Learning Deep Representations for Place Recognition in SLAM

Satyaki Chakraborty[1], Sanjoy K. Saha[1] and Aritra Mukherjee[2]

***Abstract*—Closing loops for pose graph optimization, by recognising previously mapped places is an essential step for performing Simultaneous Localisation and Mapping. The traditional approaches for recognising known places follow a feature-based bag-of-words model while discarding certain geometric and structural information. In order to improve real-time query performance, we take a slightly different approach by learning low-dimensional global representation vectors using a deep autoencoder architecture. An auto encoder can encode and decode an image to itself but in the process learns a representation of the image in a reduced feature space better than linear methods like PCA. We train a deep autoencoder architecture and take the distance from loop initiation image as the penalty thus reporting loop closures at lowest penalties. Finally we evaluate the performance of our method on the KITTI visual odometry dataset and compare it with the state-of-the-art approaches for place recognition.**

I. Introduction

In the context of robot navigation with vision, the task of Simultaneous Localization And Mapping (SLAM) is an important task. The entire SLAM process relies on recognizing the places the robot has already visited to achieve visual loop closure detection. Though SLAM is considered as a chicken-and-egg problem where simultaneous determination of location of the robot and making the map of the environment has to be done, visual cues help in determining the end of a loop travelled by the robot better than GPS location data or in places where GPS data is unavailable or highly erroneous. With further information from visual odometry or IMU (Inertial Measurement Unit) data or both, mapping of the environment under the scope of camera or LiDAR can be done with considerable accuracy. The task of visual loop closure detection can be easily described as reporting a high score when the robot comes back to the position that it started from by recognizing the scene. The main challenges in this approach are change in pose of the camera, change in illumination, change in macro level features like parked cars, movable objects etc. Traditional bag-of-words based approach works by representing the image as vector of visual patches, corners etc. but that is likely to fail when the loop initiation images change significantly. Hence a representation that captures the contextual essence of the images is necessary to detect the loop closure successfully, in general place recognition mechanism independent of minor changes in the environment is in demand.

1. Both S. Chakraborty and S. K. Saha are with the Department of Computer Science and Engineering, Jadavpur University, India, e-mail: (satyaki.cs15@gmail.com, sksaha@cs.jdvu.ac.in).

2. A. Mukherjee is with (TODO - add Aritra's designation)

Manuscript received April 19, 2005; revised January 11, 2007.

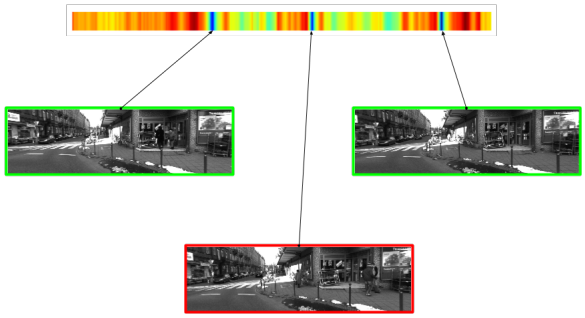

Figure 1 *Here the image frame in red border denotes the query image frame and the possible loop closure candidates are shown in green border. Above is shown a single row from the confusion matrix generated from our method. The blue regions show the best matches obtained by our approach.*

In this regard deep autoencoders allow us to devise a method to extract a much lower dimensional vector representation of an image that can ensure near accurate reconstruction. In this work, we have proposed a deep autoencoder based network to accomplish the task of place recognition.

II. Related Work

In the context of place recognition the major tasks are representing the frames with the help of visual descriptors and subsequently judging the similarity between the frames based on the descriptors. Various approaches have been followed by the researchers. Some of the major approaches are discussed as follows.

A. BoW based approaches and the FABMAP model

The BoW(Bag-of-words) approach was first successfully applied to image classification and retrieval[11]. Here, a fixed size vocabulary is used as a vector quantizer to classify descriptors in an image frame. The vectors consists of image patches which acts as features and are generally chosen randomly from image patches with textured neighbourhood. While querying the database with an image, the extracted features of the query image are converted into a vector of classes present in it. It is then used to compare the query image with an image from the

database. The vector does not store positional information but is rather a histogram of frequency, the technique of assigning a patch to a class is done by nearest neighbour approach in the multidimensional feature space, dimension being simply pixel intensity values. The vocabularies in this case are usually learned using unsupervised clustering methods like k-means or hierarchical k-means[13].

