

# Unsupervised learning to detect loops in Visual SLAM with optimized SDA

Suman Laha

*GURU GHASIDAS VISHWAVIDYALAYA, CSIT DEPARTMENT*

*KONI, BILASPUR, CG - 495009*

*sumanlaha214@gmail.com*

**Abstract-** This paper is concerned of the loop closure detection issue for visual concurrent confinement and mapping frameworks. We propose a novel approach in light of the improved stacked denoising auto-encoder (SDA), a multi-layer neural system that self-sufficiently takes in a compacted portrayal from the information in an unsupervised way. Different with the customary pack of-words based techniques, the profound system can take in the complex inward structures in picture information, while never again needs to physically plan the visual highlights. Our approach utilizes the qualities of the advanced SDA to take care of the circle identification issue. The work process of preparing the system, using the highlights and registering the likeness score is displayed. The execution of streamlined SDA is assessed by an examination consider with Fab-map utilizing information from open datasets. The outcomes demonstrate that SDA is practical for distinguishing circles at an acceptable exactness and can in this way give an elective method to visual SLAM frameworks.

**Keywords-** SLAM, SDA, Neural network, Stochastic Gradient Descent

## I. INTRODUCTION

Unsupervised learning is a profound idea that can be drawn closer from altogether different points of view, from brain science and psychological science to designing. It is regularly called "learning without a teacher"[1]. This infers a learning human, creature, or counterfeit framework watches its environment and, in view of these perceptions, adjusts its conduct without being advised to relate offered perceptions to given wanted reactions (managed Learning) or without even given any indications about the decency of a given reaction (Reinforcement learning)[2]. Unsupervised learning is by all accounts the essential system for tactile adjustment, e.g. in the visual pathway [3].

On the building side, it is a profoundly capable and promising way to deal with some functional information handling issues like information mining and learning revelation from extensive databases, or new methods of human-PC associations in which the product adjusts to the necessities and propensities for the human client by watching her conduct [4]. Essentially, the unsupervised neural learning calculations can be categorized as one of two classes [5]. The primary, expansions of the direct change coding techniques for

insights, and second, learning vector coding strategies that depend on focused learning.

The top of the line of unsupervised learning strategies are inspired by standard measurable techniques like PCA or FA, which give a decreased subset of direct mixes of the first information factors[6]. A significant number of the learning rules for PCA depend on the creator's PCA neuron show [7]. A later model in this class is that of autonomous parts, which would maximally lessen the excess between the dormant factors. This prompts the strategies of free segment examination (ICA) and visually impaired source partition (BSS) [8]. In the last procedure, an arrangement of parallel time flags, for example, discourse waveforms, electromagnetic estimations from the cerebrum, or monetary time arrangement, are thought to be straight mixes of fundamental free inert factors. The factors, called autonomous parts, are found by productive ICA learning rules [9].

The approach of fundamental PCA, FA, and ICA. The inferior of techniques is near grouping. An average application is information mining or profiling from huge databases [10]. It is important to discover what sort of common groups there are among the information records [11]. In a client profiling application, finding the bunches from an expansive client database implies all the more strongly focused on showcasing with less cost [12]. In process demonstrating, finding the important groups of the procedure state vector in genuine activity helps in determination and control. A focused learning neural system gives a proficient answer for this issue [13]. The synchronous restriction and mapping (SLAM) is viewed as one of the basic advances in self-governing robot examines and has been widely explored in the previous years[14]. In a run of the mill SLAM framework, the robot will construct a model of nature around it and gauge the direction of itself at the same time [15][16].

## II. RELATED RESEARCHES

Bo Du et.al [17] clarified profound systems have accomplished brilliant execution in taking in portrayal from visual information. In any case, the regulated profound models like convolutional neural system require extensive amounts of marked information, which are exceptionally costly to acquire. To take care of this issue, this paper proposes an unsupervised profound system, called the stacked convolutional denoising auto-encoders, which can delineate to progressive portrayals with no name data. The system, enhanced by layer-wise preparing, is developed by stacking layers of denoising auto

encoders in a convolutional way. In each layer, high dimensional element maps are created by convolving highlights of the lower layer with portions learned by a denoising auto-encoder. The auto encoder is prepared on patches removed from include maps in the lower layer to learn strong component indicators. To better prepare the expansive system, a layer-wise brightening method is brought into the model. Before each convolutional layer, a brightening layer is inserted to circle the information. By layers of mapping, crude pictures are changed into abnormal state include portrayals which would help the execution of the resulting bolster vector machine classifier. The SCDAE, an unsupervised profound system propelled by late component learning designs CNN and a change of the current fruitful system SDAE. SCDAE is built by stacking the DAEs whose parameters are optimized. Eventhough the parameters are gotten productively huge arrangement of database should be characterized.

