A Comparative Analysis of Feature Engineering Techniques for Image Retrieval

ELIA CUNEGATTI
University of Twente
e.cunegatti@student.utwente.nl

RUBEN POPPER
University of Twente
r.popper@student.utwente.nl

November 15, 2021

Abstract

We study the differences of several information retrieval systems working with images from the Mapillary Street-Level Sequences (MSLS) dataset. We perform a comparative analysis among several configurations of feature extractors and distance metrics, which are used to build visual vocabularies and calculate similarity scores. We juxtapose image features obtained through SIFT, SURF, HOG, and ORB algorithms, embedded through the bag-of-visual-words (BoVW) approach. We use these descriptors as input for the K-means clustering algorithm. We identify the appropriate image vector representation size generated by the BoVW approach, by selecting the optimal number of cluster centroids. Finally, we extract image features with convolutional neural networks (CNNs), namely the Vgg-16 and ResNet pre-trained models, and compare their performances with those of the classical computer vision algorithms. From our findings, all CNN approaches yield better results than the traditional algorithms. VGG-16 embeddings generate the best retrieval system, regardless of the distance metric used to compute similarity scores. SURF embeddings ouperform those of the other traditional algorithms, and top results are obtained when Cosine distance is used for the retrieval system implementation. Nonetheless, its image representations seem robust to the choice of distance metric.

I. INTRODUCTION

Image retrieval systems are computer programs that help users search, compare, and retrieve information matching a certain query from large databases (Long et al. 2002).

Nowadays, the large volume of data generated poses important challenges to retrieval systems. First, a retrieval system must be able to detect all the documents deemed relevant to a query. Second, the list of relevant results must be ranked according to their importance with respect to the user needs. More specifically, borrowing the terminology from the paper "Learning in Intelligent Information Retrieval" (Lewis 1991), a retrieval system is composed of four stages: indexing, query, comparison, and feedback.

Due to the amount and velocity in which data are generated today, the efficacy of a retrieval system depends on its ability to automate these four processes. In order to do so, computers need a way to analyze files autonomously in a human-like fashion, while reducing the inconsistencies introduced by human errors. This is when machine learning tools come into play. This branch of computer science provides models to identify patterns among the data, formulate predictions and perform classification tasks that supplement the needs of retrieval systems.

Although machine learning finds applications in each of the aforementioned processes of a retrieval system, in this paper we analyze the challenges related to image representation and comparison. At this stage, the performances of a system are dictated by its ability to understand the content of a file. When it comes to multimedia data, like text or images, we need a way to extract the features of a document and convert them to a numeric vector representation. We refer to this process as feature engineering.

The underlying idea is that, when represented in a n-dimensional feature space, similar images will be close to each other. A similarity measure among vectors can then be used to score the relevance of the database image files with respect to a given query vector. Documents with the highest score can thus be retrieved by the system.

For our analysis, we work with the MSLS data set and investigate the effectiveness of several approaches to image embedding. We analyze SIFT, SURF, HOG, and ORB feature extractors, as well state of the art CNNs: VGG-16 and ResNet. For the traditional algorithms, we rely on bag-of-visual-words (BoVW), implemented with several distance metrics, to em-

bed the image descriptors into a feature vector representation (Sivic and Zisserman 2003). Finally, we compare the similarity scores generated by different distance metrics to determine the relevant matches to a given user query.

The remaining of the paper is organized as follows. Section 2 presents a short survey of related work. Section 3 describes the dataset and its challenges. Section 4 provides an overview of the methodology adopted to set up the experiments. Section 4 presents the results. Section 5 concludes.

II. Related Work

Based on the type of features used to analyze documents, image retrieval systems can be categorized as text-based (TBIR) or content-based (CBIR) (Rui et al. 1999).

The former exploits tags, annotation and keywords accompanying a digital image to index documents and retrieve them from a database based on a query string (Alkhawlani et al. 2015). The manual annotation process introduces subjectivity in the indexing stage, which renders this approach non-standardized and hardly scalable to the volume of data generated today.

CBIRs rely on computer vision tools to automatically detect images' low level features such as shapes, textures, colors, spatial layouts and orientation (Long et al. 2002). This standardized features are proven to be more successful for information retrieval purposes (Kato 1992).

