Computer Vision 1 Project

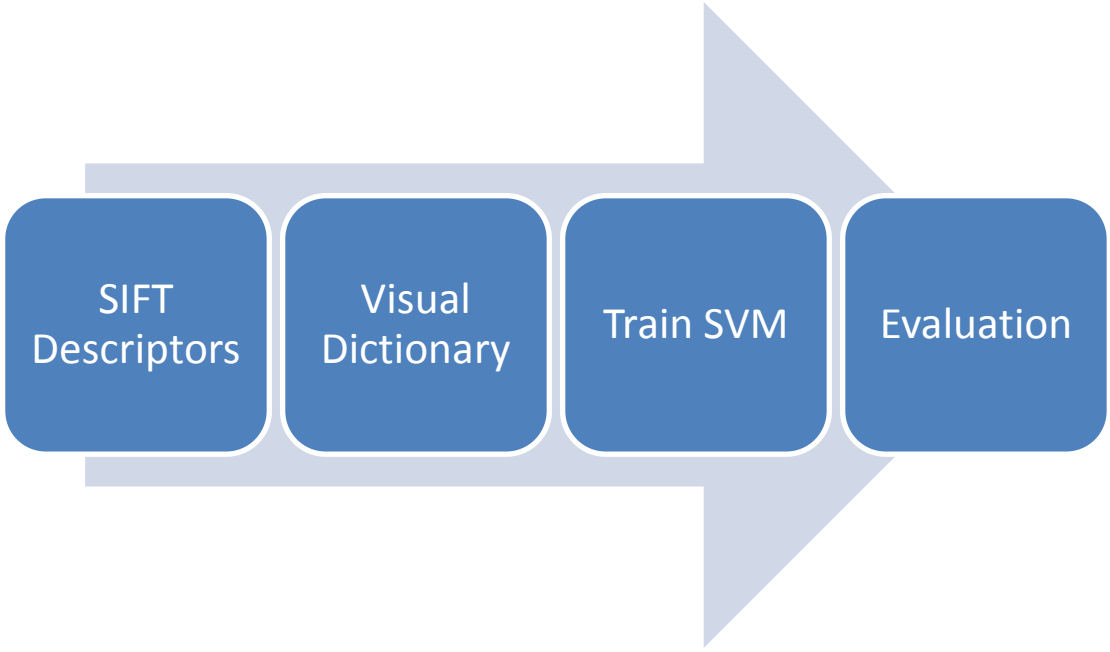
Konstantinos Lampridis (10753915)

Sanida Ioanna (10876812)

26/10/2014

Introduction

The project involves the implementation of a pipeline for image classification, The work flow looks like this:



Pipeline

SIFT Descriptors

100 SIFT descriptors were extracted from each image category (airplanes, cars, faces, motorbikes) at sampled points. Our algorithm although supports also dense sampling, rgbSIFT and opponentSIFT,

Settings

- Dense SIFT
- SIFT at key points (interest points)
- rgbSIFT
- opponentSIFT

Visual Dictionary

We created two versions of visual dictionaries with 400 and 2000 visual words respectively.

Settings

- 400 visual words
- 2000 visual words

SVMs

A binary SVM was trained for each class using a random sample of images from the training set that were not used for building the visual dictionary, Apart from that different kernel functions were tested.

Settings

- kernel
  - Linear
  - Multinomial
  - Radial basis
  - Sigmoid
- Number of images per class used for training the SVMs
  - 100
  - 200
  - 300

Evaluation

All 50 images per class were used for testing the trained SVMs, measuring the Total Precision, Average Precisions per class and building a ranked list per class according to each image's probability.

