



VISUAL SEARCH OF AN IMAGE COLLECTION

ELENA MIHALAS [EEE3032]

The report presents the results of a Matlab visual search program, which evaluates the implementation of different approaches for images similarity estimation and different distance measures.

Table of contents

Abstract _ The aim of the report is to evaluate various techniques that visually search, among a collection of images, those images which hold a content similar to the one present in the query image. Thus, it will be explained how to compute different descriptors for each image, by extracting information such as colour, edge orientation and corners, and how to estimate image similarity through different distance measures such as euclidean distance, mahalanobis distance and city block distance. Moreover, there will be applied various evaluation methodologies for each of the experiments, namely the Precision Recall (PR) statistics, the Confusion Matrix and the Mean Average Precision (MAP), in order to obtain a comprehensive framework for experimental results' assessment.

The visual search software, written in Matlab, is composed of two main scripts, namely the "cvpr_computedescriptors " and the "cvpr_visualsearch" programs. Each script performs two separate tasks. The "cvpr_computedescriptors" code iterates through each image in the dataset and creates an image descriptor by calling the function "extractDescriptor.m". While the "cvpr_visualsearch" code loads all the descriptors computed by "cvpr_computedescriptors" into ALLFEAT matrix, computes the distance between the query descriptor and the descriptor of each other image and then evaluates the overall performance by ranking the images according to their distance from the query image.

Evaluation Methodology

In this report will be discussed both the low-level visual features similarity, such as colour, texture and corners similarities, and the semantic content similarity based on categories. Furthermore, all experimental results will be evaluated qualitatively, by showing both the top 10 closest descriptors to the query descriptor and the confusion matrix for the top 15 results, and quantitatively, by providing the PR curve or the Mean Average Precision values.

Inside the Matlab program the evaluation of experiments' performance is undertaken by the "Precision_recall" and the "Confusion_matrix" functions. The "Precision_recall" function takes in input the query image and builds the ground truth for that particular query image, by marking as relevant only the images which belong to the same category as the query image. Then it computes the precision ("n" number of returned results which are relevant) and recall ("n" number of relevant results returned) metrics for the top 15 distance values as well as the average precision for that particular query. Whereas the "Confusion_matrix" function takes in input one query image at a time, determines the category to which it belongs and defines it as the main category, namely it identifies the row on the confusion matrix. Then it determines the category of each of the 15 top results and add one to each of the columns that represent the specific category of the image. Once the confusion matrix is created, it is normalised across the rows and the result is plotted by assigning to each value inside the confusion matrix a particular colour.

Apart the stated above evaluation methodologies, I have also implemented the Mean Average Precision computation in order to obtain an overall performance estimate for different experiments with respect to all possible query images taken from the dataset. Thus the MAP was computed for both low-level similarity, by simply taking the mean of 591 (total descriptors' number) Average Precision values, and for semantic content similarity, by summing up the values along the diagonal of the confusion matrix and dividing the result by the total number of categories present in the dataset.

Global Colour Histogram

In order to rank the similarity between a query image and a list of other images, it is necessary to transform the entire image in a descriptor which encompass the content information of that image. The simplest way, therefore, to compute an image descriptor is to build a normalised colour histogram that describes the overall colour distribution of that image.

Thus, the RGB space was quatised in "q" regions so that similar colour values were located in the same region of the quantised RGB space (dimensionality [q-1]* [q-1]* [q-1]). Then the values present in each region was distributed among the histogram bins, in the range [0, (q^3)-1], by representing the number in "base q" rather than in "base 10". Eventually, the obtained global colour histogram of (q^3)-1 bins was normalised by dividing each bin by the total number of pixels in order to obtain a meaningful comparison between different size images.

Spatial Grid Histogram

The grid based image descriptor is more discriminative than the global colour descriptor because it offers additional information about the colour and/or texture distribution in different regions of the gridded image. However its computation could become quite expensive if a separate global colour histogram or edge orientation histogram is built within each image cell and afterwards concatenated in a single big descriptor. Moreover, is does not exist a specific rule for the gridding type selection. In fact, when it has to be decided in how many cells to divide the image, there are usually applied simple rules of thumb which take into consideration the trade-off between the discrimination (coarse grids) and the compactness (fine grids) property of the descriptor that is going to be generated.

The Matlab code written for the computation of various descriptors, included the grid based descriptor, is "extractDescriptor". It allows to calculate three descriptors which encode different spatial information , namely the spatial colour distribution descriptor, the spatial edge orientation descriptor and the combined colour and texture descriptor.

The spatial colour distribution descriptor was computed by dividing the image in 36 equal cells and by calculating the average R,G,B values inside each cell. Thus, I obtained a descriptor of size [6x6x3] by concatenating the average values present in each cell. Whereas the spatial edge orientation histogram was obtained by convolving the luminance content inside each cell with a sobel filter in order to determine the edges. Afterwards I computed the edge strength and edge orientation and I included in the histogram bins only the edge orientations that presented a magnitude higher than 0.25 in order to ignore eventual noise. At last I concatenated the information inside individual cells after having divided each histogram by the sum of non null edge orientations.

The combined colour and texture histogram, instead, was computed by concatenating both the average colour and the edge orientation information inside each cell. The obtained descriptor was poorly compact because its dimension was quite high (3x8x6x6=864) and thus the evaluation process took a long time.

