Compact Environment-Invariant Codes for Robust Visual Place Recognition

Unnat Jain
Dept. of Computer Science
University of Illinois Urbana-Champaign
Urbana, USA
uj2@illinois.edu

Vinay P. Namboodiri
Dept. of Computer Science & Engg.
Indian Institute of Technology
Kanpur, India
vinaypn@iitk.ac.in

Gaurav Pandey
Dept. of Electrical Engg.
Indian Institute of Technology
Kanpur, India
gpandey@iitk.ac.in

Abstract—Robust visual place recognition (VPR) requires scene representations that are invariant to various environmental challenges such as seasonal changes and variations due to ambient lighting conditions during day and night. Moreover, a practical VPR system necessitate compact representations of environmental features. To satisfy these requirements, in this paper we suggest a modification to the existing pipeline of VPR systems to incorporate supervised hashing. The modified system learns (in a supervised setting) compact binary codes from image feature descriptors. These binary codes imbibe robustness to the visual variations exposed to it during the training phase, thereby, making the system adaptive to severe environmental changes. Also, incorporating supervised hashing makes VPR computationally more efficient and easy to implement on simple hardware. This is because binary embeddings can be learnt over simple-to-compute features and the distance computation is also in the low dimensional hamming space of binary codes. We have performed experiments on several challenging data sets covering seasonal, illumination and viewpoint variations. We also compare two widely used supervised hashing methods of CCAITQ [1] and MLH [2] and show that this new pipeline out-performs or closely matches the state-of-the-art deep learning VPR methods that are based on high-dimensional features extracted from pre-trained deep convolutional neural networks.

Keywords-visual place recognition; similarity learning; hashing; convolutional neural network; dynamic time warping

I. INTRODUCTION

Robots are often required to operate in environments for a long period of time ranging from days, months to even years. For autonomous operation of robots in such cases, the robot has to recognize places that it has visited before. This ability of the robot to recognize places has a wide range of applications in its autonomous navigation capabilities that include global localization and loop-closure detection. The task of VPR for a long-term autonomous visual navigation system becomes extremely challenging because over a long period of time the appearance of a place can drastically change. Traditional visual place recognition approaches like [3], [4] focused on situations where robot has to recognize places that it has recently visited, where the difference between query and database images was mainly due to different view-point of sensors. However, for a long-term autonomous visual navigation system the VPR method should be **robust** to seasonal, illumination

(a) Nordland spring-summer & summer-winter data sets: Mild to severe appearance change & no viewpoint change.

(b) Alderley Day-Night data set: Severe appearance & mild viewpoint change.

(c) St. Lucia data set: Mild appearance & severe viewpoint change.

Figure 1: Appearance and viewpoint variations in data sets on which we test our visual place recognition pipeline.

and viewpoint variations (Fig. 1). It is also desired that the VPR system could imbibe robustness to an unfamiliar variation from some learning examples. It should also be **real-time** & **storage efficient** for it to have utility for robotic applications. Additionally, it is advantageous if a visual place recognition system is implementable on a simple hardware configuration so that it can have much wider application.

Several methods that are robust to certain visual variations have been proposed in the past. Sünderhauf et al. [5] used concatenated BRIEF-gist descriptor to incorporate some robustness to viewpoint variations. Milford et al. [4] proposed to use the video sequence instead of independent images, thereby utilising the continuity constraint of consecutive images in-order to remove outliers. Lowry et al. [6] employed linear regression techniques to predict the temporal appearance of a place based on the time of the day. Neubert et al. [7] uses a vocabulary of superpixels to predict the

change in appearance of the scene.

