Image Retrieval

STIJN BERENDSE* and FREDERICO LOPES*, University of Twente, the Netherlands

Abstract

CCS Concepts: • **Information systems** → *Image search.*

Additional Key Words and Phrases: datasets, computer vision, image retrieval

ACM Reference Format:
Stijn Berendse and Frederico Lopes. 2022. Image Retrieval. 1, 1 (November 2022), 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

As the amount of data generated and consumed is ever increasing [11], information retrieval (IR) systems are more relevant than ever. The use of these systems is already widespread, with examples such as Google, Netflix, and Pubmed. As such, it is important to continuously develop and improve these systems to achieve a performance that is as high as possible, both in terms of correctness and speed.

IR systems can be used for various types of data, ranging from text to multimedia such as images. The best performing search models may not be the same for all data types, and even within a data type, different models may suit different purposes. One of these IR models is the bag-of-words model, which can be applied to both text and image instances [8]. In the latter case, various steps must be taken that can be approached in multiple ways.

Location recognition is one of the areas in which image retrieval is important. Location recognition is based around detecting characteristic features of a certain location, and using these to match other images to the same location, even if the angle or distance is slightly different. This is useful for applications like automated navigation in places where localisation such as GPS is not reliable. Cities are an example of such places, due to a concept known as urban canyons [3]. Location recognition using images can mitigate this issue by supporting localisation systems such as GPS, but to do so, the image retrieval technique must perform as well as possible to prevent incorrect or contradicting location predictions.

The resulting research question is the following:

RQ1 To what extent can performance of image retrieval in location recognition using bag-of-words be improved?

To answer this question, the following subquestions were formulated:

RQ1.1 What descriptor extraction method leads to the highest performance?

RQ1.2 What clustering method leads to the highest performance?

RQ1.3 What distance metrics lead to the highest performance?

1.1 Bag-of-words information retrieval

As mentioned, a bag-of-words model consists of various steps. In general, the first step is to tokenise the separate words from the document corpus. Once all unique words have been identified, they form the vocabulary of the model. Next, for all documents that will be searched, the document is scored before being converted to a vector. This scoring can done in various ways, for example by counting the number of occurrences for each word, and saving these counts in a vector. After the vectors have been constructed, a distance metric can be used to determine similarity between documents.

When applying a bag-of-words model to image retrieval, an image serves as the document, and characteristics of that image serve as "words". The first step in this process consists of detecting and extracting the image features. Generally, approaches for this start by detecting the features of the image. Features generally consist of artifacts like edges, corners, or blobs. Subsequently, the image is split in local patches, which may overlap with each other, which are then converted to numerical vectors. Such vectors are known as feature descriptors. For these descriptors to be usable, they must be grouped as well as possible. This can be done by performing clustering, where each cluster represents a group of descriptors that describe a similar artifact in an image. A cluster center then serves as a "codeword", which works the same as a word in a text document would. The set of codewords forms a vocabulary, known as a "codebook". This codebook can be used to determine similarity between images, by comparing the codewords contained in them.

1.2 Feature detection and extraction methods

The choice of feature detection and extraction method influences the performance of the model, as it influences the way the codewords are formed. If the generated feature descriptors are not correct, the features may not be clustered properly, even if the clustering process is done correctly. As such, there exist various approaches to this challenge, of which several will be discussed below.

1.2.1 ORB. Oriented FAST and Rotated BRIEF (ORB) was developed at OpenCV labs by Rublee et al. in 2011 as an efficient and viable alternative to SIFT and SURF [10], mainly because those are patented algorithms. ORB is built on FAST keypoint detection [9] and BRIEF descriptor extraction [1].

FAST begins by identifying keypoints. To to this, it selects a pixel p and a set of 16 surrounding pixels, if more than 8 pixels are darker or brighter than p, the pixel is selected as a keypoint. After doing this for the entire image and identifying the keypoints, FAST assigns

*Both authors contributed equally to this research.

