Learning Long-Term Invariant Features for Vision-Based Localization

Niluthpol C Mithun[1], Cody Simons[1], Robert Casey[2], Stefan Hilligardt[2] and Amit Roy-Chowdhury[1]
[1] University of California, Riverside, CA [2] VWGoA Electronics Research Lab, Belmont, CA
{nmithun@ece., csimo005@, amitrc@ece.}ucr.edu, robert.casey@samsung.com, stefan.hilligardt@nio.io*

Abstract

Constructing a feature representation invariant to certain types of geometric and photometric transformations is of significant importance in many computer vision applications. In spite of significant effort, developing invariant feature representations remains a challenging problem. Most of the existing representations often fail to satisfy the long-term repeatability requirements of specific applications like vision-based localization, applications whose domain includes significant, non-uniform illumination and environmental changes. To these ends, we explore the use of natural image pairs (i.e. images captured of the same location but at different times) as an additional source of supervision to generate an improved feature representation for the task of vision-based localization. Specifically, we resort to training deep denoising autoencoder, with CNN feature representation of one image in the pair being treated as a noisy version of the other. The resulting system thereby learns localization features which are both discriminative and invariant to illumination and environmental changes. In experiments tailored towards vision-based localization, features generated using the proposed method produced higher matching rates than state-of-the-art image features.

1. Introduction

Capturing good visual representations from images is of crucial importance in computer vision. Traditionally, a number of hand-engineered feature extraction methods such as SIFT [30], HOG [12] and SURF [7] have been developed to find distinctive features that could be considered an underlying visual representation. Over the last few years, there has been significant interest in learning rich visual representation directly using deep neural networks [53] leveraging massive amounts of labeled data. Many feature extraction methods aim to achieve a certain level of invariance to different transformations, such as affine and lighting changes. Operations like orientation histogram binning in SIFT/HOG and pooling and convolution operations in CNNs are designed to achieve invariance to small affine transformations [21]. Data augmentation is another commonly used technique with feature learning which can achieve a limited degree of invariance to certain transformations. However, most of these representations fail to satisfy long-term repeatability requirements of specific applications like vision-based localization.

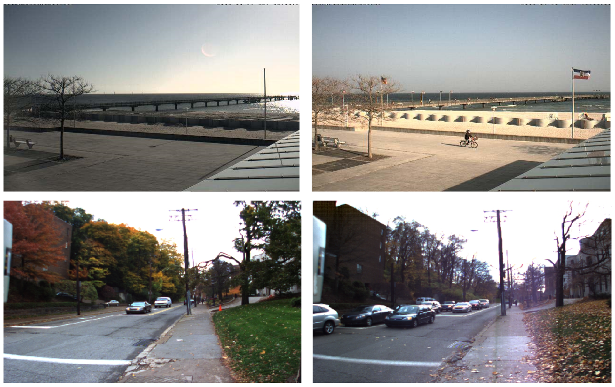


Figure 1: Each row represents the same place imaged at two different times. The goal is to identify features in the images that would allow us to determine with high confidence that these are pictures of the same location.

Vision-based localization approaches usually attempt to provide a long-term and cost-effective alternative modality to GPS [5, 8]. These localization systems usually follow a well-defined pipeline [31]. First, the perception system constructs an environmental map based on the unique, or at least, distinctive features visible at various locations of the route to be mapped. Thus, the distinctive features become associated with distinctive locations. Next, when a new image is presented to the system, the system tries to match the features associated with this location-coupled image with the database (i.e. map) of previously detected image features. The challenges of this task are many and include viewpoint changes, dynamic aspects of the environment (pedestrians, vehicles, etc.), sensor and model noise, changes due to the seasons, and illumination variance

*This work was performed when Robert Casey (currently with Samsung) and Stefan Hilligardt (currently with NIO) were with Volkswagen Group of America, Belmont, CA, USA

(Fig.1). Furthermore, many locations might not have visually distinctive features. Hence, existing feature descriptors fail to provide a level of *long-term* invariance to environmental factors, like illumination, necessary for widespread application in tasks such as location re-identification. This limits the applicability of such approaches to fair-weather scenes during a limited time window, typically during the day.

Ideally, we wish to generate a feature descriptor, such as through the discovery of common patterns in images, which is invariant to environmental changes, like illumination, that happen over extended time periods, like different times of the day. The complete space of transformations between images taken at the same location but at different times is unknown and likely non-uniform. Hence it is likely intractable to develop physics-based approaches which can accurately model such changes. There is evidence to suggest that humans typically re-identify the same location at a different time by discovering invariant patterns in the scene as well as models of how the patterns change over time [6] based on experience. We pose the question as to whether a learning-based method can be used to develop the same.

Building upon the successes of deep learning methods for feature extraction in images, we propose the use of a stacked autoencoder [50, 51] for solving this problem. We aim to model the complex non-linear transform between the images of same location through a stacked autoencoder, considering its ability to handle non-linear transforms. By considering one of the images at a particular location as a noisy version of the other at the same location, our approach can be framed in the context of a denoising autoencoder (DA), which takes a partially corrupted input whilst training to recover the original undistorted input. While in a standard DA, a predefined stochastic corruption process is applied on the input data to generate the corrupted version of input, we assume there is an unknown non-linear stochastic mapping that exists between the images of same location. By training on such corresponding data pairs, the model tries to learn discriminating features for the target task and is able to gain invariance to meaningful transformations (e.g. illumination, occlusions, sky appearance, etc). This model encodes the simplifying assumption that the appearance of different locations changes uniformly, regardless of the environment, and thus changes learned during training can be generalized to previously unseen locations.

