

Sign Language and Static-Gesture Recognition using Python

1.Introduction

Gesture recognition is an open problem in the field of Computer Vision, a field that aims on emulating Human Vision using mathematical algorithms. One of the fields of Gesture recognition is sign language recognition, where certain symbols are converted to natural language understandable to the user.

The problem statement involved detecting the alphabet represented by an image. For this purpose a dataset of raw images were given. Bounding boxes of the raw images were provided for training purposes. In this project I have solved the given problem using scikit-learn and scikit-image libraries. The main idea was to build a pipeline which first does localization of the hand in the image and then focuses in classifying the image into one of the classes.

2.Description of the approach used

2.1 Localization

The idea is to build a binary classifier to locate the hand in a raw image and feed this to the multiclass classifier developed in second part to get the alphabet represented by the image.

Dataset comprising of raw images, separated user wise is used. Each image has a corresponding bounding box given for training and testing the binary classifier. After initial training we use hard negative mining to improve the classifier. Then sliding window approach is used to find the regions of interest in new images.

To get the training data of hand and not hand images from raw ones, we use sliding window along with IOU(intersection over union method). All those windows with IOU less than 0.7 are put as non-hand images. The bounding box image when rescaled to the window size is put as a hand image. Then subsequently any three images are randomly chosen from the non-hand ones for each image to maintain a 3:1 ratio while training. The classifier used here is the Random Forest Classifier.

Now the above trained classifier along with a sliding window is used to pick regions of interest. Among all of them non-maximal suppression is used to get one window per raw image extracting the hand in each image. This is fed into the next stage of the pipeline.

2.2 Classification

Now for this part I used a Linear Support Vector Machine Classifier(SVC) for classifying the hand got from the previous part into one of the 24 classes. For training the hand was extracted from each raw image using bounding boxes and the image was labelled with its appropriate class. This was fed into the classifier.

The main part here was though sending a vectorized form of an image instead of raw pixels. To achieve this I have used a Histogram of Oriented Gradients(HoG), which worked well for the

problem given.

3. Summary

Evaluation for this was done in two parts. Localization metric used was IoU. For classification a top 5 accuracy measure was used. Localization was given a weightage of 6.5 while classification was given a weightage of 3.5. Data from User 1, User 2, User 8 and User 20 were used for evaluation. Rest all were used for training. My score was 9.14/10 for this course project.