Bare Demo of IEEEtran.cls for IEEE Journals

Michael Shell, *Member, IEEE,* John Doe, *Fellow, OSA,* and Jane Doe, *Life Fellow, IEEE*

Abstract—We present a real-time indoor relocation system based only on 2D-laser sensor. Our system processed 2D laser data into multi-channel image and occupancy grid map, which were used as the input of a Convolutional Neural Network(CNN). Our system trains a convolutional neural network to regress the 3-DOF robot pose from a channel-superposition image by multi-channel images and occupancy grid maps with no need of additional optimisation. Our algorithm can obtain approximately 2m and $5°$ accuracy indoors, in which our system distributes particles depend on particle filter algorithm. Our system only uses five layers convolution and four layers fully connected network structure, which can achieve the effect of rapid training.

Index Terms—Relocation, 2D-laser data, Multi-channel image, Occupancy grid map

I. INTRODUCTION

OVer the past few decades, Indoor robotics have made an amazing improvement. Indoor robots rely on lasers alone have been able to navigate smoothly [1], [2]. In traditional method, particle filter algorithm is used in the task of location and navigation indoors [3], [4]. However the problem of robot lost or kidnapped is to hard for traditional method to solve. One of the main techniques to solve those kind of problem is to let robot know it's pose approximately only depends on the sensor when the robot is started for the first time.

In recent years, there are more and more approaches aiming at robot relocation indoors. In [5], an idea of extending recent advances in key point recognition was presented, however, this is no information when robot start for the first time. A hypotheses was proposed in [5], which found with relocation procedure can be used to initialize a generic localization algorithm before the robot starts moving, or when the robot gets lost. However, traditional methods can not solve the problem of robot relocation in an excessively similar environment. Popular past works have considered posture regression. In [6], a network structure based on Googlenet [7] is used to regress a 6-DOF pose.The image obtained based on depth camera contains more feature points, however, this method can not be used under the condition of insufficient illumination, and it consumes a lot of computation when training the network structure based on Googlenet. Similar ideas are also introduced in [8], which is able to recover from complicated kidnap scenarios and failures online in real-time.

In order to solve the kidnapped robot problem when robot restart from any point in the similar indoor environment,
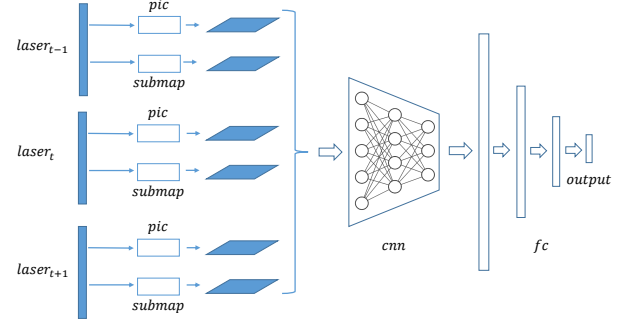
Fig. 1. System overview of the proposed relocation algorithm for robot relocalization indoors. The laser data is processed into RGB image and occupancy grid map. The image and map were superpose into 6-channel image,while 3 6-channel images,which represent laser data t-1,t,t+1respectively, were superpose into a 12-channel image.The 12-channel image is used as the input of a convolutional neural network(CNN),while the output is a 3-DOF posture including pose and orientation.

we introduce a novel relocation algorithm in this paper. Our proposed system, Relocation-Net, takes a multi-channel image and regresses the robot's 3-DOF pose relative to a scene. Fig.1 demonstrates the system structure. The algorithm is simple in the fact that it consists of a convolutional neural network(CNN) trained to regress the robot's orientation and position.

The paper's main contribution are the 2D-laser data processing method and the deep convolutional neural network robot pose regressor. We introduced a novel technique to process the 2D-laser data, which makes it possible for indoor robot to regress pose relying only on 2D-laser sensor. Our main contributions are as follows:

- A novel method to process 2D-laser data in order to increase the features of the input of CNN, which is so hard that it is impossible to regress only depend on 2D-laser data.
- A new method to regress robot 's pose indoor. To the best of our knowledge, this is the first algorithm to regress robots 3-DoF pose depends on the 2D-laser merely.

