FOUNDATIONS OF INFORMATION RETRIEVAL

Comparing Different Feature Extractor Methods for Content-Based Image Retrieval

November 2022

Alessandro Cortese
a.cortese@student.utwente.nl
University of Twente
Enschede, NL

Gianmarco Lodi
g.lodi@student.utwente.nl
University of Twente
Enschede, NL

1 INTRODUCTION

Given a user query, Information Retrieval (IR) systems search for relevant information in large collections of unstructured or semi-structured data. When these collections grow to a scale where conventional cataloguing methods are no longer sufficient, IR models becomes necessary [34]. Such systems have been so far widely applied in a variety of contexts, including web searching [14], text retrieval [37], audio retrieval [29, 40] and image retrieval [19, 21].

As far as the latter is concerned, there are several approaches to search and retrieve pictures from large databases of digital images. Traditional image retrieval techniques make use of pictures' metadata such as captions, descriptions, titles and keywords to perform text-based retrieval over these [10]. However, these metadata-based systems require human cataloguers to annotate each image, a process which is expensive, labour-intensive and inconsistent, due to inherent variability in human judgments. Moreover, with the recent explosion in the volume of digital images created each day by devices like mobile phones and surveillance cameras, such techniques appear to be no longer adequate for a number of situations.

Another widely employed approach, which has been representing an area of active research since the 1990s, is Content-Based Image Retrieval (CBIR). This IR method seeks to recognize and retrieve from a database the most similar images with respect to the query, based on their specific content. CBIR is preferred over the metadata-based approaches in many instances, since searches that just employ metadata rely on the accuracy and comprehensiveness of the annotations. However, it still suffers from a problem known as *semantic gap*, which consists in the fact that users normally want to retrieve images based on what they represent (*i.e.* their semantics), not just on the similarity between the low-level pixel values. [9, 39]

In this context, machine learning has been thoroughly investigated as a possible way to bridge this *semantic gap* [39]. In particular, deep learning approaches have been proven to model high-level abstractions in data by means of architectures composed of multiple hidden layers with non-linear activation functions, which allow them to automatically learn feature representations, thus eliminating the need for manual feature engineering.

As the specific feature representation (*i.e.* the way in which each image is encoded into a specific vector carrying information about the image itself) and similarity metric (which measures how similar is each of the database images with respect to the query) both play a crucial role in the retrieval performance of a CBIR system, we wanted to explore different configurations of these two variables in order to compare their performances with respect to the task of Visual Place Recognition (VPR). This involves matching one image of a specific location with another of the same location, possibly captured at a different moment or from a different viewpoint. VPR is related to a variety of fields, such as robotics [24], autonomous driving [11] and augmented reality [27].

Furthermore, it is necessary to keep in mind that there is no standard similarity metric that is appropriate in every situation, so it is crucial to test multiple metrics in order to choose the most suitable one for the specific application that the CBIR system is designed for. [3]

More specifically, in this work we compare the performance of three different feature extraction algorithms, namely SIFT, BRIEF and ORB, using a Bag-of-Visual-Words (BoVW) approach to encode images into feature vectors.

Moreover, we also use a pre-trained state-of-the-art Convolutional Neural Network (CNN), the VGG-16, as a feature extractor and compare the performance of the resulting embeddings against that of the BoVW ones.

The rest of this work is structured as follows: after exploring relevant related works in Section 2, we describe the dataset employed in the analysis in Section 3. We then explain the methodological aspects of our work in Section 4, focusing on the specific feature extractors (4.1) and on the experimental setup (4.2). Section 5 presents the main results, which are discussed more in depth in Section 6. Finally, conclusions are provided in Section 7.

2 RELEVANT LITERATURE

Traditionally, CBIR systems were developed to retrieve images from databases using global descriptors based on low-level features like their color [17], texture [26], and shape patterns [13]. Another widely employed approach, which has been demonstrated to find semantically richer image representations, entails aggregating features based on local invariants, such as interest or corner points, and representing images as "bags of visual words" [45, 46]. This method involves the use of some specific Computer Vision algorithms such as SIFT [23], SURF [5] and ORB [32] to extract specific keypoints for the images and to turn them into descriptors. Other related techniques, such as Fisher Vector descriptors [25] and Vector of Locally Aggregated Descriptors (VLAD) have been shown to obtain promising results [18].