In the early years of image retrieval application, most systems represented images through a single vector containing global low level features (Flickner et al. 1995). However, researchers have noticed how global features suffer from a major drawback: they struggle to reduce the semantic gap (Halappa and Sudhamani 2013). This expression refers to the way computers' interpretation of an image differs from humans', who interpret images through higher levels of abstraction that relate to the semantic meaning of an image: the objects contained, the actions or the feelings portrayed in it. (Hare et al. 2006). Conveying the semantic information contained in an image by automatically converting low level characteristics into higher levels of abstraction is a challenging task (Datta et al. 2008).

Single subregions of a picture carry different semantic contents (Halappa and Sudhamani 2013). This is why global descriptors often fail to provide CBIR systems with sufficient discriminating power (Hiremath and Pujari 2008). For this reason, researchers have focused on the analysis of local features, described as the visual descriptors extracted independently from several regions of interest of an image, the keypoints (Aly et al. 2009).

Region-based image retrieval systems (RBIR) are a class of CBIR systems that segment an image into small group of pixels to extract local features (Jagadeesh and Hiremath 2007). This approach have shown to improve retrieval performances (Rao et al. 2011). A local feature extractor algorithm is typically comprised of two items: a detector to identify keypoints, and an extractor to create visual descriptors (Tian 2013). Among the most widely used algorithms we find the Scale-Invariant Feature Transform (SIFT) designed by Lowe 2004, the Speeded-Up Robust Features (SURF) proposed by Bay et al. 2006, the Oriented FAST and Rotated BRIEF (ORB) from Rublee et al. 2011, and the Histogram of Oriented Gradient (HOG) from Dalal and Triggs 2005.

Finally, researchers proposed deep learning approaches for detecting and extracting features (Lenc and Vedaldi 2016; Yi et al. 2016). In particular, convolutional neural network (CNN) provide promising results with respect to image retrieval applications. State of the art performances were achieved on different datasets and for specific computer vision tasks by ResNet (He et al. 2016) and VGG-16 (Simonyan and Zisserman 2014).

For our analysis, we implement the most widely used traditional algorithms and CNN to perform image retrieval on the MSLS dataset.

III. Data

We work with a subset of the Mapillary Street-Level Sequences (MSLS [1]) image collection. Specifically, we use images from the city of London. For our analysis, we consider 500 query images and 1000 map images to be retrieved from the database. To test the performances of the retrieval systems implemented, a matrix of dimension 500×1000 containing binary relevance judgment is provided. More specifically, each cell ij contains the relevance judgment for query image i with respect to the database image j, taking value 1 when relevant, and 0 otherwise.

The dataset poses several challenges to retrieval applications. In particular, although restricted to the city of London, there are plenty of factors that introduce image quality and appearance changes: whether conditions, illumination, time of the day, season, moving objects such as bikes, cars and pedestrians, structural modifications such as roadworks or architectural work, camera intrinsic and viewpoints (Warburg et al. 2020).

Due to the complex nature of the data, we do not expect

[1] https://www.mapillary.com/dataset/places

our models to achieve top results with respect to matching efficiency. For this reason, our focus will not be the absolute evaluation of a particular system, but rather the relative comparison of results among different retrieval system designs.

IV. Methodology

In this section, we present the experimental set up adopted to compare the traditional algorithms and the CNNs for feature extraction. Please bear in mind that the in-depth analysis of the mathematics behind each model is out of the scope of this report. Moreover, due to the limited computing power at our disposal, we are aware that the retrieval systems developed can be further optimized.

i. Experimental Setup

Our goal is to understand how a retrieval system performances change on the basis of the documents' features fed to it. To do this, we freeze the standard architecture (as defined in Lewis 1991) of the system, and run experiments using different algorithms for keypoints identification and feature extraction. The experiments will be run in the following way:

- To define visual dictionary vocabularies, the descriptors from the traditional algorithms are clustered using K-means.
- For each set of descriptors, the most appropriate number of cluster is identified with the heuristic approach of the "Elbow Method" (Thorndike 1953). Therefore, to determine the number of the k parameter (number of centroids) of the Kmeans algorithm, we run the algorithm with 20 different values of k. For each feature extractor, we displayed the clusters' distortion [2] as a function of k, and select the elbow of the curve as optimal value of k.
- For image embeddings, the BoVW framwork is implemented to obtain vector representations of dimension equal to the optimal number of clusters determined in the previous step.
- To build the histogram of visual words for each image, we utilize Euclidean, Manhattan, Minkowski and Cosine distance metrics.
- To query the database, the same distance metrics are calculated to determine images' similarity (i.e. if Euclidean distance is used to generate the embeddings, it will also be used to determine the similarity map and query images).
- To evaluate the results, our choice of performance metrics is the following: Mean Average Precision (MAP), MAP@K and TopRecall@K [3](Warburg et al. 2020).