The proposed deep learning based robot's pose regression system can relocate the robot approximately 2m far away from the real pose of robot when robots start for the fist time indoors, while particle filter algorithm is able to operate without the kidnapped robot problem because the area to distribute particles is constrained to within 2m from the true pose.

The rest of this paper is organized as follows. Section

II reviews related work. Following a system overview in Section III, Section IV presents the proposed 2D-laser data process method. . In Section V, the CNN based robot pose regression method is introduced to refine poses predicted from the network. Experimental results are given in Section VI. The conclusions are drawn in Section VII.

II. RELATE WORKS

Robot relocation aims for let robot know where it is. Classical methods tend to use the information of surrounding environment observed by sensors. In this paper, we focus on using 2D-laser data merely to regress robot 3-DOF posture including pose and orientation.

A. Robots Relocation Indoors

In resent years, researchers have made a lot of effort to try to optimize the performance of robot relocation indoors. Serval sensors were taken into consideration. Sonar data is used in [9], which presented a method for relocation of a mobile robot indoors using sonar data. In this paper a physically-based sonar sensor model was used to characterize the geometric constraints provided by echolocation measurements of different types of objects. Individual range returns were used as data features in a constraint-based search to determine the robot's position. However,the sonar detection range is directly limited by the band loss (high-frequency radar must be used to detect the distance), and pedestrians are not perceived, and it is impossible to accurately simulate all surrounding obstacles. Paper [10] proposed an offline map construction algorithm based on map elements and key frame database which designed a mobile robot fast positioning method. Similar ideas are also introduced in [11], this paper explored the correlation between geometric entities from the perspective of map composition, and showed that maintaining the correlation and hypothesis independence between map entities can reduce these uncertainties which can obtain more accurate estimation in robot relocation. Robot relocation based ORB-SLAM2 [12] is used and improved in [13], [14], [15], [16]. Robot relocation based on orb-slam algorithm can extract more observation features and has better robustness. However, such algorithms not only consume a lot of computing resources, but also have very strict requirements on lighting conditions.

2D-laser based robot indoors location technique was presented in [17], considerable techniques are introduced with respect to the state of the robot indoors, which including Kalman Filter [18], Extended Kalman Filter [19], Unscented Kalman Filter [20]. Moreover, paper [21] proposed a particle filter-based monocular SLAM algorithm which could avoid linearization appear in [17], [18], [19], [20]. The particle filter algorithm implements recursive Bayesian filtering by a non-parametric Monte Carlo [22] simulation method, which estimates the indoor robot's pose go through initializing the particle, calculating each particle's weight, and particle resampling. Particle filtering algorithm is a traditional algorithm with strong robustness, but the robot kidnapped in similar environment is fatal.
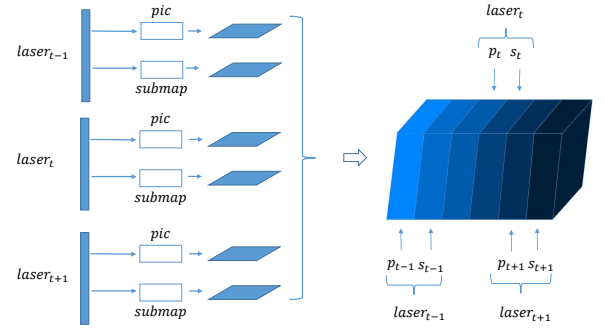

Fig. 2. System overview of 2D-laser processor We process three frames of continuous laser datas,every single-frame 2D-laser data was processed into a RGB image and an occupancy grid map.The image and map were superpose into 6-channel image,while 3 6-channel images,which represent laser data $t-1$,t,$t+1$respectively, were superpose into a 18-channel image.The 18-channel image is used as the input of a convolutional neural network(CNN).

