International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 03 | Mar-2018 www.irjet.net p-ISSN: 2395-0072

ASL FINGERSPELLING INTERPRETATION USING SVM AND MLP

Meena Ugale[1], Snehal Bende[2], Manasi Haldankar[3], Sweekritha Hegde[4]

[1] Asst. Professor, Dept. of Information Technology, Xavier Institute of Engineering, Mumbai, Maharashtra, India
[2,3,4] Dept. of Information Technology, Xavier Institute of Engineering, Mumbai, Maharashtra, India

---

Abstract – *ASL stands for American Sign Language. ASL Fingerspelling Interpretation is a system which is designed for the normal people to communicate with deaf and dumb people. Sign Language is a well- structured code gesture where every gesture has a meaning associated with it. Sign Language uses various shapes, movements and orientations of hands instead of voice or sound patterns. The ASL Interpreter works by capturing a picture of the hand gesture through an application, classifies this image based on neural networks in python followed by prediction and then displaying alphabet on the screen.*

Key Words: Hand gestures, Dataset, Classification, SVM, PHP, Android.

1. INTRODUCTION

Sign Language is a visual language. It is combination of various movements, shapes and orientations of hands. The deaf and dumb community cannot communicate with the outside world using spoken language. Sign language is not a universal language, every country has its own sign language. USA and Canada use ASL for communication amongst the deaf and dumb community. ASL fingerspelling is referred to as the American Manual Alphabet which is used to spell out the 26 different alphabets of the English language. It has its own set of rules and syntax. [1]

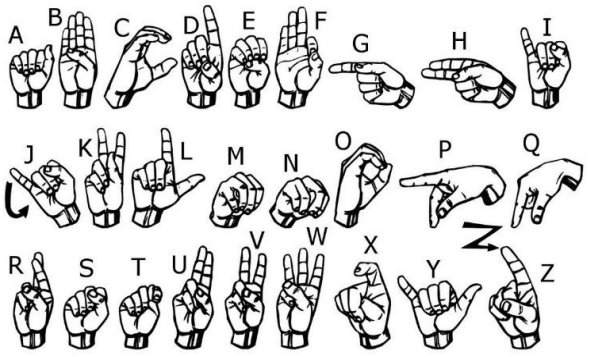

Fig -1: American Sign Language Alphabets. [5]

Figure-1 shows orientation of 26 alphabets of ASL fingerspelling. ASL fingerspelling interpreter is mainly based on neural networks for sign language recognition. A dataset is created which consist of 30 images per alphabet i.e. a total of 20 images are present in the dataset. Only static gestures are considered, the letters 'j' and 'z' are not used as they involve motion. An android application is built which is used for capturing the picture of the hand gesture. The captured image is converted into gray-scale images and then is classified. Here two methods are used for classification Support Vector Machine(SVM) and Multi-Layer Perceptron(MLP). The purpose of using different methods is to check which method gives better accuracy for prediction of alphabets.

2. METHODS

Following are the methods and algorithms used in ASL Fingerspelling Interpretation:

2.1 Image Processing

Image processing is used for performing operations like converting RGB images into gray scale images.

2.2 Classification Algorithms

2.2.1 Support Vector Machine(SVM):

SVM is a supervised learning algorithm used for outlier detection, regression and classification [2]. In SVM one of the main advantage is that different kernels can be used for decision functions. SVM is implemented in python by simply using scikit-learn. In SVM, SVC is used as it is capable of performing multi-class classification on a dataset [2].

2.2.2 Multi-Layer Perceptron(MLP):

MLP is used for deep learning in neural networks. MLP is helpful in learning non-linear models [3]. MLP is also implemented in python by using sklearn. neural_ network. MLP Classifier uses back propagation for training. MLP Classifier also helps to choose solver, number of hidden layers and learning rate that best classifies the dataset.

3. SYSTEM DESIGN

Figure-2 depicts the system design for the ASL Fingerspelling Interpretation

3.1 Android Application

Android application is used to capture an image, transfer the image to classify in python. For connectivity between android application and python code it is necessary that both server and client are on same network.