However, the main issue with this kind of methods has to do with their reliance on visual similarity, instead of semantic similarity, to retrieve the closest images with respect to a query.

Region Based Image Retrieval (RBIR) was proposed as an extension of the global feature-based approach for CBIR [16]. Its features derive from multiple sub-regions, which correspond to objects in the image. RBIR made it possible to concentrate on the specifics of the various objects in the images, which has shown to improve query accuracy and ease of spatial localization [7].

Metric learning has also been largely employed for CBIR tasks. Its main idea involves learning an optimal metric which minimizes the distance between similar images and at the same time maximizes the distance between dissimilar images [43, 44].

More recently, researchers proposed deep learning approaches as feature extractors. In particular, deep Convolutional Neural Networks (CNNs) have been proven to obtain encouraging results with respect to CBIR tasks [39]. For example, pre-trained models such as ResNet-18 [4] or VGG-16 [2] have been successfully applied for CBIR. Deep CNNs can extract feature representations from an image by taking it as input and retrieving activation values obtained from the final layers of the network, which can carry semantic information, or from the convolutional layers, which bring more spatial information [38].

3 DATASET

To conduct our analysis we used the Mapillary Street-Level Sequences (MSLS) dataset [42], a very commonly used collection of images for Visual Place Recognition tasks. In particular, we focused on a subset of 3291 database images and 2692 query images taken in the streets of London. The pictures are set in different areas of the city, at different times of the day and under varying lighting and weather conditions. In order to test our feature extractors, we exploited the provided 2692 × 3291 image similarity matrix, in which the cell (i, j) is equal to 1 if the j-th database image is relevant for the i-th query image and 0 otherwise.

Looking at the dimensions of the images in the whole dataset, we analysed the values of height and width in pixels. In particular, heights range from 256 to 341 pixels, with an average of 256.01; widths vary from 256 to 455 pixels, with an average of 362.79.

4 METHODS

We investigated different methods of detecting local keypoints within an image. As explained by Yang *et al.* (2007, [46]), “keypoints are salient image patches that contain rich local information of an image, and they can be automatically detected using various detectors and represented by descriptors”. These are then clustered in order to group those with similar descriptors, assigning them into the same cluster. Thus, each cluster acts as a “visual word”, representing the specific local pattern shared by the keypoints in that cluster. In this way we construct a visual-word vocabulary describing the different kinds of local image patterns. Therefore, an image can be represented as a “bag of visual words” (BoVW), namely a vector containing the count of each visual word in that image, which can be used as a feature vector in a number of tasks, such as classification or image retrieval. This BoVW framework is a widely established practice in Computer Vision. [46]

4.1 Feature extractors

In order to answer our research question, we compared different feature extractor methods:

1. **SIFT** (Scale-Invariant Feature Transform), in which important points in the image are selected based on sudden brightness changes in specific regions. Moreover, a staged filtering strategy is implemented to detect features in a scale space, combining a Difference of Gaussian-based keypoint detector with a descriptor based on the gradient orientation distribution in the region [12, 23]. This results in a series of 128-dimensional descriptors which are invariant to translations, rotations and scaling transformations, as well as illumination and blur [28]. The SIFT algorithm has been applied extensively for tasks such as image matching, object detection and 3D reconstruction, and has been proven to obtain good results in specific contexts [22]. However, it is very computationally heavy and as such cannot be used for most real-time applications.
2. **BRIEF** (Binary Robust Independent Elementary Features), which makes use of binary strings as a feature point descriptor. The goal of BRIEF is to be memory efficient and fast to compute: according to some studies, it can be two orders of magnitude faster than SIFT [1]. The descriptor bits in the strings can be 128, 256 or 512 and they are retrieved by comparing the intensities of pairs of points along the same lines, but without requiring a training phase [6]. However, one drawback is that it is not invariant to rotation.
3. **ORB** (Oriented FAST and rotated BRIEF), a fast and lightweight binary descriptor based on BRIEF, which is rotation invariant and resistant to noise [32]. It makes use of the FAST (Features from Accelerated and Segments Test) detector, which exploits variations in brightness around a pixel to identify important areas of the image. FAST keypoints are ordered by the Harris corner measure [31]. ORB provides an image pyramid analysis, which is a multiresolution examination of an image. This contains different versions of the same image at varying resolutions. ORB determines the orientation components of each keypoint based on variations in intensity levels, ultimately detecting fewer features but focusing on finding the most significant ones [8, 36].

