Pre-print of article that will appear in Proceedings of the Australasian Conference on Robotics and Automation 2018.

Please cite this paper as:

Stephen Hausler, Adam Jacobson, and Michael Milford. Feature Map Filtering: Improving Visual Place Recognition with Convolutional Calibration. Proceedings of Australasian Conference on Robotics and Automation, 2018.

bibtex:

@inproceedings{hausler2018FeatFilt,
    author = {Hausler, Stephen and Jacobson, Adam and Milford, Michael},
    title = {Feature Map Filtering: Improving Visual Place Recognition with Convolutional Calibration},
    booktitle = {Proceedings of Australasian Conference on Robotics and Automation (ACRA)},
    year = {2018},
}
Feature Map Filtering: Improving Visual Place Recognition with Convolutional Calibration

Stephen Hausler, Adam Jacobson and Michael Milford
Queensland University of Technology, Australia
stephen.hausler@hdr.qut.edu.au

Abstract

Convolutional Neural Networks (CNNs) have recently been shown to excel at performing visual place recognition under changing appearance and viewpoint. Previously, place recognition has been improved by intelligently selecting relevant spatial keypoints within a convolutional layer and also by selecting the optimal layer to use. Rather than extracting features out of a particular layer, or a particular set of spatial keypoints within a layer, we propose the extraction of features using a subset of the channel dimensionality within a layer. Each feature map learns to encode a different set of weights that activate for different visual features within the set of training images. We propose a method of calibrating a CNN-based visual place recognition system, which selects the subset of feature maps that best encodes the visual features that are consistent between two different appearances of the same location. Using just 50 calibration images, all collected at the beginning of the current environment, we demonstrate a significant and consistent recognition improvement across multiple layers for two different neural networks. We evaluate our proposal on three datasets with different types of appearance changes - afternoon to morning, winter to summer and night to day. Additionally, the dimensionality reduction approach improves the computational processing speed of the recognition system.

Figure 1: Top: Our feature map filtering approach removes specific feature maps that vary their activation when the scenes’ appearance changes over time. Bottom: We display the maximum activation heat maps for the stack of feature maps that survive after filtering (the top row of heat maps), and for the removed feature maps (bottom row). Notice how the removed maps are activating in response to the shadows on the road (marked by a red ellipse).

*SH is supported by a Research Training Program Stipend and ARC Future Fellowship FT140101229. AJ is supported by an Advance Queensland Innovation Partnership, Caterpillar and Mining3. MM is with the Australian Centre for Robotic Vision and was partially supported by an ARC Future Fellowship FT140101229.
1 Introduction

Visual Place Recognition, the ability to localize using just a visual sensor, is challenging due to the significant appearance change that visual scenes experience on a regular basis, including day to night, summer to winter and morning to afternoon. Both hand-crafted features, such as SURF [Bay et al., 2008] and HOG [Dalal and Triggs, 2005], and deep learnt networks have been used to attempt to solve the VPR challenge [Naseer et al., 2018; Cummins and Newman, 2008; Sunderhauf et al., 2015a; Arandjelović et al., 2018]. Both viewpoint and appearance robustness has been demonstrated when convolutional neural networks (CNNs) are used for visual place recognition [Sunderhauf et al., 2015a]. This is especially the case when a CNN is trained for recognizing a specific environment [Arandjelović et al., 2018; Kim et al., 2017; Lopez-Antequera et al., 2017]. However, this performance has the disadvantage of requiring training for all the environmental conditions that the robot is expected to experience, whereas for practical autonomy, the robot should be able to automatically, and swiftly, adjust its neural parameters to suit the current conditions.

