# Learning Deep Representation for Place Recognition in SLAM

## Machine Learning Project Report

Completed in Fulfilment of the Requirements for the Machine Learning Project Guidelines at NIIT University

Submitted By:

Ajinkya Bedekar (U101116FCS183) (D5)

Devansh Anhal (U101116FCS031) (D2)

Dhruva Agarwal (U101116FCS177) (D5)

Harsha Deuri (U101116FCS043) (D2)

# Title of Project

Learning Deep Representation for Place Recognition in SLAM

# Authors of Paper

Aritra Mukherjee, Satyaki Chakraborty, and Sanjoy Kumar Saha

# Introduction

SLAM stands for Simultaneous Localization and Mapping. It is the problem of constructing or updating a map of an unknown environment during navigation while simultaneously keeping track of the agent's location within that environment.

Place Recognition refers to accurately and efficiently recognising the location of a given input image. In the context of work done by the authors of the paper, Place Recognition means recognising whether the agent (for example, robot) has visited a particular place previously or not.

Loop closure detection is reconstruction of three-dimensional environments and estimate camera trajectory accurately. Closing loops for pose graph optimization, by recognising previously mapped places is an essential step for performing SLAM. The main objective behind pose graph optimisation is to estimate robot trajectory from relative pose measurements.

While the robot navigates, the camera sends the frames as input, which are represented using visual descriptors. Simultaneously, the robot checks for the similarity between the frames based on descriptors. If some input frame is found similar, then the loop closure is said to be detected, and thus the place is recognised as previously visited place.

There are many approaches to achieve Place Recognition in SLAM. Some of them are discussed below.

## BoW Based Approaches

In the Bag of Words (BoW) approach, the descriptors in an image frame were classified with the help of fixed size vocabulary. It was first implemented successfully for image classification and retrieval. The vectors generally comprises of images patches which are used as features and chosen randomly from image patches having textured neighbourhood. Some of the models using BoW approach include FABMAP model, SeqSLAM model, etc.

The FABMAP model takes into consideration a sequence of non-overlapping frames and checks whether each frame belongs to some already visited place. The disadvantage of this model is the problem of perceptual aliasing.

To solve the problem of perceptual aliasing in FABMAP model, SeqSLAM perform matching, based on correlation, on short sequences of frames instead of directly depending on individual frames.

The Voting based methods perform nearest neighbour search on the space made of image descriptors to identify potential matches. These methods are quite similar to the original BoW approach. The image descriptors such as SIFT, BRISK, or FREAK are sometimes used to form descriptor vector. It is necessary to reduce the feature dimensions in order to achieve fast and accurate nearest neighbour search in loop closure detection. Many methods have been proposed to achieve the same. By means of majority voting, the images which are similar are identified and using a threshold on the similarity value, loop closure is detected.

## Deep Learning Based Approaches

Several researchers have developed Convolutional Neural Network (CNN) based approaches for loop closure detection. The Overfeat network was used by a scientist, which was trained on the ImageNet dataset, to extract features from the image frames. It is possible to obtain dense representations of the images and perform a search on the low dimensional vector space using a sequence of convolution and pooling operations. The main drawback of this approach is that the network which was used, was pre-trained on the ImageNet dataset. So, the model was optimized primarily for object detection and not focussed towards place recognition which is desired for loop closure detection.

For localisation tasks, Denoising Autoencoders (DA) have also been used. The DA is used with fully connected layers to extract features for comparing structural similarity of two images.

The major challenge for this problem is to design the descriptors suitable for loop closure detection. It is dependent on the scenes and conditions under consideration. Using this as motivation, the authors relied on deep learning that has the capability to automatically extract the features and thus, can be utilized for place recognition.

# Motivation

Place recognition is one of the most fundamental topics in the computer-vision and robotics communities. Despite years of knowledge accumulated in this field, the process of identifying the location of a given image by querying the locations of images belonging to the same place in a large geotagged database, usually known as place recognition, still remains an open problem due to the various ways in which the appearance of real-world places may differ.

Place recognition has attracted a significant amount of attention in the computer-vision and robotics communities.

One major characteristic that separates place recognition from other visual recognition tasks is that place recognition has to solve condition-invariant recognition to a degree that many other fields haven’t.