Kai sun et.al [18] incorporated ELM with auto encoder has turned into another point of view for removing highlight utilizing unlabeled information. In this paper, we propose another variation of outrageous learning machine auto encoder (ELM-AE) called summed up extraordinary learning machine auto encoder (GELMAE) which adds the complex regularization to the goal of ELM-AE. A few examinations completed on genuine informational indexes demonstrate that GELM-AE beats some cutting edge unsupervised learning calculations, including k-implies, laplacian installing (LE), ghastly grouping (SC) and ELM-AE. Besides, we additionally propose another profound neural system called multilayer summed up extraordinary learning machine auto encoder (ML-GELM) by stacking a few GELM-AE to recognize more conceptual portrayals. The above strategy develops better execution and less time. Most indispensable thing that the viability of the framework must be moved forward.

Chen Lu et.al [19] examined a powerful and solid profound learning strategy known as stacked denoising auto encoder (SDA), which is appeared to be reasonable for certain wellbeing state recognizable pieces of proof for signals containing encompassing clamor and working condition variances. SDA has turned into a prevalent way to deal with accomplish the guaranteed favorable circumstances of profound design based powerful component portrayals. In this paper, the SDA-based blame conclusion strategy contains three progressive advances: wellbeing states are first isolated into preparing and testing bunches for the SDA show, a profound various leveled structure is then settled with a transmitting standard of voracious preparing, layer by layer, where sparsity portrayal and information annihilation are connected to acquire high-arrange attributes with better strength in the emphasis learning. Approval information are at last utilized to affirm the blame analysis consequences of the SDA, where existing wellbeing state ID strategies are utilized for correlation. This calculation creates better outcomes. The planning limitations and strength of the framework must be enhanced for promote advancement of process.

Ayush Tewari et.al [20]discussed a novel model-based profound convolutional auto encoder that tends to the exceptionally difficult issue of recreating a 3D human face from a solitary in-the-wild shading picture. To this end, we

join a convolutional encoder connect with a specialist outlined generative model that fills in as decoder. The center development is the differentiable parametric decoder that embodies picture arrangement systematically in view of a generative model. Our decoder takes as information a code vector with precisely characterized semantic implying that encodes nitty gritty face posture, shape, appearance, skin reflectance and scene light. Because of this better approach for consolidating CNN-based with display based face Reconstruction, the CNN-based encoder figures out how to separate semantically important parameters from a solitary monocular input picture. Out of the blue, a CNN encoder and a specialist outlined generative model can be prepared end-to-end in an unsupervised way, which renders preparing on expansive (unlabeled) genuine information achievable. Semantic significance in the code vector is implemented by a parametric model that encodes variety along the posture, shape, articulation, and skin reflectance and enlightenment measurements. Our model-based decoder is completely differentiable and enables end-to-end learning of our system. There is a need of more parameters to be considered and preparing of extensive information is required.

In [17] the large dataset has to be classified. Similarly, in [18] the effectiveness of the process has to be improved. Consequently, in [19] the timing constraints and robustness of the system has to be improved. In addition to that in [20] there is an extended need to consider more parameters and training of large data has to be done. On the whole by considering above issues in the mind a new framework is designed to overcome the issues stated.

### III. PROPOSED METHODOLOGY

Since SDA has not been widely applied in SLAM systems, we will briefly introduce the formulation of SDA at first, and then discuss our modifications. The flow of methodology can be represented as,

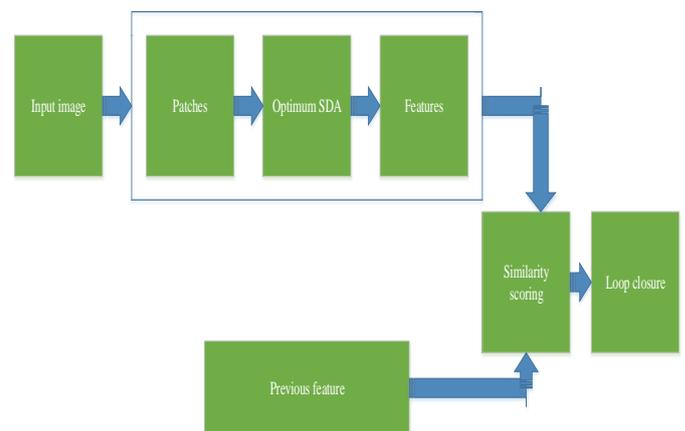


Figure 1: Proposed Methodology Architecture

#### A. SDA auto encoder

SDA is a unsupervised neural network that learns a compressed representation of the input data. It contains several end-to-end layers and each layer is a Denoising Auto-encoder (DA). In SDA, the output of each DA is used as the input in