We propose a novel solution to achieve this, by calibrating a convolutional neural network for the current environment. In state-of-the-art approaches, a neural network is re-trained for the specific environment by selecting a set of images from the new environment and re-training the model using these images [Arandjelović et al., 2018; Kim et al., 2017; Lopez-Antequera et al., 2017]. However, this requires a significant time and processing cost, so much so that typical robot platforms do not have the capability to re-train the neural model online and in real-time. We propose a method that enables a fast, computationally cheap process of filtering the collection of feature maps within a layer of a deep convolutional neural network (see Fig. 1). When a network is trained on a diverse set of images, each feature map encodes a different type of abstraction from this collection of images. For example, one map in a late convolutional layer might learn to ‘fire’ upon regions of an image containing a building. We propose a calibration procedure which removes the feature maps that do not suit the recognition between the current environment and the learnt environment. This is achieved by minimizing the L2-distance between two identical locations that appear significantly different due to a change in the environment, while maximizing the distance between two different locations that look visually similar due to having the same environmental conditions.

We demonstrate the versatility of our approach by experimenting with two different CNN architectures, HybridNet [Chen et al., 2017a] and AlexNet trained on ImageNet [Krizhevsky et al., 2012], across three different datasets which demonstrate different types of appearance variations.

The paper proceeds as follows. In Section 2, we review prior uses of convolutional neural networks for the visual place recognition task and previous methods of neural network simplification. Section 3 presents our approach, explaining our calibration procedure and computational methods in detail. Section 4 details the setup of our experimental datasets and Section 5 evaluates the performance of feature map filtering, compared to not filtering. Section 6 provides intuition as to why feature map filtering works and Section 7 summarizes our contributions and provides suggestions for future work.

2 Related Work

In early experiments using convolutional neural networks for place recognition, a feature vector is produced from a particular layer of the network, using all the information that is encoded in the activations of that layer [Sunderhauf et al., 2015a]. However, such a whole-image approach is sensitive to viewpoint variations. This was addressed by developing a landmark extraction algorithm and computing the neural responses to each landmark region in a scene [Sunderhauf et al., 2015b]. Intelligently selecting the useful information within an image is a valuable method of improving the localization performance. Rather than finding regions, LoST [Garg et al., 2018b] creates a feature vector by extracting semantically meaningful keypoints within the feature map spatial region. [Chen et al., 2017b] finds keypoints by observing the activations out of a late convolutional layer, while [Chen et al., 2018] trains a soft attention mask to select salient regions within an image to improve the selection of features used to formulate the feature vector. These keypoint feature vectors consist of the activations across all the feature maps within that layer at the spatial location of the keypoint, even if some of the feature maps are encoding visual information that is counterproductive to localizing in the current environment.

Several experiments compared the performance across different layers [Sunderhauf et al., 2015a; Chen et al., 2017a], while a number of experiments use multiple layers simultaneously [Zhang et al., 2018; Chen et al., 2018], to improve the visual recognition performance beyond the performance of a single layer. Different layers have been found to encode different types of visual features, such as color and texture in early layers, and objects and scenes in later layers [Zhou et al., 2018].

The literature discussed in the previous paragraphs optimizes either the choice of layer to use, or the choice of spatial locations across the feature map stack. The third dimension to optimize is the choice of feature maps themselves within the stack of feature maps that comprise a layer. [Guo and Potkonjak, 2017] proposes that a CNN can be simplified by pruning the selection of fea-
ture maps, which attains comparable performance while improving the computational speed of a forward-pass through the network. [Zou et al., 2018] suggests an improvement by using linear discriminant analysis to calculate the discriminability score for each feature map. They are able to remove a greater number of feature maps without causing a major reduction in accuracy. [Li et al., 2018] re-weights feature maps using a feedback process to improve the classification performance. However, they only re-weight feature maps and don’t completely remove any feature maps. The concept of improving visual place recognition by discriminatively selecting a subset of the feature maps within a convolutional layer is a gap in the literature.

Recent literature on network dissection has provided evidence that individual feature maps encode specific visual features that are relatable to the classifier outputs [Zhou et al., 2018]. In their work, the hidden convolutional layers are probed by testing an individual feature map on a pixel-wise semantic segmentation task. They discover that individual feature maps activate for different objects, scenes, textures and colors. This research underpins the motivation for this work - for example, if a particular feature map activates to man-made lighting, this feature map will confuse the localization between night and day and is better removed from the feature vector.