Authors' address: Stijn Berendse, s.e.berendse@student.utwente.nl; Frederico Lopes, f.m.alvespereiraoliveiralopes@student.utwente.nl, University of Twente, Drienerlolaan 5, Enschede, the Netherlands, 7522NB.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2022 Association for Computing Machinery.
XXXX-XXXX/2022/11-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

an individual orientation to them, depending on how the intensity levels change around the keypoint.

Subsequently, BRIEF starts by smoothing the image using a Gaussian kernel function and then selects a pair of pixels in a neighborhood around a keypoint. The first pixel is drawn from a Gaussian distribution centered around the keypoint with a stranded deviation or spread of sigma. The second pixel in the random pair is drawn from a Gaussian distribution centered around the first pixel. If the first pixel is brighter than the second, then the value is 1, otherwise 0. By repeating this process, all keypoints are converted into a binary feature vector which is a 128 to 512 bit string.[13]

Strengths of ORB are efficiency and the amount of detected features, but ORB may be less accurate[12].

1.2.2 *SIFT.*

Scale Invariant Feature Transform (SIFT) is a patented algorithm and composed of 4 different steps [7].

In the first step, scale-space peak selection, the algorithm produces new variations of the original image that are progressively smaller and blurrier, called scales. Using these, another set of images is created and used to find keypoints. This is done by comparing each pixel to its neighbours in its own scale, as well as those in two surrounding scales. If this results in a local minimum or maximum, the pixel is a potential keypoint.

Some of the keypoints identified in the first step are located along an edge, or don't have enough contrast to make them useful as features. In the second step the keypoints with the highest intensity are selected.

To find the orientation of the keypoints, a neighbourhood is taken around each keypoint location. The gradient of all the pixels in the neighborhood is computed and the orientation of the keypoint is calculated based on the gradient of these points.

In the last step, the descriptors are computed. To do this, a window around the keypoint is taken and divided into blocks of pixels. For each of these blocks, an 8 bin orientation histogram is created. This leads to a feature vector consisting of 128 bin values, which form a keypoint descriptor.

The strength of SIFT is its high accuracy, but it is patented and not as efficient as other algorithms[12].

1.2.3 *BRISK.*

Binary Robust Invariant Scalable Keypoints (BRISK) is a method for keypoint detection, description and matching and consists of two parts, Scale-space keypoint detection and Keypoint description[6].

First, scale-space pyramid layers consisting of a chosen number of octaves and intra-octaves are created. Then, FAST detector is applied on each octave and intra-octave separately to identify potential regions of interest. Next, the points belonging to these regions are subjected to a non-maxima suppression in scale-space: firstly, the point in question needs to fulfill the maximum condition with respect to its 8 neighboring FAST scores in the same layer. Secondly, the scores in the layer above and below need to be lower as well.

For the keypoint description, given the set of keypoints, the descriptor is composed as a binary string by concatenating the results of simple brightness comparison tests.

The strengths and drawbacks of BRISK are similar to ORB, high efficiency and many detected features, but possibly lower accuracy[12].

1.2.4 *AKAZE.*

Accelerated KAZE (AKAZE) is and improved version of KAZE and consists of 3 parts: computing the contrast factor, building the non-linear scale-space and detecting features[4]. For the contrast factor, the image is smoothed by a Gaussian filter and then the maximum absolute gradient value is calculated by looping over all the pixels and calculating the gradient of each one.

The scale-space levels are built by numerically solving partial differential equation iteratively. The scale-space in AKAZE is a pyramidal framework. It consists of octaves and each octave consists of sub-levels.

Finally, DoH (determinant of the Hessian) images are computed at increasing scale sizes as the sub-level increases and the features are extracted by comparing the pixels in DoH images with a surrounding. If the value of this pixel is greater than the other 8 surrounding pixels and a predefined threshold, the pixel is compared spatially with keypoints in the same sub-level to exclude the repeated keypoints that exist inside the same circle.

AKAZE strength is efficiency, but is not as proficient at detecting a large number of features, nor is it as efficient as algorithms such as ORB and BRISK[12].

