Filter Early, Match Late: Improving Network-Based Visual Place Recognition

Stephen Hausler, Adam Jacobson and Michael Milford

Abstract— CNNs have excelled at performing place recognition over time, particularly when the neural network is optimized for localization in the current environmental conditions. In this paper we investigate the concept of feature map filtering, where, rather than using all the activations within a convolutional tensor, only the most useful activations are used. Since specific feature maps encode different visual features, the objective is to remove feature maps that are detract from the ability to recognize a location across appearance changes. Our key innovation is to filter the feature maps in an early convolutional layer, but then continue to run the network and extract a feature vector using a later layer in the same network. By filtering early visual features and extracting a feature vector from a higher, more viewpoint invariant later layer, we demonstrate improved condition and viewpoint invariance. Our approach requires image pairs for training from the deployment environment, but we show that state-of-the-art performance can regularly be achieved with as little as a single training image pair. An exhaustive experimental analysis is performed to determine the full scope of causality between early layer filtering and late layer extraction. For validity, we use three datasets: Oxford RobotCar, Nordland, and Gardens Point, achieving overall superior performance to NetVLAD. The work provides a number of new avenues for exploring CNN optimizations, without full re-training.

I. Introduction

Convolutional neural networks have demonstrated impressive performance on computer vision tasks [1], [2], including visual place recognition [3], [4]. Recently, researchers have investigated optimising and improving pre-trained CNNs, by either extracting salient features [5], [6], or by 'pruning' the network [7], [8]. Network pruning is typically used to increase the computation speed of forward-pass computation; however, our previous work has provided a proof of concept that a type of pruning, dubbed "feature map filtering", can *also* improve the place recognition performance of a pre-trained CNN [9].

In feature map filtering, specific feature maps are removed, based on their suitability to identify the correct matching location across a changing environmental appearance. In this work, we propose that an early convolutional layer can be filtered to improve the matching utility of feature vectors extracted from a network's *later* layers. By performing this early layer filtering, simple visual features (e.g. textures and contours) that detract from a network's utility for place recognition across changing environmental conditions are
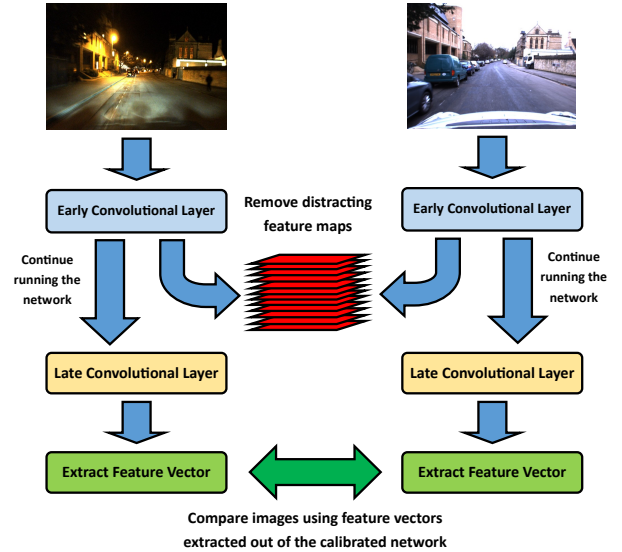
Fig. 1. Our method removes early layer feature maps that have learnt visual features that detract from place recognition performance to improve matching performance using downstream layers.

removed. Crafting a feature vector out of a later layer is beneficial, as research [3], [10] has shown that later CNN layers are more invariant to viewpoint changes. We verify the ability to handle viewpoint variations by using the Gardens Point Walking dataset, and handle condition variations using the Oxford RobotCar dataset (matching from night to day) and the Nordland dataset (matching from summer to winter).

We summarize the contributions of this work:

- We propose a novel method of performing feature map filtering (or pruning) on early convolutional layers, while extracting features for place recognition out of later convolutional layers.
- To determine the selection of feature maps to remove, we have developed a Triplet Loss calibration procedure which uses training image pairs to remove feature maps that show consistent detriment in the ability to localize in the current environment. We demonstrate experimentally that state-of-the-art performance can be achieved with as little as a single training image pair.
- We provide a thorough experimental evaluation of the effects of filtering CNN feature maps for a pre-trained neural network, exhaustively testing all combinations in the layers Conv2 to Conv5. We also include a set of experiments filtering Conv2 and using the first fully connected layer as a feature vector.

Our results also reveal the inner workings of neural networks - a neural network can have a portion of it's feature maps completely removed and yet a holistic feature vector can be extracted out of a higher convolutional layer. We also provide a visualization of the activations within a higher layer of the filtered network.

The paper proceeds as follows. In Section II, we review feature map pruning literature and discuss the application of neural networks in visual place recognition. Section III presents our methodology, describing the calibration procedure. Section IV details the setup of our three experimental datasets and Section V discusses the performance of filtering different convolutional layers on these datasets. Section VI summarizes this work and provides suggestions for future work.

II. RELATED WORK

The recent successes of deep learning in image classification [11] and object recognition [12] have encouraged the application of neural networks in place recognition. In early work, the pre-trained AlexNet [1] network is used to produce a feature vector out of the Conv3 layer [3], [13]. Rather than simply using a pre-trained network, NetVLAD learns visual place recognition end-to-end. In NetVLAD, triplet loss is used to find the optimal VLAD encoding to match scenes across both viewpoint and condition variations [4]. LoST uses the semantic CNN RefineNet [14] to select salient keypoints within the width by height dimensions of a convolutional tensor [5]. In a related work, these keypoints have been found by observing the activations out of a late convolutional layer [15]. The aforementioned examples involve improving a pre-trained neural network for place recognition, either by re-training, or selecting the most useful components out of the network activations.

Several works perform network simplification. Initially, CNNs were compressed by selectively removing specific convolutional weights. For example, Han et al. prunes convolutional weights that are below a certain threshold, then applies quantization and Huffan coding to reduce the network size by up to 49 times [16]. An alternate strategy is to remove all weights in an entire feature map, which is termed filter pruning. Li et al. uses the absolute magnitude of weights within each feature map as the pruning metric, with low magnitude maps being removed [17]. An alternate approach is to perform Linear Discriminant Analysis, to preserve the class structure in the final layer while reducing the number of feature maps in earlier layers [8]. A recent work, which is currently under review, uses structured sparsity regularization and Stochastic Gradient Descent to prune unnecessary feature maps [18]. A 'soft' filter has been used to enhance visual place recognition, where the activations across a stack of feature maps are scaled with a learnt scaling value and summed together to create a learnt feature vector. There are two works in this space: the first performs feature weighting on the last convolutional layer [19], while the second re-weights a concatenation of features out of multiple convolutional layers [20].

In the aforementioned feature map pruning literature, after filtering feature maps, the smaller network is briefly re-trained (to remove sparsity). In this work, we show that a network can be left sparse "as-is" and continues to produce coherent activations in higher network layers, even up to the fully-connected layer.

III. PROPOSED APPROACH

A pre-trained neural network, which was trained on a diverse set of images, will learn internal representations of a wide range of different visual features. However, in visual place recognition, perceptual aliasing is a common problem. Perceptual aliasing is where certain visual features make a scene visually similar to a previously observed scene in a different location. If a pre-trained network is selectively filtered on the expected environment, then visual features that contribute to perceptual aliasing can be removed, leaving the feature maps that encode visual features that can suitably match between two appearances of the same location. We use a short calibration method to prepare our feature map filter, as described in the following sub-sections.

A. Early Feature Filtering, Late Feature Extraction

In our previous work on feature map filtering [9], we filtered the feature map stack while extracting the feature vector from the same layer. While we demonstrated improved place recognition performance, this approach was not capable of optimizing for the extraction of visual features in higher convolutional layers. We hypothesize that filtering an early convolutional layer will remove distracting visual features, while crafting a feature vector out of a later layer has been shown [10] to have improved viewpoint robustness.

Our improved approach filters the feature map space within a CNN, except the network is allowed to continue after filtering. Optimizing the filter is now dependent on the triplet loss on the features extracted out of a higher convolutional layer. This also adds the concept of feedback to a neural network, by modulating early visual features with respect to higher level features.

B. Deep Learnt Feature Extraction and Triplet Loss

A feature vector is extracted out of a width by height by channel ($W \times H \times C$) convolutional tensor. To improve viewpoint robustness and increase the processing speed, dimensionality reduction is performed using spatial pooling [21]. As per our previous work [9], we again use pyramid spatial pooling and convert each $W \times H$ tensor into a vector of dimension 5, containing the maximum activation across the entire feature map plus the maximum activation in each quadrant of the feature map. For all our experiments we use the pre-trained network HybridNet [21]. HybridNet was trained with place recognition in mind, resulting in a well-performing pre-trained network with a fast forward-pass processing speed.

Triplet Loss [4] involves comparing the feature distance between a positive pair of images relative to one or more negative pairs of images. In this case, the positive pair are
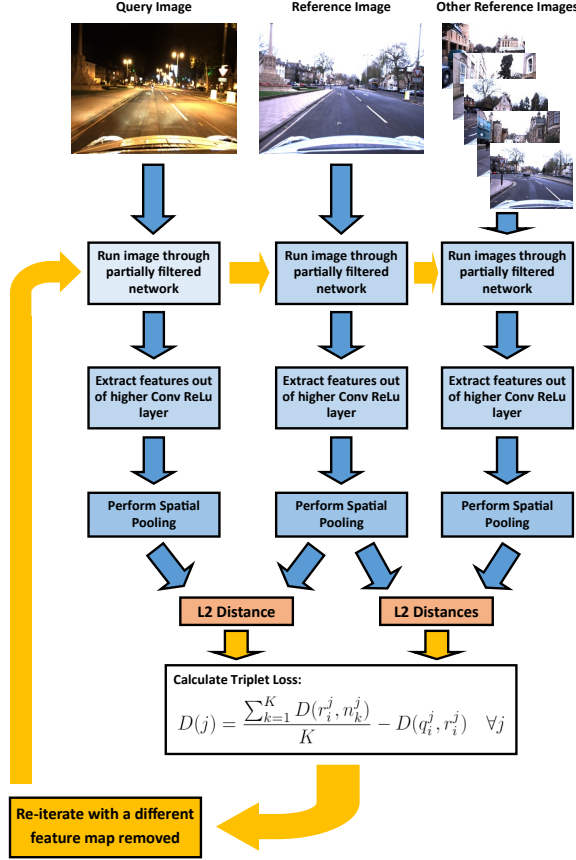
Fig. 2. Flow chart showing the Triplet Loss implementation used to rank the importance of early layer feature maps.

two images taken from the same physical location, but at different times, while the negative pairs are images taken at a similar time but at varying locations. We use one 'hard' negative pair, in this case, the second reference image is an image that is a fixed number of frames ahead of the current frame. This distance is slightly larger than the ground truth tolerance. We then have four 'soft' negatives, which are random images elsewhere in the reference dataset. Including a fixed, hard negative reduces variance in the filter calibration. As per literature best practise [4], [22], we use the L2-Distance as our optimization metric. Figure 2 shows an overview of our proposed approach.

C. Filtering Method

We use a type of Greedy algorithm [23] to determine which subset of the feature map stack suits the current environmental conditions. Our variant of the Greedy algorithm finds the worst performing (with respect to the triplet loss) feature map at each iteration of the algorithm. Normally the Greedy algorithm will terminate when the local minimum triplet loss is reached; however, to guarantee that the global minimum is found, we continue iterating until half the original feature map stack has been removed. We store the filter selection at each iteration and search for the global minimum across all iterations.

Additionally, we also implement a batch filtering extension to the aforementioned algorithm. In batch filtering, in each iteration, the four worst feature maps are discovered (based on the triplet loss) and removed before the next iteration. We can safely add this approximation because of the global minimum search. If removing four maps at once prevents the loss function from being convex, the best match can still be found due to the global search. Adding batch filtering improves the computation speed of calibration by a factor of four. The decision to terminate the search at half the maps removed is a heuristically determined trade-off between calibration processing time and localization benefits.

Determining the worst performing feature map is based on the triplet loss score out of a higher network layer. Specifically, each feature map is individually removed and the network is continued to run further into the forward pass. The triplet loss is then calculated based on the feature vectors extracted out of a higher network layer. As mentioned previously, we apply a maximum spatial pooling operation on the raw Conv ReLu activations. The purpose of this is to reduce the dimensionality of the feature vector, and to ensure the filtering process focuses on strong activations. For each pair of images in the triplet set, the L2 distance between that pair is calculated as per the equation below.

$$D(q_i^j, r_i^j) = \sqrt{\sum_{k=1}^{M}(q_i^j(k) - r_i^j(k))^2} \tag{1}$$

where M is the dimension of the filtered query feature vector q_i^j.

The equation above is repeated for the five negative pairs. The difference scores for the five pairs are then averaged together. The triplet loss is the difference between the positive pair and the averaged negative pair, across a different feature map j being removed.

$$D(j) = \frac{\sum_{k=1}^{K} D(r_i^j, n_k^j)}{K} - D(q_i^j, r_i^j) \quad \forall j \tag{2}$$

where r_i^j represents the current location filtered reference feature vector and n_i^j represents the averaged negative. j denotes the index of the currently filtered feature map. In our experiments, K is set to 5 since the set of n_k consists of one fixed reference image and four, randomly selected, reference images.

We then find the maximum distance:

$$maxval = \max_{1 \leq j \leq N} D(j) \tag{3}$$

$$worstFmap = \underset{1 \leq j \leq N}{\operatorname{argmax}} \ D(j) \tag{4}$$

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 average negative distance. To implement the batch filtering, the previous worst map from D is removed and equation 4 is repeated until the four worst feature map ids are collected.

At the end of each iteration, the weights and biases in an earlier convolutional layer are set to zero, for each weight inside the feature maps selected to be filtered. We then return the new, partially zeroed CNN to the next iteration of the algorithm. These iterations continue until half the features maps are removed. The global maximum triplet loss score is then found, and the selection of filtered feature maps at the maximum loss score are the final set of filtered maps for that calibration image.

Finally, 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 the calibration image sets. The final filtered maps are feature maps that were chosen to be removed in at least 66% of the calibration sets. This threshold was heuristically determined, using place recognition experimentation on a range of different thresholds. With this threshold, on average, approximately 25% of earlier layer feature maps are removed after filtering. We chose this metric based on the objective to find feature maps that are consistently poor at navigation, rather than feature maps that are only inefficient for a single image. This approach reduces the risk of overfitting the filter to the calibration data.

D. Place Recognition Filter Verification Algorithm

To evaluate the performance of the calibrated feature map filter, we use a single-frame place recognition algorithm. To apply the filter, every convolutional weight in a filtered feature map is set to zero. This new network is then run in the forward direction to produce convolutional activations in higher network layers. We again apply spatial pooling to the convolutional activations, producing a feature vector of length five times the number of feature maps.

The feature vectors from the reference and query traverses are compared using the cosine distance metric. While the euclidean distance was the training distance metric, our experiments revealed that it is advantageous to train the filter using the euclidean distance metric but perform place recognition using the cosine similarity metric. In early experiments, we also checked the performance by training with the cosine distance metric instead and found a reduction in the resulting place recognition performance.

The resultant difference vector is then normalized to the range 0.001 to 0.999, where 0.001 denotes the worst match and 0.999 the best match. We then apply the logarithm operator to every element of the difference vector. Taking the logarithm amplifies the difference between the best match and other, perceptually aliased matches [24]. The place recognition quality score is calculated using the method originally proposed in SeqSLAM [25], where the quality score is the ratio between the score at the best matching template and the next largest score outside a window around the best matching template. A set of precision and recall values are calculated by varying a quality score threshold value. For compact viewing, we display the localization

performance using the maximum F1 score metric, where the F1 score is the harmonic mean of precision and recall.

IV. EXPERIMENTAL METHOD

We demonstrate our approach on three benchmark datasets, which have been extensively tested in recent literature [10], [26], [27]. The datasets are Oxford RobotCar, Nordland, and Gardens Point Walking. Each dataset is briefly described in the sections below.

Oxford RobotCar - RobotCar was recorded over a year across different times of day, seasons and routes [28]. For our training set, we use 50 image sets (a positive image, an anchor and five negative images) extracted at an approximate frame rate of one frame every two seconds. Using a low frame rate ensures that the individual images show some diversity between them. Therefore, the calibration set has a duration of approximately 100 seconds, which is a realistic and practical calibration duration for a real-world application. We also experiment with a smaller number of calibration image sets, to observe the effects of using fewer calibration images.

For our test set, we use 1600 frames extracted out of the dataset, which corresponds to approximately two kilometers through Oxford. There are no training images present in the test set. The reference dataset was recorded on an overcast day (2014-12-09-13-21-02), while the query dataset is at nighttime on the following day (2014-12-10-18-10-50). We use a ground truth tolerance of 30 meters, consistent with recent publications [10], [24].

Nordland The Nordland dataset [29] is recorded from a train travelling for 728 km through Norway across four different seasons. The training set again consists of 50 images, with a recording frame rate of 0.2 frames per second. The resultant calibration duration is 250 seconds; a longer real-world duration was heuristically chosen to account for the significantly larger real-world distance of the Nordland dataset (compared to Oxford RobotCar or Gardens Point Walking).

For the experimental dataset, we use the Winter route as the reference dataset and the Summer traverse as the recognition route, using a 2000 image subset of the original videos. In our previous work [9] we used the Summer images as the reference set and the Winter images as query; we flipped the order because we found that matching from Summer to Winter to be more challenging. 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. Again the test set contains no images from the training set.

Gardens Point Walking - was recorded at the QUT university campus in Brisbane and consists of two traverses during the day and one at night, with a duration of 200 images per traverse [3]. One of the day traverses is viewed from the left-hand side of the walkways, while the second day and the night traverse were both recorded from the right-hand side. We train our filter on the comparison between the left-hand side at daytime to the right-hand side at nighttime,

using just 5 calibration images. We then use 194 images as the evaluation set and a ground truth tolerance of 3 frames.

V. RESULTS

In this section, a detailed analysis is performed on the performance of feature map filtering in visual place recognition. The results are shown using the maximum F1 score metric and we compare our early layer filter approach to three benchmarks. First, we compare against filtering the same layer as the feature vector is extracted from. To ensure a fair comparison, the same triplet loss method is used, including using five negative images. The second benchmark is the localization performance without any filtering at all. Finally, we also compare against pre-trained NetVLAD (trained on Pittsburgh 30k) [4]. NetVLAD normally outputs results as a Recall@N metric; we convert this to an equivalent F1 score by assuming a precision score of 100% and using the Recall@1 value as the recall score.

Figure 3 provides a summary of the overall place recognition performance across all three datasets. Overall, removing feature maps in the same layer as the feature vector is extracted from has a higher maximum F1 score with respect to both NetVLAD, and the same network without any filtering. Filtering the feature maps in an earlier convolutional layer produces a further improvement to the average place recognition performance.

A. Oxford RobotCar

Early layer filtering generally improves localization on the Oxford RobotCar dataset (see Figures 4 to 6). If Conv3 features are used for localization, then whether an earlier layer or the current layer is filtered is largely irrelevant. However, whichever method is used results in a significant improvement in localizing with these features. When Conv2 is filtered and Conv4 features are used, the localization experiment results in a maximum F1 score improvement of 0.8, compared to filtering on the same convolutional layer (Conv4).

In Figure 7, we varied the number of calibration images used when training the filter on the Conv2 layer. We used as little as 1 calibration image, up to 50 calibration images. We determined that the localization performance improves gradually, and even calibrating with five images in the query environment improves the place recognition performance above both NetVLAD and HybridNet without any filtering. This is particularly apparent with Conv3 features, which normally are not a suitable choice for a localization system. As a general rule, the more calibration images, this lower the risk of over-fitting the filter on the calibration data. These results indicate that even if only a single calibration image is available, our approach can provide an improvement to localization. This also indicates that there are visual features which are a detriment to place recognition across all variations in the remainder of the Oxford RobotCar dataset (from night to day), such that 1 image of the environment is sufficient to remove many of these poor visual features.
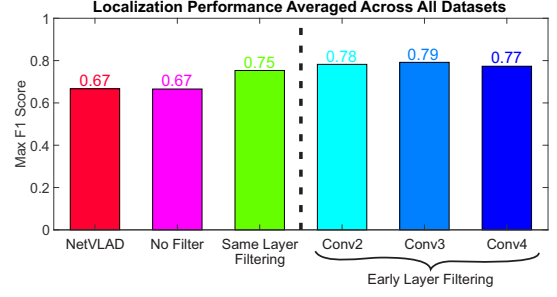

Fig. 3. Overall performance with feature map filtering, averaged across the three datasets.
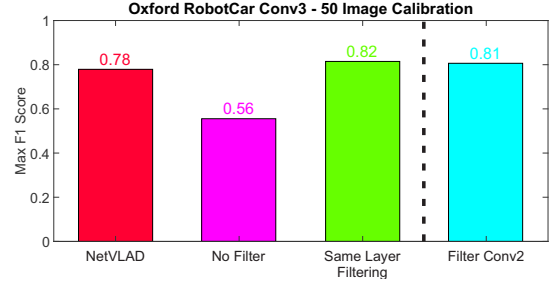

Fig. 4. Maximum F1 score for Feature Map Filtering on the Oxford RobotCar dataset, extracting features from the Conv3 layer. We compare the localization performance without any filtering (*No Filter*), with filtering the same layer from which the image descriptor is extracted (*Same Layer Filtering*), and filtering the previous layer (*Filter Conv2*). Finally we compare our results to NetVLAD.
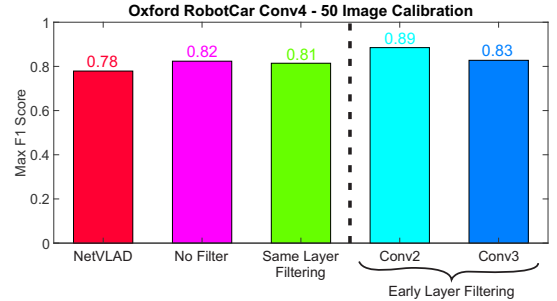

Fig. 5. Maximum F1 score for Feature Map Filtering on the Oxford RobotCar dataset, extracting features from Conv4. Again we compare the localization performance without filtering (*No Filter*), with filtering the same layer (*Same Layer Filtering*), and filtering either of the two previous layers (*Early Layer Filtering*). Finally we compare our results to NetVLAD.
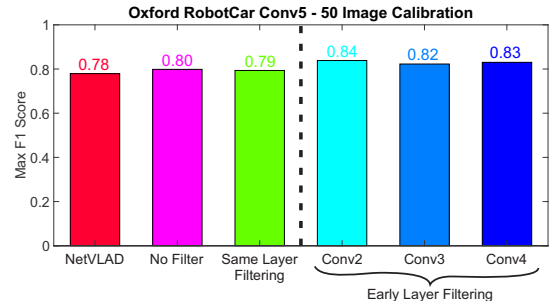

Fig. 6. Maximum F1 score for Feature Map Filtering on the Oxford RobotCar dataset, extracting features from the Conv5 layer. We compare the localization performance without any filtering (*No Filter*), with filtering the same layer from which the image descriptor is extracted (*Same Layer Filtering*), and filtering either of the three previous layers (*Early Layer Filtering*). Finally we compare our results to NetVLAD.
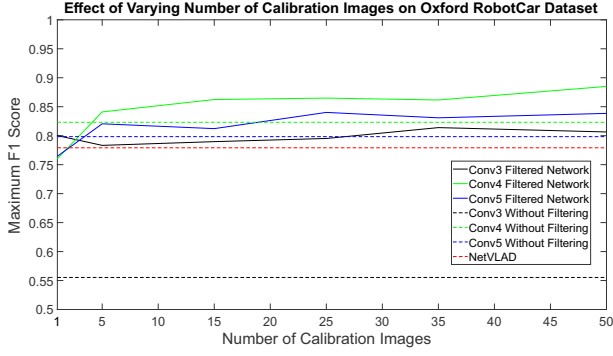
Fig. 7. In this experiment, we filter feature maps from the Conv2 layer and extract a feature vector out of one of the later layers. Increasing the number of calibration images provides a small improvement to the localization ability of the network. Even if just 5 calibration images are available, the filtering approach still beats both the baseline without filtering, and NetVLAD. With Conv3 features and a single calibration image, the maximum F1 score is higher than NetVLAD.
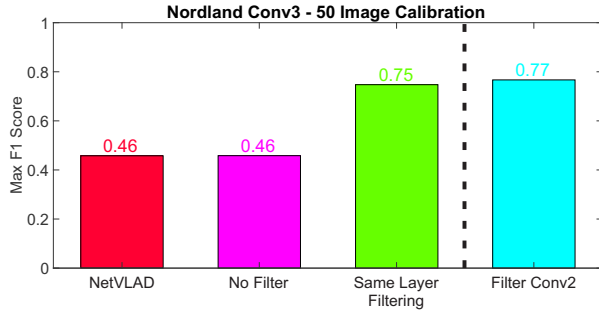

Fig. 10. Maximum F1 score for Feature Map Filtering on the Nordland dataset, extracting features from Conv5. We compare the localization performance without any filtering (*No Filter*), with filtering the same layer from which the image descriptor is extracted (*Same Layer Filtering*), filtering either of the three previous layers. (*Early Layer Filtering*), and to the comparison approach NetVLAD.

Fig. 8. Maximum F1 score for Feature Map Filtering on the Nordland dataset, extracting features from Conv3. We compare the localization performance without any filtering (*No Filter*), with filtering the same layer from which the image descriptor is extracted (*Same Layer Filtering*), filtering the previous layer (*Filter Conv2*), and to NetVLAD.
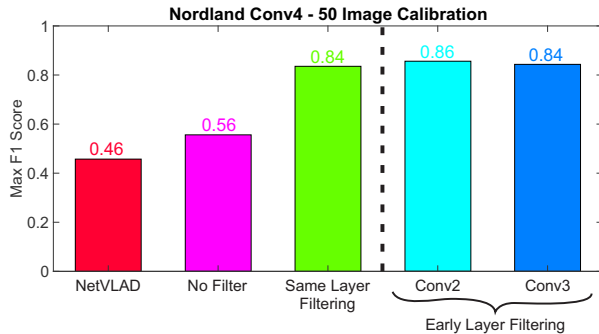

Fig. 11. Maximum F1 score for Feature Map Filtering on the Gardens Point dataset, extracting features from Conv3. We compare the localization performance without any filtering (*No Filter*), with filtering the same layer from which the image descriptor is extracted (*Same Layer Filtering*), filtering the previous layer (*Filter Conv2*), and to NetVLAD.

Fig. 9. Maximum F1 score for Feature Map Filtering on the Nordland dataset, extracting features from Conv4. We compare the localization performance without any filtering (*No Filter*), with filtering the same layer from which the image descriptor is extracted (*Same Layer Filtering*), filtering either of the two previous layers (*Early Layer Filtering*), and to NetVLAD.
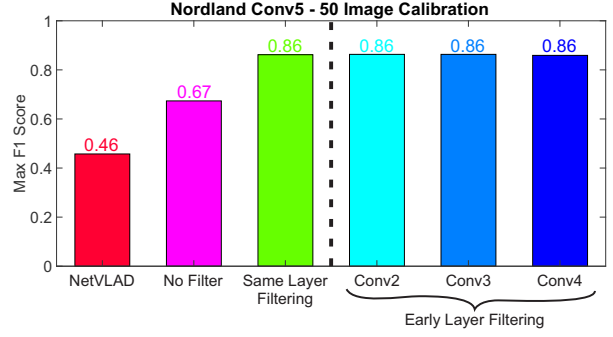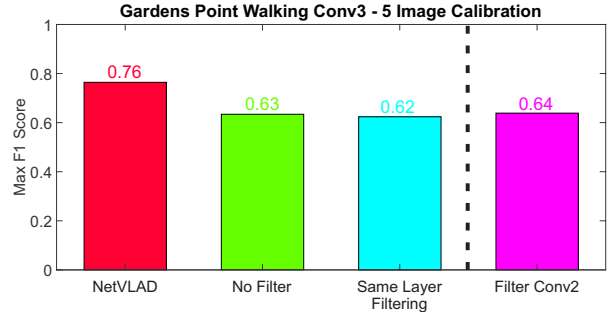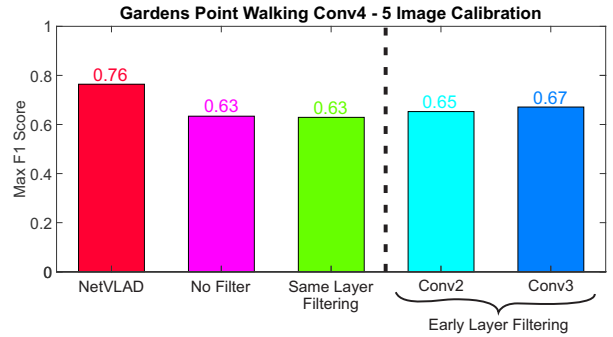

Fig. 12. Maximum F1 score for Feature Map Filtering on the Gardens Point dataset, extracting features from Conv4. We compare the localization performance without any filtering (*No Filter*), with filtering the same layer (*Same Layer Filtering*), filtering either of the two previous layers (*Early Layer Filtering*), and to NetVLAD.
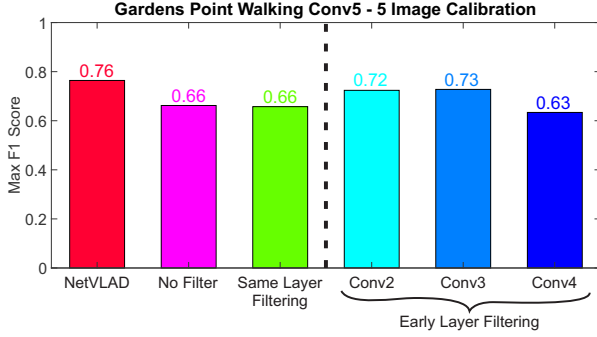
Fig. 13. Maximum F1 score for Feature Map Filtering on the Gardens Point dataset, extracting features from Conv5. We compare the localization performance without any filtering (*No Filter*), with filtering the same layer (*Same Layer Filtering*), filtering either of the three previous layers (*Early Layer Filtering*), and to NetVLAD.