Principal Component Analysis

As it was mentioned in the above chapter, the poor compactness of a descriptor could be a drawback for its widespread implementation, despite its good discriminative capacity. However, it is possible to minimise this disadvantage by reducing the dimensionality of descriptor's distributions.

The projection of a high dimensional space onto a lower dimensional space can be achieved by performing the principal component analysis (PCA). In fact, the latter identifies which dimensions express the biggest variability. Thus, once the principal eigenvalues are known, it is possible to set up the dimension of the lower dimensional space by selecting the top "n" eigenvalues above an established threshold. Therefore, the entire high dimensional space can be projected upon the "n" eigenvectors associated with the top "n" eigenvalues. Note that this transformation is known to be lossless and it reduces considerably the amount of computation during the evaluation process.

Harris Descriptor

Although, the spatial grid descriptor encompasses more detailed information from the image, it is not considered a robust method for content encoding because the similarity evaluation performance changes considerably with the variation of the cell size and with the scale and orientation of the objects of interest. Whereas the Harris descriptor encodes the presence of corners in the image, which are known to be stable and repeatable points of interest.

The matlab function that implements the harris corner detector is called “harris” and can be found inside the “extractDescriptor” function. It takes in input a gray image and the number of corners that has to be considered. Next it convolves the input image with the Prewitt filter :

$$G(x) = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}; \quad G(y) = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \rightarrow \ can\ be\ decomposed: \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} * [1 \quad 0 \quad -1]$$

The Prewitt filter detects the edges or abrupt changes in intensity and, since it can be decomposed in a product between an averaging and a differentiation kernel, it also applies a little smoothing to the filtered image in both horizontal and vertical directions.

Once the image is filtered, the corners are determined by considering the Taylor expansion of 2nd term of the sum of squared differences : $\sum [I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)]\ ^\wedge 2$. The latter expression, under appropriate approximations, becomes a matrix that expresses local intensity variations:

$$\sum \left([I_x(x_i, y_i), I_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 = [\Delta x \quad \Delta y] \begin{bmatrix} \sum (I_x(x_i, y_i))^2 & \sum I_x(x_i, y_i) I_y(x_i, y_i) \\ \sum I_x(x_i, y_i) I_y(x_i, y_i) & \sum \left(I_y(x_i, y_i)\right)^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

The characteristic of a corner is that of having the determinant of the eigenvalues matrix much bigger than its trace. Therefore, if the PCA is applied to the above matrix in order to determine its eigenvalues, it is possible to express the strength of the corners through the following expression:

$$corners = Det(M) - k * Tr(C) = \lambda_1 \lambda_2 - k * (\lambda_1 + \lambda_2), \quad where\ k = 0.04$$

However, it is not enough to determine the strength of the corners, it is also important to compute the local maxima inside a 3x3 window across the obtained matrix of corners in order to sort the strongest corners. Eventually the top “n” strongest corners are localised on the input image and a window is centred on each of those corners in order to extract average colour histograms, which are then combined together to generate a single descriptor.

Experimental Results

Requirement 1+2 _ In order to accomplish this requirement, I implemented the global colour histogram using the Euclidean distance:

$$euclidean\_distance = \sqrt{\sum \left(Query_{Descriptor} - X_{Descriptor} \right)^2}$$

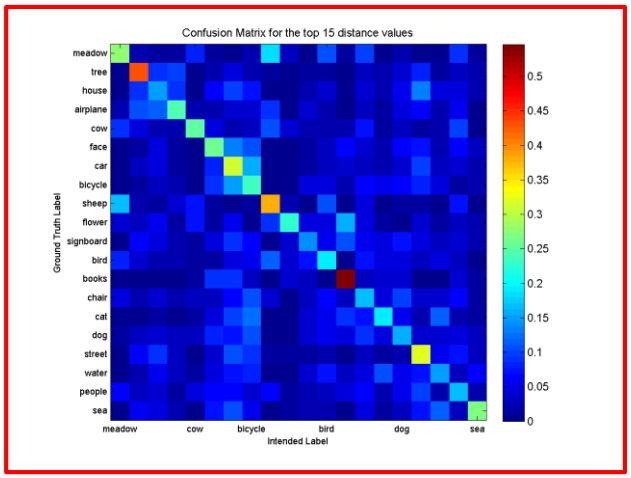
The stated above distance metric computes the Euclidean distance, in an “n” dimensional space, between the query descriptor and all other descriptors (591-1=590) in order to determine which descriptors are located closer to the query descriptor. This information is essential to determine the degree of similarity between two images because images with similar content correspond to descriptors close to one another in a high dimensional space.

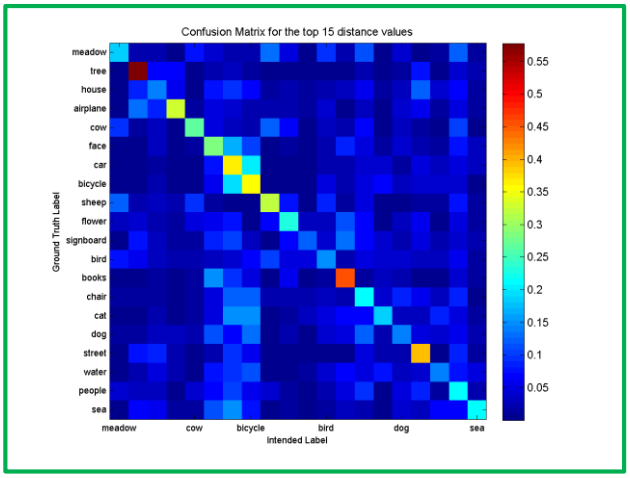
The following table shows the experimental results for three different query images at three different RGB quantisation levels (q = [4,7,10] from top to bottom):

Query Image	Top 10 results : q =[4,7,10]
	
	
	

The above results show that overall performance depends on the RGB space quantisation level. However, it is not possible to determine which quantisation generally gives the best result because the performance also depends on the degree of colouring of objects of interest. Even so, I could deduce that for colourful objects it is appropriate to use a low quantisation level, whereas for less colourful objects it is better to quantise more the RGB space because each cell will hold less colour variation.