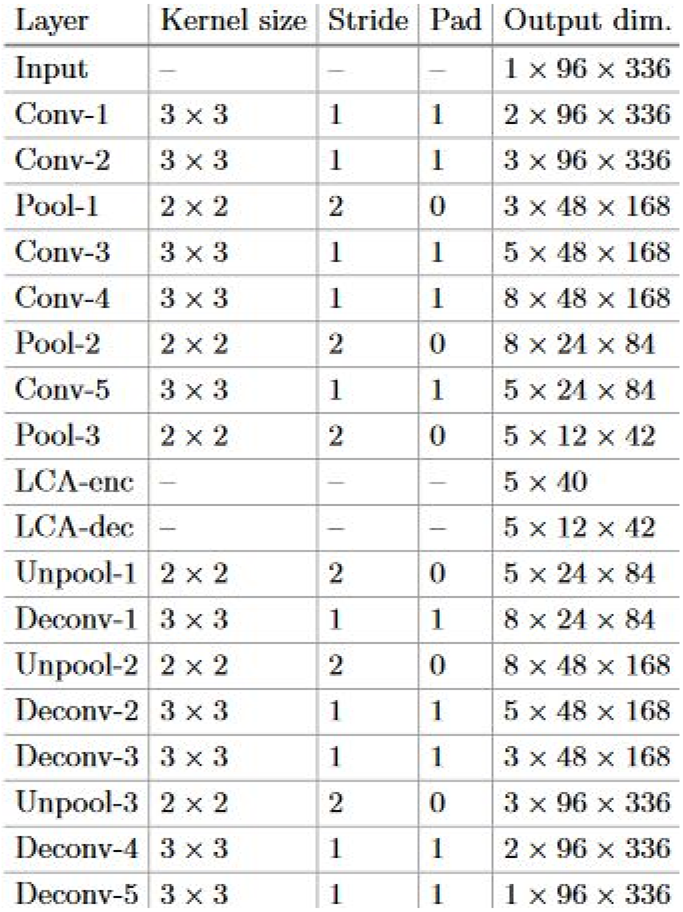
This paper provides an overview of both traditional and deep-learning-based descriptive techniques widely applied to place-recognition tasks, which is by no means exhaustive. It also presents a brief view of convolutional neural networks and the corresponding techniques used in place recognition.

# Work Done

The authors proposed an autoencoder based deep learning network that extracts a lower dimensional vector representation of an image. Using an autoencoder trained to encode and decode an image, the loop closure detection task reduces to finding the distance between the encoded vectors of the query image and the input image. When the distance falls a fixed threshold, a loop closure is said to be detected. The threshold value can be learned or tuned on the basis of previous experience about the alteration limits of the environment.

The reconstruction process here uses the switch matrix concept 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 proposed 12-layer deconvolution layer architecture, consisting of convolution layers, pooling layer, unpooling layers, deconvolution layers, and locally connected auto - encoders and decoders, is as follows.



| Layer | Kernel size | Stride | Pad | Output dim. |
|---|---|---|---|---|
| Input | – | – | – | 1 × 96 × 336 |
| Conv-1 | 3 × 3 | 1 | 1 | 2 × 96 × 336 |
| Conv-2 | 3 × 3 | 1 | 1 | 3 × 96 × 336 |
| Pool-1 | 2 × 2 | 2 | 0 | 3 × 48 × 168 |
| Conv-3 | 3 × 3 | 1 | 1 | 5 × 48 × 168 |
| Conv-4 | 3 × 3 | 1 | 1 | 8 × 48 × 168 |
| Pool-2 | 2 × 2 | 2 | 0 | 8 × 24 × 84 |
| Conv-5 | 3 × 3 | 1 | 1 | 5 × 24 × 84 |
| Pool-3 | 2 × 2 | 2 | 0 | 5 × 12 × 42 |
| LCA-enc | – | – | – | 5 × 40 |
| LCA-dec | – | – | – | 5 × 12 × 42 |
| Unpool-1 | 2 × 2 | 2 | 0 | 5 × 24 × 84 |
| Deconv-1 | 3 × 3 | 1 | 1 | 8 × 24 × 84 |
| Unpool-2 | 2 × 2 | 2 | 0 | 8 × 48 × 168 |
| Deconv-2 | 3 × 3 | 1 | 1 | 5 × 48 × 168 |
| Deconv-3 | 3 × 3 | 1 | 1 | 3 × 48 × 168 |
| Unpool-3 | 2 × 2 | 2 | 0 | 3 × 96 × 336 |
| Deconv-4 | 3 × 3 | 1 | 1 | 2 × 96 × 336 |
| Deconv-5 | 3 × 3 | 1 | 1 | 1 × 96 × 336 |

For experiments, the authors have worked with the KITTI Odometry dataset.

The confusion matrix (shown in Figure 1) represents the Euclidean distance between the learned representation vector of the image frames in the test sequence. Along the diagonal, the distance should be zero (as it is the distance with itself). By applying a threshold on the distances, (similarity) loop closure is detected. The threshold on distance should be low enough so as to avoid false detection. It should be chosen considering the change in illumination, angle of view, or even shift in dynamic objects when the robot revisits the place. Its value is taken as 5 for experiments. The image vectors that have a distance less than the threshold qualify for a loop closure.

In KITTI dataset, sequence 5 contains loop closure. The white colour denotes loop closure.