the next layer. For a single DA, it is consisted of three layers: (1) input layer  $x$ ; (2) hidden layer  $h$ ; (3) recovery layer each layer contains many fully connected nodes which are the basic elements of the network. Each node computes a simple nonlinear function (usually sigmoid) from the connected input. Let  $x$  be its input and  $a$  be the output, the function can be written as:

$$a = \sigma(w^T x + b) = \frac{1}{1 + \exp(-\sum_{i=1}^n w_i x_i - b)} \quad (1)$$

Where  $w$ ,  $b$  are the weight and bias parameters contained in a single node. Notices that the  $w$ ,  $b$  are goals of training. The well trained parameters can grab useful information from the input data. In DA, the hidden layer is fully connected to the input layer, so the output of one hidden unit  $h_j$  are:

$$h_j = f_{o_j}(x) = \sigma(w_j^T x + b_j) \quad (2)$$

$$Y_k = g_{o_k}(h) = \sigma(w_k^T h + b_k) \quad (3)$$

Put it into the matrix form, we have:

$$h = f_o(x) = \sigma(w^T x + b) \quad (4)$$

$$Y = g_o(h) = \sigma(w^T h + b) \quad (5)$$

But  $g_0$  and  $h_0$  should not be identity functions I because such a structure is useless. Finally, notice that in SDA, the hidden layer  $h$  is the real output fed to the latter layers, not  $y$ . During the training,  $f_0$  is expected to grab abstract information from  $x$ . Also, since the initial value of the parameters are randomly set and the dimension of hidden layer is also different with the input and output, the mapping function  $f$  is irreversible and one cannot directly set  $g_0$  to  $f^{-1}$ . The parameters  $W$ ,  $b$  are trained by minimizing the error of construction, which is usually measured as the cross entropy if the input  $x \in [0, 1)$  which is considered to be a drawback.

$$d = kl(x, y) = \sum_{i=1}^n x_i \log \frac{y_i}{x_i} + (1 - x_i) \log \frac{1 - y_i}{1 - x_i} \quad (6)$$

*a. Optimized SDA*

The above drawback is overcome by employing the Stochastic Gradient Descent (SGD) to solve this optimization problem. SGD divides the optimizing (training) process into many small epochs. In each epoch, the parameters are updated by a little step described by a learning rate  $\eta$  along the descent gradient direction:

$$\theta^* = \theta - \eta \cdot \frac{\partial d}{\partial \theta} \quad (7)$$

After several epochs, the object function converges to a local minima and the algorithm I stopped. The recovery layer is then removed and the data in hidden layer is used as the output (or the features). In real environments the input data captured from sensor is noisy and we do not want the noise be learned and represented. Therefore, the denoising auto-encoder (DA) is proposed which tries to reconstruct the data from a corrupted input  $\tilde{x}$ :

$$\min j = KL(x, g_o f_o(\tilde{x})) \quad (8)$$

DA is a very useful extension of auto-encoder. The corruption is usually implemented by randomly masking certain percent of  $x$  into 0 or 1, so the DA will try to predict these missing values. It is observed in previous researches that such a corruption will help the auto-encoder get a more meaningful result which will also be demonstrated in our experiments. DA has been applied in many pattern recognition tasks. Some important issues, including how to set the dimension of the hidden layer  $h$  and the initial value of the parameters, are also investigated and discussed. Generally speaking, if the dimension of  $h$  is smaller than the input  $x$ , the result of auto-encoder is just like a nonlinear principal component analysis. On the other hand, if the dimension of  $h$  is larger, the training process is an over-complete one which will obtain better result.

*b. Optimized SDA to train for loop detection*

In order to train a structure for loop closure detection, we make a few modifications on traditional DAs. Unlike other pattern recognition tasks where the input is independent image, the data in loop closure detection is captured from the sensor, usually real-time video frames. Therefore, we can take advantage of the assumption that the observation of the sensor is continuous. To clarify how the modification is added, we first introduce the whole workflow of our approach. The schematic overview of our approach is illustrated in Fig. 2.