3 Proposed Approach

We propose a novel method of calibrating a convolutional neural network for the current environment. Our calibration procedure removes the feature maps that do not suit the recognition between the current environment, and the learnt environment. This is achieved by minimizing the L2-distance between two identical locations that appear significantly different due to a change in the environment, while maximizing the distance between two different locations that look visually similar due to having the same environmental conditions (see Fig. 2). This is termed triplet loss in literature and like previous work, we also use the L2-distance as our calibration optimization metric [Schroff et al., 2015; Park et al., 2018; Arandjelovi et al., 2018].

3.1 Calibration Procedure

For each calibration scene, we extract deep-learnt features for the currently viewed scene, the corresponding reference image and a randomly selected image elsewhere in the database of reference images. We use a total of 50 calibration images, all extracted from the beginning of the query dataset - this is to mimic the real world situation where the calibration is performed prior to the robot beginning the navigation of its environment. This calibration can be achieved using pre-defined maneuvers, such as the methods described in [Jacobson et al., 2015]. These calibration triplets are used to perform feature map filtering, as explained in the following sections.

3.2 Extracting Deep Learnt Features

In a convolutional neural network, a convolutional layer tensor consists of dimension $W \times H \times C$, where $W$ and $H$ are the width and height of the data matrix and $C$ is the number of channels, otherwise termed the number of feature maps. To reduce the dimensionality of this $W \times H \times C$ feature vector, we use maximum pyramid spatial pooling [Chen et al., 2017a], which was chosen as it has both viewpoint robustness and provides a significant dimensionality reduction while keeping the key features in each feature map. In our version of pyramid spatial pooling, we convert each $W \times H$ map into a vector of length 5, consisting of the maximum activation in each map and the maximum activation in each of the four quadrants of each $W \times H$ map.

Out of a stack of feature maps within a convolutional layer, certain feature maps will activate to certain visual features in an image. For example, a feature map in a network might fire on regions of an image containing vehicles. However, in the context of visual place recognition, activations on vehicles has a negative effect, because vehicles are dynamic objects and not temporally static. This applies to other time-varying features, like snow in winter. Our goal is to search through the stack of feature maps to find the worst feature maps. We define the worst feature maps as feature maps that contain activations that vary across a change in appearance when the location does not change. We perform this search on the spatially pooled features in each feature map, for improved viewpoint robustness.

Figure 2: This diagram visually explains the triplet calibration method we employ.
3.3 Filtering Feature Maps

We use a Greedy algorithm [Fegaras, 1998] to determine which subset of the feature map stack suits the current environmental conditions. Combinatorial optimization problems are typically NP-hard, with a variety of techniques employed to produce approximate solutions in related problems such as sensor selection [Joshi and Boyd, 2009]. In our method, using Greedy causes the worst feature map to be filtered at each iteration of the algorithm, until a local maximum is reached. We chose Greedy as it runs in polynomial time and was found to converge to a satisfactory position.

To measure the feature map performance, we select each feature map individually and remove it from the feature vector before calculating the L2 (Euclidean) distance between both the images from the same location and the two images from the reference database. This results in two distance scores, one for the same location at different times of day and one for different locations at the same time of day. The result is a vector of difference scores across a different feature map being removed.

\[
D(q^i_j, r^j_i) = \sqrt{\sum_{k=1}^{M} (q^i_j(k) - r^j_i(k))^2}
\]

(1)

where \(M\) is the dimension of the filtered query feature vector \(q^i_j\).

\[
D(j) = D(r^j_i, n^j_i) - D(q^i_j, r^j_i) \quad \forall j
\]

(2)

where \(r^j_i\) represents the current location filtered reference feature vector and \(n^j_i\) represents the filtered feature vector from a random image somewhere else within the reference image database. \(j\) denotes the index of the currently filtered feature map.

