EDIS

EFFICIENT DATABASE IMAGE SEARCH

Motivation and Goals

The goal of this dissertation was to develop an efficient content-based image retrieval technique, which is a key component of an object recognition pipeline.

The algorithm receives an image as input, returning a set of the most similar images, contained in a database, as a result.

Tools

The developed application was written in the C++ programming language, and the OpenCV library was used, as it provides many computer vision algorithms.

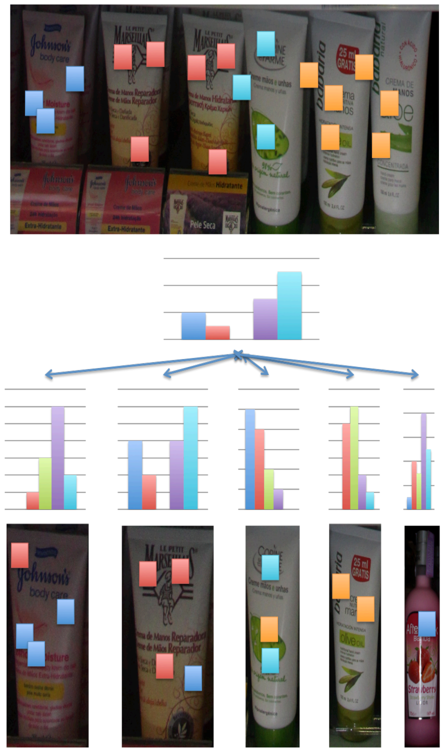

Fig1. Images are represented as histograms of visual words (Top: query image; Bottom: database similar images).

Feature Detection and Descriptor Extraction

For the comparison of images in terms of similarity, features have to be extracted from both the query image and the database images. Features consist of parts of images, such as vertices, edges or regions that remain easily identifiable despite variations in scale, noise, illumination or other image properties.

The Oriented FAST and Rotated BRIEF (ORB) algorithm was used to detect and extract descriptors from these features, which is documented by the literature as an efficient method, even when using lower powered devices such as smartphones.

Bag of Words Approach

To improve performance even further, feature descriptors extracted from the images were clustered into bags of visual words, using the k-means algorithm. The cluster for each descriptor of the image is computed depending on how far the descriptor is from the cluster centers.

Input and database images are then represented as histograms of visual

Contact

Rua Alfredo Allen, 455
4200-135 Porto, Portugal

+351 220 430 300
info@fraunhofer.pt
www.fraunhofer.pt

Visual Words

The number of visual words used when clustering, plays a big role in the final outcome. In our tests, a maximum number of 1000 features were extracted from the images, and several different numbers of visual words (k) were tested. A smaller number of images (e.g. 100) improves performance, however, the image ranking is not as good as the generated words are less distinctive.

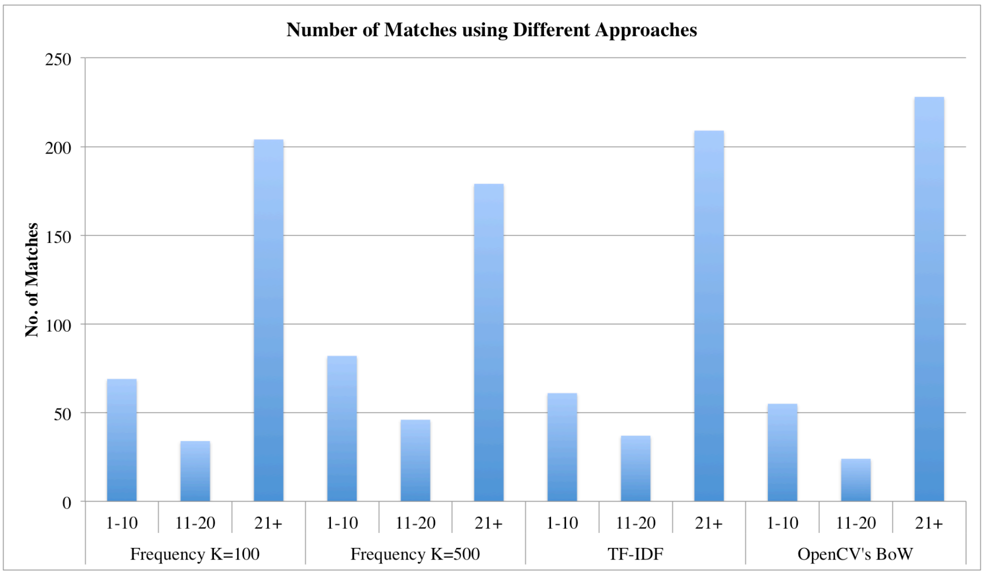

Fig2. Number of correct matches in the first 10 positions, in the 11th to 20th positions and beyond the 20th position, using 4 scenarios.

Scenarios Tested

- Custom Bag of Words implementation, using frequency histograms for image representation, with 100 visual words
- Custom Bag of Words implementation, using frequency histograms for image representation, with 500 visual words
- Custom Bag of Words implementation, using a Term Frequency-Inverse Document Frequency scoring, with 100 visual words
- OpenCV's Bag of Words implementation, with 100 visual words

words, which count how many occurrences of which word exist in the image. When database image descriptors are clustered into words, a dictionary is constructed. After normalization, the Euclidean distances between the input image's histogram and all the database image histograms are calculated, in order to measure how similar it is to each of the database images.

An alternative implementation made use of a Term Frequency-Inverse Document Frequency scoring (TF-IDF), in order to down weigh words that appear frequently throughout the image database and making the image histograms more distinctive. Another approach, based on OpenCV's Bag of Words implementation, was also developed, in order to assess how efficient it is compared to the custom implementation.

Results and Conclusions

The custom implementation of the Bag of Words algorithm provided better results overall, despite being slower.

An image dataset comprised of 125 pictures was used and several input images were tested. The number of correct matches within the first 10 results were noticeably higher when using the custom Bag of Words implementation. Using a higher number of visual words also improved results, but, as a drawback, it slowed the process significantly, in regards to both dictionary construction time and matching time. In order to achieve a matching time below 100 milliseconds, so that 10 matches can be done within the time period of 1 second, the number of visual words used in the custom implementation has to be around 80. According to the measurements made, matching time did not suffer considerably from an increase in the number of images in the database, suggesting that these values are valid even for larger databases.

In the future, in order to improve results, image matching techniques can be applied to the most similar images to remove false positive results, and an Android application can be created.