Please notice that the number of features retained by each algorithm may vary, depending on the functionality of the libraries [4] utilized. Where possible, we imposed a maximum number of descriptors to limit computational time. However, the optimal number of descriptors strictly depends on the content, size, and complexity of the images in consideration. Empirical evaluation is required in order to identify the appropriate values for each algorithm.

ii. Feature Extraction Methods

Feature extraction algorithms are composed of a detector and a descriptor. The way in which they detect keypoints and construct visual descriptors is what determines their success and limitations. Before we proceed, a remark is in order: the quality of the image representations depends on the applications they are used for and on the idiosyncratic properties of the data set. Hence, our results may differ from what the literature proposes as state of the art models.

Let us briefly describe each method the methods in chronological order:

- SIFT (Lowe 2004) and SURF (Bay et al. 2006) extract descriptors using Harris corner points and BLOB detection respectively. Corner points are keypoints invariant to affine transformations (rotation, translation and scaling), and are identified at the junction of two edges. In simpler words, a point of interest results from sudden changes in brightness around sub-regions of an image (Derpanis 2004). BLOB detection instead works by identifying areas with constant properties, thus revealing interesting points where sudden differences among blobs arise. They differ also for the dimensionality of the descriptors obtained. SIFT's descriptors are 128 dimensional vectors, whereas SURF's are 64 dimensional vectors. Because of the lower dimensionality, SURF is faster than SIFT, hence it is usually preferred over for real-time applications. Nevertheless, SIFT usually outperforms SURF

[2] In Information Theory, the squared-error distortion utilized in Kmeans is defined as the sum of the squared distances between each observation vector and its dominating centroid

[3] TopRecall@K is the percentage of queries for which we retrieved at least a correct map image among their k nearest neighbors

[4] Pyhon libraries: ORB and HOG from `scikit-image` (https://scikit-image.org/); SIFT and SURF are patented algorithms from `OpenCV` with limited functionality (https://pypi.org/project/opencv-python/)

in cluttered environments and under illumination and viewpoint changes.

- HOG extractor (Dalal and Triggs 2005) does not only spot edges to identify keypoints, but it differs from the previous approaches for its ability to determine edge's direction via the intensity gradient distributions. To achieve this, HOG starts off by dividing the image into smaller connected cells. For each cell, a histogram of gradient directions is computed based on the analysis of each pixel within the cell. A descriptor is then formed by concatenating the cell-specific histograms. Because it works on local cells, HOG descriptors are invariant to geometric and photometric transformations. To ease calculations, images were re-scaled to obtain a total of 256 descriptors, each represented as a 128 dimensional vector.

- ORB extractor (Rublee et al. 2011) follows a different approach. It relies on FAST (Features from Accelerated and Segments Test) detector, which exploits changes in brightness in the neighbourhood of a pixel to identify keypoints in the image (Rosten and Drummond 2005). Since FAST do not provide an orientation component for the keypoints, ORB algorithm builds a multi-scale representation of the image, referred to as image pyramid. The image pyramid contains several versions of the same image at different resolutions. When FAST is applied at each level of the pyramid, the keypoints identified are partially scale invariant. ORB uses changes in the intensity levels to determine the orientation components of each keypoint. Finally, this algorithm relies on rotation-aware binary-robust-independent-elementary-feature (rBRIEF) descriptor (Calonder et al. 2010), whose features are robust to photometric and geometric transformations. Overall, ORB is more time-efficient than the previous to algorithms as it works with binary descriptors of dimension 256. However, SIFT is more robust to affine transformations of the images, particularly with respect to rotation.

Let us move on to the CNN-based approaches implemented in our analysis. ResNet and VGG-16 are convolutional neural networks developed for the ILSVRC, or the *"ImageNet Large Scale Visual Recognition Challenge"* (Russakovsky et al. 2015), trained on more than 12 million images. In our analysis we use pre-trained versions of both models.

