Analysis of Fine-Grained Localisation Methods

Kelvin Lim Wan
(929715)
kelvinl3@student.unimelb.edu.au
The University of Melbourne

Kazuya Soga
(1026644)
ksoga@student.unimelb.edu.au
The University of Melbourne

***Abstract*—Location recognition refers to the task of identifying the geographical location of a photograph. This is a challenging problem, as the appearance of real-world locations can vary widely. For instances, two images pictured at the same location can appear different to each other in visual appearance due to numerous reasons, this includes: lighting conditions, the angle of the object being captured, and even differences in capturing devices. Historically, handcrafted local feature descriptors such as SIFT and ASIFT have been utilized to tackle this task. However, more recent techniques such as NetVLAD propose to 'learn' the feature descriptors. This paper proposes a comparative analysis of various techniques used for feature extraction and query prediction. We use SIFT, ASIFT, NetVLAD, and self-supervised neural networks as feature extractors, and pair them with feature matching methods such as FLANN, KNN and neural networks. Our experiments show mixed results in the performance of handcrafted and learned features. While both ASIFT and SIFT beat the self-supervised neural networks, the learned features of NetVLAD display the best results, reducing query error compared to SIFT by a factor of 3.4, while improving computational efficiency by up to 8279x.**

I. Introduction

Location recognition, also known as visual place recognition [2], location identification [6] and image matching [18], has seen considerable attention in recent years due to its practical applications in autonomous driving [12] and augmented reality [13]. Given a query image, visual place recognition aims to determine the geographical location of where the image was photographed.

Previous work in this domain have casted visual place recognition as a visual object retrieval task, where a queried image is compared to a larger dataset of geo-tagged images, and the geolocation of the image with the highest similarity to the queried image is returned. To compare the similarity of two images, SIFT for example can be used to extract the local features descriptors of the two images, then a feature matching method such as FLANN can be used to match similar features. The database image that produces the highest number of similar features is deemed the closest match, and its geotagged label is returned to the user.

More recently, after the success of AlexNet in the 2012 ImageNet challenge [9], Convolutional Neural Networks (CNNs) have been the preferred method for computer vision tasks [11]. As a result, numerous papers have proposed CNN's for visual place recognition to effectively learn and extract feature descriptors [1, 5].

In this paper, we are interested in the performance of various methods for visual place recognition. To this end, we benchmark numerous handcrafted and learned feature extractors when paired with different feature matching techniques.

II. Method

In this section, we consider several techniques to guide our task of recognising an image location. We split the techniques into two categories: Feature Engineering and Query Prediction. Feature Engineering methods take as input raw images and output image features that are representative of its location. Query Prediction methods take in those features and return the image location as (x, y) coordinates. Note that the original images are reduced to a smaller size as it allows our algorithms to compile faster while still maintaining the pixel information required for feature engineering and query prediction.

A. Feature Engineering

SIFT (Scale-Invariant Feature Transform) is a feature detection algorithm that takes an image and transforms its raw data into scale-invariant keypoints. Keypoints can be regarded as unique and interesting points in a image. They are computed using two steps: first, a difference-of-Gaussian function is used at different scales across the image to detect points invariant to scale and orientation. Next, each point is compared with its neighbourhood to detect points with low contrast or points poorly-localised along an edge. Such points are rejected and the remaining ones consitute the keypoints. Each keypoint is described by a descriptor. A descriptor is computed using a histogram of the gradients between the keypoint and its neighbouring pixels. The histogram is normalised such that the features are invariant to scale, rotation and translation and further compressed into a vector of length 128. Compression is performed such that the vector representation allows for geometric distortion and distinct lighting settings [10]. SIFT is a good feature extraction technique for image matching since two photographs taken at the same location should display similar local features ,but they might differ in scale, translation and/or rotation. Hence, detecting features invariant to these transformations facilitates matching.