This paper also introduces a dataset, which contains image pairs captured by driving through urban and suburban environments during different times of day; we shall refer to this dataset as the Localization in Changing Environment (LCE) dataset. We train our models by combining multiple datasets, including LCE dataset, CMU Visual localization (CMU VL) dataset [5], Webcam Clip Art database(WebcamDB) [25] and Phos dataset [52]. Experiments demonstrate the effectiveness of our proposed approach in illumination invariant feature extraction. Our learned model is not only generalizable across these datasets, but also shows excellent cross-dataset generalization performance in other held-out localization datasets, i.e., St Lucia [18], Nordland [44], Oxford Robocar [32] datasets. Hence, this work could be a precursor to semantic localization or a pre-processing step for finer-grade localization.

2. Related Work

Vision-based Localization. The authors in [31] and [16] offer comprehensive reviews on state-of-the-art approaches to vision-based localization. Vision-based methods usually localize [3, 20, 29, 41] or categorize [38, 40, 54] an image given a database of geo-referenced images or video streams [5, 28, 2]. Significant work has been done in recent years towards developing hand-designed feature descriptor-based localization systems [11, 26]. However, It has been observed that SURF and SIFT features [15, 49] are not robust to long-term and non-uniform illumination changes. On the contrary, global image descriptors, like GIST suffer from high sensitivity to viewpoint change [31, 35]. Features learned from deep neural networks have recently been used as robust feature extractors for visual localization. Motivated by their ability to learn generic features that are transferable to a variety of related but different visual tasks [42, 37], the authors of [10, 45, 46] utilized CNN features as holistic image descriptors and analyzed the robustness of different layers against visual appearance and viewpoint changes. They concluded that mid-level features are relatively more robust against change in appearance, while higher level features exhibit robustness against viewpoint changes [45]. Hence, the pre-trained CNN features may not be optimal for vision-based localization, as they are not specifically designed for the task. In contrast to existing works, which hinge on existing features to develop a visual localization method, our feature extraction method is tailored to achieve a high level of invariance for vision-based localization task.

Learning Correspondence. Most relevant to our work are methods which train with pairs of transformed images and infer an implicit representation for the transformation itself [21, 33, 34, 39]. In [33], to encode transformation between image pairs, bilinear models have been used and content independent motion feature was learned. The gated autoencoder has been turned into a recurrent network for sequence prediction in video in [34]. To predict future frames, recurrent neural networks have been demonstrated in [39]. These methods mainly aim to relate two temporally or spatially related images, however we aim to learn a representation for individual images which is invariant to common changes in a scene over time. Other relevant works include [1, 9], in which features have been learned for the place recognition task. [9] considered place recognition as a classification problem and trained a CNN with thousands of images of a number of specific places. Another related work is [1], which designed a layer named NetVLAD and trained with CNN feature extraction in a relevant task of place recognition. This approach has shown good performance in recognition tasks. However, the uncertainty in the localization that these methods can tolerate is much higher (tens of meters) than our application (visual localization) permits, whereas more useful results occur at the decimeter level of accuracy. Place recognition systems usually assume that the places in the map are disjoint and their views do not overlap. Hence, feature extracted from these networks may not be optimal for visual localization task.

3. Methodology

Overview. Fig 3 summarizes our proposed framework for illumination-invariant feature learning and the framework for using these learned features in a localization task. In our framework, we try to learn a feature representation which is invariant to appearance changes of a scene at different times. We assume that the training set is an exhaustive collection of possible changes in illumination conditions and not dependent on specific locations; thus, features learned during training can be generalized to previously unseen locations. The main idea is to train a DA by trying to recover the visual representation of one frame from another, which was captured at the same location but at a different time.

Once training has been completed, the trained network can then be applied to feature extraction tasks. We assume the database contains feature representation for every location along the route captured by the camera. When a new image arrives, feature representation for this sample is extracted using our trained network. Then, this feature is compared to features stored in the database to generate putative matches. In case of time-sequence data, a short time window around each location candidate is analyzed to select that match with the highest temporal consistency. Otherwise, the image with highest similarity is selected.

3.1. Feature Learning using Denoising Autoencoder

Basics of the DA. An autoencoder is a feed-forward neural network used for unsupervised learning of efficient representation for a set of data. A classical autoencoder takes the input x and maps it (with encoder) onto a hidden representation $h(x)$ through a deterministic mapping. $h(x)$ is then mapped onto the reconstruction $\hat{x}$ of the same shape as x. Then, the loss function $L(x, \hat{x})$ of the autoencoder compares the output $\hat{x}$ with the initial input x. Denoising autoencoder has been introduced with a specific approach to learn richer representation from data. In a denoising autoencoder, a preliminary stochastic mapping $x \rightarrow \tilde{x}$ is performed to corrupt the data. $\tilde{x}$ is used as input for the autoencoder (Eq.1). The reconstruction $\hat{x}$ is computed from corrupted input $\tilde{x}$ (Eq.2). However, the loss function still compares $\hat{x}$ with the initial noiseless input x, $L(x, \hat{x})$, instead of $L(\tilde{x}, \hat{x})$. Specifically,

$$h(\tilde{x}) = \sigma(a(\tilde{x})) = \sigma(W\tilde{x} + b), \tag{1}$$

Here, σ, W and b are parameters of encoder, where σ is element-wise activation function, W is the weight matrix, and b is the bias. Also,

$$\hat{x} = \bar{\sigma}(\bar{W} * h(\tilde{x}) + \bar{b}), \tag{2}$$

where, $\bar{\sigma}$, $\bar{W}$ and $\bar{b}$ are parameters of decoder.