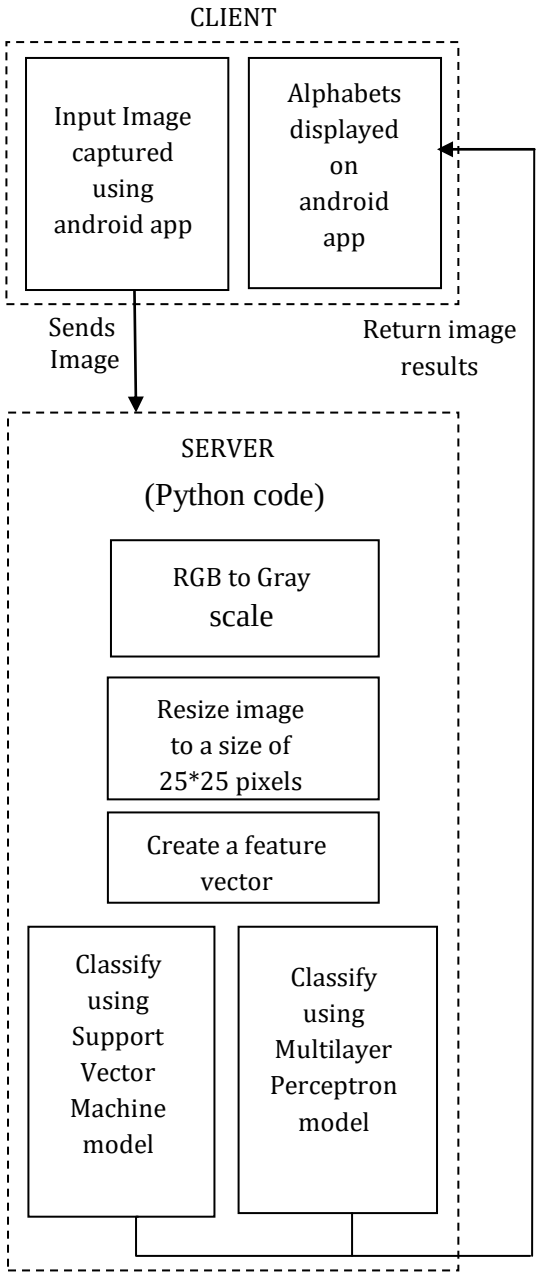

Fig -2: System Design of ASL fingerspelling interpreter

3.2 Image Processing

The dataset consists of 24 American sign language alphabets of different people, thus giving images of different size, shape, orientation of hand gestures. The images are converted into grayscale and resized in MATLAB. Then the feature vectors of the processed images are stored in a CSV file. The captured image is processed and classified in python.

3.3 Classification

In both the cases the dataset is split into training set and testing set using sklearn.model_selection.

3.3.1 Support Vector Machine(SVM):

Steps for implementing SVM:

1. import SVC from sklearn.svm
2. Select kernel using SVC.
3. Train images using fit().
4. Predict the alphabet using predict().
5. Get the accuracy from classification report.

3.3.2 Multi-Layer Perceptron(MLP):

Steps for implementing MLP:

1. Import MLPClassifier from sklearn.neural_network.
2. Set solver, number of hidden layers and learning.
3. Train images using fit().
4. Predict the alphabet using predict().
5. Get the accuracy from classification report.

The predicted results are sent to android application and are displayed on the screen.

4. IMPLEMENTATION

4.1 Training phase

4.1.1 Dataset and image processing

Dataset of 720 images is created on a black background to avoid segmentation problem. The dataset images are processed in MATLAB. They are converted into grayscale images using the rgb2gray. These grayscale images are then resized to 25*25 pixels using imresize. The grayscale dataset images are stored in an array. The values obtained are converted into a .csv file using the dlmwrite function. This CSV file of dataset is used for input of 720 image values in python.

4.1.2 Training

The CSV file is used as an input to the models used for training. The dataset is first split into training and testing sets. After splitting the dataset classification is carried out using SVM and MLP. In SVM linear kernel is used. In MLP 500 hidden layers are considered and 'lbgfs' solver is used as it is an optimizer [4].

4.2 Testing Phase:

Testing was performed on the test set to get accuracy of the training models. 20% images were taken from dataset for test set.

5. RESULTS

From 720 images 20% (144 images) images were taken for testing. Each class in the table 1 and table 2 represents an alphabet. Precision is the ratio of true positives to sum of true positives and false positives. Recall is the ratio of true positives to sum of true positives and false positives. F1-score represents harmonic mean of recall and precision. Support is number of occurrence in each class.

