SIGN LANGUAGE GESTURE RECOGNITION

Kaustav Vats
CSE Department
IIIT Delhi
New Delhi, India
kaustav16048@iiitd.ac.in

Lakshya Bansal
CSAM Department
IIIT Delhi
New Delhi, India
lakshya16240@iiitd.ac.in

Deepanshu Badshah
ECE Department
IIIT Delhi
New Delhi, India
deepanshu16144@iiitd.ac.in

I. PROBLEM STATEMENT

This project aims to compare and analyze the results and performance of different classifiers(SVM, RF, LR, CNN etc.) with different feature extraction technique for hand gesture recognition.

II. LITERATURE REVIEW

Sign language recognition and gesture recognition are two major applications for hand gesture recognition technologies. The main goal of sign language recognition is to automatically interpret sign language to help the deaf and dumb people to communicate among themselves or with normal people conveniently.

The current ways, to the best of our knowledge, to perform hand gesture and sign language recognition mainly focus on pre-processing the data and applying CNN, ANN and SVM with different changes for recognition. The following are the papers which we are referring for our project:

[1] recognizes using various feature extraction techniques like shape descriptors, SIFT and HOG individually along with SVM classifier.

[2] uses CNN (max pooling strategy) with dropout to classify images of both the the letters and digits in American Sign Language.

[3] uses CNN with stochastic pooling strategy for classification of gestures in selfie videos. CNN training is performed with 3 different sample sizes, each consisting of multiple sets of subjects and viewing angles.

III. DATASET

A. Argentinian Sign Language Dataset - 1

Dataset includes 3200 videos of 10 non-expert subjects doing 5 repetition of 64 different hand signs. Some of the common LSA hand signs were used for creating dataset. Data is divided into two sets:

- The first set contains 23 one-handed signs. Recording for this set was done in natural lighting.
- The second set contains 41 signs, 22 two-handed and 19 one-handed signs. Recording for this set was done in an indoor environment, with artificial lighting.

For both sets all subjects wore black clothes with a white background. To simplify segmentation subjects also wore fluorescent colored gloves. These simplified the dataset pre-processing by removing issues related to different skin color.

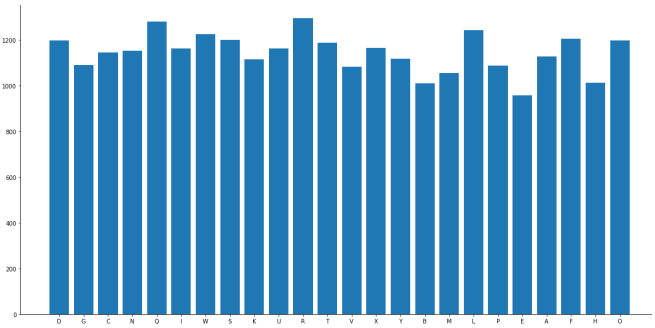
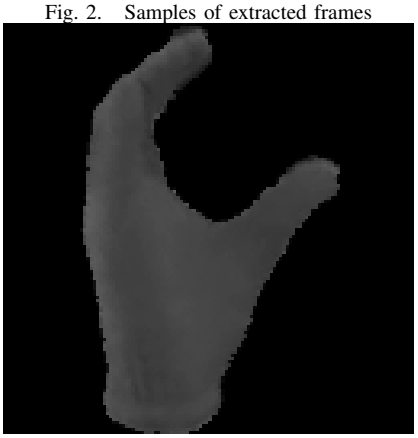


Fig. 1. Caption

Fig. 2. Samples of extracted frames



IV. PROPOSED ALGORITHM

1) We have collected videos from Argentinian Sign Language data set.
2) The relevant frames are extracted from each video.
3) Each frame is converted into grey-scale
4) Bag of words model is implemented over the extracted frames of the training videos and the histograms are computed (explained in the following paragraph).
5) SVM is trained over the training histograms computed above and classification is done on the test histograms computed from the test videos.

Feature Extraction:

1. Bag of words model for the Argentinian dataset : Sift key-points are computed for each frame and k-means clustering is applied with k = 20 clusters. We now get a bag of 20 words. Now, for each video, each Sift key-point

Fig. 3. Kaggle Mnist Dataset

Fig. 4. Kaggle Mnist data in gray scale

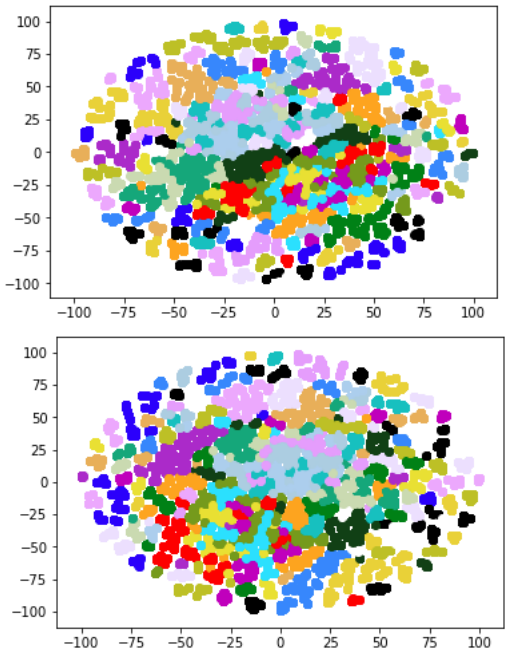


Fig. 5. T-SNE Data Visualization - original data (top), PCA reduced data (bottom)

belongs to one of these 20 clusters. Hence, for each video, we compute a histogram of these words, and this histogram is taken as the feature for this video and we have performed various techniques to understand the features variations by dimentionality reductions.

