Nicholas Maleki | nmaleki@uncc.edu

Training Bag of Features Model to Classify Images

Be sure to include the following folders and files in the folder with test.m for the program to run: variables.mat, and OpenSURF_version1c

If you wish to train a new set of images, be sure to also include displayResults.m and train.m in the same folder as ./train/. Matlab will also need access to about 2.5GB of ram to train.

Abstract

This program implements a Bag-of-Features approach developed by Saikat Basak and Ranjan Parekh [1] for adding spatial information with local appearance features for improved classification accuracy of images. Spatial information can be described as the probability of finding local appearance features within a sub-region of an image. Speeded-Up Robust Features (SURF) are extracted from sets of images and used for training and testing. Extracted local image features are plotted on a xy-coordinate plane and serve as spatial features. The classification of these spatial features is done using a k-nearest neighbors algorithm (k-NN).

The Assignment

Implement a basic bag of words model to classify images (possibly perturbed) into one of 30 categories by training and validation on the dataset we provided, which contains 30 categories.

train.m

This file is split into two parts, the training phase and testing phase. The training phase will load images from the ./train/ folder and ./val/ folders and load them into image sets which are processed. The features are extracted from these variables using OpenSURF which was chosen over Matlab's SURF program because it can process colored and rotation variant images while still outputting descriptors of length 128. Next, the output of OpenSURF is processed and sent to the next functions which will create k mean clusters and histograms of the features of these images. KNN (K Nearest Neighbors) is run on the centers found by the clustering program in order to properly classify the images.

The testing phase extract features from the validation images, builds more histograms from the images and are classified based upon the histograms and centers built in the previous phase. Once all the classification is complete the results of the first phase are saved to a variables.mat file which is used in the test.m file to classify a new set of validation images.

test.m

This given file loads validation images and passes them to the feature_extraction.m file. This file will extract the features of the validation set and return them to the test.m file. Then test.m calls your_kNN.m which will find the nearest neighbors of the validation set in comparison to the training set which was generated previously in the train.m file. This comparison returns the prediction labels of the images and results in an accuracy of about 34.67%.

Feature Extraction

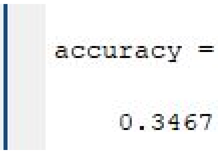
Feature extraction returns the variable ‘all_des_sample_test_uncc’ globally because it has a varying feat_dim that the test.m file cannot handle. This is the reason that feat = [1,128]; is passed to the test.m file instead of the actual features.

K Nearest Neighbors

My your_knn processes the ‘all_des_sample_test_uncc’ instead of the feat because it has a varying feat_dim that the test.m file cannot handle. KNN builds histograms from the features extracted and runs K-NN on it in order to determine the prediction labels.

The parts that were built from scratch

OpenSURF is used to extract surf features instead of matlab’s surf so it was not built from scratch. Because I used OpenSURF and not matlab’s built in functions, every other function used in this program had to be built from scratch because of the data types OpenSURF returns. Folder traversal and image set creation was all built from scratch. Creation of the K-Means clusters was built from scratch. K Nearest Neighbors function was built from scratch using a from scratch euclidean distance calculation. Histogram creation was built from scratch. Histogram and feature comparisons were all built from scratch.



```
accuracy =

    0.3467
```

References

[1] https://www.ijcaonline.org/archives/volume151/number10/basak-2016-ijca-911916.pdf