Exploring the effectiveness of a SIFT Bag-Of-Visual-Words for image similarity detection

Lucas Adelino and Heejeong Wee

Introduction

Computing image similarity is a very popular computer vision task. Possible applications include image retrieval (which itsel has many other applications), classification, as well as plagiarism and forgery detection.

For any supervised-learning approach to image similarity, extracting good features from images is of critical importance. SIFT is an influential image feature extraction algorithm. One of the main attractives of SIFT is that the features it generates are invariant to scale and rotation changes, as well as partially invariant to illumination and noise (Lowe, 1999). These advantages make SIFT potentially attractive for detecting similarity in images that may have been manipulated.

Our main interest in this topic came from plagiarism detection: namely, we were interested in exploring the effectiveness of an algorithm that detects if any two images are altered copies of one another.

Brief survey of the literature

According to Farhan et al, plagiarism is a critical issue in the educational field. Although plagiarism is an issue usually dealt with in texts, the paper mentions that there is also a method to detect plagiarism among images, and the authors introduce the SIFT algorithm in the feature detection and k-mean clustering for the match using their adaptability, scalability, and extensibility. Using 45 samples for training and 48 for testing, the research proved that this method is effective in detecting plagiarism, with a 90% matching rate and 81% of accuracy.

Bag of Visual Words(BoVW) method is a word image retrieval system that functions similarly to the Bag of Words(BoW) method that is applied to texts. Shekhar and Jawahar use BoVW representation in spotting words in images. They mention BoVW representation determines the image by computing the histogram of visual word frequencies and returning the closest image, thus making it effective in matching images. Also, it is considered to be an ideal way of retrieving images because it does not require prior learning.

Methods

Our algorithm was built using a bag-of-visual-words approach. The big-picture overview of the algorithm is as follows:

1. Extract descriptors using SIFT
2. Use K-Means to cluster features. Each cluster is a visual word.
3. Count how many visual words there are in each image. This count of visual words will be our features.
4. Test these approaches using both cosine-similarity and brute-force matching.

We will now talk in more detail about each these steps. But first, we will briefly describe the reference and testing data sets that we used

Reference and testing datasets

We used an abridged version of the DISC2021 dataset. This dataset was compiled for the 2021 Image similarity Challenge.
Our reference dataset consists of 528 images from the DISC 2021 dataset. To make up our first test set, we generated copies : text was added tof the first 50 images of the reference dataset, using a Python script. Each image was submitted to random stretching/shrinking, cropping, rotating, and brightness/saturation changes. This set was intended as a baseline, to analyze whether our approach was indeed resistant to changes in scaling, rotation, and illumination. The second test set was obtained from the DISC2021 dataset, and consists of copies of the references images that were also scaled, rotated, or brightened/darkened, but they were *also* manipulated in other forms, including adding noise and superimposing text, shapes, or other images on top of the image. This dataset was designed to test the resistance of our approach to more adversarial inputs.

SIFT descriptor extraction

We used the SIFT implementation of the OpenCV library in Python to extract descriptors for the images in our reference and test sets.

Building a visual word vocabulary

After extracting descriptors for all reference images, we employed K-Means to cluster the descriptors into visual words. We tested the accuracy of the model on test set #2 for the following values of K: 200, 300, 600, and 3000. Out of those, 3000 achieved the best results. Increasing the value of K can drastically impact computation speed, so we thought K = 3000 represented the best tradeoff between speed and accuracy.

Creating feature vectors

After clustering, we iterated over all images, counting how often each visual word appeared in that image. This vector of counts of words made up the feature vector for each image.

Testing the model

Once we built a feature matrix for all images in the reference set, we tested each test set against them. We computed four different versions of the first test set, progressively adding image manipulation:

- Version 1: Rotate only
- Version 2: Stretch/Shrink, Rotate
- Version 3: Stretch/Shrink, Crop, Rotate:
- Final version: Stretch/Shrink, Crop, Rotate, Brighten/Darken:
  For each of these versions, we computed accuracy based on both cosine similarity and brute feature matching. We likewise tested test set #2 against the reference set. For all tests, we computed accuracy based on cosine similarity as well as brute-force matching of the raw descriptors from both test sets.

Results

Test set 1

Below are the results for test set #1 for each series of image manipulations:

- Version 1 - Rotate only:
  - Cosine Similarity: 100%
  - Brute-Force Matching: 100%
- Version 2 - Stretch/Shrink, Rotate
  - Cosine Similarity: 82%
  - Brute-Force Matching: 70%
- Version 3 - Stretch/Shrink, Crop, Rotate:
  - Cosine Similarity: 62%
  - Brute-Force Matching: 68%
- Final version: Stretch/Shrink, Crop, Rotate, Brighten/Darken:
  - Cosine Similarity: 40%
  - Brute-Force Matching: 52%
  These results show that, when the image is manipulated only with resizing and rotating, cosine similarity is more effective than brute-force matching. With the addition of cropping, brute-force becomes slightly more effective. And with the addition of cropping and illumination manipulation, brute-force matching becomes reasonably more effective.

Test set 2

Test set #2 had an accuracy of 33.52%. This jumped to 40% if we counted positive matches as any match where the true image was in the top 5 candidates (rather than whether the true image

being the top 1 candidate).

Discussion

These results indicate that the SIFT/Bag-of-visual-words approach is accurate if resizing and rotation are the only types of manipulation happening. When considering cropping and rotation, brute force becomes more effectively, possibly because these manipulations confuse which words are computed in the bag-of-words; since the brute-force approach considers *all* descriptors, it may have an advantage.

The SIFT/Bag-of-visual-words approach was *not* accurate, however, on test set 2. The reason for this is likely the same as why the brute-force approach worked better with more drastic manipulation: the addition of extraneous elements in the test image may prevent the bag-of-words model from identifying the correct visual words.

References

Farhan, N. S., & Abdulmunem, M. E. (2019, March). Image Plagiarism System for Forgery Detection in Maps Design. In 2019 2nd Scientific Conference of Computer Sciences (SCCS) (pp. 51-56). IEEE.

Huang, A. (2008, April). Similarity measures for text document clustering. In Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand (Vol. 4, pp. 9-56).

Shekhar, R., & Jawahar, C. V. (2012, March). Word image retrieval using bag of visual words. In 2012 10th IAPR International Workshop on Document Analysis Systems (pp. 297-301). IEEE.