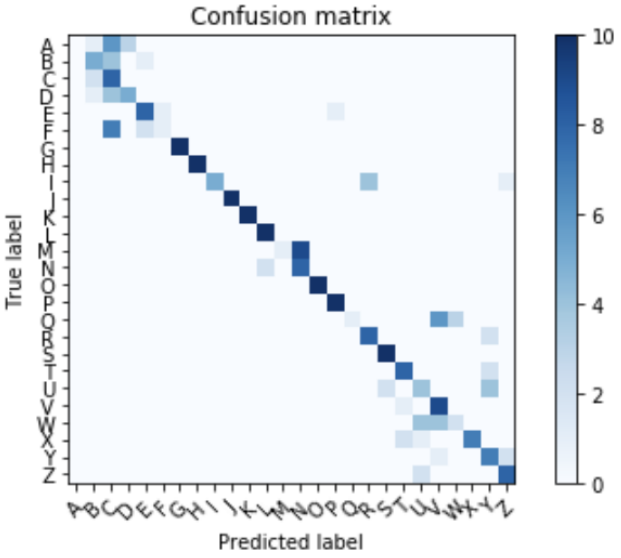

Fig. 9 LSTM Confusion matrix

C. *Fully Connected Neural Network*

The following table shows the result of Fully Connected Neural Network when model is tuned using relu activation and adam optimizer function.

TABLE III
RESULTS OF FULLY CONNECTED NEURAL NETWORK

No	H1	H2	H3	H4	H5	Accuracy
1	100	150	60	30		71.15%
2	200	500	150	97	30	75%
3	200	500	97			76.53%
4	400	656	379	147		74.23%

In the above experiment for result 1 acquired precision is 0.75, recall 0.71 and f1 score of 0.68. For result 2 obtained precision 0.84, recall 0.75 and f1 score 0.73. For result 3 obtained precision 0.78, recall 0.77 and f1 score 0.74. For result 4 obtained precision 0.82, recall 0.74 and f1 score 0.72. Confusion matrix for Fully Connected Neural Network is shown in Fig. 10.

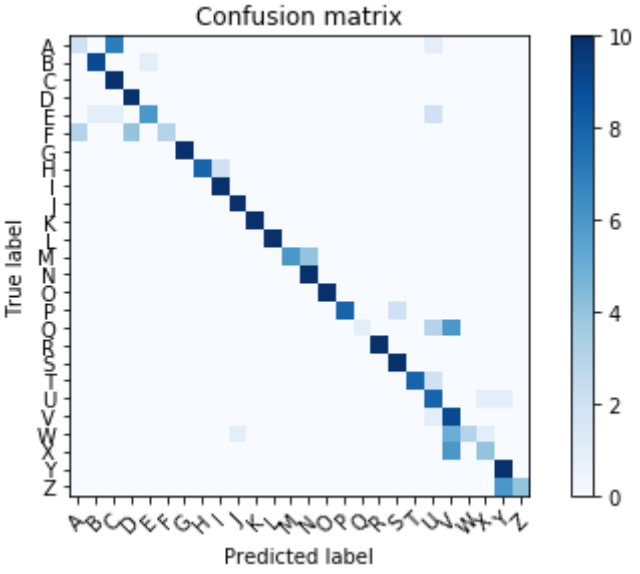

Fig. 10 Fully Connected Neural Network Confusion matrix

D. *Combination of CNN, LSTM and NN*

The following table shows the result of Fully Connected Neural Network when model is tuned using sigmoid activation and rmsprop optimizer function.

TABLE IV
RESULTS OF COMBINED CNN, LSTM AND NN

No	CNN		LSTM	Number of Neurons in Hidden Layer	Accuracy
	Kernel	Filter			
1	3	200	128	107	73.40%
2	15	250	167	83	58.46%
3	9	200	128	60	68.46%

In the above experiment for result 1 acquired precision is 0.71, recall 0.73 and f1 score of 0.7. Confusion matrix for LSTM is shown in Fig. 11.