Results based on different settings

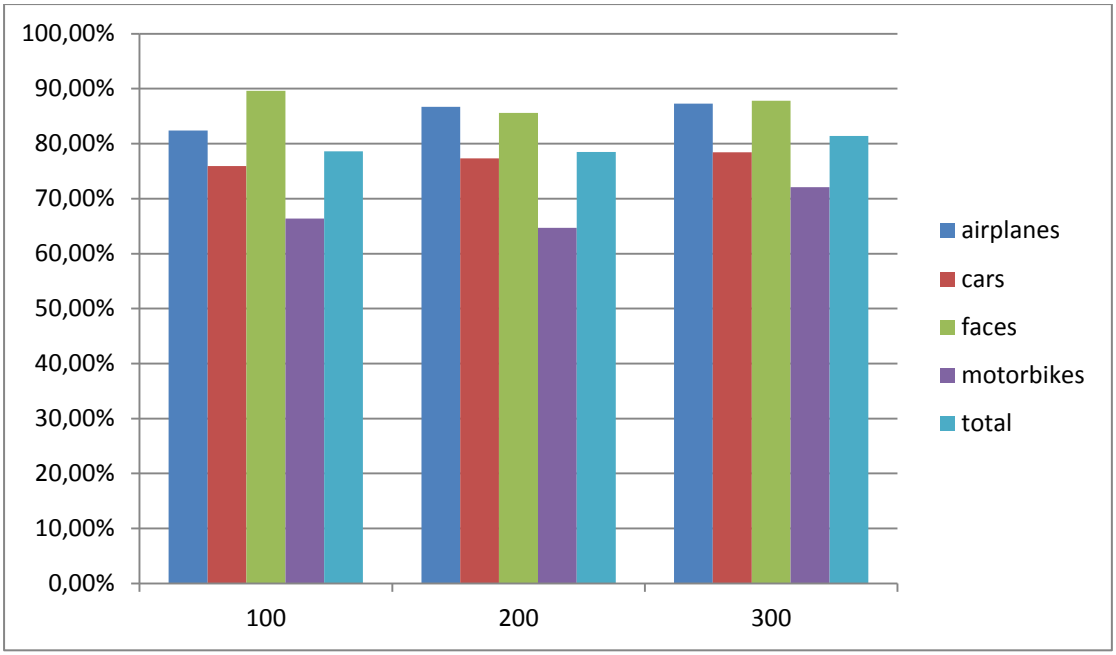
We compared results changing only one aspect at a time.

Images per class for training the SVMs

Fixed settings

- SIFT descriptors
- 100 images per class to build the visual dictionary
- 400 visual words using the matlab built-in kmeans function
- Radial Basis kernel function

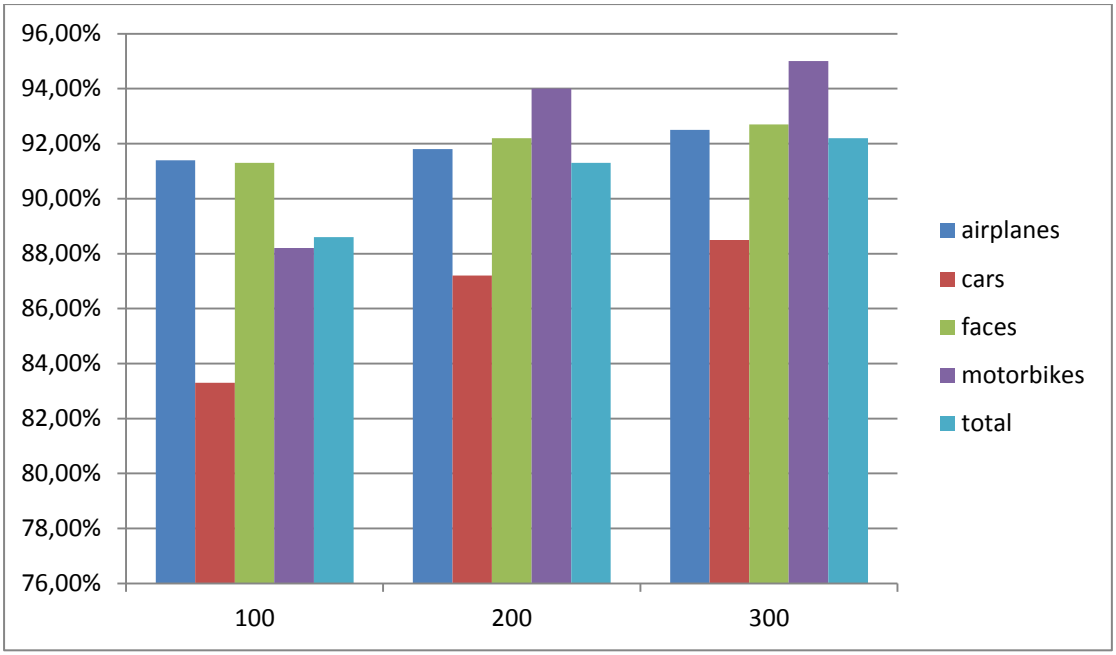
Images per class for training	**Average Precisions**				
	airplanes	*cars*	*faces*	*motorbikes*	*Total*
100	82,4%	75,9%	89,6%	66,4%	78,6%
200	86,7%	77,3%	85,6%	64,7%	78,5%
300	87,3%	78,4%	87,8%	72,1%	81,4%



