ImageSearch

# Introduction

An ImageSearch application, with its database formed using a subset of the image database NUS-WIDE, was developed in Java. A combination of four image matching algorithms – namely Color Histogram, Visual Concept Vector (and Mapping), Visual Keyword, and Text (Description) – were implemented to compute the similarities between a queried image and images within the database.

In this report, the approach in implementing the ImageSearch will be discussed. It covers various methods including (1) capturing relevant information from the training data, (2) calculating relevant probabilities or scores, and (3) combining the image matching algorithms together.

# Implementation Details

## Training Data

The training data used by the program consists of images from the NUS-WIDE database, which is provided with the ground truth, categorizing the data into at least one of the 25 classes.

Please refer to the training data summary below:

|  |  |
| --- | --- |
| **Category** | **Count** |
| Number of training images | 1221 |
| Number of test images | 250 |
| Number of unique category | 25 |

Table 1 – Summary of training data

## Information Extraction

For each image in the training database, the following information is extracted:

|  |  |
| --- | --- |
| **Item** | **Description** |
| Colour Histogram | An array of floating point numbers containing the Colour Histogram. |
| Visual Concept Vector | A Visual Concept vector of 1000-classes. |
| Visual Concept Map | A Visual Concept mapping from the 1000-classes to the 25 categories. |
| Visual Keyword | Visual Keyword extracted from 8-bit Grayscale image. |
| Text | Text Description of the image. |

Table 2 – Information extracted from training data

## Pre-Processing of Image Database

The training data forms the program’s image database. It was noticed that extracting information from every category (i.e. Color Histogram, Visual Concept, Visual Keyword, and Text) for each image took a considerable amount of time (up to 10 seconds per image). Thus to do so for all 1221 images in an on-demand manner (i.e. when a query image was given) is not practical.

Hence it was decided that the information for each image in the databases were to be pre-processed, and stored in a custom database (similar to an index). Thus when running a query against the database, the information is taken from this custom database (or index) instead of directly from the images itself.

## Colour Histogram

Color Histogram is used as one of features in ImageSearch. To obtain the Color Histogram, image representation is first converted from RGB to linear color representation, YCrCb, which then stored in an array of floating point numbers.

Given a query image, the color histogram score is obtained by computing the similarity between the query image and each of the images in the database. Bhattacharyya distance is used to compute the similarity between two color histograms. The score is stored with respect to each images in the database. Image with high score suggests that it resembles the query image in color composition.

## Visual Concept Recognition

Visual Concept Recognition is used to produce a score to determine the likelihood of the image category. The 1000-class image classification tool is used. It works by extracting different image features such as SIFT, LBP , and color histogram, and then classify the image using SVM classifier with the given model file. The tool produces a classification score, giving weights ranging from -7.0 to 7.0 (representing the likelihood that the image falls into the specific category, with 7.0 having the highest likelihood) to each of the 1000 classes.

When a query image is uploaded, the tool is used to obtain the 1000-classes scores. Thereafter, among the 1000 scores for each class, 5 classes with the highest positive scores are short-listed. (Note that only classes with positive scores are short-listed. If less than 5 classes have positive scores, then the short-list will have less than 5 classes.)

Following, we use the scores from these 5 short-listed classes to compare with the corresponding class scores of each image in the database. The comparison only occurs if the corresponding class score of the database image is also positive. If the database image’s class score is also positive, then the minimum between the database image’s class score and the query image’s class score is taken at the matching score.

Subsequently, the sum of all 5 (or less) matching scores is taken, and normalized to a score between 0 and 1. This forms the final score of the visual concept recognition feature.

## Visual Concept Mapping

Visual Concept Mapping is an extension of Visual Concept Recognition above. The 1000-classes classification specified by the tool consists of huge variations of concepts, ranging from many species of animals to many types of buildings. Most of these classifications are too specific. Therefore, a mapping from each of the 1000-classes to one or more of the 25 categories was manually generated. For example, classes like go-kart, sports car, fire engine, garbage truck, police van, race car, were all mapped to “cars” category.