The FABMAP model[12] for example considers a sequence of non-overlapping frames and checks if each frame belongs to an already visited place. This task is achieved by comparing the probability of a binary BoW vector generated from either a previously seen place or by a previously unseen place (represented by a background model). One significant issue with this approach is that whenever we see a new image frame which is very similar to two (or more) frames present in the database, the matching images attract half (or less) of the probability mass and thus the threshold for being recognized might not be achieved. Another drawback of such methods is that in order to deal with computational expenses, full feature based methods in general discard underlying structure and geometry between frames. This results in perceptual aliasing which is a very common issue with place recognition methods. This the reason why FABMAP is recommended to be operated on non-consecutive frames or else it will detect loop closure almost along the entire path. The main advantage though is that place representation by a vector of visual words is mostly generic in nature and thus the system trained in an environment performs good even in a different kind of environment.

B. SeqSLAM

In order to deal with the issue of perceptual aliasing as prevalent in FABMAP, methods like SeqSLAM[14] perform correlation-based matching on short sequences of images instead of depending directly on individual image frames. Thus, instead of finding the single most-likely location, given a query image, this method looks for the best candidate matching location within every local navigation sequence. Localisation is hence achieved by recognizing coherent sequences of such "local best matches." One significant step of the SeqSLAM algorithm is that the distance matrix is locally contrast enhanced which helps to find best matches in every local neighborhood of the trajectory instead of only one global best match. The main advantage of SeqSLAM method is its robustness to weather change and or luminosity change of the path.

C. Voting and Nearest Neighbour based approaches

Voting based methods([7], [8], [9]) perform a nearest neighbour search on the image descriptor space to identify potential matches. It is quite similar to the original bag of words approach but sometime image descriptors like SIFT[17], BRISK[18] or FREAK[19] are also used to form the descriptor vector. The selection of images that are to searched for a match are chosen in groups along the trajectory when applied to loop closure problems, rather searching all the images till the next loop closure point. In case of binary image descriptors, however it becomes difficult to perform a kNN (k-nearest neighbour) search efficiently search mainly due to high dimensionality. Methods described in [10] and [8] first project the high-dimensional feature descriptors into a low-dimensional space for fast and accurate nearest neighbour search. By means of voting similar images are identified and loop closure is detected by thresholding on the similarity value.

D. CNN based approaches

Recently Convolutional neural network based approaches have been developed for place recognition tasks. Chen et al.[15] used the Overfeat network[16] trained on the ImageNet dataset to extract features from the image frames. One significant advantage of such method is that using a sequence of linear transformations and pooling operations alternatively, it is possible to obtain dense representations of the images and perform search on the low dimensional vector space. Chen et al. further improved on this by passing the most likely matches through a spatial and a sequential filter. However, in this approach the network was pre-trained to the ImageNet dataset[24] which is optimized for object recognition rather than place recognition.

It may be noted that it is very difficult to devise the suitable descriptors for place recognition. It may vary depending on the scenes under consideration. It has motivated us to rely on deep learning that can automatically extract the features that can be utilized in optimal place recognition. In section III proposed methodology is elaborated and result is presented in section IV

III. Proposed Methodology

In this work we propose an autoencoder based deep learning network that extracts a lower dimensional vector representation of an image. With an autoencoder trained to encode and decode an image, the task of loop detection reduces to finding the distance between the encoded vectors of the query image and the input image. Whenever the distance falls below a certain threshold a loop closure can be reported. The value of the threshold can be either learned or tuned based on previous experience about the alteration limits of the environment. The reconstruction process in this case uses the concept of switch matrix which holds the position of the pixel selected during a pooling layer of the encoder so that proper mapping can be done during decoding. The methodology is detailed in the subsequent subsections.