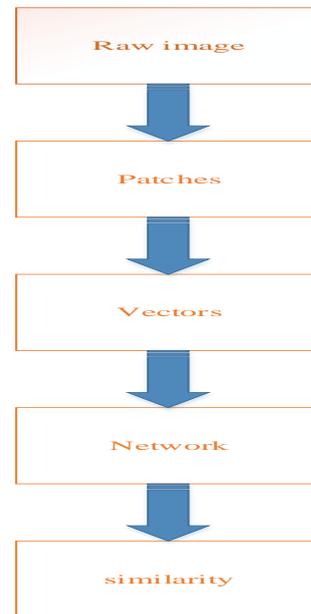


Figure 2: Schematic overview of approach

The raw observation data, i.e., gray-scale images, are first divided into small patches with size of  $s \times s$ . The patches are computed by sparse key-point detection algorithms like SIFT, FAST or ORB and are then filtered to spread over the whole image. We sort the detected key-points in the descending order of the feature response, select the first  $N$  key-points and then resize them into small image patches. The patches are vectorized and then fed to the neural network. Therefore, one input image will have  $N$  patches which form an input matrix

$XN \times s2$ . Then, SDA corrupts the input and trains a structure to reconstruct it. The final hidden layer of SDA is used as feature output layer with the dimension of NF. Therefore, once a new image comes, we can get a feature response  $ZN \times NF$  from SDA, which is used for detecting loops.

The loops are detected if we find the similarity score of two frames exceed a given threshold. In the visualized similarity matrix image, a bright block indicates there may be a loop in this area. Therefore, when a new key-frame comes, what we need to do is put it into well-trained SDA, get the feature response, compute the similarity with previous key frames, and then check if there is a loop. Regarding of this, our method is an online detection algorithm.

### B. The training dataset

Note that in order to check the loops correctly, we need a well-trained network before detecting the loops. So in these experiments we put all the images of key-frames into optimized SDA and compute the similarities, which can be viewed as off-line loop closure detection.

#### a. Pre-process of the data

The open dataset provided by has many sequences with ground-truth trajectories. Figure 4 shows the trajectory, key-frames and sample images from the dataset "freiburg\_office" where a Kinect is hold around a desk and moved to the original position at last. The trajectory of the Kinect is captured by an extrinsic tracking camera system with time-stamps. The odometry information comes at 100 Hz and the speed of video is 30 Hz. The tum dataset is designed for verifying SLAM systems and do not provide ground truth loops, so we need to compute the true loops by ourselves. In a SLAM system, key-frames are added after the registration algorithm finding the motion of camera has exceeded a certain threshold. Therefore, we divide the trajectory into a series of segments and record the ground-truth poses of key-frames:  $T_i$ ,  $i = 1, \dots, N$ . The frames between these key-frames are discarded. Then, the relative distance between each pair of poses ( $T_i$  and  $T_j$ ) is calculated:

$$D_{i,j} = \text{dis}(T_i^{-1}T_j^{-1}) + \text{angle}(T_i^{-1}T_j^{-1}) \quad (9)$$

The function  $\text{dis}(\cdot)$  and  $\text{angle}(\cdot)$  denotes the translation and rotation parts of the transform matrix. If  $D_{i,j}$  is small enough, which means that the position and heading of the camera are close, this key-frame pair is marked as a ground truth loop. The red lines in Fig. 4d show the ground-truth loops in this sequence. The trajectory of physical robot is treated in the same way. The ground-truth loops will be used to compute the precision-recall curve of algorithms.

#### b. Trained structure and loops

The images from the open dataset are fed to optimized SDA. The neural network is implemented using Theano library. In Optimized SDA, the W matrix of the first layer is often regarded as the feature detector because the hidden units compute a dot product of W and x. The column dimension of W is same as x, so it can be conveniently visualized as images with same size as input data. Because of the sparse constraint, many of the hidden units have a low average response. Only parts of them are detecting useful information. The response

of hidden units forms a nonlinear description of the image data. In the visualized figure, there are hole, edge and corner detectors which are learned during the training. Their output is regarded as features which are used for measuring the similarity of the input data. The F1 and F3 are selected from true loops while F2 is a different one. The sparse constraint makes the average response close to 0.05, and the distinctive score function makes the units that have medium response take a higher weight. In we project the feature vectors of the three key frames into 2D plane using PCA. The matches are shown in this figure as lines between the matched features. The width of such lines indicates the similarity score of them. The thick lines show strong relationships of the features while the thin ones are weak. It can be seen that the number of thick blue lines are more than green lines, which means F1 – F3 are much more similar than F1 – F2. The key-frames whose similarity exceeds a certain threshold are considered as possible loops. Hence, the feasibility of Optimized SDA-based loop closure detection is of satisfactory in this experiment.