In order to draw overall conclusions, I computed also confusion matrices for the top 15 results and Precision-Recall curves for all possible queries from the dataset:





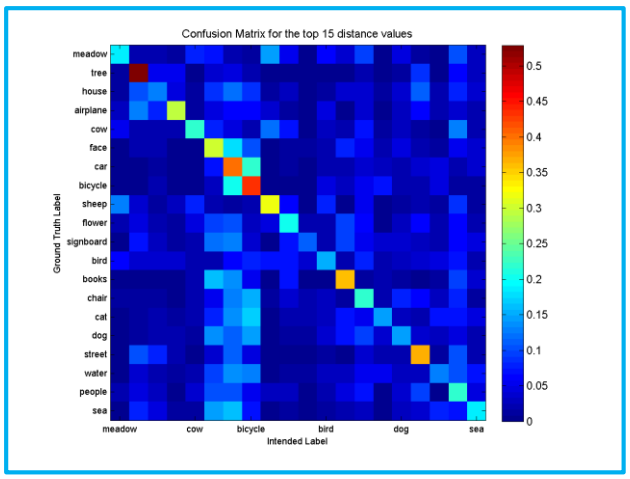


Fig.1 *The red square holds the confusion matrix for quantisation level 4, the green square holds the confusion matrix for quantisation level 7 and the blue square holds the confusion matrix for quantisation level 10.*

On the "y axis" are positioned all 20 categories, namely the ground truth. Whereas on the "x axis" are positioned the cumulative sums of top 15 distance results (categories) for all possible query images from the dataset.

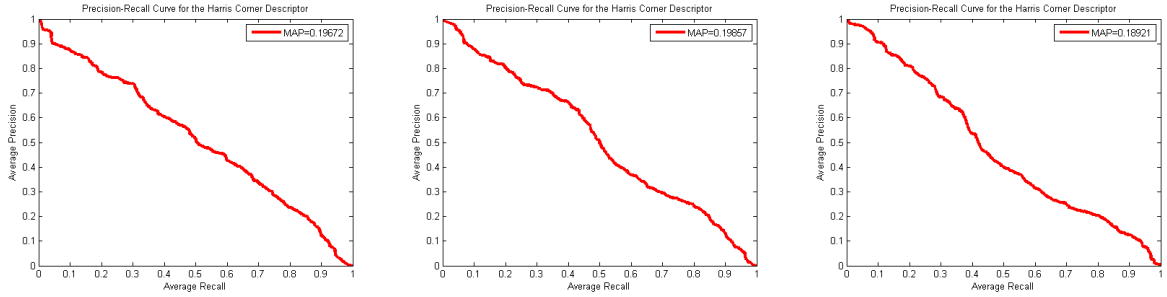
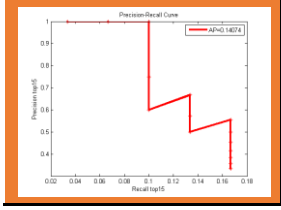
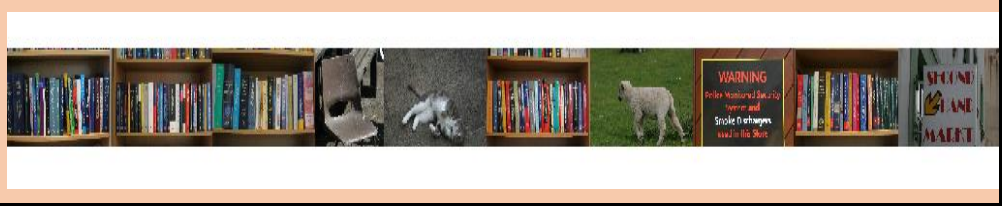


Fig.2 *The above graphs are the Precision Recall curves for all possible query images from the dataset. From left to right the quantisation level of the RGB space is [4, 7, 10] with mean average precision values of [**19.67%, 19.86%, 18.92%].***

The confusion matrices confirm the results obtained for the three selected query images. In fact, it can be noticed that the value assigned to category "books" decreases when the quantisation level increases, whereas the category "face" and "car" values increases. Furthermore, it appears that the experimental results for the provided collection of images are slightly better for quantisation level 7 and this assumption is confirmed by the MAP = 19.86% result (the highest MAP value).

Requirement 3+2_ In order to fulfil this requirement I divided the image in equal cells and I concatenated the individual histograms computed for each cell so as to obtain a single descriptor for an image. Moreover I extracted three different features from the cells, namely I analysed the contribution of colour and edge orientation to the improvement of similarity degree between two images.