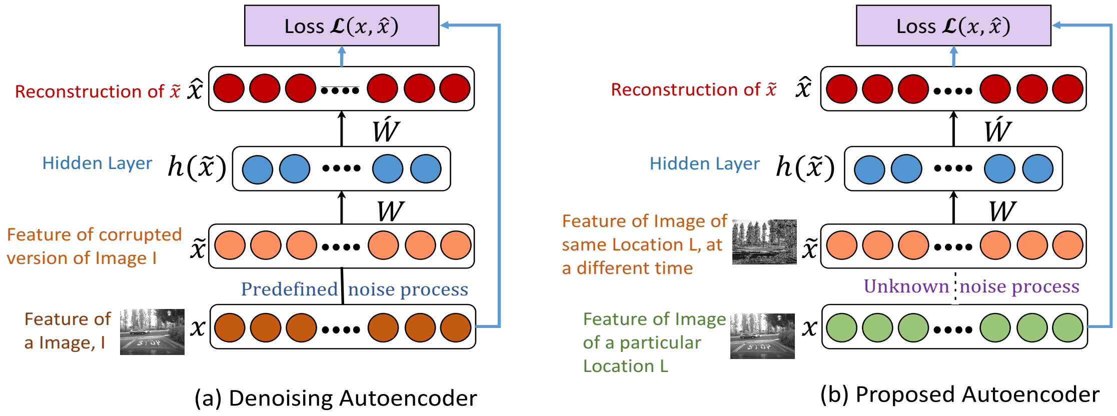
As we want to learn a shared representation between images of the same location, the objective function of the autoencoder is defined by the Euclidean loss of input feature (x_i) and the reconstructed feature ($\hat{x_i}$) with an $L2$ regularization term, shown in Eq.3 as

$$\underset{W}{\operatorname{argmin}} \frac{1}{2N} \sum_i \|x_i - \hat{x_i}\|_2^2 + \lambda \|W\|_2^2 \tag{3}$$

where, N is the size of the mini batch and λ is a hyperparameter to balance the loss and the regularization.

Adapting DA to Our Application. The main difference between our proposed autoencoder and a denoising autoencoder is in the process of generating a corrupted version of the input, as shown in Fig.2. In a denoising autoencoder, a predefined stochastic corruption process is applied to the input data to generate the corrupted version of the input. However, we consider that the image of a particular location is a corrupted version of another image of the same location taken at different time. We assume there is an unknown stochastic mapping that exists between these images, which can be thought of as modeling environmental changes such as illumination. Note, however, that there is no need to use any explicit corruption process model. Thus, we can directly use available image pairs (images of same location, but different time) as input and corrupted input. Hence, this formulation allows us to use the same equations as the denoising autoencoder described above.

However, a direct application of DA is intractable due to the high complexity of change between scenes, especially in case of limited number of training samples. We also observed this phenomenon by training convolutional autoencoder using our image pairs. Euclidean loss in image space leads to averaging all the exact positions of details of images of same location and results in a high training loss. We believe that distribution of the details are sufficient for perceptual similarity in feature representation. Hence, instead of directly using images for training autoencoder, we use deep CNN feature extracted on image pairs as input to the autoencoder. This is done since improving a feature representation for achieving desired invariance is more tractable


Loss $\mathcal{L}(x,\hat{x})$
Reconstruction of $\tilde{x}$ $\hat{x}$
$\acute{W}$
Hidden Layer $h(\tilde{x})$
W
Feature of corrupted version of Image I $\tilde{x}$
Predefined noise process
Feature of a Image, I x
Loss $\mathcal{L}(x,\hat{x})$
Reconstruction of $\tilde{x}$ $\hat{x}$
$\acute{W}$
Hidden Layer $h(\tilde{x})$
W
Feature of Image of same Location L, at a different time $\tilde{x}$
Unknown noise process
Feature of Image of a particular Location L x

(a) Denoising Autoencoder

(b) Proposed Autoencoder

Figure 2: Figure illustrates basic differences between our proposed autoencoder training and traditional denoising autoencoder training.

than trying to learn long-term invariant feature representation directly from scratch. We believe that invariance to irrelevant transformations can be achieved by measuring distances in a suitable feature space.

We apply a deep CNN to extract initial feature representation, as they are currently the state-of-the-art image features [27, 24]. The best performance among CNN models has been achieved by very deep networks with a large number of layers [43, 47], but the processing time per image in such networks can be very high. Hence, it is important to choose a CNN model based on both accuracy and scalability. It is worth mentioning that our method does not depend on this particular choice of CNN, except for the scalability issue mentioned above.

Moreover, which layer of the CNN to be chosen for initial feature extraction is also a design choice, as the mid-level features are more robust against appearance changes, while higher level features are more robust against changes in viewpoint [45]. We tried features from different layers of three pre-trained CNN, i.e., Alex-Net [24], VGG-16 [43] and VGG-19 [43] for initial feature extraction. Empirically, based on accuracy and computational time, we finally chose to use the last pooling layer (pool-5) feature of Alex-Net as the input to the autoencoder. For an image i at the input layer, we primarily extract feature x_i ($x_i \in R^d$, $d = 256 \times 6 \times 6$) from pool5 layer of the CNN and flatten it to construct a feature vector of size 9216.

We propose to use a stacked denoising autoencoder with an architecture similar to that shown in Fig.3. Our autoencoder network takes the CNN feature of an image as the input to an encoder and a decoder sequentially. The decoder output is compared with 9216 dimensional pool5 layer CNN feature of its paired image. The encoder has two hidden layers with 4500 and 2500 neurons, respectively, whereas the decoder has two layers with 4500 and 9216 neurons, respectively. The small middle layers are for learning compact semantics as well as to retain only invariant descriptors. We scale the input signals of the autoencoder to bound their magnitude between 0 and 1.

3.2. Image Matching

For visual localization task, given a query image, we need to find the most similar image from a collection of referenced images or video streams. Hence, it is important to choose a criterion for determining the similarity between two samples.

Similarity Calculation. There are several metrics to calculate the similarity between two samples. Here, we use a Gaussian kernel similarity function. First, we normalize feature vectors to have unit L2 norm. Then, given two feature vectors f_q and f_d, their similarity is measured as the score $S_{q,d}$ as follows:

$$S_{q,d} = exp(-\frac{||f_q - f_d||^2}{2\sigma^2}) \tag{4}$$

Here, $||f_q - f_d||^2$ is the squared Euclidean distance between the feature vectors f_q and f_d. We kept the free parameter σ fixed as 1. Note that the higher the value of $S_{q,d}$, the more similar are the two images. Particularly, if the two images are identical, then $S_{q,d}$ is 1. On the other hand, when there is no similarity between images, $S_{q,d}$ will be close to 0.

In order to retrieve the location of a query image I_q, we search the database for the best image match I_d. The database contains image features with corresponding ground-truth locations. More formally, the database consists of set $D = \{D_1, D_2, ...\}$ with components $D_i = (f_{d,i}, l_{d,i})$, where $f_{d,i}$ is the feature descriptor vector and $l_{d,i}$ is the location of the i^{th} database image. The estimated location l_d of the image will be the image with the highest probability within the set of all possible locations:

$$l_d = \underset{d}{\text{argmax}}(S_{q,d}) \tag{5}$$

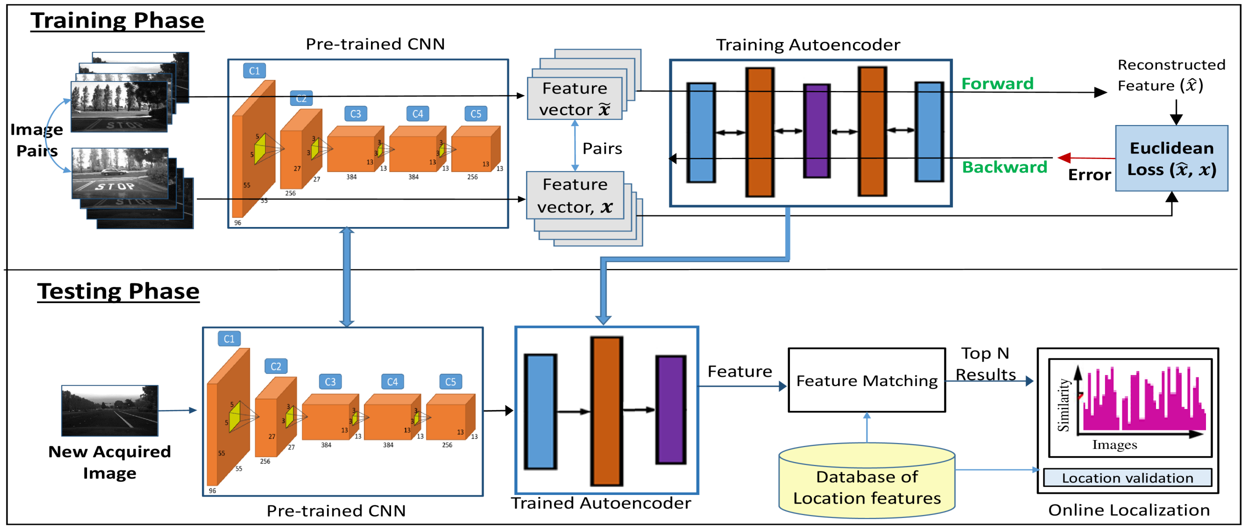


Figure 3: Our proposed feature learning and testing framework. Please see Section 3 for details.

Improving Performance by Sequence Search. For a query sample f_q, finding the database sample f_d with maximum $S_{q,d}$ usually serves as a reliable indicator of a high performance match. However, in many cases, the feature with the highest similarity S may not be from the best matching location. Encouragingly, we can exploit the available, inherent temporal information to achieve an even better result. We present a simple baseline for this approach. In such a case, after the similarity calculation, we perform a validation step with matching image *sequences* rather than individual images to improve recognition accuracy. First, we find top N possible matches based on similarity score to limit the run-time of subsequent validation steps. Then, these matches are verified based on whether they are consistent over a short time interval τ_d (e.g. $\tau_d = 1s$). If this approach returns more than one consistent result, then we choose that matching frame with the highest combined similarity score over the selected time interval. The algorithm for finding the best matching database image using sequence search is presented in Algo.1.

Algorithm 1 Find matching image in time-sequence data

```
Input: Database $D = \{D_1, D_2, ....\}$ with components $D_{\hat{t}} = (f_{d,\hat{t}}, l_{d,\hat{t}})$, where $f_{d,\hat{t}}$ is the feature of database image at relative time $\hat{t}$; Feature, $f_{q,t}$ of query Image at time $t$. Time interval for sequence matching, $\tau_d$
    Step 1: Calculate the similarity scores $S_{q,d}$ between query Image at time $t$ and all images in database.
    Step 2: Select top $N$ samples, with sufficiently large similarity score for further processing.
    Step 3: For each sample chosen in Step 2, a preceding sequence of frames with length of time interval $\tau_d$ is selected from database. Then, these database frames are compared with the frames available in $\tau_d$ time interval, prior to current query frame ($S(f_{q,t}, f_{d,\hat{t}})$, $S(f_{q,t-1}, f_{d,\hat{t}-1})$, $S(f_{q,t-2}, f_{d,\hat{t}-2}), .......$) to verify, whether they are pairwise consistent.
    Step 4: If the matching score is higher than a threshold ($\alpha_h$) for all pairs in the interval, the match is considered consistent. On the other hand, if the matching score is below certain threshold ($\alpha_L$), the candidate is rejected.
    Step 5: In case of multiple consistent match, choose the frame with highest combined similarity score in the selected time interval.
```

4. Experiments

In this section, we present quantitative (Sec. 4.3) and qualitative (Sec. 4.4) results validating our approach. We also provide details of datasets (Sec. 4.1) and implementation details (Sec. 4.2).

4.1. Datasets

We train our model combining collected image pairs from four datasets, e.g., CMU VL [4], WebcamDB [25], PHOS[52] and LCE Dataset. The LCE dataset was created by Volkswagen Electronics Research Lab. The dataset contains about 4K monochrome image pairs under various illumination conditions, which include both urban and suburban areas. We have used monochrome cameras since most production vehicles predominately employ these, instead of color cameras, due to reduced cost. In addition to illumination and environmental variations, there are also differences in the field of view for images captured of the same location. We test our approach on several visual localization datasets, i.e., Nordland [44], St Lucia [18] and Oxford Robocar dataset [32], which was not used in training. The CMU VL, LCE, St Lucia, Nordland and Oxford Robocar dataset are localization datasets. They contain several image sequences for the same route at different times. We have also utilized the PHOS dataset [52] and the WebcamDB dataset [25] for training, which include images captured of the same location using fixed cameras and webcams. Although the latter two datasets are not localization based, they served an important purpose as added sources of training data for fixed-field-of-view scenarios where there is significant visual variation in the environment.

4.2. Training Details

Implementation. Our network architecture consists of a pre-trained convolutional neural network followed by a stacked autoencoder. We use ReLU nonlinear activations throughout the network. The convolutional network module follows the popular AlexNet architecture [24]. We extract initial features from the pooling 5 layer of this CNN, features which then serve as inputs to an autoencoder. Our autoencoder contains 2 encoding and 2 decoding layers. We trained the autoencoder network with objective function listed in Eq. 3 using a stochastic gradient descent with an adaptive subgradient method called AdaGrad [14]. AdaGrad can adapt the rate of gradients based on the previous updates by computing a dimension-wise learning rate. We empirically chose to use AdaGrad after testing with AdaGrad, Adam [23] and RMSProp [48]. The autoencoder layer weights were initialized using the Xavier algorithm [17]. The Xavier initialization technique keeps the signal in a reasonable range of values through layers by automatically determining the initialization scale based on the number of input and output neurons [19]. We used a Tesla K40 GPU and implemented the network in Caffe [22].

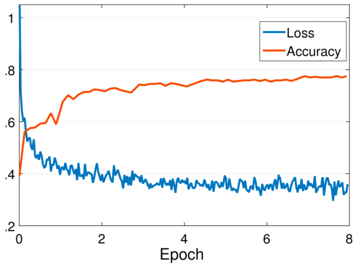


Fig. 4. Training Loss and validation accuracy of models trained on features of pool-5 layer features of our pre-trained CNN layer as a function of optimization epochs. The network converges with a loss of 0.3265 and validation accuracy 0.76.

Optimization. We collect image pairs from four datasets and train our model on feature descriptors of these pairs, as the number of pairs from a single dataset is very low. We follow an 85% / 15% split for training image pairs vs. testing. We also increase the number of training pairs using data augmentation. To this end, we add various levels of noise and vary the image contrast. Fig. 4 shows the learning curve and validation accuracy in training our network as a function of epoch. The network converges at around 8 epoch. We start training with a learning rate of 0.001 and decrease it when the training loss had reached a plateau. For the autoencoder on the input feature pairs, we use minibatches of size 100. We modified the learning procedure to handle our autoencoder training. We added a new accuracy layer, so that we can monitor whether or not the matching performance increased within the validation set. Note that this does not directly evaluate localization accuracy, as this would require comparing all possible locations to find the best matching location. However, our modified accuracy layer allows us to evaluate what percentage of image pairs in validation set has Gaussian similarity greater than a threshold.

4.3. Quantitative Results and Evaluation

Let us now turn to a quantitative verification of the robustness of the proposed feature extraction approach in a visual localization task and a comparison against state-of-the art approaches.

Compared Methods. We compare the performance of features extracted from our network against state-of-the-art methods. We evaluate the performance of our approach against extracted features from several popular CNNs. We used fc7 layer features obtained from an pre-trained Caffe implementation of AlexNet, VGG-16 and VGG-19 in comparison. We also compare our approach with a state-of-the-art method for visual localization under significant change in appearance, i.e., SeqSLAM[44] and place recognition method NetVlad [1]. We used OpenSeqSLAM code to test SeqSLAM [44]. We compare against NetVlad feature by using high performing NetVlad network (VGG-16, fVLAD with whitening) [1] for comparison. We adopt the same similarity calculation technique presented in Sec. 3.2 for all feature descriptors. Note that we could not compare with recent method [36] as the results are not available on the experimented datasets. We did not re-implement this method in our setting as it is very difficult to exactly emulate all the implementation details (e.g., requirement of all man-made structures to be labeled in images).

Evaluation Metric. The results of our proposed approach and the state-of-the-art methods are evaluated using F-measure in Table. 1 and CMC(Cumulative Match Characteristic) curve in Fig. 5. F-measure is the harmonic mean of precision and recall and it is calculated using precision and recall values obtained from the similarity matrices processed in each test. On the other hand, CMC curve is a rank based metric, which measures how well an identification system ranks the images in the test database with respect to an unknown probe image. In CMC curve, the probability of observing the correct match within a rank equal to or less than some value (y-axis) is plotted against ranks (x-axis).

Table 1: A comparison of F1-scores between our proposed method and other state-of-the-art methods on 5 datasets. We highlight the **best** and underline second best baseline method.

Dataset			Feature					
			Alex-Net	Vgg-16	Vgg-19	NetVlad	SeqSlam	Proposed
Held-Out Sets	Nordland	Winter-Summer	0.55	0.51	0.48	0.74	0.69	**0.76**
		Winter-Spring	0 .62	0.59	0.53	0.79	0.80	**0.86**
		Winter-Fall	0.58	0.54	0.51	**0.74**	0.68	**0.74**
		Summer-Spring	0.78	0.75	0.75	**0.83**	0.80	0.81
		Summer-Fall	0.94	0.92	0.90	0.93	0.84	**0.95**
		Spring-Fall	0.83	0.77	0.76	0.83	0.79	**0.86**
	St Lucia	Average	0.54	0.56	0.51	0.52	0.47	**0.61**
		Worst Trial	0.33	0.37	0.29	0.18	0.27	**0.43**
	Oxford	Average	0.64	0.62	0.62	0.62	0.41	**0.67**
		Worst Trial	**0.54**	0.43	0.48	0.23	0.14	0.46
Training Sets	CMU VL	Average	0.92	0.96	0.95	0.85	0.76	**0.98**
		Worst Trial	0.84	0.94	0.85	0.77	0.64	**0.96**
	LCE	Average	0.58	0.60	0.59	0.73	0.47	**0.78**
		Worst Trial	0.32	0.17	0.32	0.11	0.17	**0.39**

For localization datasets, a true positive is considered to be a situation where the best match is within $\pm\beta$ frames of the ground truth (β depends on the frame rate of the dataset); a false positive is where the best match falls outside these bounds. A false negative situation occurs where no match is found (i.e., the feature similarity for the best matching image pair is below certain threshold), since every scene in each dataset has a ground truth match.

Results and Analysis. Table. 1 demonstrates that in terms of F-measure, our method consistently outperforms baselines in the task of visual localization in almost all the compared datasets. For example, on the St Lucia dataset, our trained network achieves an F1-score of 0.61 compared to only 0.54 obtained by off-the-shelf AlexNet and 0.52 in NetVlad. Similar performance can be observed across all datasets.

We also show performance of our approach using CMC plot, i.e., Fig. 5 to summarize performance of our closed-set identification system. The proposed feature extractor shows similar or better performance in all datasets in almost all ranks. Although we use a simple approach for image matching, our method consistently outperform SeqSLAM in almost all experiments. SeqSLAM depends heavily on criteria that image was captured with similar speeds and minor accelerations. Hence, the performance suffers significantly when these condition are not met well.

We also trained a convolutional autoencoder (with pre-trained weights of Alexnet in encoder and the decoder weight was initialized using [13]) with all our training image pairs. Feature from the best performing model achieves F1 score of 0.91 in CMU VL, compared to 0.98 in the proposed approach and F1 score of 0.56 in St Lucia, compared to 0.61 in proposed approach. We see similar trend in all datasets. This observation supports our assumption that a direct application of DA for learning long-term invariance may be intractable due to the high complexity of change between scenes over long time.

Our trained autoencoder construct combined with baseline features improves accuracy over just using baseline features, thereby increasing appearance and viewpoint invariance in the resulting feature vectors. These results confirm two key premises of our work: 1) our approach can significantly improve image representations for visual localization and achieves better performance than conventional state-of-the-art feature extractors in this task, and 2) the popular idea of directly using features from pre-trained networks, e.g., [45, 46], can be sub-optimal since these networks, trained for object or scene classification, might be too generic for vision-based localization.