The results obtained by authors are shown below.

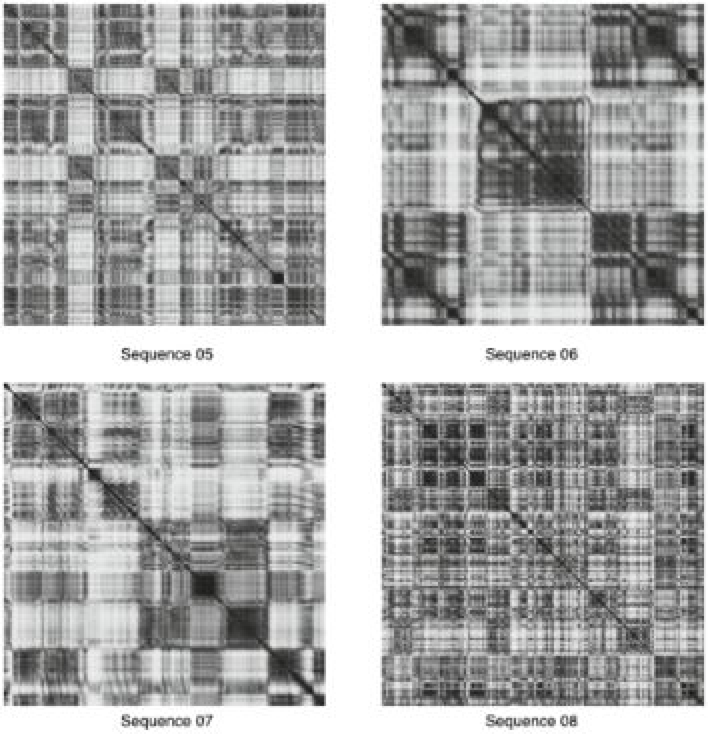


**Fig. 1.** Confusion matrices for sequences 5, 6, 7 and 8 of KITTI dataset. Darker the value, images are more similar.

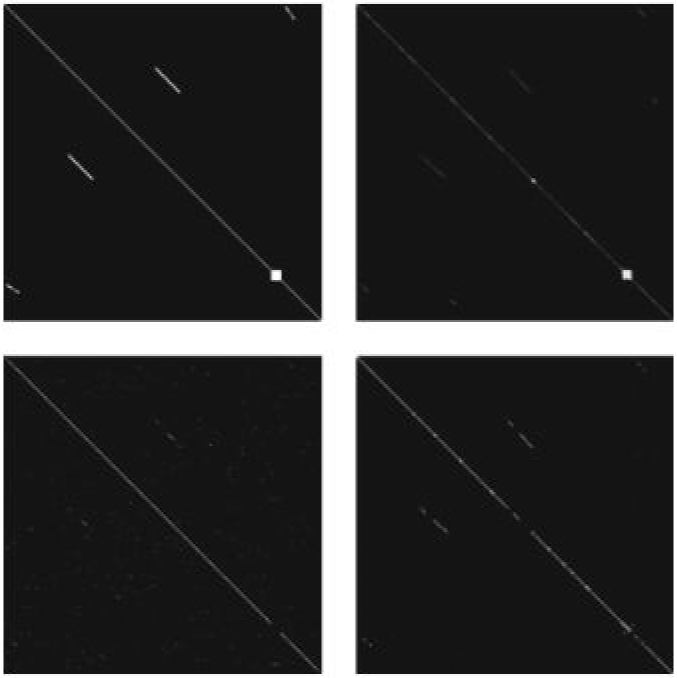


**Fig. 2.** Loop Closure Detection: ground truth matrix (top left) matrices for proposed methodology (top right), OpenSeqSlam [15] (bottom left) and OpenFabmap [4] (bottom right).

From Figure 2, it is clearly shown that like the previously proposed methods, the methodology proposed by authors detects the loop closures successfully. OpenSeqSlam suffers from over detection. Significant miss is present in both OpenSeqSlam and OpenFabmap. In comparison to this, miss and false detection both are less for the methodology proposed in the research paper. The comparison was done by thresholding and then comparing it with ground truth matrix pixel wise.

For our experiments and project, we followed a slightly different approach which is discussed below.