When given a query image, the 1000-classes classification scores will be used to determine the score of each image in the one of the 25 categories. This is done by summing the relevant 1000-classes scores for each of the 25 categories. For example, if a query image scores 0.1, 0.05, and 0.04 for race car, sports car, and police car respectively, the 25-category score will be 0.19 (sum of 0.1, 0.05, 0.04) for the “cars” category.

The Visual Concept Mapping score is then computed by comparing the respective scores for each of the categories.

## Visual Keyword (SIFT)

Visual Keyword, in particular Scale Invariant Feature Transform (SIFT), is adopted as one of the features for ImageSearch. The tool provided is capable to find features (keywords) from an 8-bit Grayscale image pgm formatted files. Each image in the database is converted to 8-bit Grayscale image using Java Advanced Imaging (JAI) ImageIO library; then its visual keywords is extracted by the tool.

When a query image is presented, the tool is ran to extract the keywords from the query image. Then a score is obtained by matching the keywords between the query image and each image in the database using matching tools. The normalized score for this feature is the number of matching keywords divided by the number of keywords of the query image.

## Text

Text Description is one of the important feature to describe an image. The text description for each image in the database (if available) is tokenized and each word is stored. Direct index is used because the number of images in the database is not very large.

Given a query image, its text description is tokenized in the same way as the database images. Then each word in the query image is cross-checked whether the database image contains the same word descriptor. The normalized score of the text feature is the number of common word descriptor divided by the number of word descriptor of the query image.

## Overall Score

The final score used for each of the images are the sum of the five scores (Visual Concept Recognition and Visual Concept Mapping contributing one score each), each multiplied by some weight.

Based on the final score, the images in the database are ranked, and the top 20 are shown to the user. The weights are obtained by running the system with different combinations and evaluating its F1-Score (will be discussed later). The best weights with the highest performance (F1-Score) are selected.

## Genetic Algorithm

For the system to work well, an optimum weight for each feature is required. Manual prediction by a human hand might not result in the optimum weight sets. To overcome this problem, a learning algorithm is adopted, namely Genetic Algorithm. It is used to learn and find the optimum weights, given initial random weights and a training data set and F1-Score is used to measure the performance of each weight set (the fitness of the gene).

The Genetic algorithm starts by reading a few specified set of weights defined by user. It creates offspring by permuting gene pairs, and crossover of each weight (chromosome) is performed on the two parents. Crossover cutting points are not fixed, and they are decided randomly on whether to pick the chromosome from the first parent or the second parent. Mutation method chosen in Genetic Algorithm is highly important in finding the optimum weights. Small steps are often successful, especially when the genes are already well adapted to the environment. However, large steps are also necessary to prevent the weights to be caught in local maxima. Larger changes can also produce good result much quicker when the genes are not well adapted yet. Thus higher probability small-steps mutation and lower probability large-steps mutation are adopted. The genes are sorted based on the best F1-Score, and 5 genes are brought forward to the next iteration.

Genetic Algorithm is chosen due to the following reasons:

* Its simplicity to implement within the program itself
* It is easy to recognize the level of significance of each feature

If other machine learning techniques are used, e.g. Maximum Entropy or Support Vector Machine, it is not possible to tell which feature contributes the most (and the least) to the performance of the ImageSearch system.

## Online (Real-World) Search Performance

During online (or real-world) search, where a new image is uploaded to ImageSearch, the time taken for the program to return the top 20 result is crucial. Despite doing every measure to lower the processing time (pre-processing, extract information offline, multi-threading), the SIFT tools seem to be taking a significant amount of time. It could take up to 30 seconds to compare the query image with each of the images in the database where the rest of the features takes up to only 1 second.

