Hand Gesture Recognition using Support Vector Machine and Bag of Visual Words model

Vipul Mehra

May 13, 2018

Abstract

This paper explores a very well known technique for image classification and recognition that is Bag of Visual Words. The process involves feature extraction using Canny Edge Detector and Scale-Invariant Feature Transform (SIFT), codebooks construction using generative model like K-Means and Vector quantization. Finally, classification is done using Support Vector Machines (SVM) using chi-squared kernel. After applying 10 cross validation the accuracy comes to be around $\approx 70\%$.

1 Data Pre-Processing and Collection

Data Set consists of images from 4 classes (A, B, C, D) additionally a class N has been added. Class N denotes the image class that is none of the A, B, C or D class. The class labels were taken from the first letter of the name of the image and was processed accordingly. File types: '.bmp, .ppm, .pgm'. (Refer Appendix A Tables for Data Info) [12]

2 Feature Selection/Engineering

Feature selection has been carried out using novel techniques like Canny Edge Detector and SIFT.

2.1 Canny Edge Detector

Canny Edge Detection is a very famous algorithm to detect edges in the images. Canny Edge detection happens in steps which includes Noise Reduction, that is the image is filtered with a derivative of the Gaussian, in simple terms Gaussian Blur is applied. As edges may have a lot of noise. Then the magnitude and the direction of the gradient is found. Which is done with the help of Sobel kernel in horizontal and vertical direction. First derivative of horizontal direction (H_x) and of vertical direction denoted by (H_y). These two images can help us find the slope of the edge as well as the direction for every pixel.

$$Edge\,Gradient(H) = \sqrt{H_x^2 + H_y^2} \tag{1}$$

$$Angle(\theta) = tan^{-1}\left(\frac{H_y}{H_x}\right) \tag{2}$$

After this Non-maximum suppression is carried out. Which simply means combining less dense multipixels that are like wide ridges into one single pixel. Last step is often called hysteresis. Two thresholds are taken as high and low any where in the image and then the high threshold is used to start edge curves and the low threshold is used continue taking them. This stage is used to decide whether the edge is really an edge or not. [5]

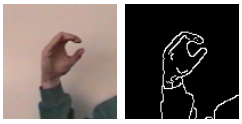

Figure 1: Image without(left) and after Canny Edge Detector (right)

2.1.1 SIFT for Keypoints and Image Descriptors

The content of the image is transformed into coordinates that are local features and are not variant to any kind of image transformation like rotation, translation or scaling the images. Sift is carried out in 4 steps that are Scale-space extrema detection that is simply searching over multiple image locations and scales. After that Keypoint localization is carried out which is simply selecting the Keypoints based on how stable they are. Then finally Orientation is assigned for each keypoint region. Which is simply choosing the best orientation for each of the keypoint. Finally, keypoint description, which is to use local image derivatives at a selected scale and orientation to describe keypoint region. Applying SIFT returns a $n \times 128$ vector, which contains the features explored by the algorithm. [6]

$$\begin{bmatrix} feature_1 \\ feature_2 \\ \vdots \\ feature_n \end{bmatrix} \tag{3}$$

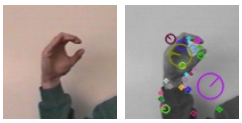

Figure 2: Image without(left) and with SIFT (right)

As can be seen the keypoints are being taken care by sift. We can find the descriptors from the keypoints which are nothing but the features.

2.1.2 Methodology to get features

After reading the image the image is converted into a gray scale image which is then filtered by Canny Edge detector and after wards SIFT is applied onto it. Even though edge detection generally looses on a lot of information, but in our system in most of the images we had to remove the background noise and detecting the edge does not lead to loosing important information. SIFT helps in deducing the keypoints and their descriptors. This helps us get important information about the image. [4, 9, 6]

Figure 3: After applying SIFT to detected edges

3 Building Bag of Visual Words for classification

3.1 Grouping Similar Features and Vocabulary or Codebook Construction using K-Means

As the training set is not very big and also to build bag of visual words it is important to cluster similar features that we got after feature engineering step. Clustering is carried out using the generative K-Means clustering and selection of K was by applying this rule: $K \approx \sqrt{number\,of\,features}$

[13]. This also helps in density estimation. After applying K-Means to the features we get the centroids or we can say words (in case of images) that are also known as codebooks or visual dictionary. This step clusters the descriptors. [7]

3.2 Representing Images or Vector Quantization

After each feature is assigned a visual word or centroid, a histogram of visual word frequencies is created. Then a TF-IDF vectorization is carried out on the histogram of visual word frequencies. This Bag of Visual Words or features are converted into GRAM matrix [1] to be fed into multi-class SVM with chi-squared kernel. The process is similar for training the classifier as well as testing the classifier. [7]

3.3 Multi-Class SVM with chi-squared kernel

3.3.1 Why Chi-Square kernel?

Radial Basis Function (RBF) kernels are useful when the functions are smooth. In the case of Bag of Visual Words, histogram of visual word frequencies are to be dealt, for which the obvious choice of a kernel would be the ones that are good at handling discretization. Therefore, in this case chi-squared kernel for the SVM is best suited. Chi-squared function is denoted by:

$$k(x, y) = exp\left(-\gamma \sum_i \frac{(x[i] - y[i])^2}{x[i] + y[i]}\right) \tag{4}$$

A strong assumption of non-negative data is assumed, which is normalized to an L-1 norm.