We then find the maximum distance:

\[
\text{maxval} = \max_{1 \leq j \leq N} D(j)
\]

(3)

\[
\text{worstFmap} = \arg\max_{1 \leq j \leq N} D(j)
\]

(4)

where \(N\) is the number of remaining feature maps.

The index of the maximum distance represents the feature map to be removed to achieve the greatest L2 difference between the images from the same location and the images from different locations. With this chosen feature map, we modify the original feature vector and remove this worst performing feature map before repeating the above algorithm for this new, filtered, feature vector.

We iterate in this fashion until a local maximum is reached, that is, the largest L2 difference between the images at same location and the images at different locations (with the images at the same location being closer in L2 space than the images at different locations). In our initial experiments we observed that the gradient towards the local maximum becomes very small prior to reaching the maximum and a significant number of feature maps are filtered out. As an alternate, less aggressive filtering algorithm, we added a gradient minimum cut-off threshold, which we set to 0.1. When removing the worst-performing feature map, if the difference between the previous iteration difference score and the current difference is less than 0.1, we stop the iteration and use the current set of remaining feature maps.

For improved robustness and to prevent outliers, we use multiple calibration images. The choice of filtered feature maps is stored for all images and after the calibration procedure is finished, the number of times a particular feature map is removed is summed across all 50 calibration images. We then find the set of feature maps that were least chosen to be filtered out, and the number of final feature maps is equal to the maximum number of remaining feature maps in the set of calibration images. This heuristic was chosen based on the principle that the choice of remaining feature maps needs to be able to encode all the features within all the calibration images, else minor variations in the current environment will cause key visual features to be missed. The filtering procedure is designed to only remove the feature maps that are irrelevant or damaging to the ability to match between the two appearances of the same location.

3.4 Place Recognition Validation Algorithm

We developed a single-frame place recognition algorithm to evaluate the improvement gained by using feature map filtering. The features extracted from both the query images and the reference database only include the particular feature maps that were chosen by the feature map filter calibration algorithm. Each query image is compared to the reference database using the cosine distance metric to create a difference vector with length equal to the number of reference templates. We then normalise the difference vector to the range 0.001 to 0.999, where 0.001 denotes a poor match and 0.999 denotes the best matching template. We calculate the quality of the best matching template using a method originally proposed in SeqSLAM [Milford and Wyeth, 2012], where the quality score is the ratio between the score at the best matching template and the next highest score outside a window around the best matching template. Precision and Recall scores are then calculated across a swept set of quality threshold values.

4 Experimental Method

We demonstrate our approach on three benchmark datasets, which have been extensively tested in recent literature [Han et al., 2017; Naseer et al., 2018].
Each dataset is briefly described in the sections below and visually shown in Figure 3.

St Lucia consists of multiple vehicular traverses through the suburb of St Lucia, Brisbane across five different times of day \cite{Glover2010}. We use the early morning traverse (190809.0845) as the reference dataset and the late afternoon video (180809.1545) as the query, with significant appearance change occurring between morning and afternoon. For the query traverse, we use 1000 images out of the original 15 FPS video. The dataset provides GPS ground truth and we use a ground-truth tolerance of 30 meters. For the calibration procedure, we extract 50 frames from the first 690 frames of the 15 FPS video. The query traverse is started after the last frame of the calibration procedure.

Nordland The Nordland dataset \cite{Sunderhauf2013} is recorded from a train travelling for 728 km through Norway across four different seasons. We use the Summer route as the reference dataset and the Winter traverse as the recognition route, using a 2000 image subset of the original videos. For the ground truth we compare the query traverse frame number to the matching database frame number, with a ground-truth tolerance of 10 frames, since the two traverses are aligned frame-by-frame. The 50 calibration images are collected from the videos immediately prior to the section we use for the 2000 image subset.