After doing thorough analysis, SIFT is not the best feature to classify the result from the database. However, it is useful to detect keywords and similarities between images that is relevant to the query image. To solve this problem, the other three features are used to get the first top 60 images, with the same scoring system as mentioned above. And then the SIFT tool is run against the 60 images and adding its normalized score (multiplied by its weight). The top 20 images are then obtained and shown to the user.

## Relevance Feedback

Simple Relevance Feedback was implemented. User can simply left click the image to give positive relevance feedback (i.e. the image is relevant), and right click the image to indicate negative feedback (i.e. image is irrelevant). It works by adding/subtracting a small percentage (in this case 10%) of each decision score in the Visual Concept 1000 classes (Section 2.4) of the query image, from the image being chosen. In graph representation, it is equivalent to moving the chosen image closer or further from the query image.

# Evaluation

Mean F1-Score is used to evaluate the performance of ImageSearch. For each query, recall, precision, and F1-Score is defined below:

Relevant images is defined as image with one or more common category/class as the query image.

The evaluation result of each individual features are as follows:

|  |  |
| --- | --- |
| **Features** | **Mean F1-Score** |
| Colour Histogram | 0.05657064043606513 |
| Visual Concept Mapping | 0.19388136345500312 |
| Visual Concept | 0.1664350524416272 |
| Visual Keyword | 0.0309762958131592 |
| Text Description | 0.06111516631656386 |
| Combined 4 features with equal weight | 0.18243447013372313 |
| Combined 4 features with the best weight | 0.24015583734453508 |
|  |  |

Table 3 – Performance of individual features of ImageSearch

Both Genetic Algorithm and naïve complete search with weights between integers 0 and 10 was done. Naïve search to find 5 sets of weights ranging from 0 to 10 inclusive incurs a cost of 11^5. Some of the best results from the naïve weight search and Genetic Algorithm are as follows:

|  |  |
| --- | --- |
| Weight of Colour Histogram | **W1** |
| Weight of Visual Concept Mapping | **W2** |
| Weight of Visual Concept Recognition | **W3** |
| Weight of Visual Keyword | **W4** |
| Weight of Text Description | **W5** |

Table 4 – Feature Weight Mapping

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **W1** | **W2** | **W3** | **W4** | **W5** | **F1-Score** |
| 0 | 9 | 10 | 0 | 1 | 0.23988 |
| 1 | 8 | 9 | 0 | 2 | 0.23370 |
| 0 | 1 | 1 | 0 | 1 | 0.23342 |
| 1 | 8 | 9 | 2 | 5 | 0.23241 |

Table 5 – Best Weights from Naïve Complete Search

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **W1** | **W2** | **W3** | **W4** | **W5** | **F1-Score** |
| 0.9755 | 897.99 | 997.47 | 0.8527 | 509.83 | 0.24339 |
| 15.368 | 704.52 | 902.89 | 8.22 | 765.26 | 0.24024 |
| 148.59 | 736.29 | 911.60 | 55.13 | 727.51 | 0.23122 |
| 95.76 | 511.31 | 848.71 | 933.64 | 933.06 | 0.22323 |

Table 6 – Best Weights from Genetic Algorithm

The low F1-score is due to the low recall caused by the limit of 20 images to be retrieved. Given a query image, there are at least 50 relevant images, and the maximum number of relevant retrieved images is 20. Therefore the maximum recall of each query is 20/(50+), less or equal to 0.4.

# Conclusion

From thorough analysis and testing, it is known that Visual Concept Recognition is the strongest feature to match images. It is able to classify an image with high accuracy, resulting in relevant image retrieval from the database. Text Description is also a strong feature to match images but subject on the correctness of the description of the image. If the description of the image is incorrect, it will lead to wrong classification of the image. Color Histogram and SIFT are the least useful features to match images. Even though they are not very useful to classify an image to the correct category, both features are able to detect visual similarities which might interest the user from the visual perspective.

Using the best weight set, ImageSearch has been able to obtain 0.24338 F1-Score. However, the third best weight from the genetic algorithm evaluation (table) will be used since it is also important to show the visual similarities to the user.