B. Deep Learning based Robots Relocation Indoors

Most of existing deep learning based robot relocation methods are vision based [6], [8], [23]. Other machine learning techniques are employed for the loop-closure problem [24] [25]. Paper [26] proposed a 2d scan matching method based on deep learning, in which using the Hector SLAM [27] to achieve a good result. However, the benefits of deep learning for relocation depends on 2D-laser merely have not been fully explored.

III. SYSTEM OVERVIEW

In this section, an CNN based robot indoors pose estimation system is briefly described. Fig. 1 shows the proposed system framework. It consists of two parts: a 2D-laser data processor and a CNN based robot pose estimation.

The CNN based pose estimation system is composed of a laser data processor with a CNN based point cloud feature extraction. The laser processor is designed to process 2D-laser data into images and occupancy grid maps, which is a novel technique to add features of the input of CNN. A CNN is designed to extract features from multi-channel images processed form laser data. Meanwhile, particle filter algorithm is not abandoned in this paper, which is used to distribute particles within a certain range around the position predicted by our system.

IV. CNN BASED RELOCATION SYSTEM

This section presents the proposed CNN based system for robot indoors relocation in detail. The cost functions designed for training the CNN are also described.

A. 2D-laser Processor

2D-laser laser is used as observation data in particle filter algorithm. Its robustness and characteristics that are not affected by the light source are its advantages as observation data in traditional algorithm. Howerer, single-frame 2D-laser data has fewer features, making it impossible to accurately regress the
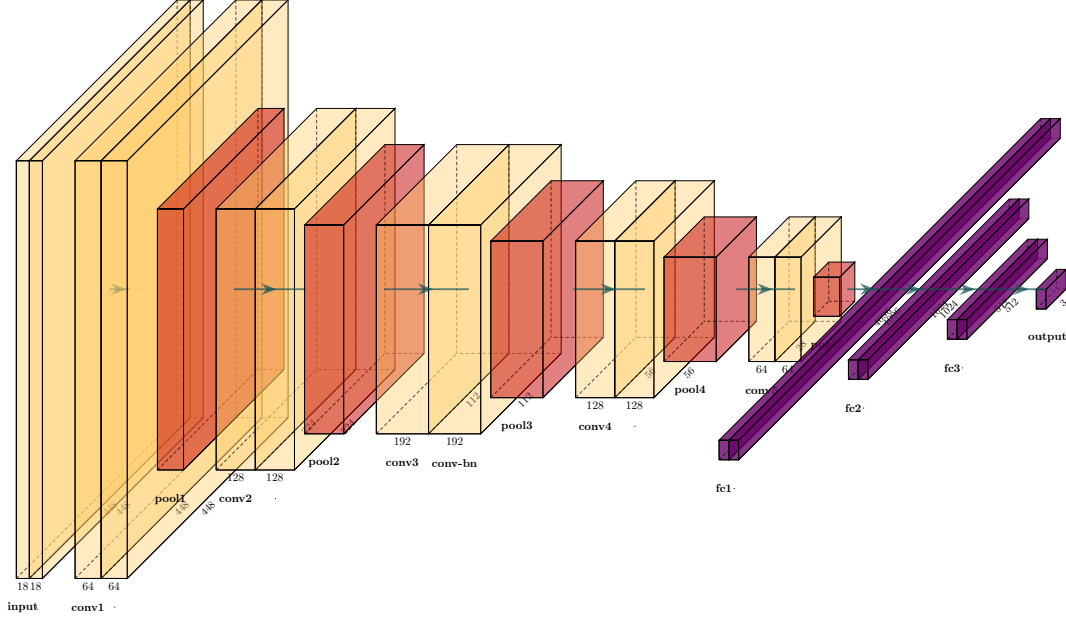
Fig. 3. Architecture of the CNN network in proposed data fusion system. The network consists of five 2D convolutional layers with three fully connected layers. The output of the network is the 3-DOF posture including pose and orientation

robot pose as an input of the deep network [26]. We presented a novel way to process 2D-laser data to increase its features significantly.