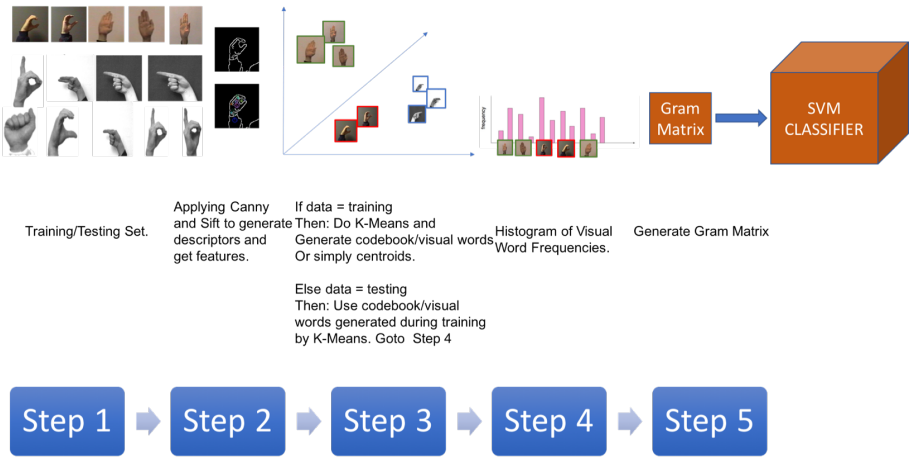

Figure 4: Whole Process of Classification

4 Conclusion

The classification accuracy is not really good taking into account deploying this model in production. With a 10 cross validation we could get an accuracy of $\approx 70\%$. This can be improved by applying techniques like deep learning and transfer learning. Which is out of scope for this project, as it explores image recognition utilizing non-neural network and transfer learning techniques. The accuracy can also be improved by adding more training data and trying different filtering and feature extracting techniques like HOG [10], SURF [11]. Also, the training data set was slightly unbalanced considering the class D due to lack of availability of data for the class D [12].

References

[1] Wikipedia, "Gramian matrix." https://en.wikipedia.org/wiki/Gramian_matrix. Accessed on 05-08-2018.

[2] A. M. Marouane Benmoussa, "Machine learning for hand gesture recognition using bag-of-words,"

[3] K. K. Pallavi Gurjal, "Real time hand gesture recognition using sift,"

[4] OpenCV, "Introduction to sift (scale-invariant feature transform)." https://docs.opencv.org/3.3.0/da/df5/tutorial_py_sift_intro.html. Accessed on 05-08-2018.

[5] OpenCV, "Canny edge detection." https://docs.opencv.org/3.1.0/da/d22/tutorial_py_canny.html. Accessed on 05-08-2018.

[6] kushalvyas, "Implementing bag of visual words for object recognition." https://kushalvyas.github.io/BOV.html. Accessed on 05-07-2018.

[7] P. Gurus, "The bag of (visual) words model." https://gurus.pyimagesearch.com/the-bag-of-visual-words-model/. Accessed on 05-04-2018.

[8] N. Villa Doria D'Angri, "Hands on advanced bag-of-words models for visual recognition." http://www.micc.unifi.it/downloads/tutorial_bow/tutorial_bow_iciap13_1.pdf. Accessed on 05-05-2018.

[9] C. Schmid, "Bag-of-features for image classification." http://www.enslyon.fr/LIP/Arenaire/ERVision/bof_classification_winter.pdf. Accessed on 05-05-2018.

[10] Wikipedia, "Histogram of oriented gradients." https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients. Accessed on 05-09-2018.

[11] Wikipedia, "Speeded up robust features." https://en.wikipedia.org/wiki/Speeded_up_robust_features. Accessed on 05-09-2018.

[12] J. T. S. H. P. Database, "Sebastien marcel - hand posture and gesture datasets." http://www.idiap.ch/resource/gestures/. Accessed on 05-06-2018.

[13] Wikipedia, "Determining the number of clusters in a data set." https://en.wikipedia.org/wiki/Determining_the_number_of_clusters_in_a_data_set. Accessed on 05-06-2018.

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13]

Appendix

A Tables

	SET A	SET B	SET C	SET D	SET N
Total Training	74	78	77	17	75
Total Testing	2	2	2	2	10

Figure 5: Data Set Information