1.3 Clustering methods

The method used to cluster the descriptors also influences the model performance, because poor clustering may lead to poorly defined codewords. Various clustering methods can be used to generate the best codewords, a number of which will be discussed briefly.

1.3.1 *k-means clustering.*

K-means clustering partitions data into k amount of clusters, so that data points in the same cluster are similar, which means, the distance between them is small. Distance between the points can by calculated by different distance metrics, where Euclidean distance is the most common.

The algorithm does this by minimizing distances within a cluster and maximizing the distances between different clusters.

1.3.2 *Mini-batch k-means clustering.*

Mini-batch k-means is a variation of k-means that uses small, randomly selected, fixed-size batches of data. In each iteration, a random sample of the data is collected and used to update the clusters. It performs better than regular k-means when working on big datasets [5].

1.4 Distance metrics

Finally, the distance metric used to determine similarity between descriptors and images can influence the performance of the bag-of-words model. Various options will be described very briefly, as they would function in a two-dimensional space.

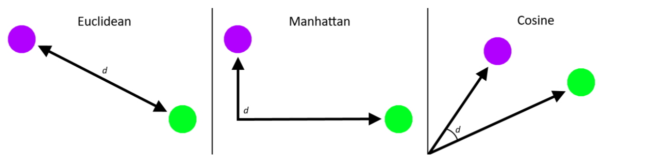

Fig. 1. A visualisation of three distance metrics, d is the distance returned

1.4.1 *Euclidean distance.*
The Euclidean distance is a metric that essentially the distance between two points as the crow flies. For two points p and q, with coordinates (p_1, p_2) and (q_1, q_2) respectively, it can be calculated as shown in equation 1. Figure 1 shows a visualisation of this metric.

$$d(p,q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2} \tag{1}$$

1.4.2 *Manhattan distance.*
The Manhattan distance is a metric that shows the distance between two points by summing the absolute distance in each dimension. For two points p and q, with coordinates (p_1, p_2) and (q_1, q_2) respectively, it can be calculated as shown in equation 2. Figure 1 shows a visualisation of this metric.

$$d(p,q) = \left|q_1 - p_1\right| + \left|q_2 - p_2\right| \tag{2}$$

1.4.3 *Cosine distance.*
The cosine distance is a metric that is calculated based on the angle difference between two points. For two points p and q, with coordinates (p_1, p_2) and (q_1, q_2) respectively, it can be calculated as shown in equation 3. Figure 1 shows a visualisation of this metric.

$$d(p,q) = 1 - \frac{(p_1 * q_1) + (p_2 * q_2)}{\sqrt{p_1^2 + p_2^2} * \sqrt{q_1^2 + q_2^2}} \tag{3}$$

1.5 Evaluating performance

Performance evaluation can be done in various ways, for example evaluating speed, accuracy, or precision. A popular metric for evaluating correctness in information retrieval is mean average precision, or mAP, being used in various benchmark challenges such as MS COCO [2]. This metric is based on taking the average precision per query, averaged over all queries. For a corpus with N queries, it can be calculated as shown in equation 4. In the equation, AP_i represents the average precision for query i.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{4}$$

2 DATASET DESCRIPTION

The data set used in this research consists of 3291 images captured on streets in London, England. The dataset was provided by the University of Twente, the original source being the Mapillary Street-Level Sequences (MSLS). MSLS is a large dataset used for place recognition, containing over 1.6 million images. Images in the dataset used are taken by various cameras, from various viewpoints, at various times, seasons and years. For each image, the absolute coordinates of the camera when the picture was taken is also provided in the dataset, but these values were not used. Provided along with the full dataset was a subset of the dataset, which contained 300 images.

When taking a single image, there are two relations another image can have to it: relevant and not relevant. An image is considered relevant for a queried image, if the location of the two images is nearby. In other words, if the found image portrays the same location as the queried image, but from a different angle or distance, it is considered relevant. An example of this can be seen in figure 2, where the left image is the image that is the query, the middle image is a relevant result, and the right image is a non-relevant result. The middle image is clearly take in a location very close to the queried image, as can be recognised using the pedestrian crossing and the building on the right. Each image can have between 0 and 30 relevant images, but the majority of images have between 1 and 10.