The following table shows the results obtained using the spatial colour distribution histograms. I chose three different query images in order to evaluate the degree of similarity between images:

PR curve	Top 10 results
	

I noticed, by analysing the above table, that grid based descriptor provide worse results for "books" category with respect to global colour descriptor, while it performs surprisingly well for "car" category. The dog image, instead, performs very badly except for the first result.

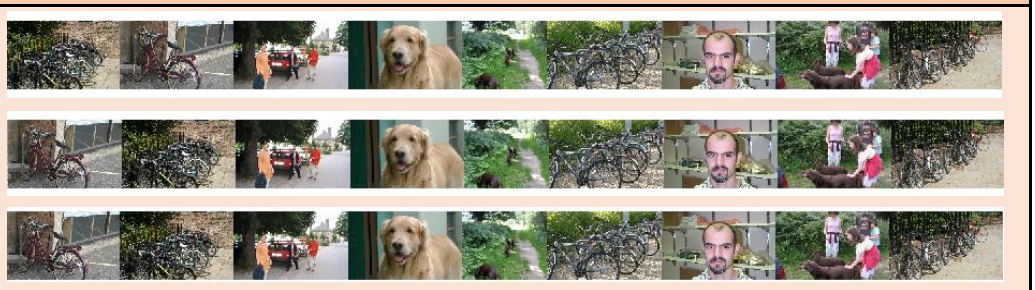
All these deductions are confirmed by the Average Precision values. In fact, AP(books) = 19.31% for quantisation level 10 (worse result for different quantisation levels) and AP(books) = 14.07% for grid based descriptor, and AP(car) = 17.11% for quantisation level 10 (best result for different quantisation levels) and AP(car) = 18.17% for grid based descriptor. Whereas AP(dog) = 9.67% which is the worse result between the selected three query images for the grid based descriptor.

The next table, instead, shows the results obtained using the spatial texture distribution histograms at different levels of angular quantisation (q_ang = [4, 8 12]):

Query Image	Top 10 results : q_ang =[4,8,12]
	
	

The above results indicate that, by increasing the angular quantisation level, generally the proximity estimation will improve significantly. However, there are also some images, such as the dog image, which show better results for low angular quantisation levels.

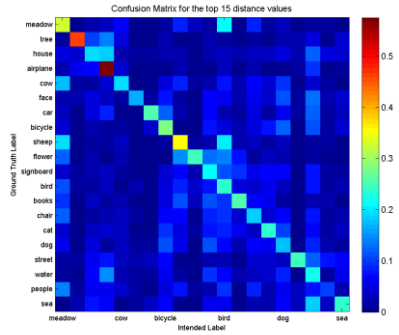
The results, which are shown in the table below, derive from the implementation, in the visual search program, of the combined colour and texture spatial distribution histograms:

Query Image	Top 10 results : q_ang =[4,8,12]
	
	
	

The above results are impressive. In fact, the combined colour and texture spatial descriptor performs very well because it takes into account both colour and edge orientation information contained inside each cell of the gridded image. Moreover, this experiment proves that the similarity estimation improves with the increase of angular quantisation level. However, it must be pointed out that there are some images that maintain a stable low score across all quantisation levels.

After having analysed the results obtained for particular query images, it would be appropriate to explore the experimental results obtained by considering each possible query image from the dataset in order to evaluate if the overall trend upholds the previous hypothesis.

Hereafter I show the computed confusion matrices and MAP values for the three different grid based descriptors:



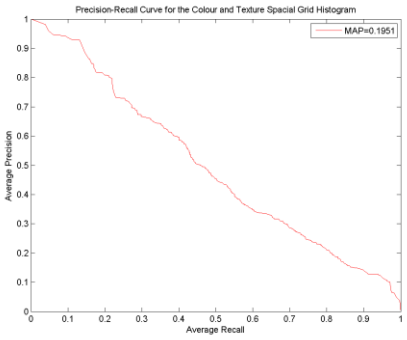
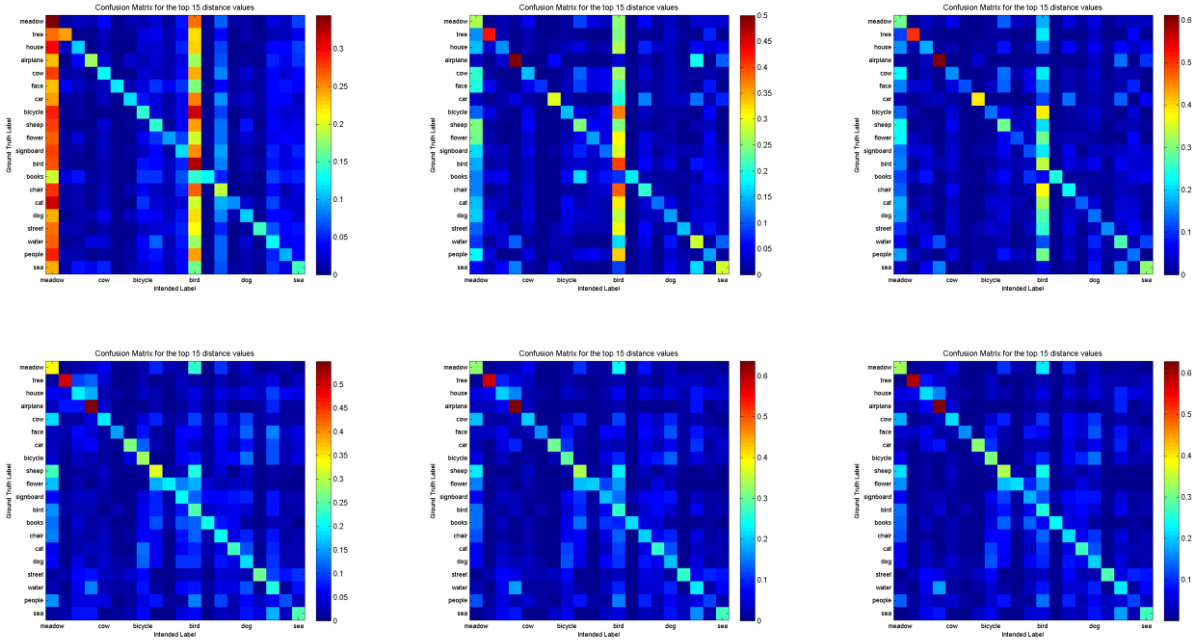