For each feature extractor, we also explored and compared different configurations. Regarding SIFT, we tried to vary the number of best features to retain (50, 100 or 200), ranked by their scores, which are measured by magnitude of their local contrast. Similarly, as far as ORB is concerned, we tested different values for the number n of keypoints to be returned (n = 50, 100, 200), keeping the best n keypoints according to the Harris corner response if more than n keypoints are detected. If not, then all the detected keypoints are returned. Lastly, regarding BRIEF, we compared the performances of three different descriptor sizes for each keypoint (128, 256, 512).

As a final comparison, we tested the performances of a CNN-based approach. Specifically, we used the **VGG-16** Convolutional Neural Network, developed for the *ImageNet Large Scale Visual Recognition Challenge* [33] and pre-trained on more than 12 million images. The number 16 refers to the number of layers of the network, which has more than 138 million parameters. In order to get a feature vector for each image, we dropped the output layer originally used to classify, through a softmax function, the input image as belonging to one of the 1000 classes of the original ImageNet dataset. Hence, the final layer of the new network becomes the second fully-connected layer, which encodes each image into a 4096-dimensional vector representing a holistic descriptor of the image content.

4.2 Setup

For each traditional feature extractor we employed the standard BoVW framework for Content Based Image Retrieval.

Firstly, we ran each algorithm to detect keypoints and obtain a series of visual descriptors for every image in the database. We then used K-means clustering on the whole set of descriptors in order to group them by their similarity. This allows us to describe the *visual words* that constitute the images. In particular, K was set to 32 and, for each iteration, the algorithm was run 5 times. The chosen model was the one with the lowest inertia.

Subsequently, each image was turned into a 32-dimensional vector corresponding to the histogram of the occurrences of the *visual words* in the image itself. Basically, for every descriptor of the image, the closest centroid (according to Euclidean distance) was computed and the counts for its corresponding index were increased by one. This means that the sum of the cells in the vector is equal to the number of descriptors of the image. The BoVW vectors were then normalized using z-score normalization. The same procedure was applied to the query images, which get normalized using the mean and variance computed on the database images.

On the other hand, regarding the VGG-16 model, we fed into the Convolutional Neural Network input layer each database and query image, extracting its corresponding 4096-dimensional feature vector.

For each query, the distance between its feature vector (be it the BoVW representation or the VGG-16 vector) and each of the vectors in the database was computed. In particular, four different distance metrics were explored:

- Euclidean distance
- Cosine distance
- Minkowski distance (with $p = 3$)
- Manhattan distance

Afterwards, the images in the database were ranked in ascending order (*i.e.* from closest to furthest) according to these distances. Finally, the performance of the systems was evaluated according to two metrics:

- **Top Recall@K (R@K)**: that is the proportion of queries for which the system was able to find at least one relevant map image among the top K closest images. In particular, we computed R@K for $K = 1, 5, 10$.
- **Mean Average Precision (mAP)**: this corresponds to the sum of the Average Precision (AP) score for each query, divided by the number of queries. The AP score summarizes the uninterpolated precision-recall curve of the query into a single value representing the average of all precisions. Therefore, the mAP approximates the average area under the precision-recall curve for the set of queries [35].

These performance metrics were chosen as they are extensively used in the literature related to Visual Place Recognition [15, 20, 42].

5 RESULTS

Table 1 presents the results obtained for each performance and distance metric considered, with respect to each feature extractor model and its configuration. Note that we rendered in bold text the best values for each descriptor extractor, underlined the best values over all BoVW models, italicized the best values with respect to VGG-16.