- ResNet (He et al. 2016) is implemented under two different architectures: one with 34 layers, which map images on a 512 dimensional feature space based on 21.282 million parameters, and one with 152 layers, which maps images on a 2048 dimensional feature space based on 58.157 million parameters. We chose to implement both to understand whether the deeper network loses generalization capabilities on our MSLS dataset.

- VGG-16 (Simonyan and Zisserman 2014), where 16 refers to the number of layers of the network, has approximately 138 million parameters. In our implementation, we removed the last fully connected and the softmax output layers to generate 4096 dimensonal feature representations. Although it achieved 92.7% accuracy in the ILSVRC competition, a major drawback of this model is its expensive memory usage and computation time. Another difference regards the depth of the network. Other versions of the network include more layers. However, when growing too deep, this model suffers from the vanishing gradient problem formulated by Hochreiter 1991. This problem refers to the computation of the partial derivatives of the errors for gradient-based and back-propagation models. When the errors become close to 0, the neural network fails to update the weights or, in other words, stops learning. Without going in too much detail, ResNet architecture solves this problem, hence allowing for deeper networks to be constructed.

V. Results

Figure 1 at the end of the report displays the plots produced for determining the value of the parameter k for the Kmeans clustering. The set of results for the traditional computer vision algorithms and the CNNs are also reported below in Table 1 & Table 2.

i. Choice of K

Before assessing the retrieval systems performances, let us analyze the number of the k parameter (number of centroids) for the Kmeans algorithm. Figure 1 comprises four plots, one for each algorithms, that are used to heuristically determine the optimal number of centroids.

According to the elbow method, $k = 45$ was selected for ORB (1a), SIFT (1b), and SURF (1c). The HOG descriptors were instead clustered into $k = 55$ centroids (1d). The similar values of k seem to reflect the overall relatedness of the descriptors extracted for each image by different algorithms. In fact, although the algorithms differ for the methods adopted to detect and extract descriptors, as well as with respect to

their computational efficiency, they are all high-performing approaches. Thus, they are expected to yield similar results when used to determine general properties of image, rather than performing specific tasks. In other words, the number of clusters identified indicates the fact that the pixel-level information extracted by the algorithms can be well summarized with image-vector representations of size 45, for ORB, SIFT and SURF, and size 55 for HOG. Intuitively, we expect HOG performances to differ more than the others, since the clustering performed on its descriptors seems to increase the dimension, hence the nuances, of the final image embedding.

ii. Model Comparison

We now discuss the retrieval systems performances with respect to the evaluation measures computed according to the distance metrics selected for the BoVW embeddings and, consistently, for the similarity scores among query and map images.

Please notice that in Table 1 & Table 2 the results are encoded to highlight the most important values. From a row-level perspective, the highest scores for each algorithm are rendered in bold. When considering the performances of each distance metric with respect to MAP@K, we italicize k^{th} top results. For TopRecall@K, we only look at the last row since, by construction, the values increase as k goes up [5].

- **ORB**: The algorithm achieves the highest MAP with the embeddings generated through the cosine distance metric. However, the best results in terms of MAP@K and TopRecall@K are obtained with the Manhattan norm. It is worth noticing that, with the exception of cosine distance, which achieves the second-best results, the values obtained by other distance metrics are significantly worst.
- **SIFT**: Cosine distance metric resulted in the best performing retrieval system according to all evaluation metrics with the exception of TopRecall@200, where Minkowski distance scored 0.4 percentage points higher (82.8% vs 82.4%). The retrieval system implemented with Euclidean distance achieved, on average, the second-best results. Overall, SIFT descriptors seem robust to the choice of distance metrics, as no substantial variation is exhibited across the columns of Table 1.
- **SURF**: Cosine distance generates the best retrieval system according to all evaluation metrics considered. Nonetheless, when implemented with Manhattan norm, the system reached near-optimal results.
- **HOG**: Top results were obtained with Minkowski distance. The second and third best models used Manhattan and Cosine distances respectively. The former distance metric achieved better results in terms of the system's average precision and top-sensitivity for high values of k. The latter, seems to improve both precision and TopRecall for low values of k. Notice that, when Euclidean distance was utilized, no significant deterioration of the retrieval system performance was observed.