### C. Defining the similarity

With the cost function  $J^*$  and proper values of balance parameters  $\beta$  and  $\gamma$ , we can obtain a set of hidden units h that grab useful, sparse, representative and denoising information from the input data. In ideal case, similar input (in the feeling of human beings) will have a similar hidden response in the network. Therefore, the obtained h can be used as distinctive features to measure the similarity of input images.

#### a. Computing the similarity

The purpose of loop closure detection is to find the same scenes in the trajectory. Assume that there are two key-frames F(1), F(2), which contain  $k_1, k_2$  features in total:

$$S = V \wedge V^T \quad (10)$$

$$S_r = \sum_{i=k}^N \lambda_i V_i V_i^T \quad (11)$$

Where h is a output of the last layer in SDA. We need to measure the similarity of these two frames. The detailed procedure is described in Algorithm 1.

<p>1 Computing the Similarity  <b>input:</b> Key-frames: F(1), F(2);  <b>output:</b> Similarity score: S;  1: Set <math>S = 0</math>;  2: Compute the average response h.  3: Compute the distinctive score of each hidden units:  <math>\delta_i = \phi(h_i)</math>  4: Match the features:  <math>M = \{mk   mk = (h(1)i, h(2)j), 1 \leq i \leq k_1, 1 \leq j \leq k_2\}</math>  5: For <math>m_k</math> in M  6: Compute the weighted distance of features:  <math>\delta_k = \ \delta^T (h_i^{(1)} - h_j^{(2)})\ </math>  7: Add to the similarity score:  <math>S += \pi(sk)</math>  8: end for  9: return S</p>
---

Figure 3: Algorithm for computing similarity

The algorithm has two parts. First, we compute the average response  $\bar{h}$  of each unit in SDA. If an unit is always activated (has a very high average response), it is likely to be an ordinary feature, such as a black block that may appear at floors, walls or the back of a chair. On the other hand, a unit that has very low response may have no useful information but only noise. Therefore, we prefer the units with medium response and regard them as distinctive features useful for recognition. The distinctive score is defined to measure such a preference, which is described by the function  $\phi(\cdot)$ . As the output of sigmoid function belongs to (0, 1), we choose the Gauss function to compute the score: where  $\mu$  and  $\sigma$  are parameters to obtain a proper shape of Gauss function. The distinctive score  $\delta$  is used as a weight vector, indicating that if the responses of distinctive units are similar, the data is likely to be a loop. Second, we match the features provided by SDA using existing algorithms. In the experiments we make use of the Brute-force match and the fast approximate nearest neighbor (FLANN) implemented in OpenCV library. The match algorithm gives a list of possible matches denoted as  $M$ . For each match, the diversity of features can be measured by the weighted distance in the feature space:

$$\delta_i = \phi(h_i) = \exp\left(-\frac{(h_i - \mu)^2}{2\sigma^2}\right) \quad (12)$$

$$\delta_k = \left\| \delta^T (h_i^{(1)} - h_j^{(2)}) \right\| \quad (13)$$

Finally, we compute an accumulating score using a similarity function  $\pi$ , whose purpose is to keep the score into a reasonable interval and to balance the effect of close and far matches.

#### IV. RESULTS AND DISCUSSION

System configuration:

- Operating System: Windows 8
- Processor: Intel Core i3
- RAM: 4 GB
- Platform: MATLAB

##### A. Data set consideration

In this section, we demonstrate several offline experiments to evaluate the effect of our approach, compared with FabMap 2.0, a well-known BoW based loop closure detection algorithm. The input data is selected from tum open dataset and “New College”, “City Centre” from Fabmap.. Note that in order to check the loops correctly, we need a well-trained network before detecting the loops.



Figure 4: Database images

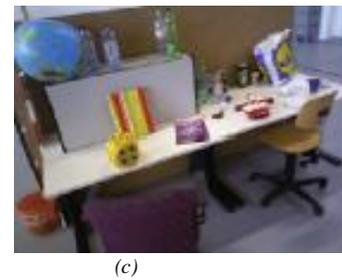
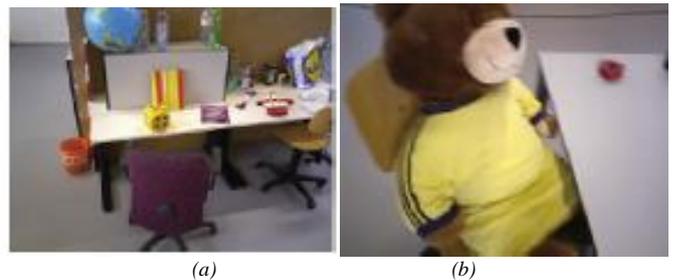


Figure 5: (a) input image  $F_1$ , (b) Image from other environment  $F_2$ , (c) Image that matches with image  $F_1$  and given as  $F_3$ .