- **ORB**: It can be easily noticed that the highest performances are reached when the feature extractor was set to retrieve the descriptors for the best 200 keypoints of every image. While observing that the values are mostly similar among the various distance metrics, it can be seen that cosine distance performed best for each performance metric, with a R@1 of 1%, a R@5 of 2.9%, in a R@10 of 4.9% and a mAP of 2.8%.
- **SIFT**: This feature extractor was found to be the best among the traditional algorithms. The configuration that retained the best 200 features for each image had the highest results, scoring 2.3% in R@1, 7.4% in R@5, 11.2% in R@10 (which is the highest score achieved among all traditional extractors) and 3.9% in mAP. Generally, SIFT outperformed ORB and BRIEF with respect to all the metrics. Also for this descriptor extractor, cosine distance gave the highest values.
- **BRIEF**: The various configurations of this algorithm performed similarly, ranging from 0.5% to 1.2% for R@1, from 2.4% to 3.6% for R@5, from 4.5% to 5.9% for R@10 and from 2.4% to 2.7% for mAP. This clearly means that the descriptor size does not have a great impact on performance. Cosine is once again the best choice for the distance metric.

Overall, cosine distance seems to be the best metric for evaluating recall and precision. Yet, in a couple of cases other distance metrics performed better. For example, regarding ORB extractor with 50 keypoints, Manhattan distance yielded slightly better results for R@1 and R@5. Also, BRIEF extractor with descriptor size equal to 128 showed that R@1 gave a better score with the Euclidean distance. In general, it is not straightforward to compute a ranking of the distance metrics, since, aside from the cosine distance, their scores are comparable.

As far as the VGG-16 model is concerned, it can be clearly remarked that its performance is way higher than that of the traditional descriptor extractors, being, in some cases, two orders of magnitude greater. For instance, VGG-16 achieves a score of 13.2% in R@1 with cosine distance, whilst the traditional descriptors' scores range from 0.7% to 2.3%. Moreover, the CNN model attained a result of 27.3% in R@5, 35.5% in R@10 (the highest score overall) and 10.6% in mAP.

6 DISCUSSION

Regarding ORB and SIFT descriptor extractors, it is evident that performances increase when more features are retained. In fact, for both of them, the highest scores are achieved when at most 200 of the best keypoints are retrieved as descriptors: encoding an image focusing on a larger quantity of salient local points carries out more information and helps in retrieving more relevant examples

		ORB			SIFT			BRIEF			VGG-16
		50	**100**	**200**	**50**	**100**	**200**	**128**	**256**	**512**	
TopRecall@1	*Euclidean*	0.0082	0.0093	**0.0100**	0.0056	0.0152	0.0197	0.0108	0.0093	0.0071	0.1233
	Cosine	0.0082	0.0085	**0.0100**	0.0082	0.0178	**0.0227**	0.0085	**0.0126**	0.0082	***0.1326***
	Minkowski	0.0074	0.0074	0.0089	0.0074	0.0134	0.0160	0.0078	0.0100	0.0078	0.1185
	Manhattan	0.0089	0.0078	0.0089	0.0082	0.0145	0.0167	0.0056	0.0059	0.0045	0.1252
TopRecall@5	*Euclidean*	0.0190	0.0241	0.0208	0.0327	0.0531	0.0605	0.0293	0.0331	0.0301	0.2507
	Cosine	0.0212	0.0271	**0.0294**	0.0394	0.0565	**0.0743**	**0.0368**	0.0364	0.0338	***0.2734***
	Minkowski	0.0204	0.0219	0.0219	0.0342	0.0490	0.0609	0.0275	0.0320	0.0297	0.2474
	Manhattan	0.0219	0.0264	0.0238	0.0327	0.0483	0.0632	0.0230	0.0245	0.0290	0.2611
TopRecall@10	*Euclidean*	0.0372	0.0398	0.0398	0.0565	0.0832	0.1036	0.0494	0.0550	0.0453	0.3273
	Cosine	0.0375	0.0438	**0.0494**	0.0684	0.0869	**0.1122**	**0.0594**	0.0591	0.0583	***0.3555***
	Minkowski	0.0349	0.0364	0.0424	0.0561	0.0828	0.0996	0.0483	0.0539	0.0453	0.3158
	Manhattan	0.0368	0.0420	0.0416	0.0606	0.0802	0.1033	0.0453	0.0461	0.0483	0.3369
Mean Average Precision	*Euclidean*	0.0237	0.0245	0.0258	0.0274	0.0318	0.0355	0.0261	0.0260	0.0254	0.0951
	Cosine	0.0249	0.0256	**0.0279**	0.0301	0.0342	**0.0390**	0.0277	**0.0278**	0.0276	***0.1062***
	Minkowski	0.0236	0.0242	0.0253	0.0270	0.0308	0.0343	0.0256	0.0260	0.0253	0.0924
	Manhattan	0.0242	0.0244	0.0259	0.0278	0.0319	0.0362	0.0243	0.0248	0.0245	0.0983