ASIFT (Affine-SIFT) is an iterated version of the SIFT algorithm. It provides additional invariance to the camera axis orientations, that is, it is more invariant to affine transformations [14]. ASIFT aids in identifying features where the objects

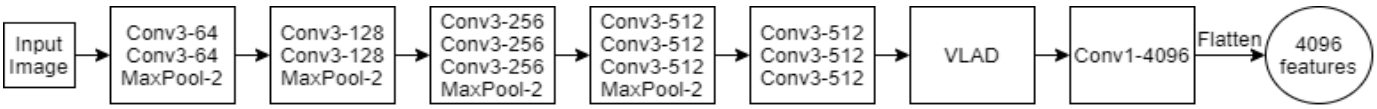

Fig. 2. CNN architecture for NetVLAD

have undergone massive distortion due to the camera's field of view.

Self-Supervised Learning on Image Rotation entails the learning of image features by training a CNN to predict the angle of rotation applied to an input image. Research from [7] shows that this trivial task actually provides powerful semantic features for further prediction. Features are extracted from one of the last layers in the CNN; a one-dimensional vector is generally preferred. We first preprocess the images by alternatively rotating each image in the training set and assign labels corresponding to their rotation. We consider four labels for the rotations of 0°, 90°, 180° and 270°. The architecture of the CNN is shown in Figure 1.

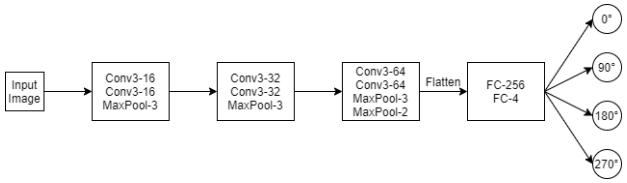

Fig. 1. CNN architecture for Self-Supervised Learning on Rotation

Self-Supervised Learning on Image Warping: Since some photographs are generally taken with a wide field of view, there is a certain degree of object distortion in those pictures. For instance, two photographs of a same scene could contain a same object that is represented differently due to warping. To counter this problem, we train a self-supervised CNN to discriminate image warping and extract the features from one of its deep layers [3]. We apply the same concept as with self-supervised learning on rotation and utilise the same CNN architecture. However, we use five labels for no distortion, left distortion, right distortion, top distortion and bottom distortion.

NetVLAD [1] is a CNN with a VLAD (Vector of Locally Aggregated Descriptors) layer positioned deep into the network. Its architecture is shown in Fig. 2. The CNN architecture is based on best practices for image retrieval so that it is robust to transformations and brightness. The VLAD layer essentially captures information about local descriptors in an image; it uses the sum of residuals between a descriptor and its corresponding cluster centre for each object. Since there is empirical evidence from [1] that NetVLAD performs well in different location recognition environments such as Tokyo and Pittsburgh, we consider it as a feature extraction method.

B. Query Prediction

FLANN-based Matcher (Fast Library for Approximate Nearest Neighbours) contains a library of algorithms optimised for fast nearest neighbour search in large datasets. It takes the descriptor generated for a test image (generally using SIFT or ASIFT) and matches it to all other features in the training images dataset using an optimisation algorithm as heuristic. The closest match in the training dataset is returned [15]. A FLANN-based Matcher is suitable for prediction given descriptors from SIFT and ASIFT since it returns the closest match with relatively small execution time. The label of the closest match is returned as the estimated location of a test image. We use the K-dimensional tree with 5 trees and 50 checks as heuristic for FLANN.

K-Nearest Neighbours (KNN) takes the features from a test image and compares it with the features from all the training images. The K nearest neighbours are determined based on a similarity measure [8]. Our implementation uses a K-value of 1 and the Euclidean distance as similarity metric. The features for a test image is input to the algorithm and its one nearest neighbour is returned. The label of that nearest neighbour is then assigned as the test image's position.

Multi-layer Perceptron: A Multi-layer Perceptron (MLP) consists of at least three layers of nodes with non-linear activation functions. It helps in distinguishing data that is not linearly separable [16]. We build an MLP with 2 hidden layers and 1 output layer that returns 2 values- the x and y coordinates of the image location. A linear activation is used in the output layer since we are performing regression to estimate the location coordinates.

C. Baseline

We define a simple CNN that takes as input a raw image and returns its location as (x, y) coordinates as a baseline. Its architecture is shown in Fig. 3. A good model should at a minimum outperform this baseline.

