Universiteit van Amsterdam

Computer Vision 1

Final Assignment

Automated Image Classification

Elise Koster, 5982448
Sharon Gieske, 6167667
David van Erkelens, 10264019

Master's Programme Artificial Intelligence
Graduate School of Informatics
University of Amsterdam

October 27, 2014

1. Introduction

Object classification is a fundamental part of Computer Vision and can be used to automate processes and provide environmental reasoning for robotics. Consequently, a high accuracy in object classification systems is invaluable for industry and science alike.
This paper outlines the results of a project analyzing different approaches to image classification using techniques from Machine Learning and Computer Vision. Multiple techniques have been implemented and tested, yielding different results.
The data section will describe the images used for training and testing, the implementation section will introduce the techniques used and the results section will describe the difference in performance for each set of techniques. The conclusion will report on the optimal combination found and the section future work will comment on possible improvements.

2. Data

The training data consists of 2000 .jpg-images in four classes (500 per class): airplanes, cars, faces and motorbikes. The group of training images is split into a vocabulary-training set and a classification training set.The test data consists of 200 .jpg-images in the same four classes, all of whom are used for testing.

3. Implementation

Instead of training classifiers on a large set of pixels, the bag-of-words approach is used. This approach first extracts features from images and subsequently uses them to build a vocabulary of visual 'words'. Each image can then be described as a set of these words, which makes training a classifier easier and faster than a pixel-by-pixel approach.

3.1. SIFT

To be able to build a bag-of-words, features need to be extracted from each training image. This is done using Scale Invariant Feature Transform (SIFT), an algorithm that detects points of interest in an image and produces descriptors of these features. Two types of SIFT are used: key-point (produces descriptors of points of interest) and dense-sampling (every n pixels a descriptor is produced). A multitude of color spaces is used: gray-scale, RGB (regular .jpg-image with three channels), normalized RGB (rgb) and opponent, where R, G, B respectively

are the pixel values per color channel, and

For rgb:

$$r = \frac{R}{R+G+B}$$
$$g = \frac{G}{R+G+B}$$
$$b = \frac{B}{R+G+B}$$

For opponent:

$$O_1 = \frac{R-G}{\sqrt{2}}$$
$$O_2 = \frac{R+G-2B}{\sqrt{6}}$$
$$O_3 = \frac{R+G+B}{\sqrt{3}}$$

Each different color space defines different intensities for each pixel in a color channel, and thus causes SIFT to return different descriptors.

3.2. K-means

Performing SIFT on the first set of training images results in a set of descriptors for each image, which are clustered into visual words using K-means. The K-means algorithm works by calculating the Euclidian distance between each data point and a cluster-center (the mean), and iteratively re-calculating the means and re-assigning the data-points until convergence. The resulting clusters form the visual vocabulary used for describing and classifying images later on.

3.3. SVM classification

Once the visual vocabulary has been built, features are extracted from a new set of training images. These features are grouped into words according to the visual vocabulary, and for each image a histogram of visual word frequencies is computed. These histograms are used as input to train four Support Vector Machines (SVMs) (one per class), using different kernel-functions. SVMs are non-parametric classifiers, which work by maximizing the margin between the decision boundary and two classes of data. After training, all test images are classified according to the SVM-models built using the training images.

4. Results

This section contains an overview and short analysis of the different results obtained during this project.

All classifications are made using a sigmoid kernel.
First, different color spaces were tested. The results can be found in the table below, and show that the gray-scale color space has much better precision than any of the other color spaces.

SIFT type	Samples:vocabulary	Samples:SVM	Clusters	MAP
Key, grayscale	50	50	400	0.7041
Key, RGB	50	50	400	0.1958
Key, normalizedRGB	50	50	400	0.1958
Key, Opponent	50	100	400	0.1946

The following table contains the per class average precisions for the table above:

Airplanes	Cars	Faces	Motorbikes
0.7219	0.7232	0.5326	0.8386
0.1394	0.3118	0.1924	0.1394
0.1394	0.3118	0.1924	0.1394
0.1402	0.3080	0.1909	0.13940

Secondly, different sizes of training sets for SVM-training were tested. As expected, higher numbers of training data yielded better results (see table below).

SIFT type	Samples:vocabulary	Samples:SVM	Clusters	MAP
Key, grayscale	50	50	400	0.7041
Key, grayscale	50	100	400	0.7362
Key, grayscale	50	150	400	0.7322
Key, grayscale	50	250	400	0.7518
Key, grayscale	50	300	400	0.7548

For completeness' sake, below is a table presenting different average precisions per SVM for the parameters in the table above. Consistently, faces yield a worse precision than the other three classes (as they did in the precisions-per-class table earlier).

Airplanes	Cars	Faces	Motorbikes
0.7219	0.7232	0.5326	0.8386
0.7937	0.7525	0.5547	0.8441
0.7419	0.7914	0.5450	0.8503
0.8226	0.7722	0.5415	0.8710
0.8311	0.7612	0.5521	0.8748

Next, the effect of the number of images to create the vocabulary for classification was tested for different sets of SVM-training data. The results are presented below. More images used for training the vocabulary ensure a better precision.

SIFT type	Samples:vocabulary	Samples:SVM	Clusters	MA
Key, grayscale	50	50	400	0.5
Key, grayscale	100	50	400	0.7
Key, grayscale	100	100	400	0.7

Airplanes	Cars	Faces	Motorbikes
0.5978	0.6869	0.4507	0.6292
0.7127	0.7743	0.5397	0.8354
0.7235	0.7723	0.5175	0.8344

5. Conclusion

As gray-scale outperformed the other color-spaces significantly, in spite of supplying less information, it is presumed that the current implementation used for color SIFT was not sufficient. Furthermore, faces seem much harder to classify than any of the other classes, in spite of those classes being much closer to each other in terms of context and object similarity.
Similarly unexpected is the worse performance in terms of precision of a larger vocabulary size. One explanation is overfitting, as with larger vocabulary sizes, more noise may be treated as true information.
On the other hand, a higher number of training image for both the SVM and the visual vocabulary ensured a larger mean average precision, which is expected as there are more positive *and* negative examples to base the models on.

6. Future work

In the current implementation, SIFT for color spaces uses an average over color channel histograms. A possible improvement (though less computationally tractable) would be to train a SVM classifier per color channel per class and build a committee of these SVMs to classify the image. Another improvement can be made by creating classifiers with different kernel functions. Due to time constraints only the sigmoid kernel is used in this project, however, in research by Chapelle et al[1]the radial basis function kernel shows an improved performance. Another suggestion for future work lies in the evaluation of dense-sampling, which could be implemented using the earlier mentioned vlfeat, as the current version was too slow to allow for testing.

[1]*Support vector machines for histogram-based image classification*, Chapelle, Olivier and Haffner, 1999