Fig.3 *The left image represents the confusion matrix for the spatial colour distribution descriptor, whereas the right image shows the PR trend when all possible query images are taken into consideration (* ***MAP = 19.51%*** *)*



	MAP for q_ang = 4	MAP for q_ang = 8	MAP for q_ang = 12
Spatial Texture Desc.	12.76%	17.45%	19.2%
Combined Descriptor	19.41%	20.78%	21.12%

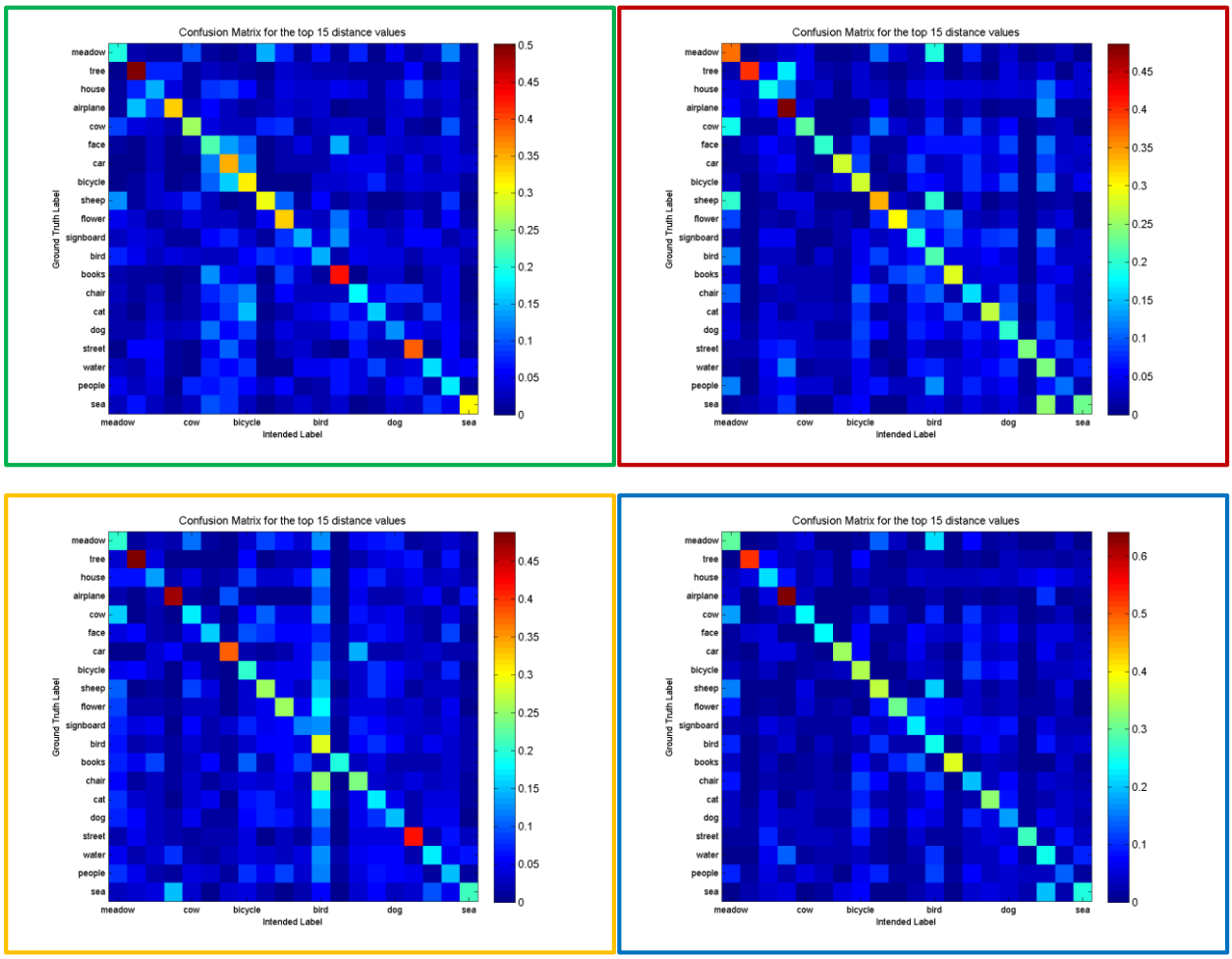
Fig.4 *The above image represents the confusion matrices for both spatial texture distribution descriptor (top row) and combined texture and colour descriptor (bottom row) for different angular quantisation levels (from left to right q_ang = [4, 8, 12]). The table, instead, holds the MAP results for each combination.*

The above results represent a summary for grid based descriptors, which states that the combined texture and colour descriptor gives the best overall results. Furthermore, it also shows that the performance improves when the angular quantisation level increases.