The novelty of the proposed work lies in the facts that:

- It uses deep learning technique to represent an image thus helping in faster and more contextual loop closure detection.
- It uses switch matrix to minimize reconstruction error due to pooling while training, thus preventing underfitting of the network.
- If trained over a significantly high number of images then the representation vector can represent different macro level features of the image as a score in different elements of the vector, thus creating a numerical description of the scene.

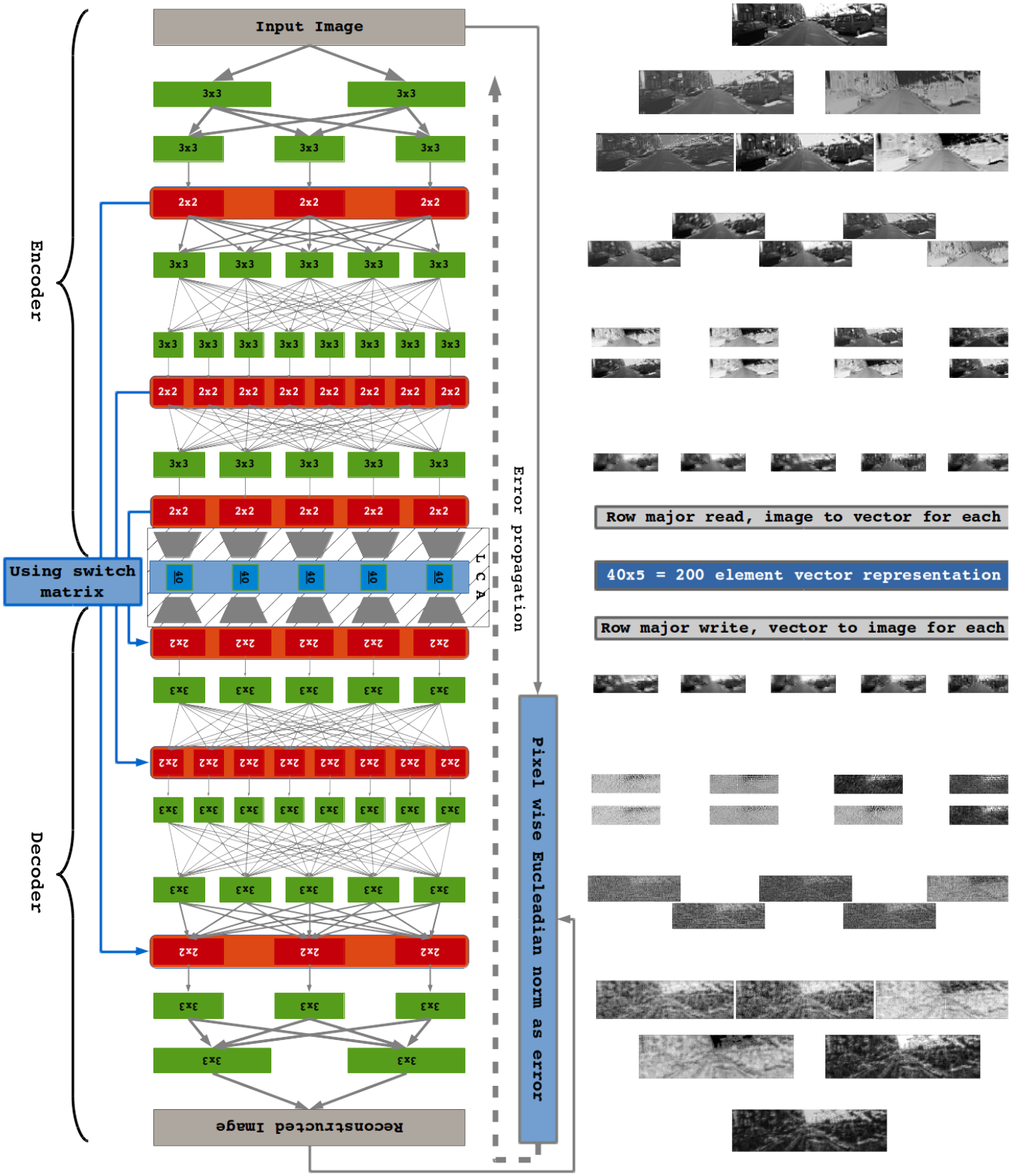

Figure 2 *The overall architecture along with the image output at each layer after convolution and deconvolution layers, the switch matrix generated for each image during encoding pooling layers are used up in unpooling layers during decoding as shown above*