Fig. 3. CNN architecture for Baseline

III. Experiments

We perform feature extraction on the raw images in the training dataset using SIFT, ASIFT, Self-Supervised Learning on both Image Rotation and Image Warping and NetVLAD. Since SIFT and ASIFT both return keypoint descriptors as features, it only makes sense to use a FLANN-based matcher. Further, the feature sets from the SIFT and ASIFT algorithms do not have a fixed size; the size depend on the number of keypoints found in the image. Using MLP or KNN on these feature sets would require padding the feature sets, which

Feature Extraction Method	Feature Extraction time (secs)	Prediction Method	Prediction Time (secs)	Error (MAE)
SIFT	86	FLANN	8776 (0.06x)	20.2 (1.78x)
ASIFT	1016	FLANN	168960 (0.00x)	27.64 (1.30x)
Self-Supervised Learning on Rotation	138.6	MLP	28.3 (20.92x)	29.75 (1.21x)
		KNN	0.33 (1793.94x)	28.41 (1.26x)
Self-Supervised Learning on Warping	127.9	MLP	27.4 (21.60x)	39.76 (0.90x)
		KNN	0.32 (1850.00x)	40.11 (0.89x)
NetVLAD	28.5	MLP	29.3 (20.20x)	16.92 (2.12x)
		KNN	1.06 (558.49x)	5.88 (6.10x)
Baseline CNN			592 (1.00x)	35.89 (1.00x)

Fig. 4. Tabular results of the benchmarks of various location recognition techniques.

would add noise to the data and consequently add bias to the model. Therefore, we only use MLP and KNN for prediction on the features extracted through Self-Supervised Learning on both Rotation and Warping and NetVLAD. FLANN-based matching would consider each input in the flattened feature array as a keypoint with the element value describing that keypoint (the descriptor is a value instead of an array) and therefore it is ignored for feature extractions methods other than SIFT and ASIFT.

A. Setup

For SIFT and ASIFT, we use their published implementation provided by Python's OpenCV2. The image transformations in Self-Supervised Learning are performed using the Scikit-image and Scipy libraries. We also import a published implementation of a NetVLAD architecture from [1] and use it with its default configuration and weights pretrained on image datasets for fine-grained localisation in Pittsburgh and San Francisco [17]. Lastly, we utilise OpenCV2's FLANN algorithm, Scikit-learn's KNN implementation and build the MLP using the Keras library. The MLP is trained with a batch size of 100 for 200 epochs. All experiments are conducted on a 3.7GHz Ryzen 9 5900x CPU, 32 GB RAM and RTX 3800 GPU.

B. Dataset

Our dataset consists of 7500 images captured inside and around an art museum (Getty Center in Los Angeles, USA). These images were taken using Google Street View. Each image in our dataset is labelled with its GPS location as (x, y) coordinates.

C. Metrics

The methods are contrasted in terms of the execution times for both feature extraction and prediction, and the Mean Absolute Error (MAE). Note that the training dataset is randomly partitioned into a holdout set for model evaluation. 80% of the dataset is used to train and predict and 20% is used to assess the method. Since the execution times are quite high, we could not perform cross-validation to get a more accurate MAE due to time restrictions.

D. Results

The results are summarised in the table shown in Fig. 4 and the corresponding plots are shown in Fig. 5.

1) Execution Times: In Fig. 5, we see that NetVLAD extracts features the fastest. This is because NetVLAD does not require training as we use pre-trained weights as opposed to the Self-Supervised methods. NetVLAD outperforms Self-Supervised Learning on Rotation and Warping by 4.86x and 4.49x respectively. Additionally, the Self-Supervised Learning methods and NetVLAD utilise the GPU, which significantly improves their computational efficiency compared to SIFT and ASIFT. ASIFT outputs features the slowest due to its additional pre-processing, which coincides with the results of [18].