Many visual place recognition methods based on deep convolutional neural network (CNN)s which are pre-trained on the task of either object recognition or scene classification, have recently been proposed [8]–[11]. The deep CNN based methods have shown to outperform previous state-of-the-art visual place recognition techniques. However, CNN feature based methods like [8] with no dimensionality reduction are slow because of computations on high dimensional feature vectors. Several dimensionality reduction techniques have been proposed to speed-up the algorithm. Milford et al. [12] uses principal component analysis (PCA) as an extra step to reduce the dimensionality of feature vectors. Sünderhauf et al. [11] utilized Gaussian random projections to obtain shorter features than the raw `conv3` CNN features. Sünderhauf et al. [10] borrows from research in unsupervised hashing methods to obtain binary codes of 8192 bits to describe the images. It is important to note that all the above dimensionality reduction methods are **unsupervised** and have lower accuracies than a visual place recognition method which uses the 'non-reduced' raw features.

Unlike the existing methods, the proposed VPR pipeline uses **supervised** dimensionality reduction to reduce high dimensional real features to compact and more semantic binary codes. All other VPR methods [10]–[12] demonstrate lower performance due to the dimensionality reduction step. On the contrary, a supervised VPR system based on the learning of compact binary codes from image features significantly improves the image retrieval performance when compared to VPR methods based on the corresponding raw features. We improve on accuracy while still using a binary code representation. This representation keeps our VPR pipeline real-time and space efficient. Also, the proposed method is **feature agnostic**. Therefore, our VPR pipeline is capable of processing both *simple* gist [13] features (whose computation requires only simple convolutions with Gabor filters) as well as *complex* deep learning based CNN features. With the boost of accuracy, our VPR method is capable of bootstrapping the accuracies of simple-to-compute features like gist to performance comparable to (usually better than) existing VPR methods based on pre-trained CNN features.

The rest of the paper is organized in the following sections:
Section II describes how gist & CNN features have been utilized for VPR research. It also motivates the need of a system capable of adaptively *learning* robustness rather than relying only on the *pre-learnt* robustness of popular image descriptors. Section III describes CCAITQ [1], the supervised hashing method that we utilize in our VPR pipeline. Section IV gives details of the experimentation - data sets, training & testing set division, assigning similarity labels, running hashing methods and testing. We explain important inferences of our experiments in section IV-C and conclude the paper with section V.

Feature	Dimension	Advantage	Disadvantage
Gist	512 or 2048	Compact global representation	Low performance in severe changes
`fc6`	4096	Robust to viewpoint variations	Susceptible to appearance variations
`conv3`	43264 (VGG-f)	Robust to severe appearance variations	Very high dimensional and susceptible to viewpoint variations

Table I: Details about popular image features for VPR

II. BACKGROUND

A. Gist based Visual Place Recognition methods

Initial feature based VPR methods (like FAB-MAP [3]) built a visual vocabulary from local SIFT or SURF descriptors and then used a bag-of-words (BoW) image descriptor to find the best matching image frame corresponding to a query frame. Later, success in scene classification based on gist features [14] led to gist features being applied to the place recognition tasks. Gist features were adapted to panoramic views for VPR [15]. Visual Loop Closing methods which earlier utilized SIFT or SURF BoW, demonstrated much superior performance after adapting gist descriptors [5], [16], [17].

B. Deep CNN based VPR

AlexNet [18] made deep CNNs popular in Computer Vision research in 2012. Studies in [19], [20] demonstrated that features extracted from deep CNN (which are pre-trained on object recognition task) can be used for generic visual recognition tasks. CNN features performed better than features which were handcrafted specific for the tasks like domain adaptation, fine grained recognition and scene recognition. Thereafter, research in VPR has almost suddenly turned to explore the power of CNN based features. Work of Sünderhauf et al. [10], [11] has extensively compared features extracted from different CNN layers, on challenging VPR data sets. Instead of extracting global CNN image descriptors like [10], [11] extracts pooled local `conv3` CNN features of fifty landmark regions and then use similarity matching of these local features to obtain the image which is the best match to a query image. Inferring from the detailed empirical study of [10], we focus our experiments on features extracted from two layers - lower level convolutional layer (third layer - `conv3`) and higher level fully connected layer (sixth layer - `fc6`).

Drawbacks of pre-trained CNN based VPR approaches:

- *Choice of layer*: [10] empirically validates the appearance invariance of `conv3` and and viewpoint invariance of `fc6` layers. However, for a new environment displaying a unknown weighted mixture of multiple visual variations, one has no insight of which layer features to utilize.

