CS 6375- Machine Learning

Sign Language recognition using Neural Network and SVM

Kashif Ahmad

Kac160230

Final term Project Report

1. **Introduction**

Sign Language recognition is an interesting machine learning problem. Sign language is used by deaf people to communicate with people and it is also a substitute for speech human computer interaction for deaf people. Nearly 94 % of the human population cannot understand sign language; hence it poses a problem for the deaf people to communicate their thoughts. The purpose of this project is to devise a system that can train itself and then be used to decipher sign language. The existing systems rely on a large amount of datasets, hence they take a lot of time to train themselves and there aren’t a large amount of datasets available. My project is aimed at using different algorithms to evaluate small datasets and come up with accurate classifiers much quicker than those that already exist. Current accuracies that exist are between 90 and 100 percent. The target was to achieve 90 percent accuracy with quicker training and classification.

1. **Problem Definition and methodology**

Nearly 94 % of the human population cannot understand sign language; hence it poses a problem for the deaf people to communicate their thoughts. The purpose of this project is to devise a system that can train itself and then be used to decipher sign language. If modeled correctly with proper image acquisition, this can be further extended to mobile applications whereby an app if pointed towards a deaf person’s hand can use the mobile’s camera to understand what that deaf person is trying to say or communicate.

The methodology used is as follows:-

1. The images are obtained via webcam and resized to [100 100 1]
2. The images are converted to black and white to remove extra noise and RGB information so that the classifiers can be trained much quickly. This information is removed using the second biggest blob detection, which is assumed to be the hand in this case. The first being the background, the wall in the case of the following images.



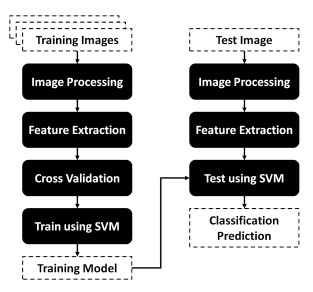






1. The multiclass - SVM and the neural Network is trained on these images.
2. The multiclass – SVM uses HOG feature recognition tool in Matlab to determine the feature of interest, in this case the area covered by hand. I tried using the boundary of the hand by providing the boundary but it did not yield the required results, hence I persisted with a black and white image.
3. The Convolution Neural Network classifies and trains on big images very fast, hence my choice of Neural Networks.

The main algorithm and steps are summarized in the following figure



1. **Experimental Evaluation**

Following are the accuracies with Data Set of size 1500. A set of 10 webcam images are taken with 5 reps, making the total sample size of 50.

|  |  |  |
| --- | --- | --- |
| **Classifier** | **Data Set size for training** | **Accuracy** |
| SVM learners with one v one | 1500 (~ 250 x 6) | 82.6% |
| Convolutional Neural Network | 1500 (~ 250 x 6) | 61.48 % |
| Convolutional Neural Network with Colored Images | 1500 (~ 250 x 6) | 54.23 % |

Following are the accuracies with Data Set of size 300. A set of 10 webcam images are taken with 5 reps, making the total sample size of 50.

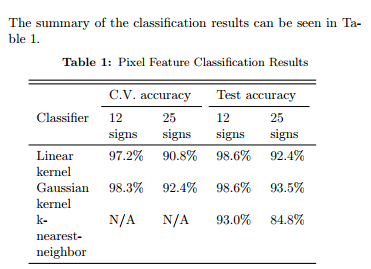
|  |  |  |
| --- | --- | --- |
| **Classifier** | **Data Set size for training** | **Accuracy** |
| SVM learners with one v one | 300(~ 60 x 5) | 86.72% |
| Convolutional Neural Network | 300 (~ 60 x 5) | 42.54 % |
| Convolutional Neural Network with Colored Images | 300 (~ 60 x 5) | 52.98 % |

Hence we can conclude that taking the RGB component out of the images and using blob estimation removes the noise and useless artifacts. The Accuracy with black and white images is greater in the case of neural networks.(inser t accuracy of coloured of SVM)

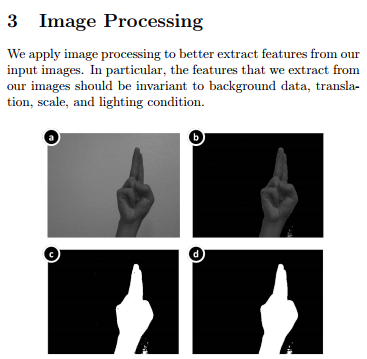
The Neural Network should provide more accuracy theoretically, but its performance suffers in this project’s case, the validation accuracy of Neural Networks is higher than that of SVM. 92% compared to the latter’s 88%.

The training set and the images classified at real time are from a different web camera, which could cause a difference in the color and the quality of the images. The training set should be a set of images from the user. However the training set in the case of this project is not from same user. Images can be optimized and variables and factors can be tweaked to achieve better marginal accuracy but couldn’t be done so due to brevity of time.

1. **Related Work**



These are the accuracies reported by a group project in Stanford. The filtration technique adopted by them is similar to what I have tried to achoeve in this project.



However they take a set of images and classify them later on. The choice of images for accuracy is not reported. My project takes webcam images at runtime and classifies them.

1. **Future Work**

The major shortcomings of my project are that it is matlab based it needs to be implemented in an application and the training should be done on the images taken for each individual, not a generic dataset and these images should improve the weights of Neural Networks and SVM classification over time.

There are other image processing techniques such as using the boundary of the hand to classify. This needs to be explored in detail since it will train the Neural Networks and SVM’s faster.

PCA and other techniques could be employed to better the quality of the image and the extraction of the region of interest.

1. **Conclusion**

This is a project that uses a small dataset to achieve a reasonable accuracy using SVM and Neural networks. Moreover it classifies images by taking inputs from the webcam during real time. In real time, accuracy suffers a great deal and hence the Accuracy achieved in this project as mentioned above is quite reasonable

1. **Bibliography**

<http://cs229.stanford.edu/proj2011/HuangHuangInterpretingAmericanSignLanguageWithKinect.pdf>

<http://cs229.stanford.edu/proj2011/ChenSenguptaSundaramSignLanguageGestureRecognitionWithUnsupervisedFeatureLearning.pdf>

<https://www.mathworks.com/help/nnet/examples/create-simple-deep-learning-network-for-classification.html>

<https://www.mathworks.com/help/vision/examples/image-category-classification-using-deep-learning.html>

<https://www.mathworks.com/help/nnet/deep-learning-image-classification.html>

<http://cs229.stanford.edu/proj2011/KurdyumovHoNgSignLanguageClassificationUsingWebcamImages.pdf>

<https://arxiv.org/ftp/arxiv/papers/1503/1503.03614.pdf>

<https://arxiv.org/ftp/arxiv/papers/0802/0802.2428.pdf>