For finding the accuracy in SVM model steps mentioned in section 3.3.1 are followed. In SVM, parameter random_state is considered as 0. Linear kernel gives best results for small datasets having features which are linearly separable.

Table -1: Support Vector System Report

Class	Alphabet	Precision	Recall	F1-score	Support
1	A	0.83	0.83	0.83	6
2	B	1.00	1.00	1.00	8
3	C	1.00	1.00	1.00	6
4	D	0.75	1.00	0.86	3
5	E	0.67	1.00	0.80	2
6	F	1.00	1.00	1.00	6
7	G	0.80	0.80	0.80	5
8	H	0.86	0.86	0.86	7
9	I	1.00	1.00	1.00	9
10	K	0.71	1.00	0.83	5
11	L	1.00	1.00	1.00	6
12	M	0.60	0.60	0.60	5
13	N	0.71	0.71	0.71	7
14	O	1.00	1.00	1.00	6
15	P	1.00	1.00	1.00	5
16	Q	1.00	1.00	1.00	6
17	R	1.00	0.92	0.96	12
18	S	1.00	0.86	0.92	7
19	T	0.83	0.83	0.83	6
20	U	1.00	0.88	0.93	8
21	V	0.80	0.80	0.80	5
22	W	1.00	1.00	1.00	5
23	X	1.00	1.00	1.00	5
24	Y	1.00	0.75	0.86	4
Avg/total		0.92	0.91	0.91	144

The accuracy obtained in SVM is 91.0%.

For finding the accuracy in MLP model steps mentioned in section 3.3.2 are followed. In MLP model the learning rate parameter alpha is considered as 1e-5. The random_state parameter for MLPCLassifier is set as 1.

Table -2: Multilayer Perceptron Report

Class	Alphabet	Precision	Recall	F1-score	Support
1	A	0.57	0.80	0.67	5
2	B	0.60	1.00	0.75	3
3	C	1.00	1.00	1.00	8
4	D	0.67	0.80	0.73	5
5	E	0.88	0.70	0.78	10
6	F	0.75	1.00	0.86	6
7	G	0.67	0.80	0.73	5
8	H	0.83	0.83	0.83	6
9	I	1.00	1.00	1.00	5
10	K	0.71	0.71	0.71	7
11	L	1.00	0.88	0.93	8
12	M	0.50	0.60	0.55	5
13	N	0.60	0.50	0.55	6
14	O	1.00	0.75	0.86	4
15	P	1.00	1.00	1.00	7
16	Q	1.00	1.00	1.00	9
17	R	0.80	0.67	0.73	6
18	S	0.57	0.57	0.57	7
19	T	0.83	0.71	0.77	7
20	U	0.67	1.00	0.80	6
21	V	1.00	0.71	0.83	7
22	W	1.00	0.67	0.80	6
23	X	1.00	0.75	0.86	4
24	Y	1.00	1.00	1.00	2
Avg/total		0.83	0.81	0.81	144

The accuracy obtained from MLP is 80.56%.

The training process of SVM is much faster than MLP. In MLP 500 hidden layers are used which reduces the speed of training process. It is observed that accuracy of SVM is more than MLP.

6. CONCLUSION

ASL Fingerspelling interpretation represents a solution that is useful for communicating with the dumb and deaf people. Two classification models are used among which SVM gives better accuracy. SVM uses less memory as compared to MLP and is faster. Various values for parameters in both the models were tested on trial and errors basis, and the values which resulted in better accuracy were selected for implementation. 91% accuracy is achieved from SVM and 80.56% accuracy is achieved from MLP. This solution is cost effective. It can further be used for classification of other sign languages around the world.

REFERENCES

[1] H. M. Ewald, Ishan Patil and Shalini Ranmuthu,"ASL Fingerspelling Interpretation on Android", in Stanford University.

[2] http://scikit-learn.org/stable/modules/svm.html [Accessed, March 8,2018]

[3] http://scikit-learn.org/stable/modules/neural_networks_supervised.html [Accessed, March 12,2018]

[4] http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html [Accessed, March 12,2018]

[5] https://qualityanswersingservice.com/american-sign-language-guide/ [Accessed, March 20,2018].