- *Dimensionality*: We validated [10]'s claim that lower depth features from `conv3` layer are invariant to severe appearance and mild viewpoint variations. Despite their good pre-learnt robustness, raw `conv3` features are not useful for VPR due to their huge dimensionality, 64896 for [18] and 43264 for [21], which significantly increases the computation time.
- *Storage size:* Each dimension for raw real-valued features is 256 bits leading to 1.3Mb size of one `conv3` feature vector. Table II gives details about the storage capacity required for storing different features.
- *Hardware requirement*: Deep CNNs have complex architecture and require much advanced hardware devices for training and feature extraction. The model of a CNN has over millions of parameters and merely loading a CNN model requires huge amount of RAM space, which is often impractical for use in many robotic vision applications.

Feature	Dimension	Size (bits)	Size of entire Alderley data (MB)
Gist512 (binary)	512	512	2 Mb
Gist2048 (binary)	2048	2048	8 Mb
`fc6` (binary)	4096	2048	8 Mb
Gist512 (raw)	512	131072	493 Mb
Gist2048 (raw)	2048	524288	1973 Mb
`fc6` (raw)	4096	1048576	3946 Mb
`conv3` (raw)	43264	11075584	41678 Mb

Table II: Comparing storage size of binary codes and their corresponding raw/precursor features. To illustrate the impact on VPR, we also calculate how much space is required to store a data set (here, Alderley Night/Day data set)

C. Learning invariance vs. pre-learnt invariance

Pre-trained CNN features acquire invariance to the variations that the CNN has been exposed to while training. Since the popular pre-trained CNNs are trained for object recognition, the variations in images of the same object might not cover environmental variations prevalent in places. For example, glaring lights in the night traversal of Alderley data set, change of texture from grassy (in summer) to snowy (in winter) are variations which are rarely seen in object recognition data sets. Such variations are not as generic as simple illumination, rotation and scale variations which pre-trained CNNs have already been exposed to, while training on data sets like ImageNet [22]. Thus, any feature based VPR system needs some bootstrapping to learn robustness against *unfamiliar* variations. We incorporate this bootstrapping by supervised hashing methods discussed later.

Is obtaining supervision easy: Obtaining GPS tagged images is cheap and easy, and can be done by any vehicle which travels with a camera and GPS. Multiple traversals of the same tracks helps create cluster of images of the same places at different times of the day and different seasons of the year. This forms the ground truth for a supervised VPR system. There are many publicly available data sets which have multiple images of the same place (under different conditions). These include Nordland [23], Alderley [4], St. Lucia [24], KITTI [25], Pittsburgh 250k [26] and Tokyo 24/7 [27] data sets. Thus, supervision is simple to incorporate in VPR and is especially important when a VPR system is put in a new environment.

III. HASHING METHODS

In efficient retrieval systems, high dimensional vectors are often compressed using similarity-preserving hash functions which map long features to compact binary codes. Hashing methods can be used to map *similar* images to nearby binary codes (with small hamming distance between them) and map *dissimilar* images to far-away binary codes (with large hamming distance between them). Two main advantages of using hashing methods for visual place recognition are:

- *Storage space*: Table II compares the storage size of real-valued features with that of binary code representations. Even though [10] applies hashing to obtain compact 8192 bit binary codes, it can preserve 95% of the accuracy obtained by using raw real-valued features. Our supervised approach obtains even shorter binary codes and simultaneously shows better performance than that obtained by using raw real-valued features.
- *Speed*: The hamming distance computation required to compare similarity between compact binary codes is much faster than euclidean or cosine distance computations required to compare very high dimensional feature vectors. Hamming distances can be efficiently calculated using low level (hardware) bit-wise operations.

Hashing methods can be categorized into - unsupervised and supervised, based on the notion of similarity that these methods preserve. Unsupervised hashing methods preserve **distance based similarity** whereas supervised methods preserve **label based similarity**.