A. Architecture

At the heart of the proposed architecture lies a deep convolutional autoencoder. It is further modified by adding a layer of locally connected autoencoders to map an image frame into a representation vector of n dimensions. The higher the value of n, the greater is the capability of the vector to encode unique macro level features of the scene in each of its elements. The choice of optimal size for the vector is subjected to further research. The value of n is empirically chosen as 200 in this work. We discuss the architecture in the following two subsections.

1) Deep Convolutional Autoencoder: Deep autoencoders were initially studied by Hinton et al.[1] for reducing the dimensionality of raw input data with neural networks. This approach was later extended for image[2] and document retrieval[3] tasks. But when working with images, fully connected autoencoders ignore local 2D image structure and hence suffer from a redundancy in learning the parameters. The visual field of the features are made to span the entire input thus destroying local structural information. In this case enforcing local connectivity and weight sharing[4] not only scales well for realistic image sizes, but also removes redundancies in the input to model discriminative representations. The architecture we used here is essentially a 13 layer deep convolutional autoencoder with only the middle layer as a layer of locally connected autoencoders. The first six layers are for encoding and the last six layers are for reconstructing the input which structurally is the mirror image of the encoding network. The features of the 7^{th} layer(the layer of locally connected autoencoders) are used as representations for the image frames. The choice of the number of convolution layer, the number of kernel in each layer and the number of pooling layers are empirically chosen, though the initial design was inspired by the Alexnet [25]. The task of convolution kernels is to learn the weights in such a manner that it can extract unique features from the images which helps in its discrimination. The advantage over hand designed kernels is that here it learns features often missed out by human judgement. The task of pooling layers is dimensionality reduction without significant loss of information. The stride of both convolution and pooling layers defines the number of pixels the kernel shifts. The pad defines the number of extra zero value pixels padded on the boundary after convolution or pooling, in general if the convolution kernel is a square of dimension $2n + 1$ then the pad is n. The figure III-A shows the entire architecture and the condition of an input image at each layer of the encoding and decoding process. It is to be noted that all the output of the previous convolution layer is fed to all the convolution process in the next layer. A pooling is done for dimensionality reduction of images after convolution, and so the pooling layers are shown as connected to each convolution layer before it. The number of pooling kernels shown in the figure III-A is just symbolic, the number of kernel in each pooling layer is one.

In order to aid reconstruction, we use the 'switch matrix' method ([5] & [6]) in the decoding layers. As deep learning involves huge amount of matrix computation so to speed up the process without significant loss of accuracy pooling technique is used. The image size to next convolution layer is diminished by selecting one pixel value of the next layer input, from a patch of the output image of the previous convolution layer. Max-pooling selects the pixel value which is maximum within the patch. During max-pooling, switches store the locations of the cells from which the values are selected and this information is used for reconstructing the original image during unpooling in the decoding layers. Without such an approach a random number is used to put back the pixel in proper place thus unwillingly increasing the reconstruction error even when the final representation was good enough. It is to be noted that switch matrix is a transient by-product during the pooling stage of an image which can only be used during the unpooling of that image during an epoch, thus thought the switch matrix holds some of the information during encoding, the information is unusable for a unified representation learning. Table 1 shows the complete architecture in tabular form.