3 METHODOLOGY

In short, the methodology for this research consisted of preparing the dataset, creating and evaluating a baseline model as a control group, and creating various models where certain aspects were changed with regards to the baseline model. The performance of each model was then evaluated and compared to the performance of the baseline model. This performance comparison was first performed on the data subset, after which a second comparison was performed using the top performing variation versus the baseline on the full dataset. The code used for this research is available in a GitHub repository[1].

3.1 Dataset preparation

The first step in the methodology was preparing the dataset. Because the dataset contains all required information already, there were few required steps in this process, namely splitting the dataset in a training set and a test set, as well as converting the images to greyscale. For the full dataset, there were a number of images without any relevant query results. As such, these were removed from the dataset, bringing the number of query images down from 2692 to 2646. For both the subset and full dataset, a 70/30 train and test split was used.

3.2 Creation and evaluation of the baseline model

The next step was preparing a baseline for the model performance. This baseline was taken by training the bag-of-words model with ORB descriptor extraction, k-means clustering and the Euclidean distance metric, as shown in table 1. This configuration was chosen as the baseline because it was provided in the Foundations of Information Retrieval course. First, this baseline was evaluated for the data subset of 300 images, then for the full dataset of 3291 images.

Table 1. The baseline model setup

Model aspect	Approach	Parameters
Feature extraction	ORB	n_keypoints = 50
Clustering technique	k-means	k = 32, n_runs = 5
Distance metric	Euclidean	n.a.

3.3 Creation and evaluation of variations

Next, the approaches discussed in section 1 were used to create model variations. Each variation was created by changing a single aspect of the model, and fitted to the data subset of 300 images. Subsequently, each variation was evaluated, after which the best performing variation was fitted to the full dataset containing 3291

[1]https://github.com/Frederico-lp/Information-Retrieval

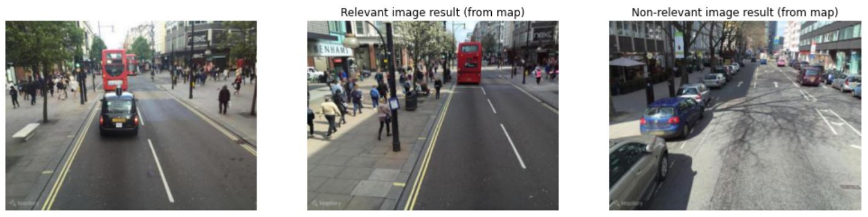
Relevant image result (from map)
Non-relevant image result (from map)

Fig. 2. Three example images from the dataset, left to right: a queried image, a relevant image result, and a non-relevant image result

images. This score was then compared against the baseline performance. Parameters for the clustering methods were the same as the baseline model, unless specified otherwise.

4 RESULTS

The results for the baseline model were a mAP of 0.0151 for the data subset, and a mAP of 0.0056 for the full dataset.

The results of the various model variations that were created can be seen in table 2. A clear performance difference is visible compared to the baseline model for all variations. The best performing variation consisted of SIFT, mini-batch k-means clustering with Euclidean distance, at a mAP of 0.0583.

The mAP of the best performing variation on the full dataset was 0.0153.

Table 2. The mAP scores of the model variations on the data subset

	Extractor	Clustering method	Distance	mAP
1	SIFT	k-means	Euclidean	0.0434
2	SIFT	mini-batch k-means	Euclidean	0.0434
3	SIFT	mini-batch k-means	Manhattan	0.0445
4	SIFT	mini-batch k-means	Cosine	0.0351
5	AKAZE	mini-batch k-means	Euclidean	0.0274
6	BRISK	mini-batch k-means	Euclidean	0.0327
7	ORB	mini-batch k-means	Euclidean	0.0340
8	SIFT	mini-batch k-means(k=100)	Euclidean	0.0583
9	SIFT	mini-batch k-means(k=250)	Euclidean	0.0524
10	SIFT	k-means(k=50)	Euclidean	0.0496
11	SIFT	k-means(k=100)	Euclidean	0.0546
12	SIFT	k-means(k=100)	Manhattan	0.0456

5 CONCLUSION

To conclude, the results of this research showed that performance of a bag-of-words model for image retrieval can be improved by a factor 3.8 on a small dataset and a factor 2.7 on a large dataset, when evaluating for mean average precision. This was achieved by using a combination of SIFT as the feature extraction approach, mini-batch k-means with $k = 100$ as the clustering method, and Euclidean distance as the distance metric. Of these three aspects, the feature extraction method had the largest influence on the performance improvement.