B. CNN based Pose Regression

In the reason that convolutional layers can be interpreted as transforming inputs into feature representation effectively, a neural network, in this paper, was designed to learn useful features representation of the multi-channel image automatically. The features extracted form multi-channel image went through five convolutional layers and was pulled into 1 dimension. The fully connected layers followed map the learned distributed feature representation to the sample tag space. The output of the last fully connected layers is a $1*3$ vector including $1*2$ pose and $1*1$ orientation.

TABLE I
CONFIGURATION OF THE CNN

Layer	Kernel Size	Padding	stride	Nunber of Channerls
conv1	7	SAME	2	64
conv2	5	SAME	2	128
conv3	5	SAME	2	192
conv4	3	SAME	2	128
conv5	3	SAME	2	64

The configuration of the CNN is outlined in TABLE I. The input of the network is a multi-channel image processed form 2D-laser data. There are five convolutional layers along with five pooling layers, each of which has a rectified linear unit (leaky-ReLU [28]) activation. Following the last convolutional layer, four fully connected layers are used to extract a global feature and regress the pose we need.

C. Cost Function

The cost function is designed to train the CNN so that its predicted poses are close to the ground truth. Specifically, the cost function contains two parts: a pose error with an angle error.
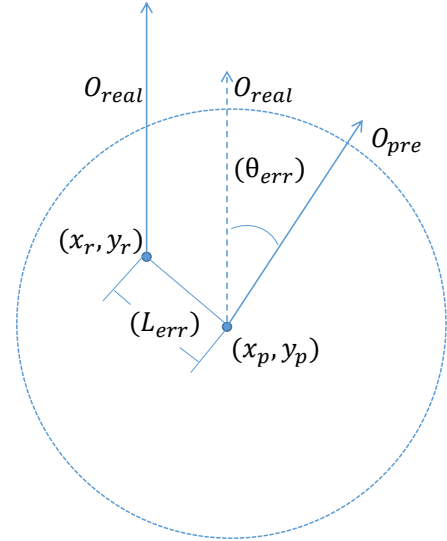

Fig. 4. Illustration of the way to calculate the errors between the predicted location and the ground truth.

1) Pose error : Pose error is formulated as the euclidean distance between the predicted positions the ground truths. The function aims to minimize the euclidean distance between the ground truth poses and the estimated ones from the network.

The error function is as follows:

$$L_{err} = \frac{1}{n}\sum_{i=1}^{n}\{||\hat{x}-x||_2^2 + ||\hat{y}-y||_2^2\} \qquad (1)$$

where $||\cdot||_2^2$ is the 2-norm, and n is the batch size.

2) Orientation error : Two methods are considered to calculate the orientation error in this paper.

a) quaternions: The first way is shown in Formula2,we used the form of quaternion to calculate the error between the predicted value and the ground truth.

$$\Theta_{err} = \frac{1}{n}\sum_{i=1}^{n}\beta||\hat{q} - \frac{\hat{q}}{||q||_2}|| \qquad (2)$$

where $||\cdot||_2^2$ is the 2-norm, and n is the batch size,and q means the quaternion representing the orientation,and β is a factor to balance the weight of positions and orientations.

Quaternions is a good choice to represent orientation, because arbitrary 4-D values are easily mapped to legitimate rotations by normalizing them to unit length.

b) angle: The second way is shown in Formula3,we used form of angle to calculate the error between the predicted value and the ground truth.

$$\Theta_{err} = \frac{1}{n}\sum_{i=1}^{n}\beta||\hat{\theta} - \theta||_2^2 \qquad (3)$$

where $||\cdot||_2^2$ is the 2-norm, and n is the batch size,and θ means the angle representing the orientation,and β is a factor to balance the weight of positions and orientations.