4.4. Qualitative Results

As an understanding aid, we now present work on visualizing what our networks learn. We follow the method [55] for examining occlusion sensitivity of classification networks. In this method, different portions of the image are systematically covered up with a gray square and the change in feature representation is monitored. Each pixel in the heatmap corresponds to the change in representation. It can be seen in Fig. 6 that a pre-trained AlexNet focuses mostly on recognizing objects and shapes useful for discovering different categories. In contrast, our autoencoder network is trained for learning invariant representation for visual localization task; thus features can be more abstract, as long as they are repeatable. Hence, our network learns to assign lower importance to potentially confusing features, such as moving objects (e.g.,cars, people), which are not

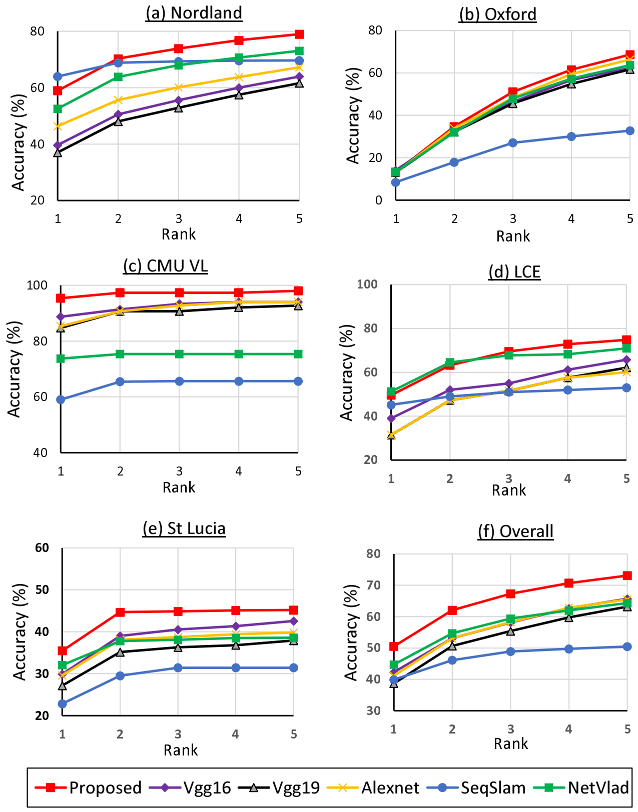


Figure 5: A comparison of our method vs baselines and sate-of-art methods based on Cumulative Matching Characteristic(CMC) curve on 5 datasets. The plots show the recognition percentage for rank 1 to 5. The corresponding dataset names are given above each plot.

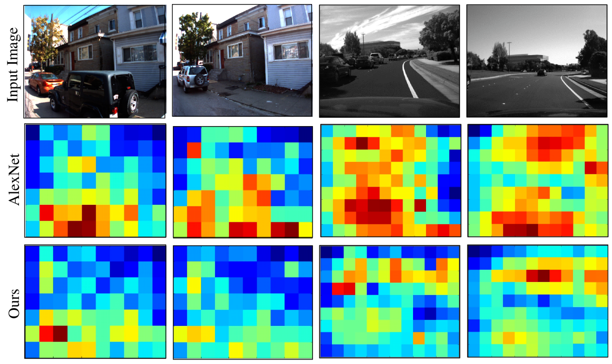


Figure 6: Figure shows one image pair from CMU-VL dataset and one pair from LCE dataset in 1st row. The corresponding heatmaps for the images from AlexNet and our network are shown in 2nd and 3rd row.

statically identifiable features of specific locations. Instead the network becomes trained for invariance to illumination and other global changes in the location.

In Fig. 7, we show some challenging queries and most similar images to these queries based on different feature extractors. These queries are hard for all feature extractors, because of significant change in appearance due to change in illumination, seasonal effect and appearance of different non-static objects in scene. However, our feature extractor was fairly successful in most of the cases, compared to other state-of-the art feature based methods. The fourth query was one case in which our approach failed. This was overall a very difficult query. Along with non-uniform illumination change and appearance of new different non-static objects, the viewpoint also changed significantly.

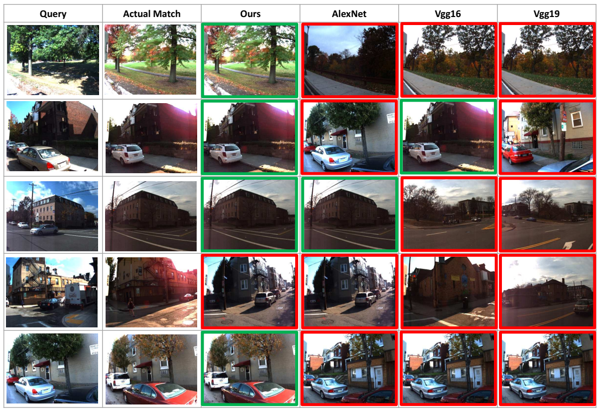


Figure 7: Example of retrieved best matching image for 5 difficult query image from CMU VL dataset. Each row contains query image (1st column), actual matching image(2nd column) and best matching result based on different feature descriptor (3rd, 4th, 5th and 6th column). The green and red border around result indicate correct and incorrect matching, respectively.

5. Conclusions

In this paper, we presented a novel method for learning illumination invariant features for vision-based localization. This feature extraction allows highly accurate location recognition amidst significant scene changes. To accomplish invariant feature learning, we employ pairs of images captured of the same location but at different times. These pairs train deep neural network models to learn commonsense understanding of how locations change appearance over time. Specifically, the model is tuned to extract discriminative features specific to a particular location, features with invariance to common changes in scene appearance, such as illumination, occlusion, sky, etc. Experimental results demonstrate that the proposed method significantly improves image matching accuracy over state-of-the art image-feature-based methods. Moreover, the proposed approach is scalable and is expected to generalize to previously unseen locations, based on experimental results.

Acknowledgments: This work was partially supported by Volkswagen Group of America.

References

[1] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, pages 5297–5307. IEEE, 2016.

[2] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, and E. Romera. Towards life-long visual localization using an efficient matching of binary sequences from images. In *ICRA*, pages 6328–6335. IEEE, 2015.