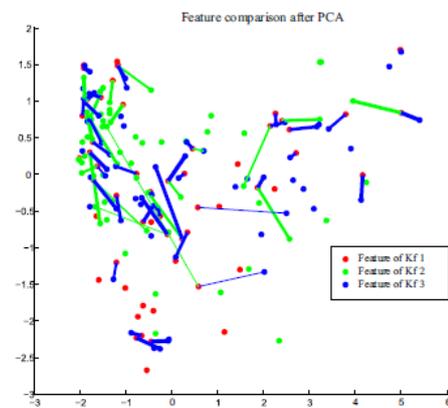


Figure 6: Feature matching of three images

The precision rate of our proposed methodology provides effect with increase in recall intrudes increase in precision rate. The below tabulation 1 defines it

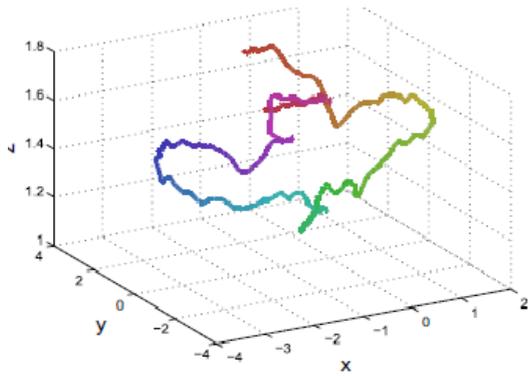
TABLE I  
PRECISION RATE OF OPTIMIZED SDA

precision	Recall
0	1
0.1	1
0.2	1
0.3	1
0.4	1
0.5	1
0.6	1
0.7	1
0.8	1
0.9	1
1	0.3

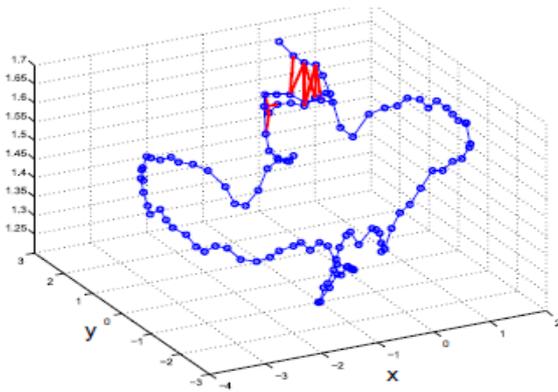
On the basis of above tabulation the graph is plotted the path of curve track in graph gives the precision rate of method



Figure 9: Graph for precision rate of optimized SDA



(a)



(b)

Figure 7: (a) ground trajectory, (b) loop detection in trajectory, red lines indicates loop closure

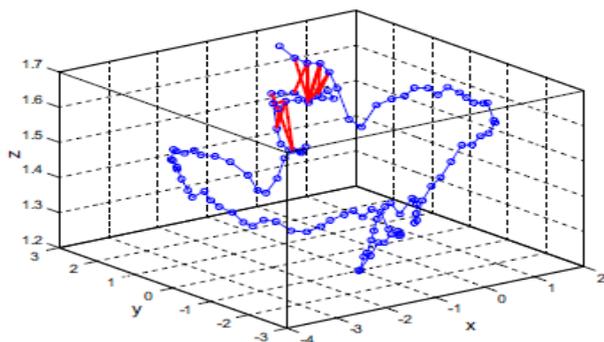


Figure 8: Loop detection using optimized SDA

**B. Performance analysis**

The performance of the system can be analyzed by using three parameters,

- Precision rate
- Training time
- Detection time

(a) Precision rate of optimized SDA

(b) Training time

The time taken to train dataset is given as the training time. The training time of system is decreased which yields better results. The tabulation is given below in table 2.

TABLE II  
TRAINING TIME OF OPTIMIZED SDA

Nodes in hidden layer	Training time(s)
2500	29.02
1500	19.04
1000	14.13
500	9.8

The graph for above tabulation is plotted below given in fig 10.