Requirement 4+2_ The current requirement was fulfilled by applying the principal component analysis to the set of extracted descriptors and then by computing the projection of all features onto a dimensional space of size fifteen. Furthermore, I implemented the following equation in order to compute the Mahalanobis distance:

$$mahalanobis\_distance = \sqrt{\sum \frac{\left(Query_{Descriptor} - X_{Descriptor}\right)^2}{eigenvalues}}$$

I have considered only the RGB space quantisation level 7 and the angular quantisation level 8, in order to compute the overall performance for each of the descriptors stated in the above sections. Therefore, I obtained four confusion matrices and four MAP results:



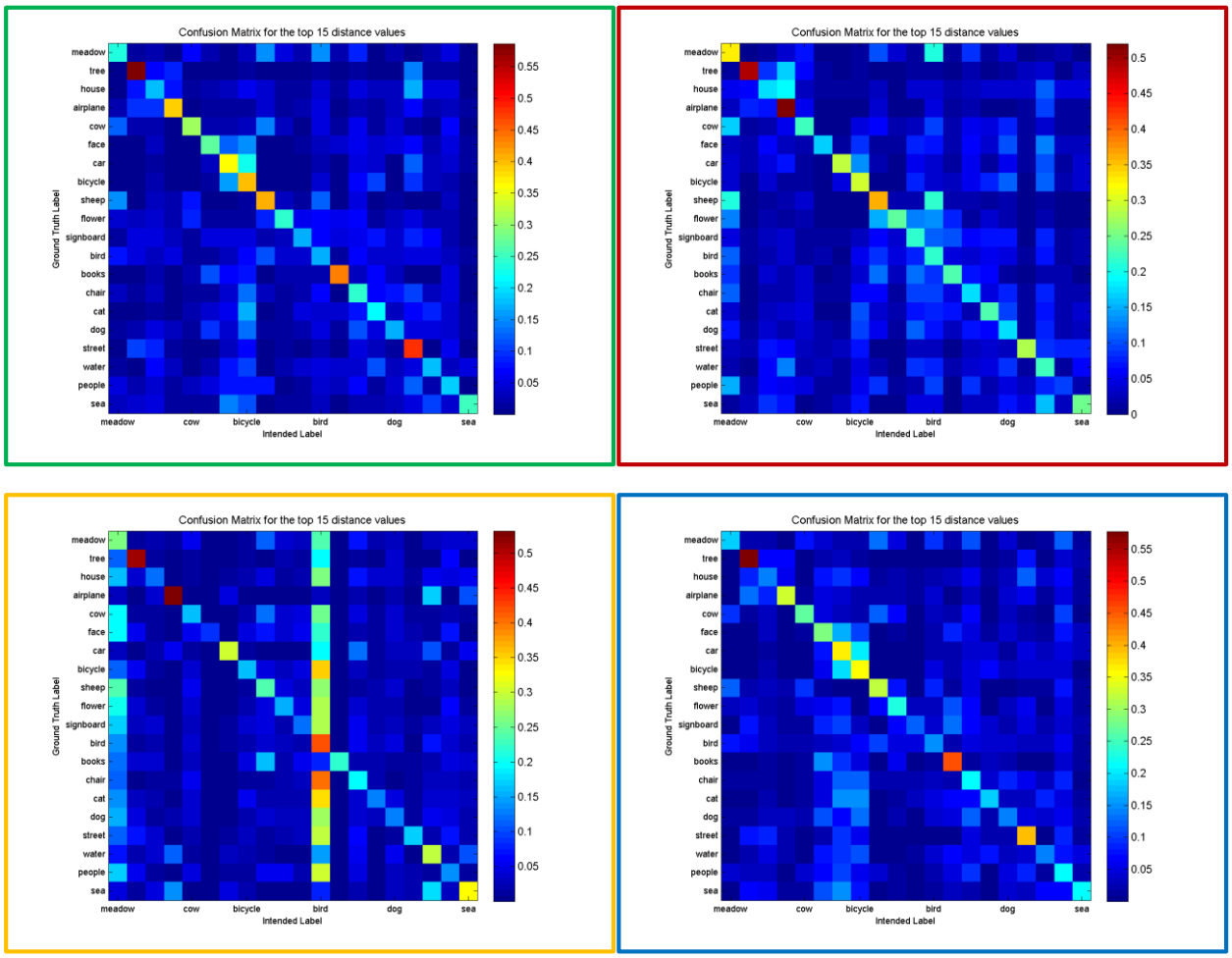
	Global Colour D.	Spatial Colour D.	Spatial Texture D.	Combined Desc.
MAP results	19.28%	19.01%	18.08%	21.81%

Fig.5 *The green square holds the confusion matrix for the global colour descriptor, the red square holds the confusion matrix for the spatial colour descriptor, the yellow square holds the confusion matrix for the spatial texture descriptor and the blue square holds the confusion matrix for the combined colour and texture descriptor. The table, instead, holds the MAP results for each descriptor.*

The above results show that Mahalanobis distance is more accurate in identifying the closest images to the query image in the reduced dimensional space. In fact, quite all MAP results are higher than the corresponding MAP results computed with the Euclidean distance, except for the global colour descriptor. Moreover, same as in the Euclidean distance case, the combined colour and texture descriptor proves to be the best descriptor to apply to images in order to obtain a better parting into categories in the "n" dimensional space.