The fastest prediction method is KNN, followed by MLP and then FLANN. For each test image, KNN computes the Euclidean distance between the test features and each of the training features. As a result, its time complexity is directly proportional to both the number of features training instances. However, we precompute the Euclidean distances of each training instance so that upon prediction, the KNN only needs to compute the Euclidean distance of the test instance and then compare it to the ones for the training instances. In contrast, the time complexity of MLP prediction is directly dependent on the number of parameters in the neural network. For the Self-Supervised Learning features, we select 32 neurons per hidden layer since the input size is 256 and for the NetVLAD features of size 4096, we have 128 neurons in each hidden layer. Consequently, we have a numerous parameters in both networks since a large number of neurons imply a large number of parameters, which results in slow computation. For instance, the prediction times for both KNN and MLP are smaller for the Self-Supervised Learning features of size 256 as opposed to the NetVLAD features of size 4096. FLANN has

[1]https://github.com/crlz182/Netvlad-Keras

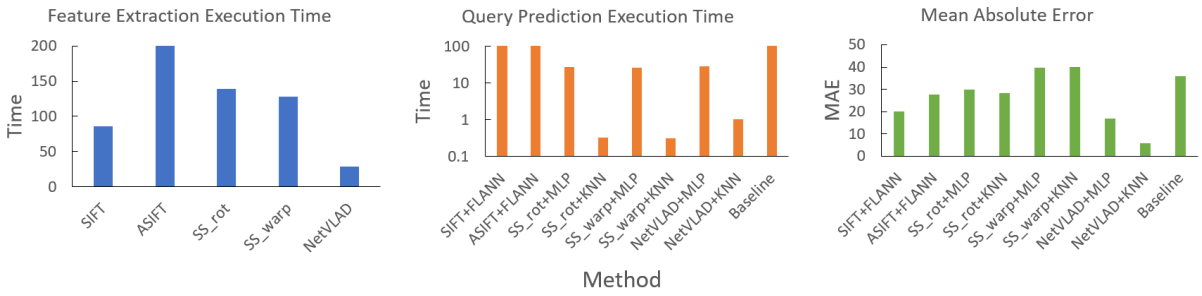

Fig. 5. Visual results of the benchmarks of various location recogniton techniques.

the slowest computation time since each descriptor from SIFT and ASIFT have a length of 128 and the number of descriptors depends on the number of keypoints found, which is 67 for SIFT and 879 for ASIFT on average. This results in feature sets of high dimensionality and therefore high computational complexity.

2) Query Accuracy: Observing the results from Fig. 5, the combination of NetVLAD for feature extraction and KNN results in the lowest MAE. NetVLAD reduces the error from reduction from Self-Supervised Learning on Rotation and Warping by 79% and 85% respectively, when predicted using KNN. This is because NetVLAD extracts features that are more indicative of the image location since the weights used were trained on a supervised dataset that was purposefully built for fine-grained localization of images. Comparing the prediction methods, we can see that regardless of the feature extraction technique, MLP displays a higher error than KNN. This is because MLP is a neural network, which means that the it learns the relationship between each feature and the label (location). This could be misleading if, for instance, two images that are located far apart both have paintings in their raw image which are extracted as features; the network may average the coordinates of the two locations and 'assign' it to the painting features. KNN simply returns the location of the training image that is most similar in terms of features, which makes more sense. SIFT with FLANN and ASIFT with FLANN, both outperform all other combinations except for the methods using NetVLAD as feature extraction method. This is because the concept of keypoints, descriptors and feature matching is compatible with our task of image matching.

3) Qualitative Analysis: SIFT and ASIFT with FLANN imply finding important points in images, matching those points in a test image to the points in training images, and returning the location of the training images with the highest number of matches. This technique is very intuitive and makes sense in our context. Even though computing the keypoints and their descriptors is a lengthy process, the results are generally accurate. However, FLANN is specifically designed for keypoint descriptors. It would not make sense to generalize the method to the other features extracted. Features from Self-Supervised Learning methods are not implicitly indicative of image locations. They are rather simple characteristics of an image. That is why their results are not as desirable as the other feature extraction techniques. They would perform better in another environment such as object recognition. The NetVLAD architecture, together with the pretrained weights, is explicitly designed for image localization, which explains why the corresponding results are adequate. MLP, with its last layer performing regression is not quite appropriate for our task of image matching. Perhaps a modification to perform classification would return better results. However, we would need much more training data to tailor the parameters weights since there are 1499 distinct labels in the existing dataset and only 7500 training images (only 5x more).