As for the CNNs, the results can be summarized as follows:

- **VGG-16**: Top results were achieved, under all evaluation measures considered, by the retrieval system implemented with Cosine distance.
- **ResNet-34**: Again, Cosine distance resulted in the best performing retrieval system implemented with ResNet-34 image representations
- **ResNet-152**: Despite being the second best CNN model, it provides the least consistent results. Regarding MAP@K, top performances were obtained, for the first three values of k, using Euclidean norm. For $k = 50$, $k = 100$, and $k = 200$ Cosine distance obtained the highest scores. As for TopRecall@K, we obtained even more contrasting results. All distance metrics achieved at least one top score, depending on the value k.

iii. Discussion

For the traditional computer vision algorithms, SURF images embeddings achieved the best performances in our retrieval application. More specifically, the algorithm scores a MAP of 7% with Cosine distance and outperforms the others in terms of MAP@K, for every value of k. As for TopRecall@K, SURF achieves better results than the other algorithms for all values of k, with the exception of $k = 50$, $k = 100$, $k = 200$, where it is outperformed by SIFT.

The second-best performing models are SIFT and HOG. Although HOG reaches higher MAP (4.1% vs 3.8%), the results in terms of MAP@K and TopRecall@K are very close. Once again, for the last three values of TopRecall@K, SIFT outperforms HOG. ORB descriptors generated the worst performing retrieval system according to all evaluation metrics considered, excluding MAP@K and TopRecall@K, for $k = 1$ and $k = 5$.

Regarding the CNN, all the networks considered substantially outperformed the traditional algorithms. VGG-16 results in the optimal retrieval system. It scores 13.2% in MAP, 2 percentage points higher than ResNet-152, 4.1 higher than ResNet-34. For all values of k considered in MAP@K &

[5] In the above paragraph and for the discussion that follows, k refers to the k^{th} image retrieved based on the ordered similarity score computed

TopRecall@K, VGG-16 achieves the best results, followed by ResNet-152, and ResNet-34 at last. VGG-16 superior performances may be attributable to the larger size of its feature vectors. Overall, ResNet-34 generates the poorest image embeddings, which result in lower values for all performance and distance metrics under consideration.

When accounting for the time efficiency of the traditional algorithms, we conclude that SURF should be preferred over the others. Not only it achieves higher performances, but does it more efficiently. We draw a different conclusion for the CNN-based approaches. Although VGG-16 resulted in the optimal model for our retrieval application, ResNet-152 achieved near-optimal results more efficiently.

VI. Conclusion

We proposed an experimental design to investigate differences in retrieval system performances based on: a) the feature extractor utilized, b) the optimal dimension of image embeddings for the traditional algorithms (defined by the parameter k of Kmeans clustering), c) The distance metric used to generate BoVW representations and determine similarity scores.

The final image representation of SURF, SIFT, and ORB extractors, according to the elbow method, was 45; HOG descriptors were instead summarized by 55 dimensional vectors.

CNN-based approaches outperformed the traditional computer vision algorithms. Among the CNNs, VGG-16 image features resulted in the top performing retrieval system regardless the distance measure used to compute similarity scores. As for the traditional computer vision models, SURF image embeddings generated by the BoVW approach using cosine distance yielded the highest values for all the performance measures considered.

Manhattan norm increases the retrieval accuracy of ORB descriptors, whereas Minkowski distance seems particularly suitable for the HOG embeddings. For all other models, cosine appeared to be the preferred distance metric.

Please keep in mind that all the results obtained are inherently dependent on this specific retrieval application. The idiosyncrasies of the MSLS data set prevent our results to be generalized to other retrieval tasks.

Furthermore, due to time and computational constraints, the maximum number of descriptors had to be set, where possible, a priori. Notice also the different clustering techniques could be compared to validate the robustness of the image representations obtained. The additional experiments required to identify the optimal size of the algorithms' output and the soundness of the Kmeans embeddings are left for future work. Finally, a more rigorous quantitative analysis to determine the retrieval systems efficiency is in order if one of the approaches mentioned in this paper were to be selected for real-time applications.

REFERENCES

M. Alkhawlani, M. Elmogy, and H. El-Bakry. Text-based, content-based, and semantic-based image retrievals: A survey. *International Journal of Computer and Information Technology*, 4:58–66, 01 2015.

M. Aly, P. Welinder, M. Munich, and P. Perona. Automatic discovery of image families: Global vs. local features. pages 777 – 780, 12 2009. doi: 10.1109/ICIP.2009.5414235.

H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. volume 3951, pages 404–417, 07 2006. ISBN 978-3-540-33832-1. doi: 10.1007/11744023_32.