Layer	kernel size	stride	pad	Output dim.
Input	-	-	-	1x96x336
Conv-1	3x3	1	1	2x96x336
Conv-2	3x3	1	1	3x96x336
Pool-1	2x2	2	0	3x48x168
Conv-3	3x3	1	1	5x48x168
Conv-4	3x3	1	1	8x48x168
Pool-2	2x2	2	0	8x24x84
Conv-5	3x3	1	1	5x24x84
Pool-3	2x2	2	0	5x12x42
LCA-enc	-	-	-	5x40
LCA-dec	-	-	-	5x12x42
Unpool-1	2x2	2	0	5x24x84
Deconv-1	3x3	1	1	8x24x84
Unpool-2	2x2	2	0	8x48x168
Deconv-2	3x3	1	1	5x48x168
Deconv-3	3x3	1	1	3x48x168
Unpool-3	2x2	2	0	3x96x336
Deconv-4	3x3	1	1	2x96x336
Deconv-5	3x3	1	1	1x96x336

TABLE I. THE FIRST HALF OF THE NETWORK CONSISTS OF CONVOLUTION (CONV) AND POOL LAYERS, FOLLOWED BY ENCODING AND DECODING (DONE BY THE LOCALLY CONNECTED AUTOENCODERS) AND FINALLY A NUMBER OF DECONVOLUTION (DECONV) AND UNPOOLING LAYERS.

2) Locally Connected Autoencoders: The feature maps at the output of the 6^{th} layer are passed through a layer of locally connected autoencoders (LCA) to learn a further lower dimensional representation. An LCA is a fully connected 2 layer feedforward neural network where the number of input neurons is equal to the number of output neurons and the the number of hidden neurons is equal to the dimension of the autoencoder, which in this case is 40. Each of the 5 feature maps are passed through an autoencoder and projected into a representation vector of 40 dimensions. All such representations are stacked on top of one another to form the 200 dimensional representation of the image frame. Using local connections in stead of using a fully connected layer not only helps capture distinguishing features from each feature map separately but also reduces the number of parameters to be learned.

B. Training Methodology

Training and testing was done on the KITTI Odometry dataset[23]. The training was done in two phases – a pre-training phase followed by a global fine-tuning phase, discussed as follows.

1) Greedy Layer-wise Unsupervised Pretraining: The greedy unsupervised pretraining proceeds in a layerwise fashion. Keeping in mind the difficulty of jointly training a deep neural network architecture with respect to a global objective, at this stage, each layer is pretrained in an unsupervised fashion by taking the output of the previous layer and producing a new representation as output. This phase is called layer-wise because only the parameters of one layer are updated at a time keeping the others fixed. The first layer of the autoencoder architecture takes an image as input and tries to reconstruct the same. The next layer then takes the activations of the first layer as input and tries to reconstruct the same. In this way pretraining proceeds in a bottom-up fashion from the first layer to the last.

Parameters learnt this way serve as a good initialisation for each layer of the network. In the next stage, the network is then fine tuned and the parameters are updated with respect to some global criteria.

2) Global Fine-tuning: Unsupervised pretraining has been extensively used where the fine tuning phase is supervised. However it has been shown, that for autoencoder networks, unsupervised pretraining improves test accuracy significantly when the fine tuning phase is also unsupervised[1]. Similar to their work, our global fine tuning phase is also unsupervised. Erhan et al.[20] studied why the unsupervised pretraining produces a good initialisation, by considering the trajectories of the neural network during the supervised fine-tuning phase. After the full autoencoder network is trained with respect to some global objective, the output of the LCA (locally connected autoencoder) layer are used as the learnt representations of the images in the dataset.

IV. Results and Analysis

Without the LCA (encoding and decoding) layers, the network is able to achieve much less reconstruction error on the test set as the dimensionality reduction is avoided at the LCA layer thus loosing less amount of information. However the main aim of our network is to learn a lower dimensional representation of the input image. Hence with the introduction of the LCA layers, even though reconstruction error increases to some extent, useful low dimensional representations can be learnt which can be used to identify loop closures. We hence show the reconstructed outputs of an image sample during test time using the network described in section III-A1 and the corresponding network without the LCA layers.