A. Unsupervised hashing

Distance based similarity means that images are considered *similar* if their feature vectors have small distance (euclidean, cosine etc.) between them. Unsupervised hashing methods output binary codes which aim to preserve the original distances in real-valued feature space. They leverage no other information except the image features, therefore, their performance is lower than original real-valued features. Despite slightly lower performance, they help overcome two of the issues with CNN features - dimensionality and storage size. Locality sensitive hashing (LSH) [28], [29] is a widely used unsupervised hashing method which preserves cosine distance between original data points. Random hyperplane adaptation of LSH in [30] is most commonly used. We too

Figure 2: Comparing hashing methods - CCAITQ [1], MLH [2] and LSH [28].

use it in this work to replicate results of previous LSH-based VPR approaches [10].

B. Supervised hashing

Supervised hashing leverage supervision of labels. Using the knowledge from labels, their performance is higher than original real-valued features. For the task of long-term visual place recognition, feature vectors of spatially proximal places might not be nearby in the feature space. For example, feature vector corresponding to winter image of a place X might be closer to the feature vector corresponding to summer image of place Y (rather than that of place X). In such cases externally provided labels can support supervised hashing, which defines a better notion of similarity. We use the well known technique of Canonical Correlation Analysis combined with Iterative Quantization (CCAITQ) [1] to perform supervised hashing for robust VPR. Another supervised hashing technique that was developed at the same time is Minimal Loss Hashing (MLH) [2]. In our comparison in section IV-A we observe that for the task of VPR, CCAITQ performs better than MLH and is also more computationally efficient. Hence, we use CCAITQ in the proposed VPR method.

CCAITQ [1] is a linear hashing method which combines Canonical Correlation Analysis (CCA) [31] and Iterative Quantization (ITQ) techniques to eventually obtain binary codes. CCA is the supervised analog of the PCA, which learns a matrix $W \in \mathbb{R}^{d \times k}$ where d is the dimensionality of original features and k is the desired dimensionality of output binary codes. This matrix helps in finding the directions in which the feature vectors and the label vectors have maximum correlation. The input feature matrix is $X = [x_1; x_2; ...x_n] \in \mathbb{R}^{n \times d}$ where n is the number of data points and x_i's are rows of features. The aim of CCA is to

learn the matrix W such that $V = XW$ transforms feature vectors (rows of X) to a more semantic real space. After obtaining transformed representations $v_i \in \mathbb{R}^k$ (ith row of $V = XW$) from the CCA step, we ought to quantize this representation to obtain binary codes. This can be directly done by using indicator function on each of the dimensions: $f(v) = 1_{\mathbb{R}+}(v)$. However, a better binary embedding is obtained by rotating the features obtained after the CCA step in such a manner that the quantization loss is minimized. Gong et al. [1] describe a Procustean approach to solve this quantization problem by minimizing the following loss function:

$$\arg\min_R \|f(VR) - VR\| \text{ s.t. } R'R = RR' = I$$

By solving this minimization problem we obtain the orthogonal matrix R which minimizes the quantization loss.

IV. EXPERIMENTATION

We perform experiments on four data sets chosen for the variety of variation in them (details of data sets is given in table III). Each data set has two traversals of the same route - database and query traversal, with appropriate ground truth match. In Nordland data set the frames of the database and query traversal are synchronized i.e. i^{th} winter frame matches the ith summer frame, whereas the ground truth in Alderley and St. Lucia data sets is provided externally using a frame matching matrix (**fm**). More generally, **fm**$(i) = j$ stores that the ith training frame in query traversal corresponds to the same locations as j^{th} training frame in the database traversal. We use some portion of both the traversals for training CCAITQ to learn the transformation matrices W and R as described in section III-B.

For extracting gist feature descriptors we use the original implementation[1] of gist, made available by Oliva & Torralba. For extracting deep CNN features we use the `matconvnet` toolbox made available by VGG group.