Fixed settings

- SIFT descriptors
- 100 images per class to build the visual dictionary
- 400 visual words using the matlab built-in kmeans function
- Linear kernel function

Images per class for training	**Average Precisions**				
	airplanes	*cars*	*faces*	*motorbikes*	*Total*
100	91,4%	83,3%	91,3%	88,2%	88,6%
200	91,8%	87,2%	92,2%	94,0%	91,3%
300	92,5%	88,5%	92,7%	95,0%	92,2%



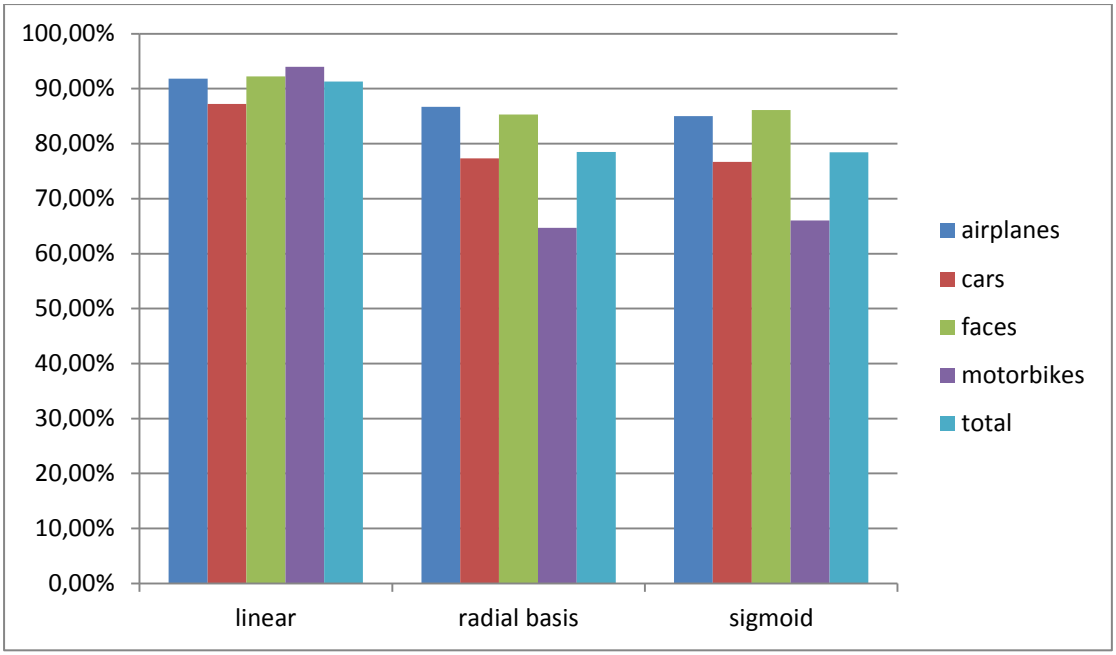
As we can see the number of images used for training the SVMs does not play such a significant role in the Total Average Precision of our model. The trade-off between increasing the number of images for improving the results and the time needed to process them is not worth it.

Kernel

Fixed settings

- SIFT descriptors
- 100 images per class to build the visual dictionary
- 400 visual words using the matlab built-in kmeans function
- 200 images per class for training the SVMs