In this paper we describe the convolutional neural network (convnet) we train to estimate robot pose directly from a multi-channel image image. Our network outputs a pose vector V_o, given by a 2D robot position x and orientation represented by angle θ:

$$V_o = [x, \theta] \qquad (4)$$

We chose angles as our orientation representation, because arbitrary 1D values are easily mapped to legitimate rotations in 2D space. This is a simpler process than the 4D values normalization and orthonormalization required of rotation matrices.

V. EXPERIMENT RESULTS

In this section, we evaluate the performance of the proposed system. The CNN model is trained by using data collected from a true environment, and tested on a real Pioneer 3-DX robot running in the same environments. As we know,the particle filter algorithm derives the pose of the indoor robot through four steps, which including particle initialization, particle motion update, particle observation update and calculate the posture in the end [17]. Here we use the posture calculated from the last step of the particle filter algorithm as the ground truth of CNN and use the single-frame laser data which is used to calculate the posture as the data to be preprocessed into multi-channel image.
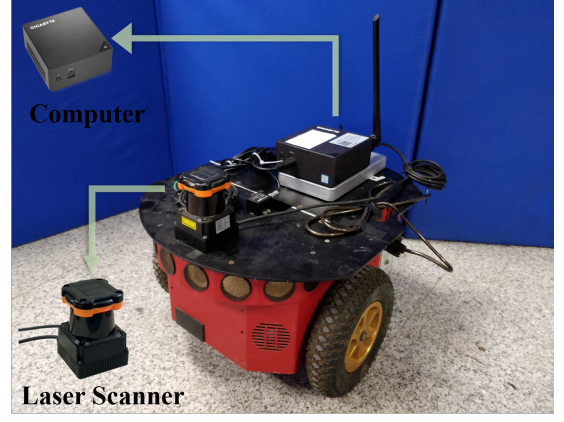

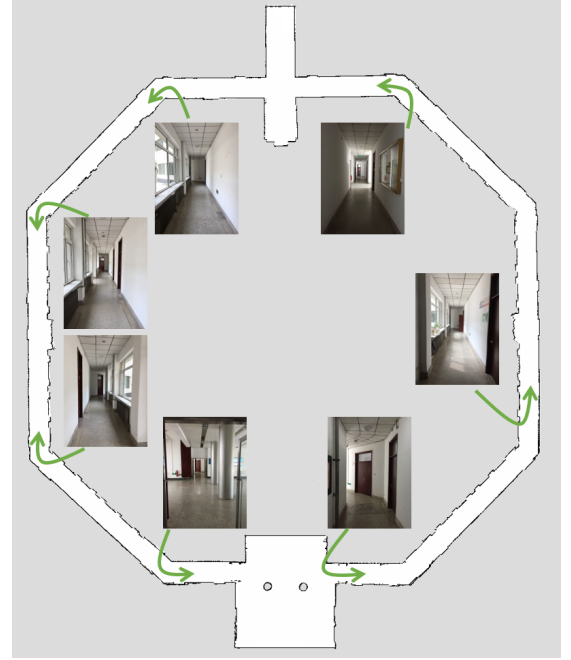Fig. 5. The setup of the mobile robot equipped with a Hokuyo laser scanner.

Fig. 6. Priori map I of the real environment used to collect the data set.The white pixels represent the area without obstacles,the black pixels represent areas with obstacles and the gray pixels represent unknown area.

Fig. 7. Priori map II of the real environment used to collect the data set.The white pixels represent the area without obstacles,the black pixels represent areas with obstacles and the gray pixels represent unknown area.
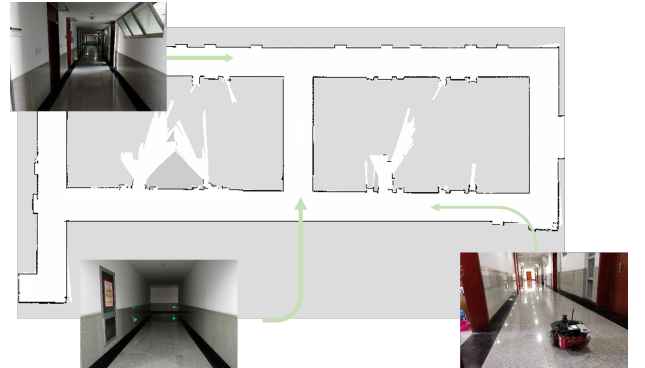
A. Dataset and Training

In order to obtain enough data for training, we gather 2D laser point clouds with poses in a real environment indoors with symmetrical similar structure as is show in Fag. 6. There is a d Pioneer 3-DX robot mounted with a Hokuyo UTM-30LX 2D laser scanner (shown in Fig. 5). The maximum linear and angular velocities of the robot are 0.6m/s and 1.0rad/s, respectively. The maximum measurement range of the laser scanner is 30.0m, and wide angle is 270. The ground truth positions $[x, y, \theta]^T$ and the range measurements are recorded at the speed of 3Hz. The environment used to collect the data sets is as shown in Fag. 6.

To validate that the convnet is regressing pose beyond that of the training examples we show the performance of the system in test dataset.The results shows that our performance exceeds this we conclude that the convnet is successfully able to regress pose beyond training examples (shown in Fag.8).
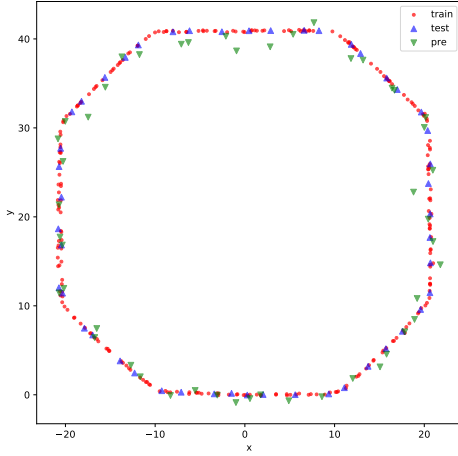

Fig. 8. Magnified view of a sequence of training (red) and testing (blue) data for scene I. We show the predicted robot pose in green for each testing frame. This shows our system can interpolate robot pose effectively in space between training frames.
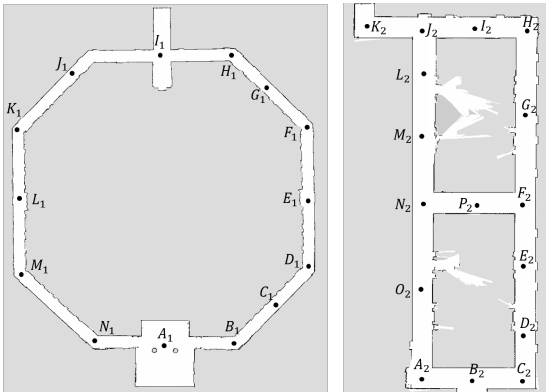

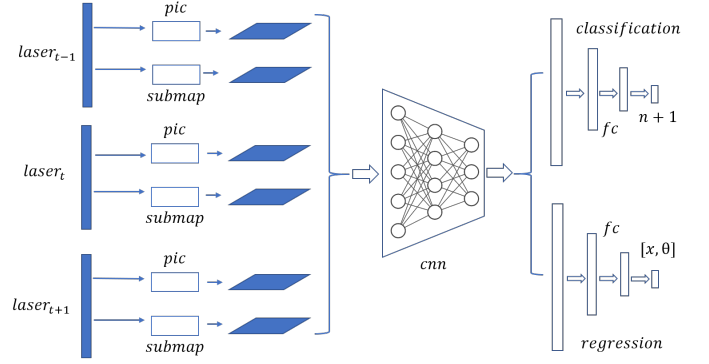Fig. 9. shows the poses selected to test both particle filter algorithm and **Relocation-Net**.

Fig. 10. System overview of the Improved relocation algorithm for robot relocalization indoors. Part regression is used to predict robot pose. Part classification is used to distinguish the scene robot in.
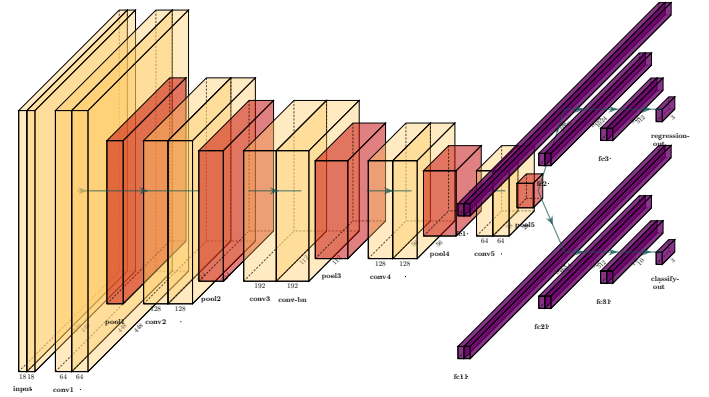

Fig. 11. System overview of the Improved relocation algorithm for robot relocalization indoors. Part regression is used to predict robot pose. Part classification is used to distinguish the scene robot in.

B. System Performance

In order to prove the effectiveness of the our system, we conducted repeated experiments in the locations shown in Fag.9. We tested particle filter algorithm and our particle filter algorithm optimized by our algorithm.

As is shown in TABLE.II, TABLE.III, we tested effectiveness of the our system. We selected test sites in two real scenarios and conducted 10 tests in each location, and the test results were shown in the table. In the process of robot relocation, the traditional particle filter algorithm cannot solve the kidnapped robot problem in the similar structure environment, but our system can avoid the step of global particle distribution in the process of particle filter algorithm, so the kidnapped robot problem in the similar environment can be fundamentally solved.

C. Different Forms of the Input

In order to obtain a stable system and use the least computational resources, we used 3-channel, 6-channel, 12-channel and 18-channel images as the input of the neural network respectively, and compared their average measurement accuracy and computational resource consumption on the same test set(Shown in Table.IV and Table.V).

TABLE II
SYSTEM PERFORMANCE IN ENVIRONMENT I

Site A_1-N_1		Way of Testing	Testing Time Each Site				A_1-N_1		Test Success Rate				
Particle Filter Algorithm		Rotate in Situ	10						2/140				
Relo-Net + Particle Filter		Rotate in Situ	10						139/140				

	A_1	B_1	C_1	D_1	E_1	F_1	G_1	H_1	I_1	J_1	K_1	L_1	M_1	N_1
Particle Filter Algorithm	1/10	0	0	0	0	0	0	0	1/10	0	0	0	0	0
Relo-Net + Particle Filter	1	1	1	1	9/10	1	1	1	1	1	1	9/10	1	1

TABLE III
SYSTEM PERFORMANCE IN ENVIRONMENT II

Site A_2-P_2		Way of Testing	Testing Time Each Site				A_2-P_2		Test Success Rate					
Particle Filter Algorithm		Rotate in Situ	10						6/160					
Relo-Net + Particle Filter		Rotate in Situ	10						158/160					

	A_2	B_2	C_2	D_2	E_2	F_2	G_2	H_2	I_2	J_2	K_2	L_2	M_2	N_2	O_2	P_2
Particle Filter Algorithm	0	2/10	0	0	0	1/10	0	0	0	3/10	0	0	0	0	0	0
Relo-Net + Particle Filter	1	1	1	1	1	1	9/10	1	1	1	1	1	9/10	1	1	1

TABLE IV
DIFFERENT FORMS OF CNN INPUT IN SCENE I

The Input of CNN	Scene	Mini Batch Number	Training Time	Training Frames	Testing Frames	Mean Accuracy on Test Set
3-channel	I	6	10h	4803	4180	5.543m
6channel	I	6	14h	4803	4180	**1.920m**
12-channel	I	6	22h	4803	4180	1.900m
18-channel	I	6	26h	4803	4180	1.912m

