

Assignment Three Report

March 17, 2019

Sharat Vyas

Approach:

My approach is quite simple and fast, the only drawback is that it was difficult for me to cross the 40% accuracy mark. The project is driven by the *test.m* file. For each test image, I provide the set of features we will use to classify the image. The feature-extraction is done in the *feature_extraction.m* file. I performed histogram of oriented gradients (HOG) feature extraction. I used the '*extractHOGFeatures*' method to extract the features. I passed some name-value pair arguments to the function which greatly helped me improve my accuracy. I set the Cell-size to 32 and set the number of orientation histogram bins to 100. Once I have returned the image features for the Validation set, the *test.m* files tries to classify the images by calling *your_kNN.m* and passing the feature vectors. In *your_kNN.m*, before classifying the validation set, I load the training images and extract their features using '*feature_extraction.m*'. Once the training features are loaded, I fit a kNN model to those features using MATLAB's '*fitcknn*' method where I again specify some name-value pair arguments to increase the accuracy. I train the model on the fly instead of trying to save the model since training the kNN is a matter of a few seconds in this case. Once trained, I classify the validation-set features and then return them back to '*test.m*' where the accuracy is computed.

Challenges:

I faced quite a few challenges in this assignment. Firstly, I tried to originally go about by implementing all the components for the bag-of-visual words myself, starting from feature-extraction all the way to clustering. After learning that we can use some of MATLAB's in-built functionality, I set out to try and utilize the '*bagOfFeatures*' functionality that MATLAB provides but I ran into an issue where if I tried to use the '*bagOfFeatures*' function, I would have to specify my own feature-extraction method so that there would be consistency between the features used for training and testing. However if I were to do this, MATLAB expects a 2D

matrix return-value while we are required to return a 1D vector for *'test.m'*. Another issue I faced was getting the accuracy up. I had some difficulty even getting the accuracy to 40% and was unable to find a way to pass the 40% threshold. I tried using an SVM instead of kNN but that ended up getting similar results and taking about 15 mins to train.

Conclusion:

Overall, I enjoyed this assignment even though I had a fairly low accuracy. In hindsight, if I had more flexibility on the assignment, I would likely use MATLAB's *'bagOfFeatures'* method which would handle the feature extraction using SURF which looks to better. Another option would be to not use the bag-of-visual words approach and train a neural network to perform a similar task. In my opinion, I think the most important part of this assignment was the feature-extraction and I felt limited to using HOG due to the requirement to return a vector. I think having the flexibility to be able to use other feature-extractors would have drastically increased the accuracy.