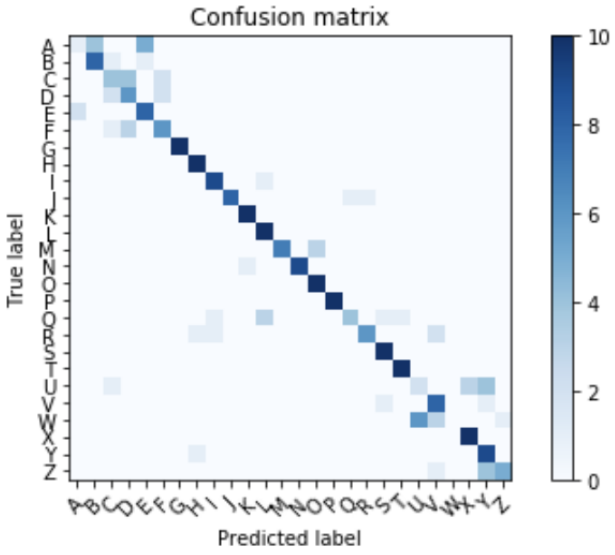

Fig. 11 Combination of CNN, LSTM and NN Confusion matrix

E. *Ensemble Methods*

Results obtained for Ensemble Methods are shown in Fig. 12.

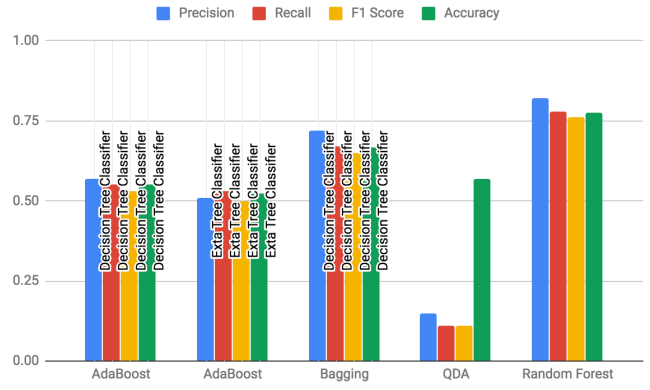

Fig. 12 Results of Ensemble Methods

From the results shown in Fig. 12 we can conclude that random forest gives the highest accuracy of 77.69%.

F. *Naive Bayes*

Results obtained for Naive Bayes are shown in Fig. 13.

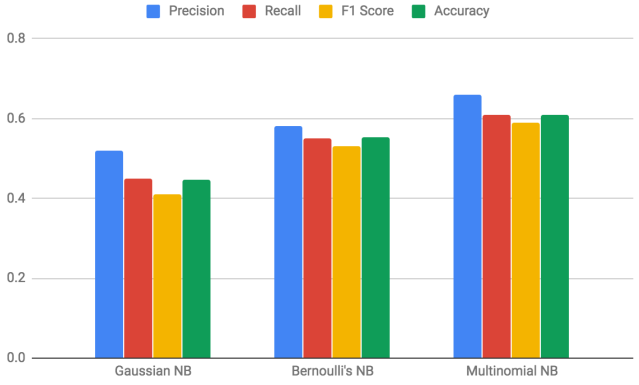

Fig. 13 Results of Naive Bayes

From the results shown in Fig. 13 we can conclude that Multinomial Naive Bayes gives the highest accuracy of 60.96%.

G. *K-Nearest Neighbor*

The following table shows the result of K-Nearest Neighbor when model is tuned using 1 neighbor. When I tried to increase number of neighbors the accuracy was descreasing.

TABLE V
RESULTS OF K NEAREST NEIGHBOR

Precision	**Recall**	**F1 Score**	**Accuracy**
0.69	0.59	0.58	58.65%

H. *Semi-supervised*

Results obtained for Ensemble Methods are shown in Fig. 14. Kernel used is KNN and N-neighbors for label propagation is 2 and for label spreading is 3.

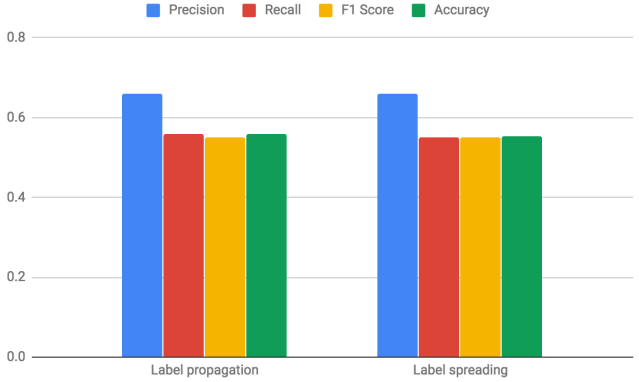

Fig. 14 Results of Semi-supervised

I. SVM

Results obtained for SVM are shown in Fig. 15.