We consider each training image of both traversals as a label. Label vector (for CCAITQ) of a particular training image i has $1's$ for $i \pm margin$ frames of the same traversal and also **fm**$(i) \pm margin$ frames of the other traversal. Thus, the obtained binary code of a given frame learns similarity to neighbouring frames (**variation in viewpoint**) and also learn similarity to corresponding frames in the other traversal (**variation in appearance**). We employ the original implementation of CCA-ITQ[2].

A. Comparing supervised hashing methods

The works of MLH [2] and CCAITQ [1] have performed well in comparison to other methods of supervised hashing such as spectral hashing and binary reconstructive embedding. We therefore restrict ourselves in this paper to

---

[1] http://people.csail.mit.edu/torralba/code/spatialenvelope
[2] http://slazebni.cs.illinois.edu/research/smallcode.zip

Figure 3: Comparing effect of code length (in bits) on recall (in %). Our model can convert raw real valued features (green) to binary codes (red) which boosts the performance. Binary codes obtained by popular unsupervised hashing method (LSH) (blue) have lower performance than the corresponding raw real-valued features.

these two popularly used supervised hashing methods. Fig. 2 shows a comparison between the two hashing methods. We also compare it to the baseline of unsupervised hashing method LSH (used in recent VPR methods [10], [11]). We find MLH is intractable for learning more than 256 bit binary codes, requiring more than a day of training time on our data sets on a 16Gb i7 computer. On the contrary, CCAITQ requires only few minutes to train over 4096D `fc6` features for data sets of considerable size. Moreover, it always outperforms MLH's performance. Hence, we conduct experiments and report results using CCAITQ in our supervised VPR pipeline.

B. Bit-wise study of supervised VPR

We compare the recall performance of binary codes for different code lengths varying from 32 to 2048 bits for the most challenging Nordland winter-summer data set. While testing our VPR pipeline, we extract binary codes for each test set image of query and database traversal (using learnt W and R matrices). For every query image code we find the closest database image code (in hamming space) and count it a true positive if it is within the *margin* around ground truth match. CCAITQ algorithm outputs binary codes with lengths lesser than the dimensionality of the raw features. Hence, 3b does not go beyond 512 bits (learnt from 512D gist features). The black benchmark is calculated using `conv3` raw features, which is the best performing feature as suggested in [10]. We compare `conv3` features with our VPR pipeline's results, obtained from much smaller features - simple gist and `fc6` CNN features. We observed

that the learnt binary codes (red) perform significantly better than the corresponding raw features (green). Fig. 3c shows the VPR system with the proposed modification helps outperforms `conv3` raw features [10] by using only 2048 bit binary codes. Hence we are able to **bootstrap simple-to-compute & low dimensional gist features to match the performance of pre-trained CNN based VPR systems** and *simultaneously* **reduce dimensionality & storage space**.

C. Comparing precision-recall performance

Precision-Recall (PR) curves are used to compare image retrieval techniques. VPR is similar to image retrieval and research in VPR [4], [8], [10]–[12] plot PR curves by altering a parameter. Achieving high precision with high recall is desired. Hence, farther a PR plot is from the axes, better the performance. The procedure for plotting PR curve for a VPR system is described ahead. Each query image has $n = 2 * margin + 1$ ground truth positives. We retrieve top k matches for a query image out of which m ($\leq k$) are true positives. We use $Precision = m/k$ and $Recall = m/n$, which is averaged over all query images to make the PR plot. The value of k is varied from 1 to the total number of test images to obtain multiple points on the curve. We extract different top matches for each VPR method and compare the performance in fig. 4. We observe that the recognition performance of the binary codes (coloured solid lines) learnt over all three features - 512D gist, 2048D gist and 4096D `fc6`, improve over their corresponding raw feature versions (coloured dashed lines). Moreover, **2048 bit binary codes learnt over 2048D gist features shows better**

(a) Nordland summer-spring data set (A: Mild, V: No)

(b) Nordland summer-winter data set (A: Severe, V: None)

(c) Alderley Night-Day data set (A: Severe, V: Mild)

(d) St. Lucia 01-10 data set (A: Mild, V: Severe)

Figure 4: Precision-Recall curves of data sets exhibiting difference mixture of appearance (A) and viewpoint (V) variations

(or marginally less) performance than the benchmarking raw `conv3` features (solid **black**).

