COMPUTER VISION 1 FINAL PROJECT BAG-OF-WORDS BASED IMAGE CLASSIFICATION

maurits bleeker (10694439) & jörg sander (10881530)

28 March 2016

1 INTRODUCTION

In hindsight more elaborated results would have been possible if

- we would have had more computational resources;
- we were better a code *debuggers*;
- we would have used more than 50 images per category for training (not performed due to our limited computational resources).

Frankly, we made two other important mistakes

- we tested the complete pipeline too late;
- we underestimated the volume aka workload of this final assignment.

We made a mistake with the normalisation of the images histograms. By a programming mistake the histograms where normalised by the length of the data set used. Therefor the training data set (±1800 images) was normalised way differently then the training set. Therefor when classifying the test images all the predicted labels where -1. It took use too much time to solve this problem. Hence, we had not enough time to test the model on all the color models, dense/point sampling for all vocabulary sizes.

The previous *excuses* boils down to the fact that we are not satisfied with the results we present in this report but we still hope that you, the reader, gets the impression that we tried hard to succeed. We literally *wasted* a lot of time on a very *tiny* programming bug which took us days to debug, unfortunately.

We managed to investigate the following Bag-of-Words models, the results can be found in table 1 (see the supplied HTML templates for more details).

Table 1: MAP values of the Bag-of-Words models

Visual Vocabulary size **SIFT type**	400	800	1200	1600
Intensity SIFT (key points)	0.93	0.94	0.95	0.93
Intensity SIFT (dense)	0.99		0.99	
rgb SIFT (key points)	0.93			
opponent SIFT (key points)	0.92			

2 IMPLEMENTATION DETAILS

We build a MATLAB function **pipeline** that can be used to construct, train and test a model. For example when invoking the function with *pipeline(50, 'train', 'rgb', 'point', 400)* a Bag-of-Words model with 400 visual words is created based on intensity and rgb colorSIFT descriptor extraction of 200 images (50 of each image category) using a key point sampling method.
The function assumes that the *ImageData* directory is situated in the current working directory.

We are using the *vl_kmeans* function of the VLFeat library to compute the codebook. The classifiers are computed by means of a Support Vector Machine and implemented by means of MATLAB's standard function *fitcsvm*. We are using a *radial basis function* as Kernel Function. The models are tested with MATLAB's standard *predict* function.

The dense feature description extraction uses a step and block size of 20 pixels [1].

2.1 Implementation of rgbSIFT

According to van de Sande and Gevers [2] in the normalized RGB color model, the chromaticity components r and g describe the color information in the image (b is redundant as $r + g + b = 1$). Therefore the rgbSIFT descriptor features are calculated for the r and g color channels of the normalized RGB color model.

The computed features based on the rgbSIFT extraction are fused with the intensity based descriptors for a particular image by taking the unique set of features.

2.2 Implementation of opponentSIFT

The opponent SIFT descriptors are based on the O_1 and O_2 channel because the O_3 channels is equal to the intensity information. Again for a particular

[1] Dense sampling was implemented with the *vl_dsift* function of the VLFeat library.

[2] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluation of color descriptors for object and scene recognition. In IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, 2008.

image the extracted features are fused with the intensity based descriptors by taking the unique set of features.

3 TRAINING

As mentioned in the introduction we used 50 images from each category to compute the visual vocabulary. We are aware of the fact that this is quite a low number. In order to train the four classifiers we used all 200 images that were provided for test purposes. Therefore each classifier is fed with 50 positive and 150 negative examples. The models are stored on disk and used during the test/prediction phase [3]

We discovered that due to the non-deterministic step of creating the visual vocabulary (sampling 50 images from each category) it is important that the testing of the classifier is done with the same visual vocabulary that was used during training.

4 RESULTS

The *mean average precision* (MAP) results are shown in table 1 for the models we were able to implement and test. The best results for the intensity based models were obtained for dense sampling with a codebook size of 1200 words. The overall performance for the intensity based models is good, the obtained MAP values are between 0.93 and 0.99.

The models using the colorSIFT descriptors (in addition to the intensity information) where only tested for a visual vocabulary of 400 words. The results are comparable to the intensity based models that use the same codebook size.

Looking at the average precision for the separate image categories one can notice that the *cars* are consistently classified less accurate (see the HTML templates for a detailed overview of all the results).

When comparing the point with dense sampling we can conclude that the MAP is much higher then when using point sampling. Even tough we used a very high block size(20 pixel) for the dense sampling the obtained results are extremely high. We assume that the same results will hold for a vocabulary size of 800 and 1600.

Due time limitations we did not manage to create a dataset with a vocabulary size larger then 1600. Also it was not possible to run a lot of experiments for dense sampling. This because if we reduces the block size for the dense sampling the experiments would take to long.

[3] the earlier mentioned *pipeline* function can be also used for testing.