Requirement 5+2_ In order to accomplish the last requirement, I inserted a new distance metric in the Matlab software, namely the City Block distance: $L1_{norm} = \sum \left| Query_{Descriptor} - X_{Descriptor} \right|$. Moreover, I wrote an additional function in order to compute the Harris Corner descriptor (images with evidence of corners' detection in Appendix).

The results obtained with the city block distance are visualised in the following figure. I have considered only the RGB space quantisation level 7 and the angular quantisation level 8. Therefore, I obtained four confusion matrices and four MAP results:

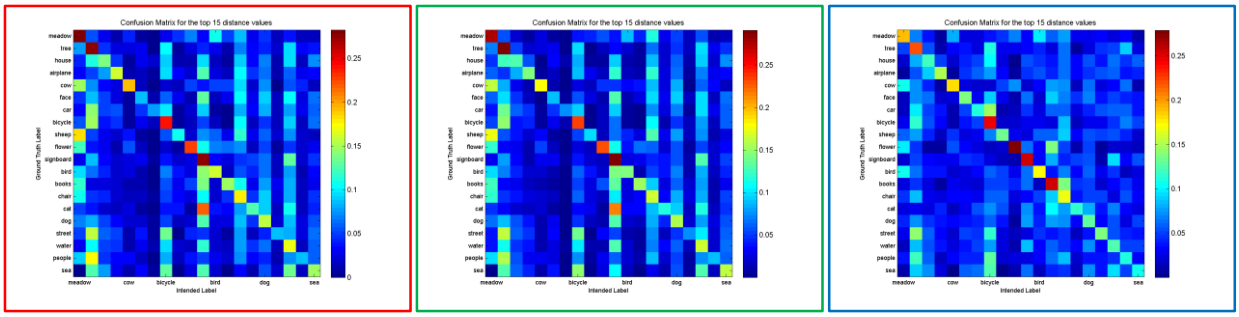


	Global Colour D.	Spatial Colour D.	Spatial Texture D.	Combined Desc.
MAP results	22.41%	19.75%	18.79%	19.86%

Fig.6 *The green square holds the confusion matrix for the global colour descriptor, the red square holds the confusion matrix for the spatial colour descriptor, the yellow square holds the confusion matrix for the spatial texture descriptor and the blue square holds the confusion matrix for the combined colour and texture descriptor. The table, instead, holds the MAP results for each descriptor.*

The obtained results indicate that L1 norm distance works even better than Mahalanobis distance for quite all previously discussed descriptors, except for the combined colour and texture descriptor. It is worth noticing that the performance obtained with the global colour descriptor and the city block distance is much better compared with the other performances, whereas the performance obtained with the combined descriptor is the worse one.

Hereafter I give evidence of results obtained using the Harris Corner Descriptor:

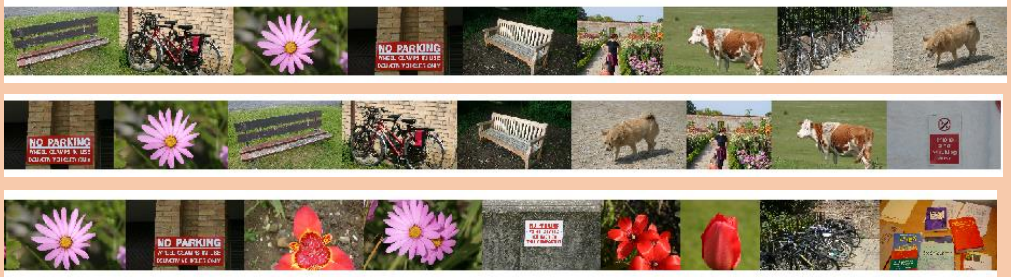


	City Block Dist.	Euclidean Dist.	Mahalanobis Dist.
MAP results	13.38%	13.2%	13.04%

Fig.6 *The red square holds the confusion matrix for city block distance, the green square holds the confusion matrix for the Euclidean distance and the blue square holds the confusion matrix for the mahalanobis distance. The table, instead, holds the MAP results for each distance.*

From the above results, I could deduce that Harris Corner descriptor is not sufficiently discriminative and it is not an appropriate descriptor choice in most of the cases. However, I also noticed, by analysing the above confusion matrices, that it works quite well for at least four different categories, such as "meadow", "tree", "flower" and "signboard". Thus, I selected two different images from the "flower" category and one image from the "signboard" category in order to check if the single search confirms the overall result.

Below I present a table containing the single search results for three different query images using all previously discussed distance metrics:

Query Image	Top 10 results : L1 norm, Euclidean, Mahalanobis
	
	
	

The few images that I selected as query images confirm the confusion matrices results but not the MAP results. In fact, I obtained very good performance using the Mahalanobis distance, whilst the MAP result for the Mahalanobis distance is the worst one. Moreover, it is worth to mention that, by using the Mahalanobis distance, the categories become less interchangeable and therefore there are less indetermination in assigning one image to a certain category.