Table 1: Comparison of the results - Performance for each feature extractor and its configuration, for different distance metrics. Highlighted in bold are the best values for each algorithm with respect to performance metrics (best ones regarding all BoWV models are also underlined, best ones regarding the VGG-16 CNN are also italicized).

given a query. On the other hand, looking at BRIEF scores, it can be observed that different descriptor sizes do not impact on the general performance. This can be due to the fact that, in this case, we are not varying the number of retrieved keypoints, but just their binary descriptors' dimensionality.

Although BRIEF and ORB have a worse overall performance than SIFT, they are much faster at extracting features from images, both being binary descriptors that are implemented with efficiency and speed as their main peculiarities. That makes them also more suitable for online applications.

As concerns Mean Average Precision, it can be noticed that it is the metric with the lowest variance among the others, with respect to all the algorithms tested: for the traditional ones, mAP ranges from 2.3% to 3.9% and it barely reaches 10% with the VGG-16 model. This is most likely due to the fact that, unlike TopRecall@K, mAP is a metric that, averaging the AP through all the queries, has particularly good discrimination and stability. [35]

7 CONCLUSIONS

We investigated three different traditional feature extractors algorithms and a state-of-the-art CNN for the task of Visual Place Recognition using a subset of the MSLS dataset containing images taken in the city of London. As far as the traditional Computer Vision algorithms are concerned, we modified a specific parameter in order to derive three different configurations: this was the best features to retain for SIFT and ORB, while for BRIEF we changed the descriptors' size. The closest images in the database were then retrieved according to four different distance metrics: Euclidean distance, cosine distance, Minkowski distance and Manhattan distance. Finally, we evaluated the performance of the Image Retrieval task according to two widely used metric in this field: Top Recall@k and Mean Average Precision.

The results we obtained clearly show that the VGG-16 Convolutional Neural Network largely outperforms the traditional feature extractors, obtaining the highest R@K and mAP with respect to every distance metric.

Among the three traditional feature extractor algorithms, SIFT embeddings led to the highest results with respect to every metric, whereas ORB and BRIEF obtain comparable results, often surpassing each other in different metrics.

As far as the performance metrics are concerned, cosine distance stands out as the best distance measure with highest results across all the descriptors. The other metrics achieved comparable scores.

More work could be done to investigate how the retrieval performance of these approaches changes by varying other parameters, such as the k in K-means clustering or the normalization technique applied on the data. Also, exploring different clustering tecniques (*e.g.* spectral clustering, hierarchical clustering, etc.) could be useful to understand whether that has an impact on the overall system performance. Another aspect which we omitted is the use of specific statistical tests to validate the robustness of our findings.

Moreover, the MSLS dataset comes with a further set of relevant judgements, aside from the image similarity matrix that we exploited, which is the Field of View matrix. This specifies a degree of similarity defined in the continuous interval [0,1], explaining how much each query is similar to the database images. In future works, it could be interesting to make use of these values in order to get a more detailed performance evaluation.

Overall, our findings confirm that the deep learning approach is very effective at learning good feature representations for our CBIR task. This appears to be a very promising way to address the *semantic gap* problem inherent in image retrieval [41]. However, it should be noted that such methods are not devoid of limitations. For example, they generally require extremely large amounts of data and computing power during their training phase. Even though pre-trained models exist and are widely adopted, features learned from deep Neural Networks are specific to the dataset they have been trained on, which makes this approach difficult to use in the case of niche applications with very specific image content. On the other hand, traditional Computer Vision algorithms are more general and perform the same for any image [30].