Next we show the corresponding confusion matrices that are generated from the representations learnt by the network with LCA layers and the network without LCA layers in Figs 5.1 and 5.2 respectively. Even though the result for the network with LCA layers is relatively more noisy compared to the results obtained from the network without the LCA layers, it is to be kept in mind that in the former case the representations

Figure 4.1

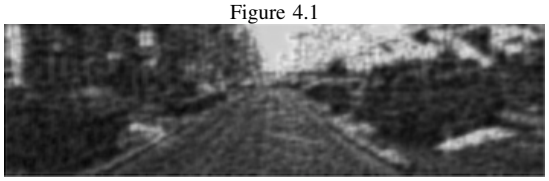

Figure 4.2

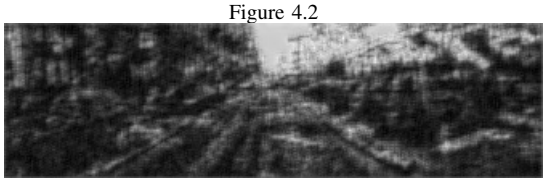

Figure 4.3

Fig. 4.1 shows a sample input image from the Karlsruhe drive sequence. Fig. 4.2 shows the reconstructed output from a network without the LCA layers. Fig. 4.3 shows the same from the network discussed in section III-A1

are 200 dimensional feature vectors whereas in the latter case those are of 2520 dimensions. This threshold can be learnt or manually tuned. In this case however we have manually chosen the threshold and reported the variation of precision with recall on the KITTI odometry dataset sequence 5. We have also compared our result against that of FabMap [12] and SeqSLAM [14], both codes being available as opensource. As mentioned earlier in section 3.2 the ground truth in confusion matrix format was generated from trajectory format by checking for overlapping image sequences in a snall geographic radius of 3 meters. The fig 6 shows all the matrices after applying a low-pass threshold of 5 on the intensity scale of 0 to 255 and then inverting the same. It should be mentioned that our system processed approximately 150 frames per second. It can be observed that only our method was successful in detecting when the vehicle was waiting at a turn with the camera running thus producing the box on the diagonal in the lower right corner.

V. Conclusion and Future Work

Place recognition is not only helpful in loop detection in SLAM but also for content based image retrieval when the subject of the image is a natural scene. The bag of word approach used in FABMAP though straight forward in concept, suffers from inaccuracy when there a drastic change in the landmark set occurs. it also suffers from perceptual aliasing. The seqSLAM approach is better than feature based approach and is robust to seasonal and temporal change of the path. The CNN based approach takes context into account by learning features from it and representing a scene by a vector, but the CNN used was trained on a more universal dataset. In our approach the network is a multi layer autoencoder and is trained to reconstruct the image but in the process it learns a vector representation of the image which in turn is used for loop closure detection. The advantage of this approach

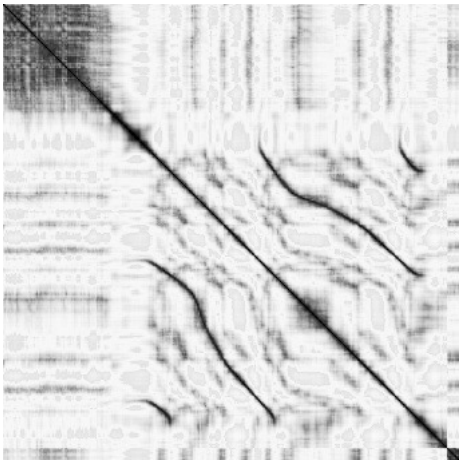

Figure 5.1 *Confusion matrix of Karlsruhe drive sequence '2010_03_09_drive_0081' obtained from the network with LCA layers. Representation vectors have 200 dimensions.*