Conclusions

This report contains the description on how were computed the descriptors: global colour, spatial colour distribution, spatial texture distribution and Harris corner. Furthermore, it explains which evaluation methodologies were applied in order to classify random query images from the dataset and it also discusses the descriptor's projection onto a lower dimensional space. Eventually, the presented experimental results were useful to determine which type of descriptor combined with a particular distance metric performed better.

In the first experiment, I implemented the global colour descriptor and I evaluated the performance using the Euclidean distance. The MAP results that I obtained are [19.67%, 19.86%, 18.92%] for quantisation level [4, 7, 10] respectively. Thus, I could deduce that in average the quantisation level 7 gives satisfying results. However, the single search, which I performed for some query images, presented some discrepancies. In fact, I noticed that very colourful objects need a low quantisation level (between 4 and 7) whilst less colourful objects need a high quantisation level in order to determine well the similarity of the query image to other images. This effect must be due to the necessity to narrow the cubes inside the RGB space for the less colourful images in order to group the few colours in separate cubic regions and therefore to increase the discrimination between one category and other categories containing objects with different colour distribution.

In the second experiment, I extracted three different descriptors from three different query images using the Euclidean distance. I obtained MAP = 19.51% for the spatial colour descriptor, MAP = [12.76%, 17.45%, 19.2%] for the different angular quantisation levels q = [4, 8, 12] of the spatial texture descriptor and MAP = [19.41%, 20.78%, 21.12%] for the same range of angular quantisation levels of the combined colour and texture descriptor. This overall MAP results together with the presented single search images prove that combined colour and texture descriptor offers the best estimations and identifies with more precision images similar to the query image. However, when the angular quantisation level is low, the spatial colour descriptor performs better than the other two descriptors. This is because the edge orientation of pixels inside each cell is distributed among few bins and the descriptor is too generic for its texture content.

In the third experiment, I projected the entire high dimensional feature space onto a fifteen dimensional space using the eigenvalues and eigenvectors obtained by applying the principal component analysis. Then, I implemented the Mahalanobis distance in order to estimate the classification performance. Thus, I obtained MAP = 19.28% for the global colour descriptor, MAP = 19.01% for the spatial colour descriptor, MAP = 18.08% for the spatial texture descriptor and MAP = 21.81% for the combined colour and texture descriptor. From the above results, I deduced that projecting the features along the dimensions that show greater variance and applying the Mahalanobis distance, which takes into consideration those high variances, increases considerably the overall performance. However, there is an exception for the global colour descriptor, which proves to perform better when combined with the Euclidean distance rather than with the Mahalanobis distance. This must be cause by the absence of true grouping in the high dimensional space of features belonging to the same category. In fact, the Mahalanobis distance compresses the space proportionally to the variance along each dimension of the entire distribution before computing the difference between the query image and a candidate image. Therefore, this step might worsen the overall result, if the entire distribution presents a rather uniform distribution as in case of the global colour feature distribution.

In the fourth experiment, I implemented the city block distance and I obtained the following results: MAP = 22.41% for the global colour descriptor, MAP = 19.75% for the spatial colour descriptor, MAP = 18.79% for the spatial texture descriptor and MAP = 19.86% for the combined colour and texture descriptor. These results prove the superior performance of the city block distance over the other two distance metrics. Indeed, I suppose that this must be cause by an indefinite mixing of features belonging to different categories in the same regions of the high dimensional space. In fact, the obtained MAP result for the combined colour and texture descriptor is the worst result among other distance metrics because the stated above descriptor results to be more discriminative. Therefore, it is likely that in the high dimensional space similar objects tend to group together and thus the proximity estimation tends to worsen when L1 norm is adopted as distance metric.

Finally, in the fifth experiment, I extracted the Harris corner descriptor from each image and I applied all three distance metrics in order to assess its performance. I obtained the following MAP results: 13.38% for the city block distance, 13.2% for the Euclidean distance and 13.04% for the Mahalanobis distance. The above stated results are very low and indicate that Harris corner descriptor offers a bad classification, especially if the evaluation is based on the Mahalanobis distance. However, I also noticed, by analysing the confusion matrices, that Euclidean distance metric generates a lot of results outside the matrix' diagonal, whereas Mahalanobis distance results are mostly concentrated on the diagonal. This phenomenon indicates that Mahalanobis distance metric's discriminative power is higher, namely it associates with more precision one image to the right category, even if the MAP value is the lowest among the other distance metrics. This discrepancy must be caused, indeed, by the average operation, which cancels the information of few very good results for certain categories. In fact, it is worth to mention that there are some types of categories for which the Harris Corner descriptor works very well and which are mainly characterised by a high frequency content with a lot of edges and corners, such as "medow", "tree", "flower", "signboard", "bicycle" and "books".

BIBLIOGRAPHY

- J. Collomosse, *Computer Vision and Pattern Recognition*, Features and Matching Lecture Notes, Guildford: University of Surrey, 2016.
- J. Collomosse, *Computer Vision and Pattern Recognition*, Features and Matching (part I) Lecture Notes, Guildford: University of Surrey, 2016.
- Microsoft Research (MSVC-v2) dataset of 591 images: http://research.microsoft.com/en-us/um/people/antcrim/data_objrec/msrc_objcategimagedatabase_v2.zip.
- J. Collomosse, '*cvpr_computedescript*ors' and '*cvpr_visualsearch'* skeleton Matlab Programs, Guildford: University of Surrey, 2016.

APPENDIX

In this appendix are attached some images that give evidence of Harris Corner detection: