Convolutional Neural Networks for the Representation of Natural Scenes with Large Seasonal Variations

BMVC 2016 Submission # 202

Abstract

This paper focuses on the evaluation of deep convolutional neural networks for the analysis of images of natural scenes subjected to large seasonal variation as well as significant changes of lighting conditions. The context is the development of tools for long-term natural environment monitoring with an autonomous mobile robot.

We report various experiments conducted on a large dataset consisting of a weekly survey of the shore of a small lake over two years using an autonomous surface vessel. This dataset is used first in a place recognition task framed as a classification problem, then in a pose regression task and finally the internal features learned by the network are evaluated for their representation power.

All our results are based on the Caffe library and default network structures where possible.

1 Introduction

Long-term natural environment monitoring using visual inspection is the process of collecting images of an outdoor landscape over a long duration with respect to the natural dynamics of this environment. In our case, we are specifically considering the weekly observation of a lake shore over multiple years using an autonomous boat programmed to follow the shore at a constant distance while recording images. As such, our images depict scenes which are combinations of close-up trees, far-away trees, bushes, lawns, water and sky, with a high

Mar. 14

June 25

Aug. 12

Figure 1: Examples of variation in appearance of a section of the lake shore from winter to summer. The significant variation in the vegetation and lighting conditions makes place recognition particularly challenging.

level of similarities. Because we work in a natural setting, this environment is subjected to seasonal changes (trees blooming, leaves falling, ...), structural changes (cut branch, fallen trees, mowed lawns) and weather variations impacting the lighting condition, spectrum and incidence. Figure 1 gives an example of a relatively easy group of images in our dataset.

To assess the difficulty of interpreting these images, in a previous work, we evaluated the time required by a human to decide if two images correspond to the same place albeit at different time of the year. For some image pairs, our test subjects took more than 30 seconds to confirm their answer.

One of the challenge of natural environment monitoring is to be able to compare the appearance of vastly varying outdoor settings with the purpose of, at least, detecting unnatural changes (intruders, structural damages, ...) and, ideally, to classify changes according to their nature.

In this context, this paper focuses on the evaluation of deep convolutional neural networks (CNN, [7]) applied to images of natural environments subjected to large seasonal variations. The task we set ourselves has three stages. In a first stage, we evaluate the ability of a CNN to recognize a place independently of the seasonal and lighting changes. This problem is framed as a classification task where each class is a location around our lake shore and a standard network structure can be used. In a second stage, we now consider a CNN trained to predict the pose (or view point) of the camera (2D position and heading) using an adapted network structure suitable for a regression task. Finally, in a third stage, we evaluate the quality of the internal representation learned by the convolutional layers of the CNN to describe a place independently of its seasonal appearance as well as the generalizability of the resulting features. This third stage is important for the potential of this internal representation to be used to detect changes with respect to a season-invariant representation of a place.

In summary, this paper contribution is two-fold. First, it introduces a relatively unique long-term natural environment monitoring dataset to the computer vision community. Secondly and more importantly, this paper is a benchmark on the performance of CNN for the very particular task of processing images of natural environments subjected to large seasonal changes and natural lighting conditions variation.

All the results presented in this paper have been trained using the Caffe library [9] with a default network structure where possible.

2 Dataset and Experimental Setup

As mentioned earlier, this paper based its evaluation of CNN for natural environment on a particular dataset we have been creating since August 2013. Since then, every 8 to 15 days, we operate an autonomous surface vessel (Kingfisher from Clearpath Robotics, see fig. 2) around a small lake next to our campus. This lake is 400m long by 200m wide, with a small island and a total perimeter of 1km. Its shores are covered with trees, from small bushes to tall full-grown trees, some at the water line, others further away, grass areas are mixed with the shrubberies and a small scenic trail runs around the lake. Some of the places (as in fig. 1) have office buildings in the background.

Every survey we collect contains images acquired by a pan-tilt surveillance camera (704×480 pixels) at 10Hz with a slight JPEG compression. The boat runs autonomously at a constant distance of 10m to the shore (lattice-type local planner) and at a bit less than 1km/h. This means that a survey is a collection of close to 40'000 images, acquired with the camera

Figure 2: The Kingfisher on a very smooth lake. The pan-tilt camera is housed in the white dome at the back of the boat, just behind the laser scanner used for navigation.

pointing to the port or starboard side of the boat (i.e. $\pm 90^o$ from the direction of travel). In addition to the images, we record all the boat sensor data: position from GPS, heading from compass, pitch and roll angle from IMU although they can be neglected and proximetry from the laser range finder (not used beyond the on-board controller in this study).

In this study, we are considering 80 surveys from the second half of 2013 up to the end of 2015. This corresponds to potentially a bit more than 3'000'000 images collected over 80km of autonomous navigation.

The particularity of our dataset is that most of our images depicts natural scenes combining some water, trees and shrubberies at various distances, sometimes grass areas and/or far-away buildings, and sky. All of these elements are challenging for computer vision: the lake surface acts as a somewhat deformable mirror, sometimes very smooth and reflective and at other times not reflective at all due to wavelets. Additionally, flooding events means that the water line can move by up to 1m in some surveys. Trees are challenging for three reasons, first these ones do not always have leaves, second they are fractal self-similar structures, and last they are 3D semi-transparent structures whose appearance is very sensitive to view point, especially in winter. Finally, the sky varies with the weather and the sun position. Because we run the boat on the perimeter of the lake at different times of the day and as long as it is not raining (to avoid water drops on the camera dome), our images are also sometimes affected by sun-glares or very challenging dynamic range requirements.

3 Related Works

It is becoming well known that the traditional approach to data association, i.e., point–based feature matching, is unreliable in unstructured environments. It is more applicable the more structured an environment is. Point–based features can be associated well indoors, but special care has to be taken as they are applied in urban environments (e.g., street-view) [2, 21]. They lose representational power as the environment changes with night [16], rain [4], and shadows [5]. This means that in some natural environments, like lake shores, point–based feature matching is sporadic even among images from the same survey, and is unreliable between different surveys [8].

The lack of a dominant method for data association in outdoor environments has led to a number of new approaches. All of them function using some form of information beyond the capabilities of point–based features. Image sequence [3, 6, 13, 14, 15], image patch [11, 24], and whole image [1, 17] techniques are becoming increasingly dependable. There are,

however, still shortcomings among them. A common limitation is robustness to changes in viewpoint among some approaches based on sequences or whole images. This may not be a factor in monitoring applications, however, since surveys are captured from similar trajectories; the viewpoint and the scale are relatively stable between images (see e.g., [12]).

In the recent years, deep neural networks have become very popular methods for solving both classification and regression problems because technical difficulties related to their training have been overcome. In the context of image analysis, convolutional neural networks have been around for several decades because they benefit from inherent regularities in images to constrain the trained architecture and their architecture regularize more general deep neural networks. In the recent years, state of the art performances were achieved with deep convolutional neural networks on various machine learning tasks in the context of computer vision [10, 20]. Of particular interest for our study, deep convolutional neural networks have been successfully applied to place recognition [23] (a classification task), pose regression [18] and viewpoint estimation [22]. Finding the best neural network architecture for solving a given machine learning problem can be very challenging. In this study, we consider the CaffeNet architecture which is an implementation in Caffe[9] of the AlexNet convolutional neural network[10]. This network had state of the art classification accuracy on the ImageNet Large Scale Visual Recognition Challenge.

4 Methodology

4.1 Data Pre-Processing

The dataset consists in a bit more than 3'000'000 images collected during 80 surveys between the second half of 2013 up to the end of 2015. We consider two experimental setups: a classification task and a regression task. For the classification task, a label is affected to each image based on the pose of the robot. The pose, consisting in the position (from the GPS) and the heading (from the compass) of the robot, is discretized. The position of the robot is discretized into 2.5 meters squares positioned around the lake on a 350 m by 600 m grid centered on a reference point at the center of the lake. The heading is discretized in non overlapping angular sectors of 10 degrees. This led to a total of 1'209'600 possible classes.Only a fraction of those possible classes represents images from our dataset and some of the obtained classes were underrepresented. Therefore the dataset was sub-sampled to ensure the balance of the classes. Namely, we kept only the classes with a number of images at least 50% of the largest class, containing 1750 images. This represents a total of 295 classes with approximately 1'000 images for most classes for a total of 300'000 images on the training set and 5000 images selected randomly for the testing set. For the regression task, the same set of images was used and the labels were defined from the pose of the robot. One possibility would have been to use the position and heading of the robot as labels but this would imply to define a specific loss taking into account the angular nature of the heading. Although feasible in Caffe, this requires an in-depth modification of the library. Instead, we decided to use an Euclidean loss and therefore defined the labels as a four-component vector with the position of the robot (from GPS) and the position of the point 10 meters away from the robot along the optical axis of the camera. One potential drawback of this approach is that the regression problem becomes more complicated than if we were to predict the position and heading since it requires the regressor to predict a specific location along the heading. However it turns out that despite this constraint, the regressor performed reasonably well

(see section 5.2). Finally, in order to ease the definition of the learning rate and to speed up convergence of learning, the labels to be regressed are normalized and centered.

4.2 Convolutional neural network architecture and training

In this study, we used the AlexNet convolutional neural architecture[10] trained in Caffe[9]. The network consists in five convolution layers, five pooling layers, seven rectified linear unit layers, two normalization layers and three fully connected layers. Minor modifications were required to train successfully the AlexNet network. For the classification task, the size of the mini-batches is decreased to 32 samples and learning rate was decreased at a regular rate 10 times throughout training. The output layer of the fully connected part of the architecture uses a soft-max transfer function to get a probability distribution over the labels and the loss is the cross-entropy classification loss. For the regression task, a linear output transfer function is considered and the loss is the Euclidean loss. Experimentally, it was required to consider a lower learning rate than AlexNet which otherwise lead to a divergent loss. As we shall see in the result section, several strategies for setting the learning rate are considered. For both the regression and classification problems, the architecture is trained with the default CaffeNet settings, namely stochastic gradient descent, a momentum set to 0.9 and a weight decay to 0.0005.

4.3 Extracting season invariant representations

Being able to classify an image as belonging to one part of the lake with its viewpoint or to regress from it the pose of the boat are of interest by themselves. However, one of the objectives of the study was also to extract season invariant representations in order to detect the changes of the lake shore. This part of the study was done only from the network trained in the classification task. For every class of the selected dataset, a prototypical image was computed by averaging all the filter responses, at a given depth of the network, of all the images belonging to the considered class. A query image is then propagated through the network up to the depth where the prototypes have been computed, the responses of all the filters at that depth are then averaged and this representation is compared to the computed prototypes. The quality of the prototype images is then assessed from the cosine similarity between the representation of the query image and the prototypes; labeling a new image is then performed by picking the class whose prototype has the largest similarity with the averaged representation of an image.

5 Results

5.1 Classification

Our best training on the aforementioned dataset was made with 300 000 iterations over mini-batches of size 32 (down from the default 256 for better accuracy), with a learning rate decreasing at a fixed rate ten times during the course of training. The full training took 5 days to finish on a Tesla K20C machine, and attained 70% accuracy on the test set, on a top-1 classification basis. Such results are satisfactory considering the similarities of natural scene images. It should be noted that classification was not the main goal, but a good classification will intuitively lead to better class representations.

We can thus extract the trained filters responses to see how images are processed by the network. The `conv1` layer will mostly learn edges, namely the skyline and the waterline. `Conv2` detects foliage, and convolutional layers 3 through 5 contain low-level features that are harder to interpret properly. We tested prototype generation on all convolutional layers, as well as the last pooling layer `pool5`, shown in fig. 3.

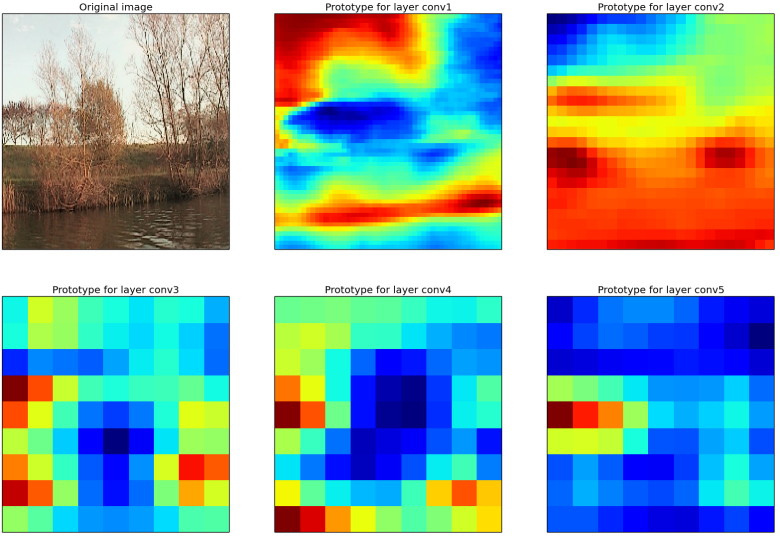

Figure 3: Example of prototypes for a given image

We generated prototypes for all classes, and tested whether an image can be recognized only using its class prototypes. Testing over a thousand random images and measuring using a cosine distance shows that all layers can be used as suitable descriptors, with distances to the wrong prototypes being indubitably larger than distances to the right prototype on average (see table 1, column 2 and 3 and fig. 4)

Layer	Overall Ratio (complete dataset)	Precision (%) top-20	Ratio (seen) (1^{st} half dataset)	Ratio (unseen) (2^{nd} half dataset)
`conv1`	1.76	43.9%	1.70	1.03
`conv2`	2.99	42.3%	0.93	0.96
`conv3`	1.86	34.8%	1.40	0.94
`conv4`	1.79	08.4%	1.16	1.00
`conv5`	1.68	57.3%	1.32	0.98
`pool5`	1.86	52.0%	1.42	0.95

Table 1: Ratio of median distance of a random image to the wrong prototypes over median distance to the correct prototype. 2^{nd} column refers to the ratio achieved using the complete dataset for training. 3^{rd} column give the percentage of successful top-20 classification using the distance to the prototypes of the full dataset. 4^{th} column is similar but using only half of the classes. 5^{th} column evaluate the generalization performance by evaluating images from the classes not used for training.

We also trained the same network on half the classes, to test for generalization capabilities. We wanted to test whether the network learned how to transform an image into its seasonal-invariant representation. In the case of the classes observed in the training set, the

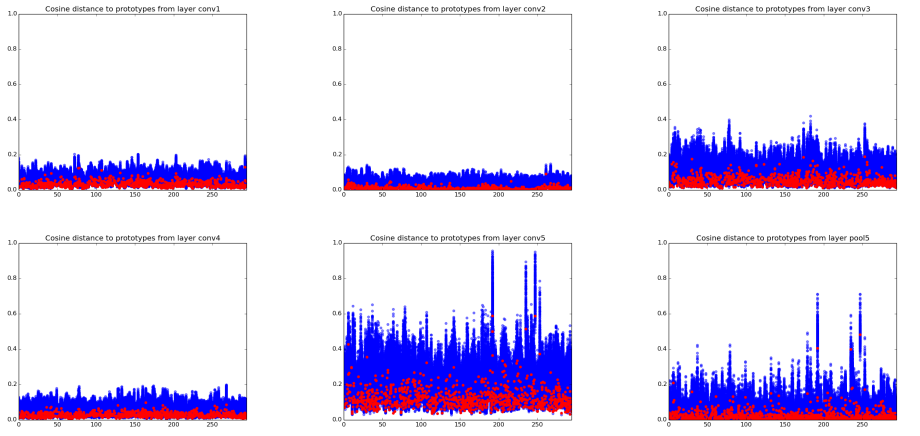

Figure 4: Red dots represent the distance from a random class image to its class prototype, blue dots are distances to other class prototypes. Graphs show layers conv1 through pool5.

representation results are analogous to the full dataset. However, the internal features are not discriminative when applied on images from unseen classes (table 1, column 4 and 5). It seems that the network learns how to efficiently discriminate between its known classes, rather than truly learning seasonal variations.

5.2 Regression

The objective of the regression task is to perform localization with performance comparable to GPS system. The parameters of the training had to be tuned for the network to perform at its best for the regression task.

The layers of the network are initialized using normal distributions. The initialization variances had to be changed from 0.01 to 0.1 for Conv1 and to 0.05 for Conv2, Conv3, Conv4 and Conv5, so that the weights are big enough to propagate information through the network while still ensuring convergence. The initial learning rate and its evolution policy heavily depends on the loss which is used. For the regression task, very high values and a diverging behaviour were observed with the Euclidean loss at the beginning of the training. The learning rate on the last fully connected layer had to be decreased from 0.01 to 5e-04 and the weight decay from 5e-4 to 2.5e-6 to avoid these effects. The learning rate policy was set to the "step" policy from Caffe and it was chosen to decrease the learning rate by half every 25 000 iterations. In our case this corresponds to the number of iterations required to observe the stabilization of the loss after each exponential decrease.

In this context, we tested three different approaches. The first one consists in training every convolution layer with the same learning rate. The second one consists in fine-tuning the learning rate of the Conv1 layer based on the results given by the first approach. The third one consists in loading the weights of the convolution layers from the classification training. In order to compare our results with those three approaches, we refer to the following loss function :

$$Loss = \frac{0.5}{scale^2} * \sum_{i=1,\,4} (label_i - prediction_i)^2 \quad (1)$$

The first approach allowed to achieve an average error of 19.3 meters per label as it is shown in fig. 5.a. The loss computed on the test dataset is plotted in green and the loss computed on the train dataset is plotted in blue. However after the training, the convolution layers filters did not exhibit any specific features (fig. 6.a). Thus, it appears that the regression was only supported by the fully connected layers. In our case where the goal is to build a season-invariant representation of natural scenes this approach did not reach our expectations.

The second approach led us to push further the difference of behavior between the fully-connected layers and the convolution layers. By increasing the learning rate for Conv1 from 0.01 to 0.02 and doubling its weight decay, we forced the convolution layers to take part in the regression. This method resulted in being more successful than the previous one. The best average error we achieved was 18.3 meters (fig. 5.b). Some natural environment features can be identified in the convolution filters (fig. 6.b).

The last approach used the same learning rate settings as the first approach. It was observed that the convolution weights decreased during the training and ended with a distribution similar to the second approach. The best average error achieved was 20.9 (fig. 5.c). However the convolution filters retrieved exhibited different structures than the second approach (fig. 6.c). Based on this result it can be concluded that the convolution weights learned during the classification training could not be reused for the regression task. The weights needed for this task required a smaller variance and the final model presented the worst results among the three approaches. Consequently, learning from normally distributed convolutional gains seems to be more efficient in the case of a regression.

a) First approach b) Second approach c) Third approach

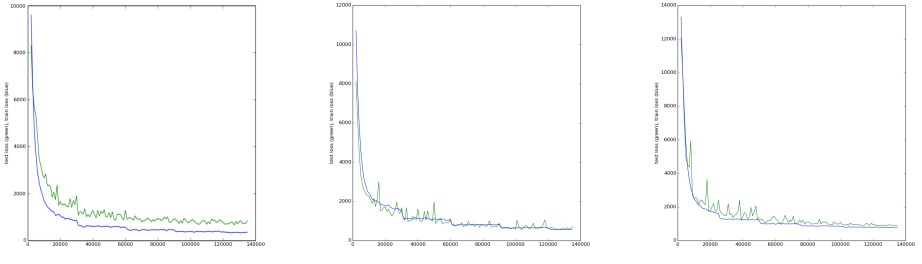

Figure 5: Loss Function for 135 000 iterations.

a) First approach b) Second approach c) Third approach

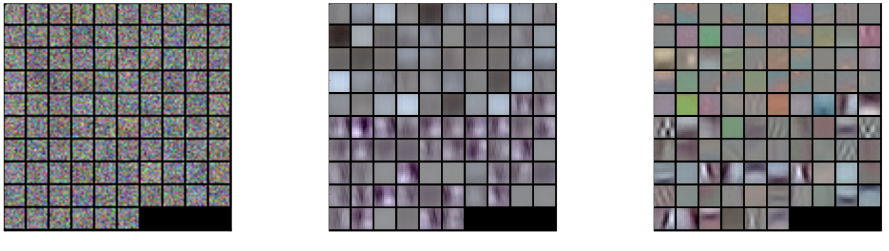

Figure 6: Conv1 filters after 135 000 iterations.

The best results regarding the loss values on the train and test datasets were achieved with the second approach. After 150000 iterations, the prediction errors are centered Gaussian-like distributions with a standard deviation of 11.8 meters on X and 21 meters on Y. Considering that performing this operation with the human brain is found to be challenging for images from a natural environment, the resulting precision is very acceptable. The labels were displayed to be compared to the predicted positions (fig. 7.a and b). The red and green dots represent the original labels and the blue and purple dots represent the predicted labels. The error between X and Y coordinates of the predicted and original labels was also plotted on fig. 7.c and was found to be centered on the origin without significant bias.

a) Map of original labels b) Map of predicted labels c) 2D error of the predicted label position

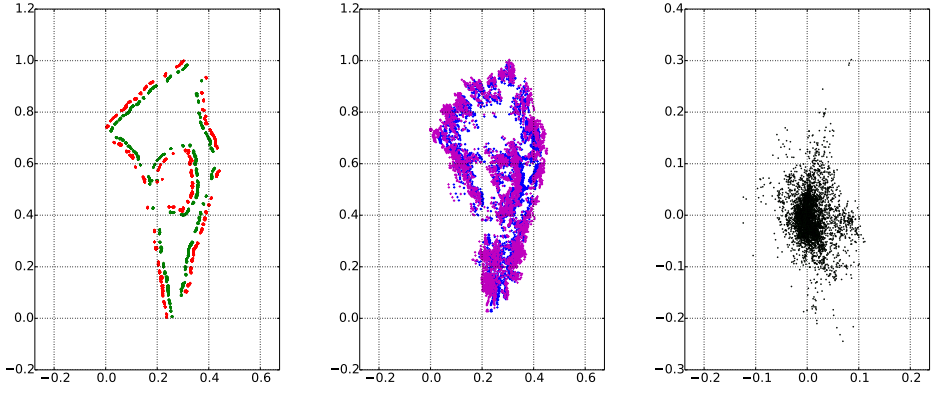

Figure 7: Map of predicted positions and original labels.

6 Conclusions

This paper evaluated the performance of convolutional neural network on a place recognition task and a pose prediction task for natural environments under large seasonal changes, in the context of a long-term autonomous monitoring problem. To this end, we presented an original dataset consisting in several million images taken at weekly interval on the shore of a small lake over two years.

Water and sky appearance inconsistency, as well as the strong seasonal changes of vegetation and the weather-dependent lighting conditions proved to be manageable both for the classification task (70% precision) and for the pose regression task (20m standard deviation over 1km of shore line). However, it turned out that using the standard network architecture did not result in learning generalizable features leading to a season-invariant representation of the environment. Turning towards more general network architectures as in [19] would probably be appropriate but it turned out that Caffe was too limited to explore this possibility within this study.

References

[1] Roberto Arroyo, Pablo F Alcantarilla, Luis M Bergasa, and Eduardo Romera. Towards life-long visual localization using an efficient matching of binary sequences from images. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 6328–6335. IEEE, 2015.

[2] Chris Beall and Frank Dellaert. Appearance-based localization across seasons in a Metric Map. In *6th PPNIV*, Chicago, USA, September 2014.

[3] Winston Churchill and Paul Newman. Experience-based navigation for long-term localisation. *IJRR*, 32(14):1645–1661, 2013.

[4] Aurélien Cord and Nicolas Gimonet. Detecting unfocused raindrops: In-vehicle multipurpose cameras. *Robotics & Automation Magazine, IEEE*, 21(1):49–56, 2014.

[5] Peter Corke, Rohan Paul, Winston Churchill, and Paul Newman. Dealing with shadows: Capturing intrinsic scene appearance for image-based outdoor localization. In *IROS*, pages 2085–2092. IEEE, 2013.

[6] 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.

[7] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, August 2013.

[8] Shane Griffith, Paul Drews, and Cédric Pradalier. Towards autonomous lakeshore monitoring. In *International Symposium on Experimental Robotics (ISER)*, 2014.

[9] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[11] Colin McManus, Ben Upcroft, and Paul Newman. Scene signatures: Localized and point-less features for localization. In *RSS*, Berkeley, USA, July 2014.

[12] Michael Milford, Jennifer Firn, James Beattie, Adam Jacobson, Edward Pepperell, Eugene Mason, Michael Kimlin, and Matthew Dunbabin. Automated sensory data alignment for environmental and epidermal change monitoring. In *Australasian Conference on Robotics and Automation 2014*, pages 1–10. Australian Robotic and Automation Association, 2014.

[13] Michael J Milford and Gordon 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*, pages 1643–1649. IEEE, 2012.

[14] Michael J Milford, Gordon F Wyeth, and DF Rasser. Ratslam: a hippocampal model for simultaneous localization and mapping. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 1, pages 403–408. IEEE, 2004.

[15] Tayyab Naseer, Michael Ruhnke, Cyrill Stachniss, Luciano Spinello, and Wolfram Burgard. Robust visual slam across seasons. In *IROS*, 2015.

[16] Peter Nelson, Winston Churchill, Ingmar Posner, and Paul Newman. From Dusk till Dawn: Localisation at Night using Artificial Light Sources. In *ICRA*, 2015.

[17] Peer Neubert, Niko Sünderhauf, and Peter Protzel. Superpixel-based appearance change prediction for long-term navigation across seasons. *Robotics and Autonomous Systems*, 69:15–27, 2015.

[18] Tomas Pfister, Karen Simonyan, James Charles, and Andrew Zisserman. Deep Convolutional Neural Networks for Efficient Pose Estimation in Gesture Videos. In Daniel Cremers, Ian D. ReidZ, Hideo Saito, and Ming-Hsuan Yang, editors, *ACCV (1)*, volume 9003 of *Lecture Notes in Computer Science*, pages 538–552. Springer, 2014. ISBN 978-3-319-16864-7.

[19] A. Radford, L. Metz, and S. Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv preprint arXiv:1511.06434*, 2016.

[20] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556, 2014.

[21] Elena Stumm, Christopher Mei, and Simon Lacroix. Probabilistic place recognition with covisibility maps. In *Intelligent Robots and Systems (IROS)*, pages 4158–4163. IEEE, 2013.

[22] Hao Su, Charles R. Qi, Yangyan Li, and Leonidas J. Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[23] Niko Sünderhauf, Feras Dayoub, Sareh Shirazi, Ben Upcroft, and Michael Milford. On the Performance of ConvNet Features for Place Recognition. *CoRR*, abs/1501.04158, 2015.

[24] Niko Sunderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft, and Michael Milford. Place recognition with ConvNet landmarks: Viewpoint-robust, condition-robust, training-free. *Proceedings of Robotics: Science and Systems XII*, 2015.