M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. volume 6314, pages 778–792, 09 2010. ISBN 978-3-642-15560-4. doi: 10.1007/978-3-642-15561-1_56.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, 2, 06 2005.

R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40:5:1–5:60, 2008.

K. G. Derpanis. The harris corner detector. York University, 2004.

M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: The qbic system. *Computer*, 28:23 – 32, 10 1995. doi: 10.1109/2.410146.

K. Halappa and D. Sudhamani. Content based image retrieval -a survey. *International Journal of Advance Research in Computer Science*, 4:14–20, 01 2013.

J. Hare, P. Sinclair, P. Lewis, K. Martinez, P. Enser, and C. Sandom. Bridging the semantic gap in multimedia information retrieval: Top-down and bottom-up approaches. 187, 01 2006.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

P. Hiremath and J. Pujari. Content based image retrieval using color, texture and shape features. pages 780 – 784, 01 2008. ISBN 0-7695-3059-1. doi: 10.1109/ADCOM.2007.21.

S. Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. 04 1991.

P. Jagadeesh and P. Hiremath. Content based image retrieval based on color, texture and shape features using image and its complement. *International Journal of Computer Science and Security*, 1, 12 2007.

T. Kato. Database architecture for content-based image retrieval. In *Electronic Imaging*, 1992.

K. Lenc and A. Vedaldi. Learning covariant feature detectors. 05 2016. ISBN 978-3-319-49408-1. doi: 10.1007/978-3-319-49409-8_11.

D. D. Lewis. Learning in intelligent information retrieval. In *ML*, 1991.

F. Long, H. Zhang, and D. D. F. Feng. Fundamentals of content-based image retrieval. *Multimedia Information Retrieval and Management*, 01 2002. doi: 10.1007/978-3-662-05300-3_1.

D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60: 91–, 11 2004. doi: 10.1023/B:VISI.0000029664.99615.94.

M. Rao, B. Rao, and D. Govardhan. Content based image retrieval using dominant color and texture features. *International Journal of Computer Science and Information Security*, 01 2011.

E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. volume 2, pages 1508 – 1515 Vol. 2, 11 2005. ISBN 0-7695-2334-X. doi: 10.1109/ICCV.2005.104.

E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: an efficient alternative to sift or surf. pages 2564–2571, 11 2011. doi: 10.1109/ICCV.2011.6126544.

Y. Rui, T. S. Huang, and S.-F. Chang. Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, 10(1): 39–62, 1999. ISSN 1047-3203. doi: https://doi.org/10.1006/jvci.1999.0413. URL `https://www.sciencedirect.com/science/article/pii/S1047320399904133`.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*, 09 2014.

Sivic and Zisserman. Video google: a text retrieval approach to object matching in videos. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1470–1477 vol.2, 2003. doi: 10.1109/ICCV.2003.1238663.