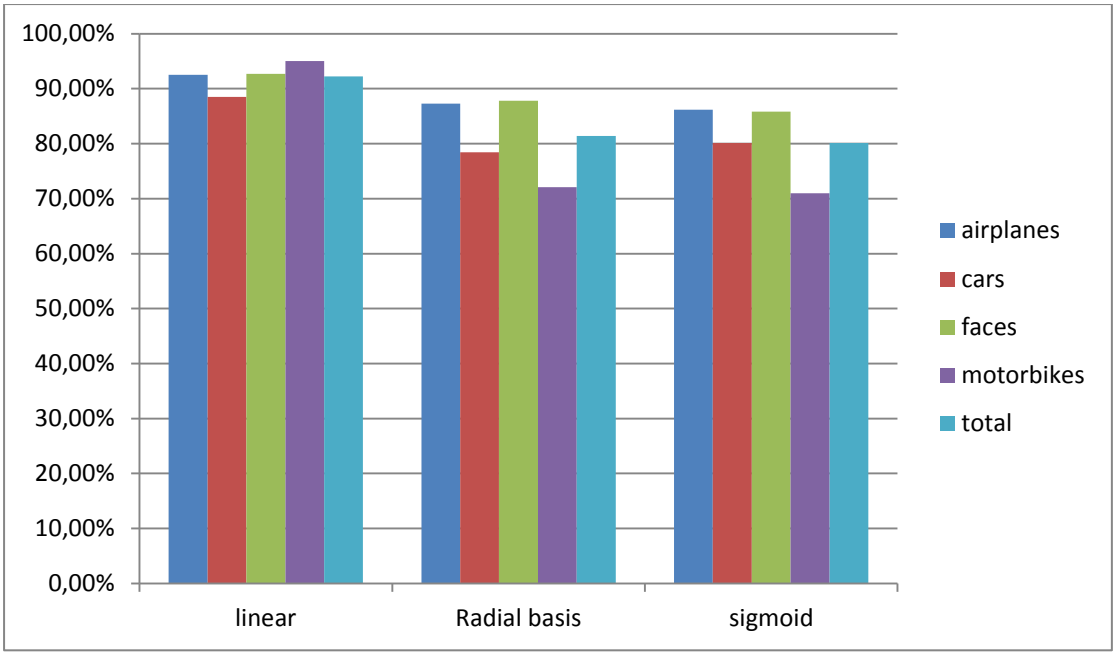
Kernel type	**Average Precisions**				
	airplanes	*cars*	*faces*	*motorbikes*	*Total*
Linear	91,8%	87,2%	92,2%	94,0%	91,3%
Radial basis	86,7%	77,3%	85,3%	64,7%	78,5%
Sigmoid	85,0%	76,7%	86,1%	66,0%	78,4%



Fixed settings

- SIFT descriptors
- 100 images per class to build the visual dictionary
- 400 visual words using the matlab built-in kmeans function
- 300 images per class for training the SVMs

Kernel type	**Average Precisions**				
	airplanes	*cars*	*faces*	*motorbikes*	*Total*
Linear	92,5%	88,5%	92,7%	95,0%	92,2%
Radial basis	87,3%	78,4%	87,8%	72,1%	81,4%
Sigmoid	86,2%	80,1%	85,8%	71,0%	80,1%



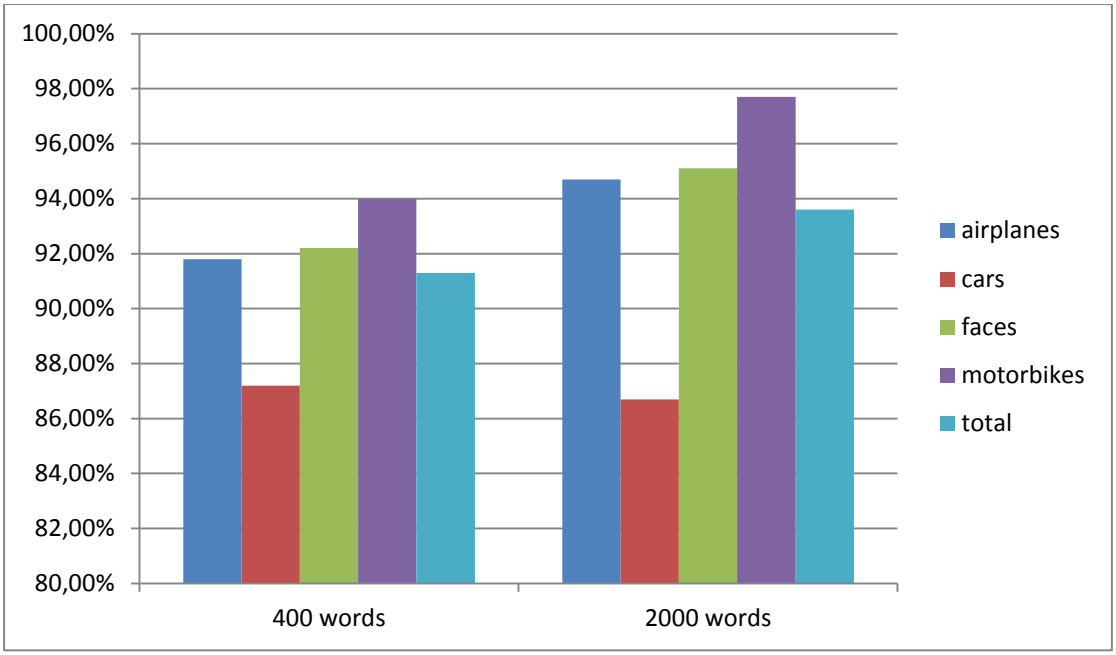
A significant rise (10%) in the Total Average Precision of our model occurs when we use a linear function for the SVMs kernel. Radial basis function performed similarly with the sigmoid, Results for polynomial function are not included since it yielded really poor results.