Oxford RobotCar - RobotCar was recorded over a year across different times of day, seasons and routes \cite{Maddern2017}. We use an approximately 2 km route through Oxford, matching from an overcast day (2014-12-09-13-21-02) to night on the next day (2014-12-10-18-10-50). We down sample the original frame rate by a factor of three and start both traverses at the same location, corresponding to 1534 query images. We use a ground truth tolerance of 40 meters, consistent with a recent publication \cite{Garg2018a}. Calibration images are collected from the dataset prior to commencing the place recognition experiment.

5 Results

To produce our results, we run our filtering algorithm on layers Conv3 through to Conv6 of HybridNet and layers Conv2 through to Conv5 of AlexNet. By experimenting on multiple layers, the layer where filtering provides the greatest value can be found. The place recognition performance is evaluated using a single-frame matching algorithm and the F1 score metric is used to quantitatively measure the performance. In Tables 1 to 6, we compare the number of feature maps pre and post filtering and display the percentage filtered across different layers, networks and datasets.

5.1 St Lucia

| Layer | Map Count | Filtered Map Count | % |
|-------|-----------|--------------------|---|
| Conv-3 | 384       | 199                | 52% |
| Conv-4 | 384       | 223                | 58% |
| Conv-5 | 256       | 153                | 60% |
| Conv-6 | 256       | 162                | 63% |

For HybridNet on St Lucia, filtering the stack of feature maps improves the localization performance across all layers (Fig. 4). This is to be expected when framed with respect to the original training data. HybridNet was trained on a collection of security cameras over time in disparate locations \cite{Chen2017a}, thus certain feature maps would have learnt to encode visual features that enable matching between summer and winter while others learn to match from morning to afternoon. Since the class output of HybridNet classifies images to
a particular location, this encoding is consistent even at higher network layers.

When filtering is applied to AlexNet, unlike HybridNet, not all layers find a major improvement after filtering. Only Conv2 and Conv3 find a significant improvement using filtering (Fig. 5). Also, a larger number of feature maps are filtered for the same gradient cut-off threshold. Since AlexNet is trained on a wider variety of images that are not applicable to visual place recognition (such as images of clothing), a larger proportion of feature maps need to be removed in the higher network layers (refer to Tables 1 and 2).

5.2 Nordland

Like our experiment on the St Lucia dataset, when filtering is applied to HybridNet, our recognition performance improves consistently across all four layers (Fig. 6). In this dataset, which has a greater appearance change, a larger number of feature maps are filtered for all four layers (see Table 3). From the results we can also infer that the higher network layers are more appearance invariant, since proportionally less feature maps require filtering.

When AlexNet is applied to the Nordland dataset, a larger proportion of feature maps require filtering (see Table 4). As can be seen in Figure 6 for Conv2, Conv3 and Conv4, feature map filtering improves the baseline place recognition performance. The improvement is particularly apparent for Conv2. In related works [Sunderhauf et al., 2015a; Sunderhauf et al., 2015b; Chen et al., 2017a], Conv2 is not considered for place recognition and our baseline results reflect the typically poor performance using Conv2. However, when filtering is used, the place recognition performance exceeds that of Conv5 baseline.

5.3 Oxford RobotCar

It is worth noting that for the same gradient cut-off threshold, more feature maps are filtered on the Oxford RobotCar dataset (see Table 5). We hypothesize that this is because this dataset has the greatest appearance variation of night to day. Filtering only improves Conv3 by a noticeable margin on the Oxford dataset (refer to Fig. 7). A possible explanation for this is a mismatch between the scene categories observed in the calibration images and the scenes observed in other sections of the dataset. For example, the calibration route occurs through an urban street with no vegetation, while

Figure 5: Maximum F1 score for Feature Map Filtering for AlexNet on the St Lucia dataset. We compare the filtered feature map recognition performance for four convolutional layers.

Table 2: Number of feature maps pre-filtering and post-filtering for AlexNet on St Lucia

| Layer | Map Count | Filtered Map Count | %  |
|-------|-----------|--------------------|----|
| Conv-2 | 256       | 129                | 50%|
| Conv-3 | 384       | 174                | 45%|
| Conv-4 | 384       | 169                | 44%|
| Conv-5 | 256       | 113                | 44%|

Figure 6: Maximum F1 score for Feature Map Filtering for HybridNet on the Nordland dataset. We compare the filtered feature map recognition performance for four convolutional layers.