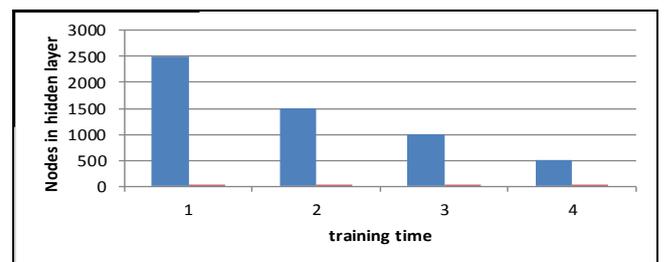


Figure 10: Graph for training time of optimized SDA

(c) Detection time

The time taken to detect the similarity is given by the detection time and the time taken for the detection is decreased for better performance

TABLE III  
DETECTION TIME OF OPTIMIZED SDA

Nodes in hidden layer	Detection time(s)
2500	0.327
1500	0.208
1000	0.174
500	0.16

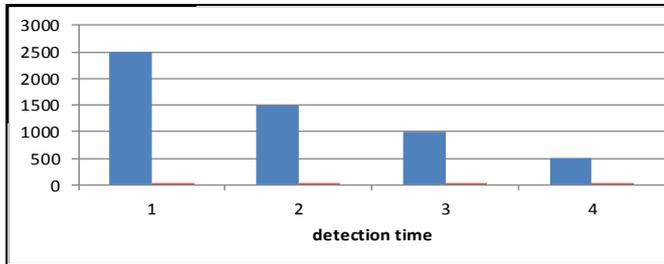


Figure 11: Graph for Detection time of optimized SDA

C. Comparison results

(a) precision rate of Fab-map and SDA

TABLE IV  
PRECISION RATE OF FAB-MAP AND SDA

precision	Recall(Fab-map)	Recall(SDA)
0	1	1
0.1	0.7	1
0.2	0.4	1
0.3	0.3	1
0.4	0.27	1
0.5	0.175	1
0.6	0.1	1
0.7	0.75	1
0.8	0.5	1
0.9	0.4	1
1	0	0.4

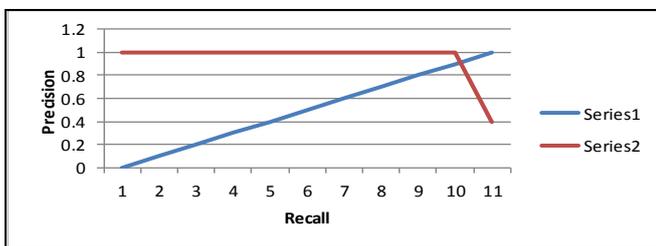


Figure 12: Graph for Precision rate of SDA and fab-map

(b) precision rate of SDA and optimized SDA

TABLE V  
PRECISION RATE OF OPTIMIZED SDA AND SDA

precision	Recall(Optimized SDA)	Recall(SDA)
0	1	1
0.1	1	1
0.2	1	1

0.3	1	1
0.4	1	1
0.5	1	1
0.6	1	1
0.7	1	1
0.8	1	1
0.9	1	1
1	0.3	0.4

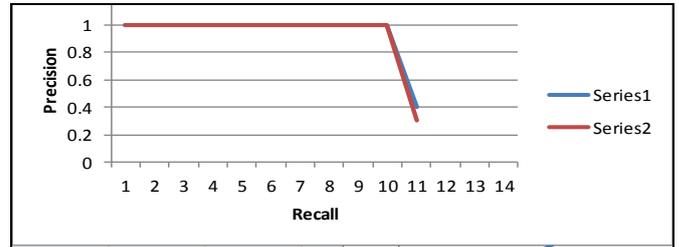


Figure 13: Graph for Precision rate of optimized SDA and SDA

(c) Comparison of precision rate of Fab-map, SDA, optimized SDA

TABLE VI  
THE PRECISION RATE OF OPTIMIZED SDA, SDA AND FAB-MAP

Precision	Recall(Optimized SDA)	Recall(SDA)	Recall (Fab-map)
0	1	1	1
0.1	1	1	0.7
0.2	1	1	0.4
0.3	1	1	0.3
0.4	1	1	0.27
0.5	1	1	0.175
0.6	1	1	0.1
0.7	1	1	0.75
0.8	1	1	0.5
0.9	1	1	0.4
1	0.3	0.4	0



Figure 14: Graph for precision rate of optimized SDA, SDA and Fab-map

D. Result analysis

Thus by utilizing Optimized SDA for visual SLAM systems the precision rate is increased such that the correctness or accuracy of our methodology is validated. Similarly, by employing this method the training time for dataset and detection time is tends to be reduced which additional advantage to our method. Thus these parameters evaluate the better performance of our system.