KNN (with K=1) is the most intuitive prediction method; given some features, it returns the image that is the most similar. This is very appropriate for our task since if a test image looks the most similar to a specific training image, it is most likely to have been photographed in the same place. With good features, this method performs the best theoretically.

IV. Conclusion

In this paper, we benchmark and provide a comparative analysis of various feature extraction and image matching methods for the task of location recognition. From our evaluations, we find that NetVLAD displays the best results in GPS location accuracy when paired with KNN, outperforming the baseline CNN by 6.1x in error reduction, while running for 559x less time. On the other end of the spectrum, using MLP for feature matching produces subpar performance for all learned feature extractors (NetVLAD, self-supervised learning). For the handcrafted descriptors, we found surprisingly that ASIFT performed worse than SIFT, increasing error by 37% while also taking 12x more time.

For future work, including additional benchmarks for feature descriptors such as histogram of oriented gradients (HOG) [4] or even state of the art learned descriptors such as SuperPoint [5] would make for a more thorough evaluation. Furthermore, as mentioned in Section III-C, we were not able to perform cross-validation due to time constraints, therefore a benchmark that included this step would provide more accurate results.

References

[1] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly su-

pervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[3] Frank M Candocia. Simultaneous homographic and comparametric alignment of multiple exposure-adjusted pictures of the same scene. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 12(12):1485–94, 2003.

[4] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[5] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018.

[6] Zhuoyue Gao, Lin Chai, and Lizuo Jin. Location recognition based on image local feature matching. In *MIPPR 2019: Automatic Target Recognition and Navigation*, volume 11429, page 114290Q. International Society for Optics and Photonics, 2020.

[7] Philipp Gräbel, Ina Laube, Martina Crysandt, Reinhild Herwartz, Melanie Baumann, Barbara M. Klinkhammer, Peter Boor, Tim H. Brümmendorf, and Dorit Merhof. Rotation invariance for unsupervised cell representation learning. *Bildverarbeitung für die Medizin 2021 Informatik aktuell*, page 42–47, 2021.

[8] Mohammad R Homaeinezhad, SA Atyabi, E Tavakkoli, Hamid Najjaran Toosi, Ali Ghaffari, and Reza Ebrahimpour. Ecg arrhythmia recognition via a neuro-svm–knn hybrid classifier with virtual qrs image-based geometrical features. *Expert Systems with Applications*, 39(2):2047–2058, 2012.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[10] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, nov 2004.

[11] Juncheng Ma, Keming Du, Feixiang Zheng, Lingxian Zhang, Zhihong Gong, and Zhongfu Sun. A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. *Computers and electronics in agriculture*, 154:18–24, 2018.

[12] Colin McManus, Winston Churchill, Will Maddern, Alexander D Stewart, and Paul Newman. Shady dealings: Robust, long-term visual localisation using illumination invariance. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 901–906. IEEE, 2014.

[13] Sven Middelberg, Torsten Sattler, Ole Untzelmann, and Leif Kobbelt. Scalable 6-dof localization on mobile devices. In *European conference on computer vision*, pages 268–283. Springer, 2014.

[14] Jean-Michel Morel and Guoshen Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.

[15] Marius Muja and David Lowe. Flann-fast library for approximate nearest neighbors user manual. *Computer Science Department, University of British Columbia, Vancouver, BC, Canada*, 2009.

[16] A. Sadr, N. Mohsenifar, and R.S. Okhovat. Comparison of mlp and rbf neural networks for prediction of ecg signals. *IJCSNS*, 11(11):124, 2011.

[17] A. Torii, J. Sivic, M. Okutomi, and T. Pajdla. Visual place recognition with repetitive structures. 2015.

[18] Jian Wu, Zhiming Cui, Victor S Sheng, Pengpeng Zhao, Dongliang Su, and Shengrong Gong. A comparative study of sift and its variants. *Measurement science review*, 13(3):122, 2013.