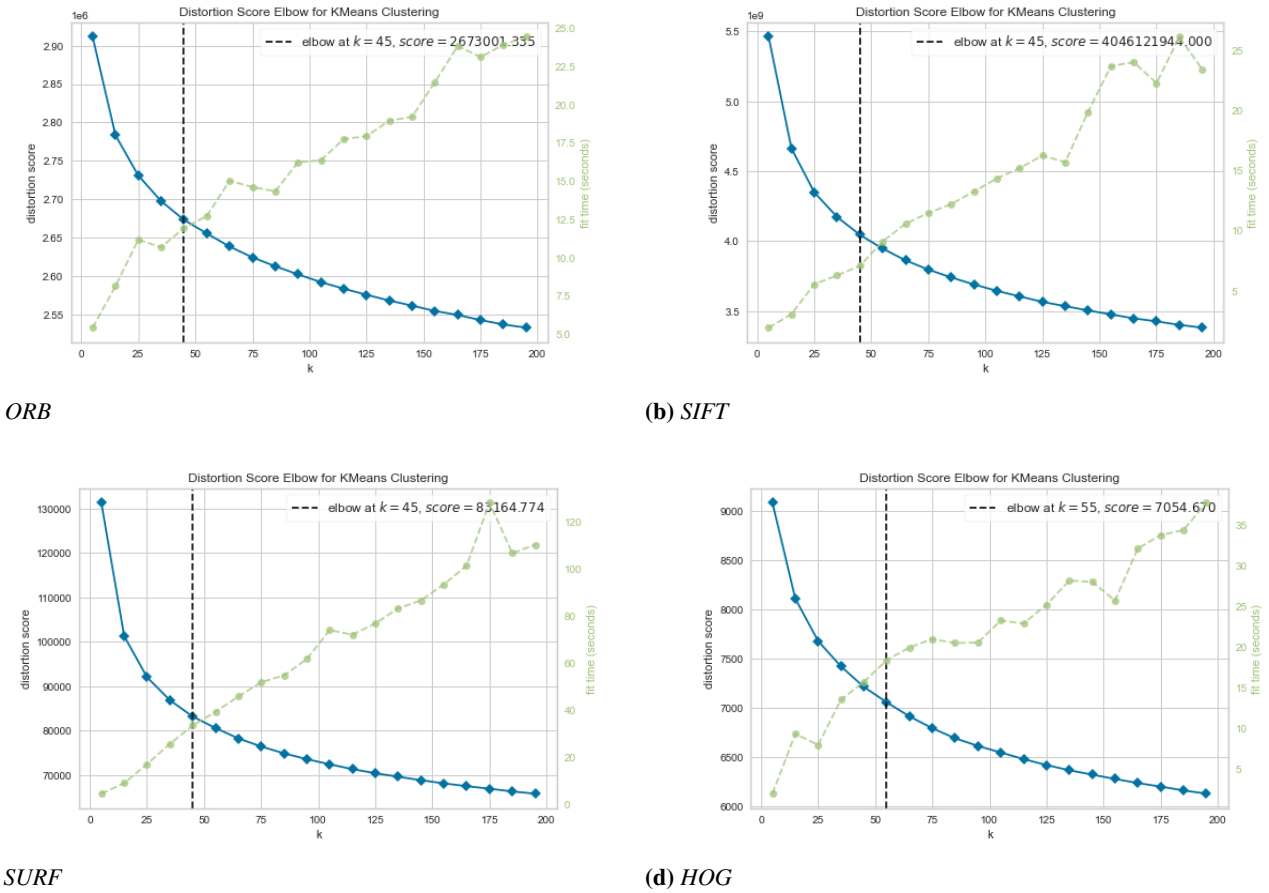
R. L. Thorndike. Who belongs in the family. *Psychometrika*, pages 267–276, 1953.

D. Tian. A review on image feature extraction and representation techniques. *International Journal of Multimedia and Ubiquitous Engineering*, 8:385–395, 01 2013.

F. Warburg, S. Hauberg, M. López-Antequera, P. Gargallo, Y. Kuang, and J. Civera. Mapillary street-level sequences: A dataset for lifelong place recognition. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2623–2632, 2020.

K. Yi, Y. Verdie, P. Fua, and V. Lepetit. Learning to assign orientations to feature points. pages 107–116, 06 2016. doi: 10.1109/CVPR.2016.19.

A. APPENDIX



(a) *ORB*

(b) *SIFT*

(c) *SURF*

(d) *HOG*

Figure 1: *Kmeans elbow curves for the clustering performed on the algorithms' descriptors*