2. For Kaggle MNIST dataset, we have used sift and HOG techniques for the feature extraction and obtained images data with lesser number of features. For each feature extractor, the following is done:

- Only PCA is applied over the features.
- Only LDA is applied over the features.
- LDA is applied after PCA
- PCA is applied after LDA.

The above 4 techniques are tested for each of the 4 classifiers that we have used. The classification accuracy obtained and the interpretation is reported in the results section.

V. CONCLUSION

- HOG features were performing better than the Bag of visual words.
- Random forest's accuracy was highest with complete features(Accuracy was reduced after PCA and LDA)
- Applying PCA projects the data into dimension of higher variance. LDA projects data into projection where the inter-class distance is maximum and intra class distance is minimum. However, this is dependent on the number of discriminant used since less number of discriminant are used then dimnesions might not have inter-class sepearability and the classifier accuracy may reduce.

VI. RESULTS

The results obtained on the mnist dataset of hand gestures using various classifiers with HoG feature extractor are reported in tables 6, 7, 8, 9 .

We observed that Bag of visual words was not performing well for LSA64 dataset. We got around 24.32% Accuracy on Validation dataset. Some deep learning architecture can be used for Videos. It is important to note that :

- It can be noted that apply PCA to the data results in an accuracy greater than or equal to the accuracy obtained when the original data is classified. Hence, the projected data has better variance of data. This can be seen from 5 where in the original data, the labels can be seen to be spread out randomly in the plot. However, once PCA is applied, data points of a label can be seen closer together in the plot.
- Applying LDA (20 linear discriminators) increases the accuracy in 3 out of 4 (except Logistic Regression) classifiers used .
- Applying LDA after PCA decreases the accuracy of Random forest and Logistic Regression, meaning that the inter class distance in the resulting projection of the data is worse than with PCA.

	N components	Training Accuracy (%)	Testing Accuracy (%)
PCA	192 (no reduction)	81.23	61.85
	180	91.90	73.25
	100	91.45	76.28
LDA	1	21.12	19.38
	10	84.72	74.39
	20	93.81	82.75
	23 (upper bound)	95.01	81.88
PCA → LDA	100 (PCA), 20 (LDA)	90.12	78.20
LDA → PCA	20 (LDA), 10 (PCA)	84.72	73.29

Fig. 6. Accuracies obtained by using Gaussian Naive Bayes

	N components	Training Accuracy	Testing Accuracy
PCA	192 (no reduction)	98.77	87.15
	180	98.77	87.06
	100	98.35	86.01
LDA	1	20.30	15.85
	10	86.90	74.37
	20	97.15	87.16
	23 (upper bound)	98.10	81.92
PCA → LDA	100 (PCA), 20 (LDA)	94.50	77.76
LDA → PCA	20 (LDA), 10 (PCA)	90.12	78.20

Fig. 7. Accuracies obtained by using Logistic Regression

	N components	Training Accuracy	Testing Accuracy
PCA	192 (no reduction)	100	91.23
	180	100	89.36
	100	100	90.40
LDA	1	99.91	15.01
	10	100	76.85
	20	100	84.31
	23 (upper bound)	100	83.11
PCA → LDA	100 (PCA), 20 (LDA)	100	85.52
LDA → PCA	20 (LDA), 10 (PCA)	100	77.14

Fig. 8. Accuracies obtained by using, Random Forest

	N components	**Training Accuracy**	**Testing Accuracy**
PCA	192 (no reduction)	84.70	73.43
	180	85.43	74.14
	100	89.73	80.34
LDA	1	24.10	19.32
	10	98.61	79.71
	20	99.93	87.11
	23 (upper bound)	99.96	88.44
PCA → LDA	100 (PCA), 20 (LDA)	99.10	83.40
LDA → PCA	20 (LDA), 10 (PCA)	98.67	79.82

Fig. 9. Accuracies obtained by using SVM

REFERENCES

[1] Juhi Ekbote, Mahasweta Joshi, Indian Sign Language Recognition using SVM and ANN classifiers

[2] Vivek Bheda, N. Dianna Radpour, Using Deep Convolutional Networks for Gesture Recognition in American Sign Language

[3] G.Anantha Rao, K.Syamala, P.V.V.Kishore, Deep Convolutional Neural Networks for Sign Language Recognition