Table 3: Number of feature maps pre-filtering and post-filtering for HybridNet on Nordland

| Layer | Map Count | Filtered Map Count | %  |
|-------|-----------|--------------------|----|
| Conv-3 | 384       | 150                | 39%|
| Conv-4 | 384       | 167                | 43%|
| Conv-5 | 256       | 125                | 49%|
| Conv-6 | 256       | 150                | 59%|

Table 4: Number of feature maps pre-filtering and post-filtering for AlexNet on Nordland

| Layer | Map Count | Filtered Map Count | %  |
|-------|-----------|--------------------|----|
| Conv-2 | 256       | 90                 | 35%|
| Conv-3 | 384       | 132                | 34%|
| Conv-4 | 384       | 160                | 42%|
| Conv-5 | 256       | 111                | 43%|

is particularly apparent for Conv2. In related works [Sunderhauf et al., 2015a; Sunderhauf et al., 2015b; Chen et al., 2017a], Conv2 is not considered for place recognition and our baseline results reflect the typically poor performance using Conv2. However, when filtering is used, the place recognition performance exceeds that of Conv5 baseline.
later in the dataset, the road travels past a park. Conv3 encodes more generic visual features which are captured during the calibration route.

Figure 7: Maximum F1 score for Feature Map Filtering for AlexNet on the Nordland dataset. We compare the filtered feature map recognition performance for three convolutional layers.

Table 5: Number of feature maps pre-filtering and post-filtering for HybridNet on Oxford RobotCar

| Layer | Map Count | Filtered Map Count | % |
|-------|-----------|--------------------|---|
| Conv-3 | 384       | 117                | 30% |
| Conv-4 | 384       | 137                | 36% |
| Conv-5 | 256       | 112                | 44% |
| Conv-6 | 256       | 134                | 52% |

When our calibration procedure is applied to AlexNet, the same trend continues - the larger appearance variation causes a greater proportion of feature maps to be filtered out (refer to Table 6). For the Conv2 layer, three quarters of the original stack of feature maps are removed and in doing so, the maximum F1 score increases from 0.41 to 0.69. This is further evidence that our proposed approach is successfully finding the feature maps that are consistent across the appearance change. Again, the higher level layers gain no localization benefit from feature map filtering, however an improvement is still made to the compute time.

6 Discussion

6.1 Localization Improvement
A key result from our experimentation is that filtering provides a considerable improvement to earlier convolutional layers. Early layers have been shown to encode simple visual features while later layers encode objects and regions that are associated with the final class outputs [Zhou et al., 2018]. Our results show that filtering object types has less of an advantage, since objects within a scene are typically less affected by environmental changes than lower level visual features, such as the color of the leaves of a tree. When an early layer is filtered, filters that encode a visual feature that is impacted by the change in environment is removed, leaving only the visual features that remain consistent over time. The feature maps selected by our approach can be visually seen in Figure 10. We also show examples where our filtering approach enables localization when the baseline of not filtering causes an incorrect place hypothesis (see Fig. 11).

6.2 Computational Improvement
Our improved F1 scores across most layers on both HybridNet and AlexNet is particularly significant when
Figure 10: On the Oxford dataset and using Conv2 of AlexNet, 62 feature maps are selected post-filtering. Using MATLAB’s deepDreamImage function, we visualize the types of visual features that the chosen feature maps respond to (left-hand montage) and compare against a selection of filtered feature maps (right-hand montage). Notice the similarity between the images in the left-hand montage and the presence of many ‘line-based’ filters. This is explainable considering that street lighting is designed to illuminate road markings and road markings are typically straight line segments.

compared to the quantity of feature maps that are removed. As can be seen in the six tables, our filter algorithm filters, on average, 51% of all feature maps when HybridNet is used and 61% when AlexNet is used. This is a significant reduction of information and yet we achieve improved localization performance and significantly improve the place recognition computation time. For example, using Conv3 of HybridNet requires an average of 68 ms to match a query image to a reference database of 1442 images (on a standard desktop PC). When filtering is used, this drops to 43.9ms, 64% of the original time per frame. This is even more apparent with Conv2 of AlexNet on the Oxford RobotCar dataset, where the processing time halves from 81ms to 41ms.