REFERENCES

[1] Prashant Aglave and Vijaykumar S Kolkure. 2015. Implementation of High Performance Feature Extraction Method Using Oriented Fast and Rotated Brief Algorithm. *Int. J. Res. Eng. Technol* 4 (2015), 394–397.

[2] Ahmad Alzu'bi, Abbes Amira, and Naeem Ramzan. 2017. Content-based image retrieval with compact deep convolutional features. *Neurocomputing* 249 (2017), 95–105.

[3] Ahmad Alzu'bi, Abbes Amira, and Naeem Ramzan. 2015. Semantic content-based image retrieval: A comprehensive study. *Journal of Visual Communication and Image Representation* 32 (2015), 20–54.

[4] Swarnambiga Ayyachamy, Varghese Alex, Mahendra Khened, and Ganapathy Krishnamurthi. 2019. Medical image retrieval using Resnet-18. In *Medical Imaging 2019: Imaging Informatics for Healthcare, Research, and Applications*, Vol. 10954. SPIE, 233–241.

[5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. 2006. Surf: Speeded up robust features. In *European conference on computer vision*. Springer, 404–417.

[6] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. 2010. Brief: Binary robust independent elementary features. In *European conference on computer vision*. Springer, 778–792.

[7] Chad Carson, Megan Thomas, Serge Belongie, Joseph M Hellerstein, and Jitendra Malik. 1999. Blobworld: A system for region-based image indexing and retrieval. In *International conference on advances in visual information systems*. Springer, 509–517.

[8] Payal Chhabra, Naresh Kumar Garg, and Munish Kumar. 2020. Content-based image retrieval system using ORB and SIFT features. *Neural Computing and Applications* 32, 7 (2020), 2725–2733.

[9] John P Eakins. 2002. Towards intelligent image retrieval. *Pattern Recognition* 35, 1 (2002), 3–14.

[10] Peter GB Enser. 1995. Progress in documentation pictorial information retrieval. *Journal of documentation* (1995).

[11] Sourav Garg, Tobias Fischer, and Michael Milford. 2021. Where is your place, visual place recognition? *arXiv preprint arXiv:2103.06443* (2021).

[12] Carsten Griwodz, Lilian Calvet, and Pål Halvorsen. 2018. Popsift: A faithful SIFT implementation for real-time applications. In *Proceedings of the 9th ACM Multimedia Systems Conference*. 415–420.

[13] Venkat N Gudivada and Vijay V Raghavan. 1995. Content based image retrieval systems. *Computer* 28, 9 (1995), 18–22.

[14] Venkat N Gudivada, Vijay V Raghavan, William I Grosky, and Rajesh Kasanagottu. 1997. Information retrieval on the world wide web. *IEEE Internet Computing* 1, 5 (1997), 58–68.

[15] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. 2021. Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14141–14152.

[16] Ryota Hinami, Yusuke Matsui, and Shin'ichi Satoh. 2017. Region-based image retrieval revisited. In *Proceedings of the 25th ACM international conference on Multimedia*. 528–536.

[17] Anil K Jain and Aditya Vailaya. 1996. Image retrieval using color and shape. *Pattern recognition* 29, 8 (1996), 1233–1244.

[18] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. 2010. Aggregating local descriptors into a compact image representation. In *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 3304–3311.

[19] Chih-Chin Lai and Ying-Chuan Chen. 2011. A user-oriented image retrieval system based on interactive genetic algorithm. *IEEE transactions on instrumentation and measurement* 60, 10 (2011), 3318–3325.

[20] María Leyva-Vallina, Nicola Strisciuglio, and Nicolai Petkov. 2021. Generalized contrastive optimization of siamese networks for place recognition. *arXiv preprint arXiv:2103.06638* (2021).

[21] Chuen-Horng Lin, Rong-Tai Chen, and Yung-Kuan Chan. 2009. A smart content-based image retrieval system based on color and texture feature. *Image and vision Computing* 27, 6 (2009), 658–665.

[22] Tony Lindeberg. 2012. Scale invariant feature transform. (2012).

[23] David G Lowe. 1999. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, Vol. 2. Ieee, 1150–1157.