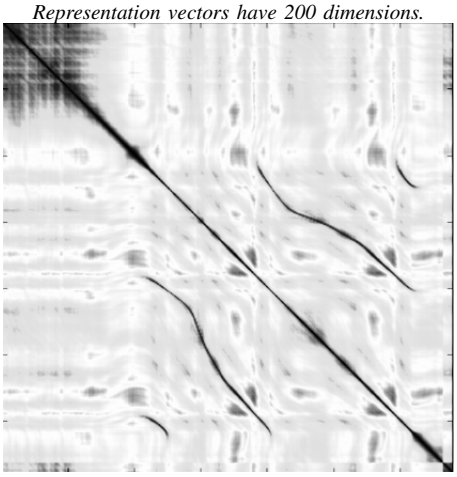

Figure 5.2 *Confusion matrix of Karlsruhe drive sequence '2010_03_09_drive_0081' obtained from the network without LCA layers. Here the representation vectors have 2520 dimensions.*

is that vectors generated for two frames of the same scene which differ geometrically but are similar contextually and by content, are quite close to each other. Thus the approach works in general place recognition tasks also and holds the promise to be extended to context and content based image matching problems. Though training an autoencoder is a time intensive process but evaluating it is very fast due, which again can be accelerated drastically by use of GPUs, making it feasible to be used for real time SLAM.

REFERENCES

[1] G.E.Hinton and R.R.Salakhutdinov, *Reducing the Dimensionality of Data with Neural Networks* Science 313.5786 (2006): 504-507.

[2] A Krizhevsky, GE Hinton, *Using very deep autoencoders for content-based image retrieval.* ESANN, 2011

[3] Hinton, Geoffrey, and Ruslan Salakhutdinov. "Discovering binary codes for documents by learning deep generative models." Topics in Cognitive Science 3.1 (2011): 74-91.

[4] Masci, Jonathan, et al. "Stacked convolutional auto-encoders for hierarchical feature extraction." Artificial Neural Networks and Machine LearningICANN 2011. Springer Berlin Heidelberg, 2011. 52-59.

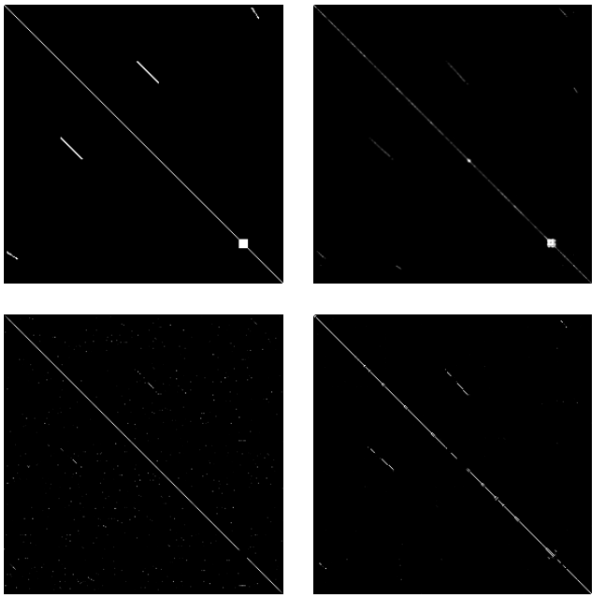

Figure 6 *The figure describes all the confusion matrices from the experiments after a threshold of 5 in intensity scale is applied, (a)top left is the ground truth, (b)top right is the output from OpenFabmap (c)bottom left is the output from OpenSeqSlam and (d)bottom right is outputof our method.*

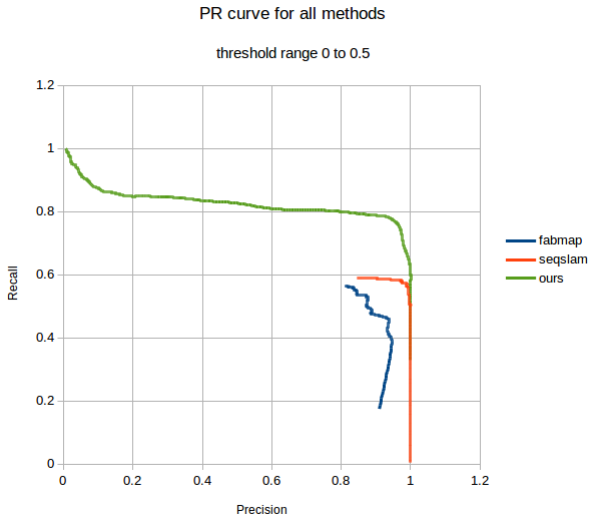

Figure 7 *The precision recall curve of the other methods with ours, note at higher precision our method has highest recall thus emphasizing the quality of our method. However this result is solely based on experiments done on sequence 5 of the KITTI visual odometry dataset. A larger dataset must be dealt with for a more thorough and extensive evaluation is needed for generalising the parameters of the neural network.*

[5] Zeiler, Matthew D., et al. "Deconvolutional networks." Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010.