B. Nordland

The striking result from these experiments (Figures 8 to 10) is the magnitude of improvement added with filtering, which is much greater than the other two datasets. We hypothesize that this railway dataset can be easily encoded using a set of calibration images. The environmental appearance of both summer and winter traverses changes little over the dataset, unlike the Oxford RobotCar dataset, where street lighting makes the environmental change more dynamic.

The choice of layer to filter is mostly indeterminate on this dataset. Because there are no viewpoint variations on this dataset, the max-pooling operations between layers makes little difference to the localization performance. Therefore, once the distracting visual features are removed from any layer in the network, the choice of layer to extract a feature vector from becomes largely irrelevant.

C. Gardens Point Walking

To test our theory that early layer filtering is advantageous for viewpoint variant datasets, we used the left-hand and right-hand Gardens Point Walking traverses. As expected, the higher Conv5 layer achieved the highest localization performance, attaining a maximum F1 score of 0.73 if Conv3 is filtered first (see Figures 11 to 13). There is a decent gap between filtering Conv3 and filtering Conv5, with a improvement in F1 score of 0.7. The improvement in F1-score using just 5 calibration images indicates that the early layer filtering process is particularly useful when moderate viewpoint variations are present. NetVLAD performs well on this dataset and beats any of our filters. NetVLAD is designed for viewpoint invariance in mind, by virtue of the learnt VLAD clustering and use of features out of the final convolutional layer. The gap between our approach and NetVLAD becomes small when Conv3 was filtered and the feature vector was produced from Conv5, using just five calibration images.

D. Fully Connected Features

We performed a final experiment, to consider using feature map filtering to optimize a feature vector formed using the first full-connected layer. We directly used the activations within the ReLu layer after the first fully-connected layer as the feature vector. We filtered Conv2 using the exact same triplet loss method on the three datasets, and show the results in Figure 14 below. Notice how the result with a random set of filtered maps is worse than the baseline performance. This result shows that feature map filtering is beneficial because of the objective function, and not because of any inherent benefits of reducing the dimensionality of the network.
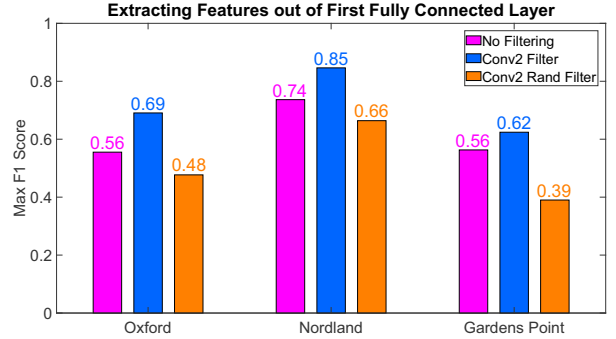

Fig. 14. Maximum F1 Scores for fully connected layer features, both without any earlier filtering and also with filtering the Conv2 layer. We also include results where the same number of feature maps are removed, but with a random map selection.

E. Visualization of Feature Map Filtering

By plotting the maximum activation coordinates onto a heat map, we can observe the change in activations after the network is filtered. As shown in Figure 15, the filtering process effects the spatial position of the maximum activations. We found that the magnitude of these maximum activations are still different, however, the location of the maximum activations within the spatial region of the feature maps becomes more accurate, between environments.
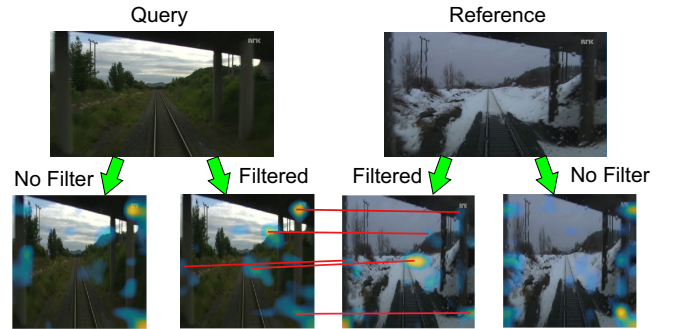

Fig. 15. Visualization of Conv5 maximum activations both without filtering, and after filtering the Conv3 layer, on the Nordland dataset. The red lines indicate spatially consistent maximum activations, between the filtered maps of the reference and query images.

VI. DISCUSSION AND CONCLUSION

In this paper, we presented an early filtering, late matching approach to improving visual place recognition in appearance-changing environments. We showed that CNNs

tend to activate in response to features with little utility for appearance-invariant place recognition, and show that by applying a calibrated feature map filter, these distracting features are removed from the localization feature vectors. Our results indicate that filtering an *earlier* layer of the network generally results in better performance than filtering the same layer that the feature vector is extracted from. We also provide a case where we filter the Conv2 layer and extract features out of the first fully connected layer, demonstrating the versatility of early layer filtering.