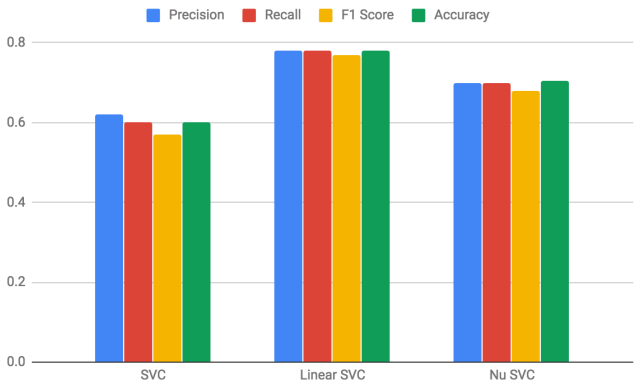

Fig. 15 Results of SVM

From the results we can conclude that highest accuracy is obtained from Linear SVC as 77.88%.

J. Decision Tree

Results obtained for Decision Tree are shown in Fig. 16. Maximum leaf node used is 520.

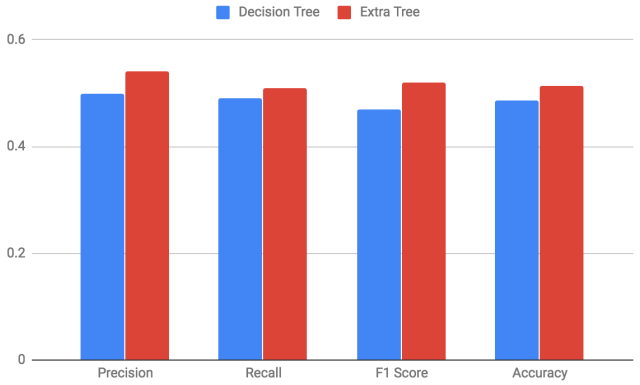

Fig. 16 Results of Decision Tree

K. Logistic Regression

Results obtained for Logistic Regression are as shown in the table.

TABLE VI
RESULTS OF LOGISTIC REGRESSION

Precision	Recall	F1 Score	Accuracy
0.77	0.77	0.75	76.53%

From all the results above we can conclude that Linear SVC (77.88%), Random Forest (77.69%) and Neural Network (76.53%) are top three accuracies among all the classification models.

V. RELATED WORK

In [1] authors have implemented American Sign Language recognition. The problem they have addressed is same as presented in this work. For the image preprocessing skin color segmentation is used which is also used by me in this project. Authors of [1] have used SVM for predicting alphabet character from image and have received 89.54% accuracy in real-time environment. However their solution have limitation that input image should be taken in bright illuminated area. Also they have only done recognition using SVM and not provided strong reason that why they haven't used any other model. Whereas I have implemented different machine learning models and compared accuracy. The best accuracy I received from --- model is still less than accuracy authors have received in [1] because I have used images taken in dark or less illuminated area.

REFERENCES

[1] S. Lahoti, S. Kayal, S. Kumbhare, I. Suradkar, V. Pawar, "Android based American Sign Language Recognition System with Skin Segmentation and SVM", *9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* , pp 1-6, 2018

[2] Colah, Understanding LSTM Networks. [Online] Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[3] Wikipedia, Convolutional neural network. [Online] Available: https://en.wikipedia.org/wiki/Convolutional_neural_network#cite_note-LeCun-1

[4] Neural Network Classification [Online] Available: https://www.solver.com/xlminer/help/neural-networks-classification-intro

[5] R. Gandhi, Support Vector Machine—Introduction to Machine Learning Algorithms [Online] Available: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

INDIVIDUAL CONTRIBUTION

- Prepared proposal
- Did state of art study
- Implemented machine learning models and found accuracy.
- Prepared report and presentation and visual aids.
- Did model tuning to speed up the learning process.

LEARNING

- When I was using RGB image directly the training process was taking too long to get completed and also accuracy of the model was low. So I did skin color segmentation that converted RGB image to binary image. That increased speed of training process as well as increased accuracy.
- Implemented combination of CNN, LSTM and Fully connected neural network models and achieved good accuracy
- Learned different models available for classification
- Learned how number of layers affects the accuracy of model