Performance Measures	ORB				SIFT				SURF				HOG			
	Euc.	Manh.	Mink.	Cos.	Euc.	Manh.	Mink.	Cos.	Euc.	Manh.	Mink.	Cos.	Euc	Manh.	Mink.	Cos.
MAP	0.014	0.018	0.014	**0.021**	0.031	0.031	0.03	**0.038**	0.049	0.052	0.049	**0.07**	0.035	0.037	**0.041**	0.038
MAP@K																
$k = 1$	0.008	**0.046**	0.008	0.042	0.03	0.03	0.02	**0.04**	0.048	0.072	0.054	**0.076**	0.03	0.032	**0.04**	0.034
$k = 5$	0.015	***0.076***	0.028	0.037	0.049	0.05	0.043	**0.058**	*0.075*	*0.092*	0.084	**0.108**	0.05	0.057	**0.058**	**0.058**
$k = 10$	0.021	**0.061**	0.031	0.035	*0.056*	*0.056*	*0.049*	**0.064**	0.08	0.091	*0.086*	***0.11***	0.051	0.059	***0.066***	*0.06*
$k = 50$	*0.028*	**0.059**	0.031	0.035	0.055	0.054	0.05	***0.062***	0.072	0.081	0.074	**0.094**	*0.053*	*0.06*	***0.066***	0.058
$k = 100$	0.026	**0.056**	*0.033*	0.029	0.051	0.05	0.046	**0.056**	0.064	0.07	0.067	**0.085**	0.048	0.056	**0.059**	0.05
$k = 200$	0.022	**0.039**	0.023	0.025	0.044	0.043	0.041	**0.048**	0.057	0.062	0.058	**0.079**	0.043	0.049	**0.05**	0.045
TopRecall@K																
$k = 1$	0.008	**0.046**	0.008	0.042	0.03	0.03	0.022	**0.04**	0.048	0.072	0.054	**0.076**	0.03	0.032	**0.04**	0.034
$k = 5$	0.024	**0.118**	0.082	0.062	0.086	0.094	0.082	**0.104**	0.14	0.154	0.14	**0.18**	0.092	0.108	**0.1**	0.106
$k = 10$	0.076	**0.128**	0.1	0.062	0.156	0.158	0.144	**0.162**	0.206	0.198	0.2	**0.24**	0.132	0.154	**0.186**	0.136
$k = 50$	0.236	**0.282**	0.19	0.242	0.428	0.428	0.394	**0.466**	0.46	**0.47**	0.464	0.448	0.386	0.34	**0.442**	0.386
$k = 100$	0.386	**0.416**	0.336	0.354	0.614	0.612	0.608	**0.632**	0.6	0.61	0.608	**0.622**	0.524	0.498	**0.582**	0.574
$k = 200$	0.612	0.626	***0.646***	0.63	0.816	0.808	***0.828***	0.824	0.74	0.736	0.736	***0.778***	0.686	0.632	0.716	***0.752***

Table 1: *Model Comparison: traditional computer vision algorithms for keypoints detection and feature extraction. Descriptors are embedded with BoVW framework using different distances. Final embedding size is set to the optimal number of centroids identified through Kmeans clustering. Similarity scores among images used in the performance evaluation are computed with several distance metrics*

Performances	**VGG-16**				**RESNET-34**				**RESNET-152**			
	Euc.	Manh.	Mink.	Cos.	Euc.	Manh.	Mink.	Cos.	Euc.	Manh.	Mink.	Cos.
MAP	0.097	0.097	0.097	**0.132**	0.079	0.072	0.072	**0.091**	0.1	0.097	0.097	**0.112**
MAP@K												
$k = 1$	0.136	0.136	0.136	**0.174**	0.098	0.084	0.084	**0.104**	**0.148**	0.126	0.126	0.122
$k = 5$	*0.181*	*0.178*	*0.178*	***0.218***	0.137	0.121	0.121	**0.151**	***0.185***	0.17	0.17	*0.179*
$k = 10$	0.176	0.177	0.177	**0.217**	*0.138*	*0.127*	*0.127*	***0.156***	**0.179**	0.17	0.17	0.17
$k = 50$	0.148	0.146	0.146	**0.181**	0.12	0.111	0.111	**0.129**	0.151	*0.146*	*0.146*	**0.152**
$k = 100$	0.129	0.128	0.128	**0.161**	0.108	0.1	0.1	**0.117**	0.137	0.132	0.132	**0.14**
$k = 200$	0.117	0.116	0.116	**0.142**	0.096	0.089	0.089	**0.105**	0.119	0.114	0.114	**0.125**
TopRecall@K												
$k = 1$	0.136	0.136	0.136	**0.174**	0.098	0.084	0.084	**0.104**	**0.148**	0.126	0.126	0.122
$k = 5$	0.298	0.284	0.284	**0.33**	0.224	0.206	0.206	**0.252**	0.276	0.262	0.262	**0.292**
$k = 10$	0.354	0.36	0.36	**0.408**	0.294	0.298	0.298	**0.342**	0.358	0.352	0.352	**0.372**
$k = 50$	0.59	0.596	0.596	**0.642**	0.562	**0.568**	**0.568**	**0.568**	0.62	**0.622**	**0.622**	0.612
$k = 100$	0.724	0.734	0.734	**0.77**	0.714	**0.734**	**0.734**	0.696	0.748	**0.75**	**0.75**	0.744
$k = 200$	0.85	0.848	0.848	***0.876***	0.844	0.838	0.838	***0.856***	0.852	0.848	0.848	***0.88***

Table 2: *Model Comparison: CNN models performances with different distance metrics used for similarity scores computation*