D. Leveraging continuity information using contrast enhanced Dynamic Time Warping

While most VPR methods are similar to an image retrieval for each query image, some methods like [4], [32] leverage the fact that we always have a sequence of images as opposed to a single query image. The common assumption being that the two traversals have no negative velocities (reverse/backward travel). Authors of [4] suggest using Dynamic Time Warping (DTW) [33] in future work to tackle variations in velocities. We apply DTW over our binary codes (red dashed) to show an improvement in results. The results are compiled in fig. 5 where the individual image-level VPR method (black circles) explained till now gives slightly incongruous (non continuous) retrievals. Cost of a path has two factors - number of elements (length of path) and cost of each element (divergence from ground truth). We contrast enhanced the DTW distance matrix, thus, giving more weight to the factor of divergence from ground truth. Hence, we found such a cost function (blue dots) to give better performance on non-diagonal trajectories (zoomed in plots in fig. 5).

E. Implementation details

Additional details of implementation needing explanation are as follows:

- *CCAITQ is applied over pre-ReLU layer features instead of post-ReLU layer features:* Rectified linear unit (ReLU) layer: $f(x) = max(x,0)$ add the non-linearity quotient to deep CNNs. Since [10], [11] directly utilize pre-trained CNN features without doing any learning, they may choose to use features after ReLU layers (post-ReLU) which are much sparser than pre-ReLU. Studies in [20] have shown that SVM learning over CNN features works better when we use pre-ReLU features. Since our task of supervised hashing method is equivalent to learning multiple (equal to the length of binary codes) hyperplanes, we too use pre-ReLU CNN features in our experiments. For features requiring no learning, we report results using post-ReLU (same as [10], [11]).

- *Use of VGG-f:* VGG-f [21] has five convolutional, three fully connected layers and takes a 224×224 image (all same as AlexNet [18]). While the old AlexNet architecture is configured rigidly to accommodate training distributed over two GPUs, VGG-f gets rid of such dependency. With more minor variations, it is shown

Data set	Database traversal	Query traversal	Variation present	Train frames	Test frames	Sample rate (fps)	Margin (± frames)
Nordland	Summer	Spring	A: Mild, V: None	$1 - 10k$ (both traversals)	$11k - 16k$ (both traversals)	2	5
Nordland	Summer	Winter	A: Severe, V: None	$1 - 10k$ (both traversals)	$11k - 16k$ (both traversals)	2	5
Alderley	Night	Day	A: Severe, V: Mild	$1 - 9k$ (day) $1 - 11301$ (night)	$10k - 14607$ (day) $12051 - 16960$ (night)	25	10
St. Lucia	1st traversal	10th traversal	A: Mild, V: Severe	$1 - 10k$ (10) $1 - 11145$ (01)	$11001 - 16k$ (10) $12145 - 17512$ (01)	15	10

Table III: Experimental details and variations present in data sets used for evaluation

Figure 5: Impact of leveraging sequencing or continuity information for Alderley data set.

to give better performance than AlexNet on recognition tasks. Hence, we report results using this variant and achieve better performance for both raw features as well as binary codes.

- *Supervised hashing not applied on* `conv3` *layer:* Whether we choose VGG-f or AlexNet variant of the deep CNN, the dimensionality of `conv3` layer is too high to tractably run CCAITQ (MLH is even slower). Both aims of our VPR pipeline - Speed and storage efficiency (Sec. III), are incongruent to the use of `conv3` layer features. Supervised hashed codes that we obtain from gist or `fc6` CNN features are compared to the raw `conv3` layer (black plots in fig. 3 and 4).