[6] Noh, Hyeonwoo, Seunghoon Hong, and Bohyung Han. "Learning deconvolution network for semantic segmentation." Proceedings of the IEEE International Conference on Computer Vision. 2015.

[7] Jegou, Herve, Matthijs Douze, and Cordelia Schmid. "Hamming embedding and weak geometric consistency for large scale image search." Computer VisionECCV 2008. Springer Berlin Heidelberg, 2008. 304-317.

[8] Lynen, Simon, et al. "Placeless place-recognition." 3D Vision (3DV), 2014 2nd International Conference on. Vol. 1. IEEE, 2014.

[9] Stewnius, Henrik, Steinar H. Gunderson, and Julien Pilet. "Size matters: exhaustive geometric verification for image retrieval" Computer VisionECCV 2012. Springer Berlin Heidelberg, 2012. 674-687.

[10] Bosse, Michael, and Robert Zlot. "Keypoint design and evaluation for place recognition in 2D lidar maps." Robotics and Autonomous Systems 57.12 (2009): 1211-1224.

[11] Sivic, Josef, and Andrew Zisserman. "Video Google: A text retrieval approach to object matching in videos." Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on. IEEE, 2003.

[12] Cummins, Mark, and Paul Newman. "FAB-MAP: Probabilistic localization and mapping in the space of appearance." The International Journal of Robotics Research 27.6 (2008): 647-665.

[13] Arai, Kohei, and Ali Ridho Barakbah. "Hierarchical K-means: an algorithm for centroids initialization for K-means." Reports of the Faculty of Science and Engineering 36.1 (2007): 25-31.

[14] Milford, Michael J., and Gordon F. Wyeth. "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights." Robotics and Automation (ICRA), 2012 IEEE International Conference on. IEEE, 2012.

[15] Chen, Zetao, et al. "Convolutional neural network-based place recognition." arXiv preprint arXiv:1411.1509 (2014).

[16] Sermanet, Pierre, et al. "Overfeat: Integrated recognition, localization and detection using convolutional networks." arXiv preprint arXiv:1312.6229 (2013).

[17] Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). "Speeded-up robust features (SURF)." Computer vision and image understanding, 110(3), 346-359.

[18] Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011, November). "BRISK: Binary robust invariant scalable keypoints." In 2011 International conference on computer vision (pp. 2548-2555). IEEE.

[19] Alahi, A., Ortiz, R., & Vandergheynst, P. (2012, June). "Freak: Fast retina keypoint." In Computer vision and pattern recognition (CVPR), 2012 IEEE conference on (pp. 510-517). IEEE.

[20] Erhan, Dumitru, et al. "Why does unsupervised pre-training help deep learning?." Journal of Machine Learning Research 11.Feb (2010): 625-660.

[21] Geiger, Andreas, Julius Ziegler, and Christoph Stiller. "Stereoscan: Dense 3d reconstruction in real-time." Intelligent Vehicles Symposium (IV), 2011 IEEE. IEEE, 2011.

[22] Geiger, Andreas, Martin Roser, and Raquel Urtasun. "Efficient large-scale stereo matching." Asian conference on computer vision. Springer Berlin Heidelberg, 2010.

[23] Geiger, A., Lenz, P., Stiller, C., Urtasun, R. (2013). "Vision meets robotics: The KITTI dataset" The International Journal of Robotics Research, 0278364913491297.

[24] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., Fei-Fei, L. (2009, June). "Imagenet: A large-scale hierarchical image database" In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on (pp. 248-255). IEEE.

[25] Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks" In Advances in neural information processing systems (pp. 1097-1105).