7 Conclusion and Future Work

This paper proposes a novel method of performing convolutional network calibration for visual place recognition, without requiring any computationally intensive re-training of the neural network parameters. We achieve this by filtering the set of feature maps produced by a layer within a CNN, by minimizing the L2 distance between the current scene and the corresponding reference image while maximizing the distance between the reference image and another reference image elsewhere in the database. Our feature map filtering approach has two key advantages: improved localization ability in changing environments, and improved computation speed. Our results demonstrate a considerable localization improvement for earlier network layers, with the greatest improvement on the Oxford RobotCar dataset, matching from night to day, using the Conv3 layer on HybridNet and the Conv2 layer on AlexNet. Our calibration procedure resulted in an improvement in HybridNet’s Conv3 F1 score from 0.56 to 0.81 and AlexNet’s Conv2 F1 score from 0.41 to 0.69.

Future work will devise a method of performing feature map filtering in real-time, without requiring any prior calibration. This could be achieved by devising a method of classifying the type of visual feature a particular feature map is activating and specifically filtering the set of classes that are only occurring in the query traverse and not present anywhere in the reference traverse (such as street lighting at night-time). Also, our feature map calibration strategy using Greedy could be replaced with an alternative heuristic, to further improve the optimization quality. Finally, feature map filtering may also have applications in other computer vision tasks, as this approach could be used to quickly prepare a deep, generically trained CNN for a very specific task without re-training the network weights.

Figure 11: Examples on St Lucia, Nordland and Oxford where filtering the stack of feature maps enables successful localization. The baseline match was generated using all the feature maps in Conv3 of HybridNet and the filter match is the correct location hypothesis when the filter calibration procedure is applied.

References

[Arandjelovi et al., 2018] Relja Arandjelovi, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(6):1437–1451, 2018.

[Bay et al., 2008] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). Computer Vision and Image Understanding, 110(3):346–359, 2008.
[Chen et al., 2017a] Z. Chen, A. Jacobson, N. Sunderhauf, B. Upcroft, L. Liu, C. Shen, I. Reid, and M. Milford. Deep learning features at scale for visual place recognition. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pages 3223–3230. IEEE, 2017.

[Chen et al., 2017b] Z. Chen, F. Maffra, I. Sa, and M. Chli. Only look once, mining distinctive landmarks from convnet for visual place recognition. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 9–16, 2017.

[Chen et al., 2018] Z. Chen, L. Liu, I. Sa, Z. Ge, and M. Chli. Learning context flexible attention model for long-term visual place recognition. IEEE Robotics and Automation Letters, 3(4):4015–4022, 2018.

[Cummins and Newman, 2008] Mark Cummins and Paul Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. The International Journal of Robotics Research, 27(6):647–665, 2008.

[Dalal and Triggs, 2005] N. Dalal and B. 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 vol. 1, 2005.

[Fegaras, 1998] L. Fegaras. A new heuristic for optimizing large queries. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 1460:726–735, 1998.

[Garg et al., 2018a] Sourav Garg, Niko Sunderhauf, and Michael Milford. Don’t look back: Robustifying place categorization for viewpoint- and condition-invariant place recognition. In 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018.

[Garg et al., 2018b] Sourav Garg, Niko Sunderhauf, and Michael Milford. Lost? appearance-invariant place recognition for opposite viewpoints using visual semantics. Proceedings of Robotics: Science and Systems XIV, 2018.