V. CONCLUSION

To the best of our knowledge, our work is the first to introduce the successful learning methods of supervised hashing to the Visual Place Recognition research. The application is a perfect fit as there is a need for both *learning* and *compact representation* for any VPR to be robust and real-time, respectively. There is the alternative to learn these compact embeddings by training a CNN for this task. Training of CNNs require a long time and also require much complex hardware capable systems. We conclude that for a widespread application of VPR, supervised hashing is ideal as it is both quick to train and outputs compact binary representation of images.

REFERENCES

[1] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916–2929, Dec 2013.

[2] M. Norouzi and D. J. Fleet, "Minimal loss hashing for compact binary codes," in *ICML*, 2011, pp. 353–360.

[3] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[4] M. J. Milford and G. F. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May 2012, pp. 1643–1649.

[5] N. Sünderhauf and P. Protzel, "Brief-gist-closing the loop by simple means," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 1234–1241.

[6] S. M. Lowry, M. J. Milford, and G. F. Wyeth, "Transforming morning to afternoon using linear regression techniques," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3950–3955.

[7] P. Neubert, N. Sunderhauf, and P. Protzel, "Appearance change prediction for long-term navigation across seasons," in *Mobile Robots (ECMR), 2013 European Conference on*. IEEE, 2013, pp. 198–203.

[8] Z. Chen, O. Lam, A. Jacobson, and M. Milford, "Convolutional neural network-based place recognition," *arXiv preprint arXiv:1411.1509*, 2014.

[9] P. Neubert and P. Protzel, "Local region detector + cnn based landmarks for practical place recognition in changing environments," in *Mobile Robots (ECMR), 2015 European Conference on*, Sept 2015, pp. 1–6.

[10] N. Sünderhauf, F. Dayoub, S. Shirazi, B. Upcroft, and M. Milford, "On the performance of convnet features for place recognition," *CoRR*, vol. abs/1501.04158, 2015.

[11] N. Suenderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford, "Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free," in *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.

[12] Z. Chen, S. Lowry, A. Jacobson, Z. Ge, and M. Milford, "Distance metric learning for feature-agnostic place recognition," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, Sept 2015, pp. 2556–2563.

[13] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[14] C. Siagian and L. Itti, "Rapid biologically-inspired scene classification using features shared with visual attention," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 2, pp. 300–312, 2007.

[15] A. C. Murillo and J. Kosecka, "Experiments in place recognition using gist panoramas," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 2196–2203.

[16] Y. Liu and H. Zhang, "Visual loop closure detection with a compact image descriptor," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 1051–1056.

[17] G. Singh and J. Kosecka, "Visual loop closing using gist descriptors in manhattan world," in *ICRA Omnidirectional Vision Workshop*. Citeseer, 2010.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[19] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," *arXiv preprint arXiv:1310.1531*, 2013.

[20] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: An astounding baseline for recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2014.

[21] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *British Machine Vision Conference*, 2014.

[22] 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)*, vol. 115, no. 3, pp. 211–252, 2015.

[23] N. B. Corporation, "Nordlandsbanen: minute by minute, season by season," https://nrkbeta.no/2013/01/15/nordlandsbanen-minute-by-minute-season-by-season/.

[24] M. Warren, D. McKinnon, H. He, and B. Upcroft, "Unaided stereo vision based pose estimation," in *Australasian Conference on Robotics and Automation*, G. Wyeth and B. Upcroft, Eds. Brisbane: Australian Robotics and Automation Association, 2010.

[25] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[26] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi, "Visual place recognition with repetitive structures," in *CVPR*, 2013.

[27] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," in *CVPR*, 2015.

[28] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. ACM, 1998, pp. 604–613.

[29] A. Gionis, P. Indyk, R. Motwani *et al.*, "Similarity search in high dimensions via hashing," in *VLDB*, vol. 99, no. 6, 1999, pp. 518–529.

[30] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*. ACM, 2002, pp. 380–388.

[31] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, no. 3/4, pp. 321–377, 1936.

[32] E. Pepperell, P. I. Corke, and M. J. Milford, "All-environment visual place recognition with smart," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1612–1618.

[33] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE transactions on acoustics, speech, and signal processing*, vol. 26, no. 1, pp. 43–49, 1978.