An initially unexpected, yet interesting finding is that the performance of the models decreases substantially when applied to the full dataset instead of the smaller data subset. The hypothesis is that the larger number of possibly relevant images increases the relative problem complexity more than the extra training data benefits the model.

In this research, attempts were made to use different clustering methods like mean-shift, spectral clustering, Ward hierarchical clustering, agglomerative clustering, DBSCAN, and OPTICS. However, due to high execution times and memory size limits none of these were successful.

While the performance improvement achieved is significant, the score is still very low. This is most likely because the bag-of-words approach is a simplistic one for the complexity that the challenge of location recognition poses. Recommendations for future work include using convolutional neural networks, as the sophistication of these more closely matches the complexity of the challenge.

REFERENCES

[1] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. 2010. BRIEF: Binary Robust Independent Elementary Features. In *Computer Vision – ECCV 2010*, Kostas Daniilidis, Petros Maragos, and Nikos Paragios (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 778–792.
[2] MS COCO. n.d.. Detection evaluation. https://cocodataset.org/#detection-eval
[3] Y.J. Cui and S.S. Ge. 2001. Autonomous vehicle positioning with GPS in urban canyon environments. *Proceedings - IEEE International Conference on Robotics and Automation* 2, 1105 – 1110 vol.2. https://doi.org/10.1109/ROBOT.2001.932759
[4] Lester Kalms, Khaled Mohamed, and Diana Göhringer. 2017. Accelerated Embedded AKAZE Feature Detection Algorithm on FPGA. In *Proceedings of the 8th International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies* (Bochum, Germany) *(HEART2017)*. Association for Computing Machinery, New York, NY, USA, Article 10, 6 pages. https://doi.org/10.1145/3120895.3120898
[5] Aman Kharwal. 2021. *Mini-batch K-means Clustering in Machine Learning*. https://thecleverprogrammer.com/2021/09/10/mini-batch-k-means-clustering-in-machine-learning/
[6] Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart. 2011. BRISK: Binary Robust invariant scalable keypoints. In *2011 International Conference on Computer Vision*. 2548–2555. https://doi.org/10.1109/ICCV.2011.6126542
[7] David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60, 2 (2004), 91–110.
[8] Wisam A. Qader, Musa M. Ameen, and Bilal I. Ahmed. 2019. An Overview of Bag of Words;Importance, Implementation, Applications, and Challenges. In *2019 International Engineering Conference (IEC)*. 200–204. https://doi.org/10.1109/IEC47844.2019.8950616
[9] Edward Rosten and Tom Drummond. 2006. Machine learning for high-speed corner detection. In *European conference on computer vision*. Springer, 430–443.
[10] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. 2011. ORB: An efficient alternative to SIFT or SURF. In *2011 International Conference on Computer Vision*. 2564–2571. https://doi.org/10.1109/ICCV.2011.6126544
[11] John Rydning. 2022. Worldwide IDC Global Datasphere Forecast, 2022–2026: Enterprise organizations driving most of the data growth. https://www.idc.com/getdoc.jsp?containerId=US49018922
[12] Shaharyar Ahmed Khan Tareen and Zahra Saleem. 2018. A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. https://doi.org/10.1109/ICOMET.2018.8346440
[13] Deepanshu Tyagi. 2019. *Introduction to ORB (Oriented FAST and Rotated BRIEF)*. https://medium.com/data-breach/introduction-to-orb-oriented-fast-and-rotated-brief-4220e8ec40cf