The experimental results show that a network layer can be severely pruned and yet continue to be run in the forward direction with coherent and effective activations in a later layer. Our approach also does not re-train after pruning, unlike many previous work in the space [7], [8], [17], [18]. Therefore this research shows that, while later layers are directly impacted by the complete removal of early features, the removal of up to 50% of these early features does not cause a catastrophic collapse of activation strength in later layers. Note that we did find that removing more than 50% of feature maps in an early layer dramatically increased the risk of localization instability, as the later activations experience significantly reduced activation strength. Also, removing too many feature maps during calibration risks overfitting the training data. Our results show that a small number of feature maps can be selectively pruned from an early convolutional layer, to optimize localization in the current environment. The approach is also practical from a training perspective: our results show that state-of-the-art performance can be achieved even with a single training image pair.

The work discussed here could be improved by making feature map filtering end-to-end, with the filters learnt by back-propagation. Normally a hard assignment of filtering is not differentiable, however, a soft filtering approach could be applied when training the filter. Further work will also investigate the use of feature map filtering to improve object detection and image classification. If an early layer can be filtered for the benefit of a later convolutional layer, or even a fully-connected layer, then it stands to reason that a filter could be learnt to optimize the final classifier output.

REFERENCES

[1] A. Krizhevsky, I. Sutskever *et al.*, "ImageNet classification with deep convolutional neural networks," in *Advances In Neural Information Processing Systems*, vol. 2, 2012, pp. 1–9.

[2] K. He, X. Zhang *et al.*, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 770–778.

[3] N. Snderhauf, S. Shirazi *et al.*, "On the performance of convnet features for place recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 4297–4304.

[4] R. Arandjelovi, P. Gronat *et al.*, "Netvlad: Cnn architecture for weakly supervised place recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1437–1451, 2018.

[5] S. Garg, N. Suenderhauf *et al.*, "LoST? Appearance-Invariant Place Recognition for Opposite Viewpoints using Visual Semantics," in *Proceedings of Robotics: Science and Systems XIV*, 2018.

[6] Z. Chen, L. Liu *et al.*, "Learning Context Flexible Attention Model for Long-term Visual Place Recognition," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4015–4022, Oct 2018.

[7] J. Guo and M. Potkonjak, "Pruning ConvNets Online for Efficient Specialist Models," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2017-July. IEEE, 2017, pp. 430–437.

[8] J. Zou, T. Rui *et al.*, "Convolutional neural network simplification via feature map pruning," *Computers and Electrical Engineering*, vol. 70, pp. 950–958, 2018.

[9] S. Hausler, A. Jacobson *et al.*, "Feature Map Filtering: Improving Visual Place Recognition with Convolutional Calibration," in *Australasian Conference on Robotics and Automation (ACRA)*, 2018.

[10] S. Garg, N. Suenderhauf *et al.*, "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.

[11] O. Russakovsky, J. Deng *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[12] T.-Y. Lin, M. Maire *et al.*, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla *et al.*, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[13] N. Suenderhauf, S. Shirazi *et al.*, "Place Recognition with ConvNet Landmarks: Viewpoint-Robust, Condition-Robust, Training-Free," in *Robotics: Science and Systems XI*, vol. 11, 2015.

[14] G. Lin, A. Milan *et al.*, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 5168–5177.

[15] Z. Chen, F. Maffra *et al.*, "Only look once, mining distinctive landmarks from ConvNet for visual place recognition," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 9–16.

[16] S. Han, H. Mao *et al.*, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding," *arXiv preprint arXiv:1510.00149*, Oct 2015.

[17] H. Li, A. Kadav *et al.*, "Pruning Filters for Efficient Convnets," in *International Conference on Learning Representations*, 2017.

[18] S. Lin, R. Ji *et al.*, "Towards Compact ConvNets via Structure-Sparsity Regularized Filter Pruning," *arXiV preprint arXiv:1901.07827*, Jan. 2019.

[19] H. Noh, A. Araujo *et al.*, "Large-scale image retrieval with attentive deep local features," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 3476–3485.

[20] J. Mao, X. Hu *et al.*, "Learning to fuse multiscale features for visual place recognition," *IEEE Access*, vol. 7, pp. 5723–5735, 2019.

[21] Z. Chen, A. Jacobson *et al.*, "Deep learning features at scale for visual place recognition," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3223–3230.

[22] A. Loquercio, M. Dymczyk *et al.*, "Efficient descriptor learning for large scale localization," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3170–3177.

[23] 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)*, vol. 1460, pp. 726–735, 1998.

[24] S. Hausler, A. Jacobson *et al.*, "Multi-process fusion: Visual place recognition using multiple image processing methods," *IEEE Robotics and Automation Letters*, 2019.

[25] 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 (ICRA)*, 2012, pp. 1643–1649.

[26] F. Han, X. Yang *et al.*, "SRAL: Shared Representative Appearance Learning for Long-Term Visual Place Recognition," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1172–1179, 2017.

[27] T. Naseer, W. Burgard *et al.*, "Robust Visual Localization Across Seasons," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 289–302, 2018.

[28] W. Maddern, G. Pascoe *et al.*, "1 year, 1000 km: The Oxford RobotCar dataset," *International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.

[29] N. Sünderhauf, P. Neubert *et al.*, "Are we there yet? challenging seqslam on a 3000 km journey across all four seasons," in *Workshop on Long-Term Autonomy at ICRA 2013*, 2013.