[Glover et al., 2010] A. J. Glover, W. P. Maddern, M. J. Milford, and G. F. Wyeth. Fab-map + ratslam: Appearance-based slam for multiple times of day. In 2010 IEEE International Conference on Robotics and Automation, pages 3507–3512. IEEE, 2010.

[Guo and Potkonjak, 2017] J. Guo and M. Potkonjak. Pruning convnets online for efficient specialist models. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pages 430–437. IEEE, 2017.

[Han et al., 2017] F. Han, X. Yang, Y. Deng, M. Rentschler, D. Yang, and H. Zhang. Srsl: Shared representative appearance learning for long-term visual place recognition. IEEE Robotics and Automation Letters, 2(2):1172–1179, 2017.

[Jacobson et al., 2015] A. Jacobson, Z. Chen, and M. Milford. Autonomous multisensor calibration and closed-loop fusion for slam. Journal of Field Robotics, 32(1):85–122, 2015.

[Joshi and Boyd, 2009] S. Joshi and S. Boyd. Sensor selection via convex optimization. IEEE Transactions on Signal Processing, 57(2):451–462, 2009.

[Kim et al., 2017] H. J. Kim, E. Dunn, and J. Frahm. Learned contextual feature reweighting for image geolocalization. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3251–3260, 2017.

[Krizhevsky et al., 2012] A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems, volume 2, pages 1097–1105, 2012.

[Li et al., 2018] Xin Li, Zequn Jie, Jiashi Feng, Changsong Liu, and Shuicheng Yan. Learning with rethinking: Recurrently improving convolutional neural networks through feedback. Pattern Recognition, 79:183–194, 2018.

[Lopez-Antequera et al., 2017] Manuel Lopez-Antequera, Ruben Gomez-Ojeda, Nicolai Petkov, and Javier Gonzalez-Jimenez. Appearance-invariant place recognition by discriminatively training a convolutional neural network. Pattern Recognition Letters, 92:89–95, 2017.

[Maddern et al., 2017] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. The International journal of robotics research., 36(1):3–15, 2017.

[Milford and Wyeth, 2012] M. J. Milford and G. F. Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In 2012 IEEE International Conference on Robotics and Automation, pages 1643–1649. IEEE, 2012.

[Naseer et al., 2018] Tayyab Naseer, Wolfram Burgard, and Cyrill Stachniss. Robust visual localization across seasons. Robotics, IEEE Transactions on, 34(2):289–302, 2018.

[Park et al., 2018] C. Park, J. Jang, L. Zhang, and J. Jung. Light-weight visual place recognition using convolutional neural network for mobile robots. In 2018 IEEE International Conference on Consumer Electronics (ICCE), pages 1–4, 2018.
[Schroff et al., 2015] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 815–823, 2015.

[Sunderhauf et al., 2013] N. Sunderhauf, P. Neubert, and P. Protzel. Are we there yet? challenging seqslam on a 3000km journey across all four seasons. In Proc. of Workshop on Long-Term Autonomy IEEE International Conference on Robotics and Automation (2013). IEEE, 2013.

[Sunderhauf et al., 2015a] N. Sunderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford. On the performance of convnet features for place recognition. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4297–4304. IEEE, 2015.

[Sunderhauf et al., 2015b] N. Sunderhauf, 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 Robotics: Science and Systems, volume 11, 2015.

[Zhang et al., 2018] Qiang Zhang, Li Zhuo, Jiafeng Li, Jing Zhang, Hui Zhang, and Xiaoguang Li. Vehicle color recognition using multiple-layer feature representations of lightweight convolutional neural network. Signal Processing, 147:146–153, 2018.

[Zhou et al., 2018] B. Zhou, D. Bau, A. Oliva, and A. Torralba. Interpreting deep visual representations via network dissection. IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 1–1, 2018.

[Zou et al., 2018] J. Zou, T. Rui, Y. Zhou, C. Yang, and S. Zhang. Convolutional neural network simplification via feature map pruning. Computers and Electrical Engineering, 70:950–958, 2018.