## V. CONCLUSION

This paper is concerned of the loop closure detection problem for visual SLAM systems. We propose a method that takes advantage of the stacked auto-encoder (SDA), a kind of well-studied deep neural network, to learn a nonlinear representation of the raw input data. The network is trained in the unsupervised way and represent the data by the response of hidden layers, which is used to compare the similarity of images. Finally, we use the similarity matrix to check the possible loops in the video sequence. Better precision rate is achieved through that better performance is evaluated.

## REFERENCES

- [1] N. Srivastava, E. Mansimov, and R. Salakhudinov. "Unsupervised learning of video representations using lstms." In International conference on machine learning, pp. 843-852, 2015.
- [2] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, & A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," IEEE International Conference on Robotics and Automation (ICRA), pp. 3357-3364, 2017.
- [3] M.A. Alsheikh, S. Lin, D. Nivato, and H.P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications." IEEE Communications Surveys & Tutorials, vol. 16, no. 4, pp.1996-2018, 2014.
- [4] G. Huang, S. Song, J.N. Gupta, and C. Wu. "Semi-supervised and unsupervised extreme learning machines." IEEE transactions on cybernetics, vol. 44, no. 12, pp.2405-2417, 2014.
- [5] M. Långkvist, L. Karlsson, and A. Loutfi. "A review of unsupervised feature learning and deep learning for time-series modeling," Pattern Recognition Letters, vol. 42, pp.11-24, 2014.
- [6] Seuret, Mathias, et al., "Pca-initialized deep neural networks applied to document image analysis," Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on, vol. 1, 2017.
- [7] K. Kourou, T.P. Exarchos, K.P. Exarchos, M.V. Karamouzis, and D.I. Fotiadis. "Machine learning applications in cancer prognosis and prediction." Computational and structural biotechnology journal, vol. 13, pp.8-17, 2015.
- [8] L. Bottou. "From machine learning to machine reasoning." Machine learning, vol. 94, no. 2, pp.133-149, 2014.
- [9] A. Tripathy, A. Agrawal, and S.K. Rath. "Classification of Sentimental Reviews Using Machine Learning Techniques," Procedia Computer Science, vol. 57, pp.821-829, 2015.
- [10] Gao, Xiang, and Tao Zhang, "Unsupervised learning to detect loops using deep neural networks for visual SLAM system," Autonomous Robots, vol. 41, no. 1, pp. 1-18, 2017.
- [11] J. Behmann, A.K. Mahlein, T. Rumpf, C. Römer, and L. Plümer. "A review of advanced machine learning methods for the detection of biotic stress in precision crop protection," Precision Agriculture, vol. 16, no. 3, pp.239-260, 2015.
- [12] K. Lai, L. Bo, and D. Fox. "Unsupervised feature learning for 3d scene labeling. In Robotics and Automation (ICRA)." 2014 IEEE International Conference on pp. 3050-3057, 2014. May.
- [13] M. Nickel, K. Murphv, V. Tresp, and E. Gabrilovich. "A review of relational machine learning for knowledge graphs," Proceedings of the IEEE, vol. 104, no. 1, pp.11-33, 2016.
- [14] B. Agarwal, and N. Mittal. "Text classification using machine learning methods-a survey." In Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012), pp. 701-709, 2014.
- [15] Buyval, Alexander, Ilya Afanasyev, and Evgeni Magid, "Comparative analysis of ROS-based monocular SLAM methods for indoor navigation," Ninth International Conference on Machine Vision (ICMV 2016), Vol. 10341, 2017.
- [16] A. Holzinger. Interactive machine learning for health informatics: when do we need the human-in-the-loop, Brain Informatics, vol. 3, no. 2, pp.119-131, 2016.
- [17] H.B. Kazemian, and S. Ahmed. "Comparisons of machine learning techniques for detecting malicious webpages." Expert Systems with Applications, vol. 42, no. 3, pp.1166-1177, 2015.
- [18] J. Lu, S. Young, I. Arel, and J. Holleman, "A 1 TOPS/W analog deep machine-learning engine with floating-gate storage in 0.13  $\mu\text{m}$  CMOS," IEEE Journal of Solid-State Circuits, vol. 50, no. 1, pp.270-281, 2015.
- [19] G. Gautam, and D. Yadav, "Sentiment analysis of twitter data using machine learning approaches and semantic analysis." In Contemporary computing (IC3), 2014 seventh international conference on, pp. 437-442, 2014.
- [20] Zhao, Hang, et al., "Loss functions for image restoration with neural networks," IEEE Transactions on Computational Imaging, vol. 3, no. 1, pp. 47-57, 2017.