Size of visual dictionary

Fixed settings

- SIFT descriptors
- 100 images per class to build the visual dictionary
- 200 images per class for training the SVMs
- Linear kernel function

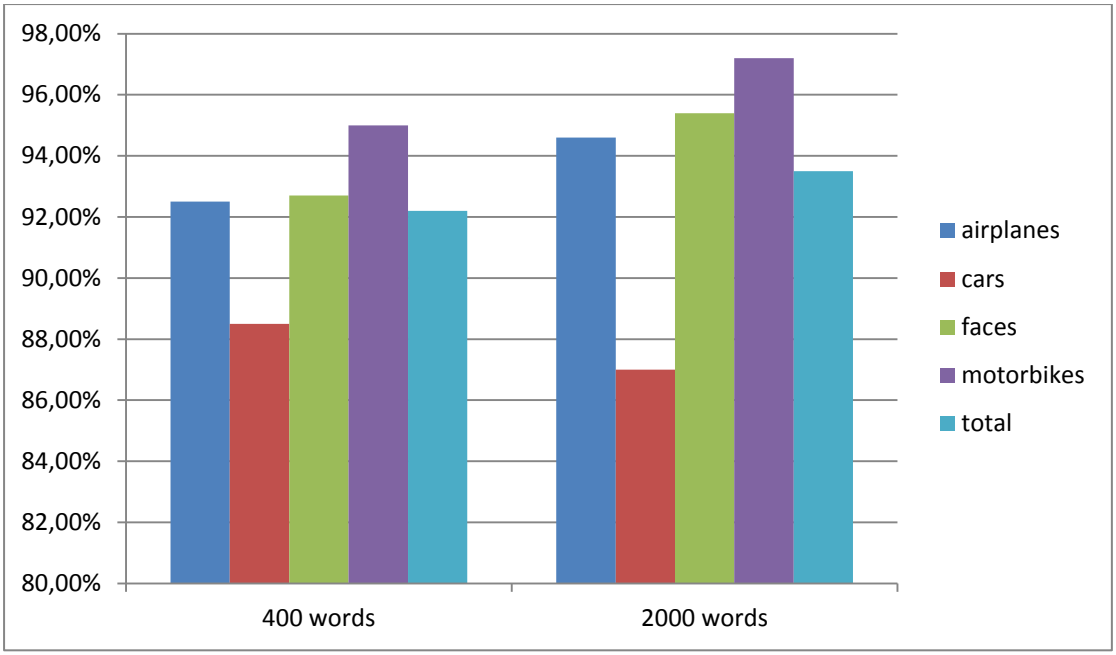
Dictionary size	**Average Precisions**				
	airplanes	*cars*	*faces*	*motorbikes*	*Total*
400 words	91,8%	87,2%	92,2%	94,0%	91,3%
2000 words	94,7%	86,7%	95,1%	97,7%	93,6%



Fixed settings

- SIFT descriptors
- 100 images per class to build the visual dictionary
- 300 images per class for training the SVMs
- Linear kernel function

Dictionary size	**Average Precisions**				
	airplanes	*cars*	*faces*	*motorbikes*	*Total*
400 words	92,5%	88,5%	92,7%	95,0%	92,2%
2000 words	94,6%	87,0%	95,4%	97,2%	93,5%



It is obvious that increasing the visual dictionary results in a small improvement concerning the Total Average Precision. For some reason the average precision for the “cars” class dropped which is the most “unpredicted” class.

Code

The simple script “run_me” runs the whole system with settings:

- 100 images per class for building the visual dictionary
- SIFT at key points
- 2000 visual words
- 300 images per class for training the SVMs
- Linear kernel function

The “classification.m” function should be placed in the same folder as the “data” folder.

Conclusion

We can conclude that image classification is a quite challenging problem. In most cases 100% precision is impossible to achieve. The various settings tested in our case showed that the most important aspect for improving precision is the choice for the kernel function. Linear kernel significantly yielded better results. Apart from that, increasing the size of the visual dictionary raised precision but after certain point diminishing returns started to kick in. Finally, the amount of images used for training the classifiers proved to be of little importance to the final outcome of the pipeline.