1. All the necessary modules were imported.

```
import os
import json
from glob import glob
import numpy as np
import seaborn as sns
import scipy.io as sio
import tensorflow.compat.v1 as tf
import matplotlib.pyplot as plt
from tensorflow.python.platform import gfile
from IPython.display import Image
```

2. We first took all the files in jpg format and traversed through them.

```
filenames                                                              =
sorted(glob('C:\\Users\\ajink\\Downloads\\ML\\05\\image_0\\*.jpg'))[
:]
representations = []
```

3. We used a pre-trained model named as ‘classify_image_graph_def.pb’ for feature extraction.

```
with gfile.GFile('classify_image_graph_def.pb', 'rb') as f:
    graph_def = tf.GraphDef()
    graph_def.ParseFromString(f.read())
```

4. We used a pre-defined layer of tensorflow named as ‘inception’ which was used to imitate the Deconvolution Net proposed in the paper.

```
def forward_pass(fname, target_layer = 'inception/pool_3:0'):
    g = tf.Graph()
    image_data = tf.gfile.FastGFile(fname, 'rb').read()
    with tf.Session(graph = g) as sess:
        tf.import_graph_def(graph_def, name = 'inception')
        pool3 = sess.graph.get_tensor_by_name(target_layer)
         pool3 = sess.run(pool3, {'inception/DecodeJpeg/contents:0':
image_data})
        return pool3.flatten()

for fname in filenames:
    print(fname)
    frame_repr = forward_pass(fname)
    representations.append(frame_repr.flatten())
```

5. Ground Truth Poses of Sequence 05 from KITTI Dataset was used in mat extension and loaded using scipy.io, which was further plotted on the graph.

```
fig, (ax1, ax2) = plt.subplots(ncols = 2)
default_heatmap_kwargs = dict(xticklabels = False, yticklabels =
False, square = True, cbar = False)

GROUND_TRUTH_PATH                                                    =
os.path.expanduser('C:\\Users\\ajink\\Downloads\\ML\\kitti05GroundTr
uth.mat')
gt_data = sio.loadmat(GROUND_TRUTH_PATH)['truth'][:]
sns.heatmap(gt_data, ax = ax1, **default_heatmap_kwargs)
ax1.set_title('Ground Truth')
```

6. The confusion matrix was built from the image frames traversed.

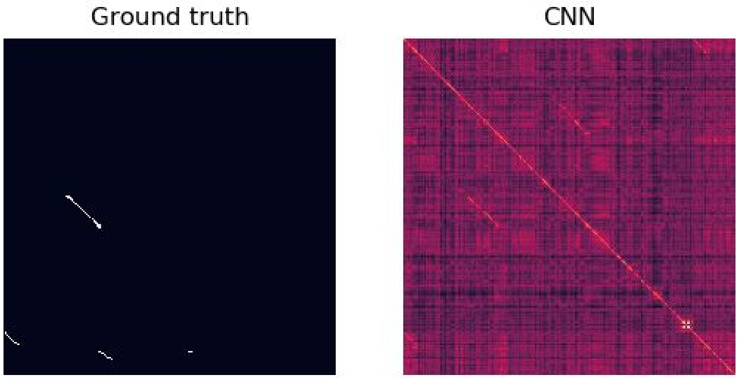
```
def normalize(x):
    return x / np.linalg.norm(x)

def build_confusion_matrix():
    n_frames = len(representations)
    confusion_matrix = np.zeros((n_frames, n_frames))
    for i in range(n_frames):
        for j in range(n_frames):
            print(i, j)
            confusion_matrix[i][j] = 1.0 - np.sqrt(1.0 -
np.dot(normalize(representations[i]),
normalize(representations[j])))
    return confusion_matrix

confusion_matrix = build_confusion_matrix()
sns.heatmap(confusion_matrix, ax = ax2, **default_heatmap_kwargs)
ax2.set_title('CNN')
```

7. The final graph was saved in a PNG file.

```
fig.show()
fig.savefig('KITTI.png')
```



# Application of Project

Place recognition a number of applications, ranging from autonomous driving and robot navigation to augmented reality and geo-localizing archival imagery. These applications are explained below.

- Self Driving Car : A self driving car, if capable of learning the places that it has already visited can build for itself a personal map with a 3-dimensional view.
- Unmanned Aerial Vehicles (UAVs) : These drones and vehicles, if get the ability to recognise different locations, can be expected to reach their destination without any human aid.
- Robot Vacuum Cleaners : These robots are programmed to check every location for dirt and dust. If they get the feature of learning places/rooms that they have already visited, they need not work in the same room more than once.
- Gaming : Many games include bots which are also an important part of the story. By giving them this knowledge of already visited places, we can make the games more challenging for people by increasing the competition.

# Conclusion

The authors have proposed a deep learning autoencoder network that can represent an image frame with significantly lower dimension. The contextual and spatial information is considerably preserved. Therefore, such representation is useful for applications like loop closure detection in SLAM. In the proposed approach, the authors tried to integrate the best of both the previously available approaches (unsupervised feature learning in DAs and weight sharing in CNNs) in a deconvolution net (composed of deconvolution and unpooling layers).

The main advantage of this approach is that the vectors generated for two frames of the same scene which are different geometrically but are similar contextually and by content, are quite close to each other.

Thus, the proposed approach works well in general place recognition tasks also and can be extended to context and content based image matching problems.