[3] G. Baatz, K. Köser, D. Chen, R. Grzeszczuk, and M. Pollefeys. Leveraging 3d city models for rotation invariant place-of-interest recognition. *International Journal of Computer Vision*, 96(3):315–334, 2012.

[4] H. Badino, D. Huber, and T. Kanade. Visual topometric localization. In *Intelligent Vehicles Symposium (IV)*, pages 794–799. IEEE, 2011.

[5] H. Badino, D. Huber, and T. Kanade. Real-time topometric localization. In *ICRA*, pages 1635–1642. IEEE, 2012.

[6] P. W. Battaglia, J. B. Hamrick, and J. B. Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110(45):18327–18332, 2013.

[7] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV*, pages 404–417. Springer, 2006.

[8] M. A. Brubaker, A. Geiger, and R. Urtasun. Lost! leveraging the crowd for probabilistic visual self-localization. In *CVPR*, pages 3057–3064. IEEE, 2013.

[9] 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. *arXiv preprint arXiv:1701.05105*, 2017.

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

[11] M. J. Cummins and P. M. Newman. Fab-map: Appearance-based place recognition and mapping using a learned visual vocabulary model. In *ICML*, pages 3–10, 2010.

[12] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893. IEEE, 2005.

[13] A. Dosovitskiy and T. Brox. Inverting visual representations with convolutional networks. In *CVPR*, pages 4829–4837. IEEE, 2016.

[14] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[15] P. Furgale and T. D. Barfoot. Visual teach and repeat for long-range rover autonomy. *Journal of Field Robotics*, 27(5):534–560, 2010.

[16] E. Garcia-Fidalgo and A. Ortiz. Vision-based topological mapping and localization methods: A survey. *Robotics and Autonomous Systems*, 64:1–20, 2015.

[17] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, volume 9, pages 249–256, 2010.

[18] 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 *ICRA*, pages 3507–3512. IEEE, 2010.

[19] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis. Learning temporal regularity in video sequences. In *CVPR*, pages 733–742. IEEE, 2016.

[20] J. Hays and A. A. Efros. Im2gps: estimating geographic information from a single image. In *CVPR*, pages 1–8. IEEE, 2008.

[21] D. Jayaraman and K. Grauman. Learning image representations tied to ego-motion. In *ICCV*, pages 1413–1421. IEEE, 2015.

[22] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, pages 675–678. ACM, 2014.

[23] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[25] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan. Webcam clip art: Appearance and illuminant transfer from time-lapse sequences. In *ACM Transactions on Graphics*, volume 28(5), page 131, 2009.

[26] H. Lategahn, J. Beck, B. Kitt, and C. Stiller. How to learn an illumination robust image feature for place recognition. In *Intelligent Vehicles Symposium (IV)*, pages 285–291. IEEE, 2013.

[27] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1–14, 1995.

[28] J. Levinson, M. Montemerlo, and S. Thrun. Map-based precision vehicle localization in urban environments. In *Robotics: Science and Systems III*, volume 4, pages 1–8, 2007.

[29] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide pose estimation using 3d point clouds. In *ECCV*, pages 15–29. Springer, 2012.

[30] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, volume 2, pages 1150–1157. IEEE, 1999.

[31] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2016.

[32] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, page 0278364916679498, 2016.

[33] R. Memisevic. Learning to relate images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1829–1846, 2013.

[34] V. Michalski, R. Memisevic, and K. Konda. Modeling deep temporal dependencies with recurrent grammar cells. In *NIPS*, pages 1925–1933, 2014.

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

[36] T. Naseer, G. L. Oliveira, T. Brox, and W. Burgard. Semantics-aware visual localization under challenging perceptual conditions. In *ICRA*. IEEE, 2017.

[37] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, pages 1717–1724. IEEE, 2014.

[38] A. Pronobis, B. Caputo, P. Jensfelt, and H. I. Christensen. A discriminative approach to robust visual place recognition. In *IROS*, pages 3829–3836. IEEE, 2006.

[39] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014.

[40] A. Rottmann, Ó. M. Mozos, C. Stachniss, and W. Burgard. Semantic place classification of indoor environments with mobile robots using boosting. In *AAAI*, volume 5, pages 1306–1311, 2005.

[41] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *ICCV*, pages 667–674. IEEE, 2011.

[42] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *CVPR Workshops*, pages 806–813. IEEE, 2014.

[43] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[44] N. Sünderhauf, P. Neubert, and P. Protzel. Are we there yet? challenging seqslam on a 3000 km journey across all four seasons. In *Workshop on Long-Term Autonomy, ICRA*, page 2013. IEEE, 2013.

[45] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford. On the performance of convnet features for place recognition. In *IROS*, pages 4297–4304. IEEE, 2015.

[46] 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. *Robotics: Science and Systems XII*, 2015.

[47] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9. IEEE, 2015.

[48] T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2), 2012.

[49] C. Valgren and A. J. Lilienthal. Sift, surf and seasons: Long-term outdoor localization using local features. In *EMCR*, 2007.

[50] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, pages 1096–1103. ACM, 2008.

[51] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

[52] V. Vonikakis, D. Chrysostomou, R. Kouskouridas, and A. Gasteratos. A biologically inspired scale-space for illumination invariant feature detection. *Measurement Science and Technology*, 24(7):074024, 2013.

[53] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, pages 2794–2802. IEEE, 2015.

[54] J. Wu and J. M. Rehg. Where am i: Place instance and category recognition using spatial pact. In *CVPR*, pages 1–8. IEEE, 2008.

[55] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833. Springer, 2014.