[24] Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J Leonard, David Cox, Peter Corke, and Michael J Milford. 2015. Visual place recognition: A survey. *IEEE Transactions on Robotics* 32, 1 (2015), 1–19.

[25] Bingpeng Ma, Yu Su, and Frédéric Jurie. 2012. Local descriptors encoded by fisher vectors for person re-identification. In *European conference on computer vision*. Springer, 413–422.

[26] Bangalore S Manjunath and Wei-Ying Ma. 1996. Texture features for browsing and retrieval of image data. *IEEE Transactions on pattern analysis and machine intelligence* 18, 8 (1996), 837–842.

[27] Carlo Masone and Barbara Caputo. 2021. A survey on deep visual place recognition. *IEEE Access* 9 (2021), 19516–19547.

[28] Darshana Mistry and Asim Banerjee. 2017. Comparison of feature detection and matching approaches: SIFT and SURF. *GRD Journals-Global Research and Development Journal for Engineering* 2, 4 (2017), 7–13.

[29] Dalibor Mitrović, Matthias Zeppelzauer, and Christian Breiteneder. 2010. Features for content-based audio retrieval. In *Advances in computers*. Vol. 78. Elsevier, 71–150.

[30] Niall O'Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco Hernandez, Lenka Krpalkova, Daniel Riordan, and Joseph Walsh. 2019. Deep learning vs. traditional computer vision. In *Science and information conference*. Springer, 128–144.

[31] Edward Rosten and Tom Drummond. 2005. Fusing points and lines for high performance tracking. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, Vol. 2. Ieee, 1508–1515.

[32] 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*. Ieee, 2564–2571.

[33] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision* 115, 3 (2015), 211–252.

[34] Mark Sanderson and W Bruce Croft. 2012. The history of information retrieval research. *Proc. IEEE* 100, Special Centennial Issue (2012), 1444–1451.

[35] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*. Vol. 39. Cambridge University Press Cambridge.

[36] Ashish Sharma, Anmol Mittal, Savitoj Singh, and Vasudev Awatramani. 2020. Hand gesture recognition using image processing and feature extraction techniques. *Procedia Computer Science* 173 (2020), 181–190.

[37] Howard R Turtle and W Bruce Croft. 1992. A comparison of text retrieval models. *The computer journal* 35, 3 (1992), 279–290.

[38] Maria Tzelepi and Anastasios Tefas. 2018. Deep convolutional learning for content based image retrieval. *Neurocomputing* 275 (2018), 2467–2478.

[39] Ji Wan, Dayong Wang, Steven Chu Hong Hoi, Pengcheng Wu, Jianke Zhu, Yongdong Zhang, and Jintao Li. 2014. Deep learning for content-based image retrieval: A comprehensive study. In *Proceedings of the 22nd ACM international conference on Multimedia*. 157–166.

[40] Avery Wang et al. 2003. An industrial strength audio search algorithm.. In *Ismir*, Vol. 2003. Citeseer, 7–13.

[41] Huafeng Wang, Yehe Cai, Yanxiang Zhang, Haixia Pan, Weifeng Lv, and Hao Han. 2015. Deep learning for image retrieval: What works and what doesn't. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. IEEE, 1576–1583.

[42] Frederik Warburg, Soren Hauberg, Manuel Lopez-Antequera, Pau Gargallo, Yubin Kuang, and Javier Civera. 2020. Mapillary street-level sequences: A dataset for lifelong place recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2626–2635.

[43] Kilian Q Weinberger and Lawrence K Saul. 2009. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research* 10, 2 (2009).

[44] Lei Wu and Steven CH Hoi. 2011. Enhancing bag-of-words models with semantics-preserving metric learning. *IEEE MultiMedia* 18, 1 (2011), 24–37.

[45] Lei Wu, Steven CH Hoi, and Nenghai Yu. 2010. Semantics-preserving bag-of-words models and applications. *IEEE Transactions on Image Processing* 19, 7 (2010), 1908–1920.

[46] Jun Yang, Yu-Gang Jiang, Alexander G Hauptmann, and Chong-Wah Ngo. 2007. Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the international workshop on Workshop on multimedia information retrieval*. 197–206.