TABLE V
DIFFERENT FORMS OF CNN INPUT IN SCENE II

The Input of CNN	Scene	Mini Batch Number	Training Time	Training Frames	Testing Frames	Mean Accuracy on Test Set
3-channel	II	6	9h	4750	3500	4.540m
6channel	II	6	14h	4750	3500	**1.87m**
12-channel	II	6	22h	4750	3500	1.845m
18-channel	II	6	26h	4750	3500	1.834m

D. Multi-scene Mixed Training

It is well known that indoor robot location relies on prior map. In order to verify that our system can not only regress robot posture, but also could identify the scene where the robot is. We improved our system so that it can classify scenarios. Fig.10 shows the improved network architecture, part regression is used to predict robot pose and art classification is used to distinguish the scene robot in. We found that training individual networks to regress position and classify scene separately performed poorly compared to when they were trained simultaneously. We classified and annotated the same number of data sets in scenario 1 and scenario 2, and merged them into a new data set to serve as the input of the improved neural network.

We found that the improved neural network structure had no effect on the accuracy of posture regression, but it increased the training time by about 30 %, and the classification accuracy reached 98.6%.

E. Real-Time Performance

We test our system on a IPC(Industrial Personal Computer) (Intel Core TM Dual Core i7H-6500 and 6GB RAM) for the real-time performance. Our system doesn't need to run in real time, it just needs to predict the robot's position when the robot starts up. Our system only consumes 0.5s without the GPU.

F. Discussion

This paper proposes a method for indoor robot relocation based on convolutional neural networks. Firstly, a 2-D laser data preprocessing method is proposed to increase the position information in the input of neural network , and then the convolutional neural network model is designed to regress the robot pose. This method is based on the improvement of the traditional particle filtering algorithm. In the particle initialization step, the particles are distributed near the real position of the robot, thereby solving the "particle abduction problem" that occurs when the indoor robot restarts at any point. This paper also improves the neural network structure of the design, adding a scene classification module, so that the robot can identify the scene.

Since the subsequent particle filtering algorithm relies too much on the position predicted by the neural network, if the predicted position is wrong, the result of the robot positioning failure will occur.

VI. CONCLUSION AND FUTURE WORK

We present, to our knowledge, the first application of deep convolutional neural networks to relocate robot pose indoors.

We show that the kidnapped robot problem can be avoided during particle initialization. We showed that such networks preserve ample pose information in their feature vectors. Most importantly, we demonstrated that indoor robot repositioning can be achieved by relying only on two-dimensional laser sensors.

In future work, we will consider the issue of sensor fusion. We consider the use of fused data from odometers and two-dimensional lasers or fusion data of sonar and two-dimensional lasers as inputs to neural networks, which will increase the input characteristics of neural networks. It is obvious that a finite neural network has an upper bound on the physical area that it can learn to localize within, which we are going to find this limit.

REFERENCES

[1] Kai Lingemann, Andreas Nüchter, Joachim Hertzberg, and Hartmut Surmann. High-speed laser localization for mobile robots. *Robotics and autonomous systems*, 51(4):275–296, 2005.

[2] Michael Csorba. *Simultaneous localisation and map building*. PhD thesis, University of Oxford Oxford, 1997.

[3] Dieter Fox, Wolfram Burgard, Frank Dellaert, and Sebastian Thrun. Monte carlo localization: Efficient position estimation for mobile robots. *AAAI/IAAI*, 1999(343-349):2–2, 1999.

[4] Farhad Aghili and Chun-Yi Su. Robust relative navigation by integration of icp and adaptive kalman filter using laser scanner and imu. *IEEE/ASME Transactions on Mechatronics*, 21(4):2015–2026, 2016.

[5] Brian Patrick Williams, Georg Klein, and Ian D Reid. Real-time slam relocalisation. In *ICCV*, volume 7, pages 1–8, 2007.

[6] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015.

[7] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[8] Manohar Kuse and Shaojie Shen. Learning whole-image descriptors for real-time loop detection andkidnap recovery under large viewpoint difference. *arXiv preprint arXiv:1904.06962*, 2019.

[9] Jong Hwan Lim and John J. Leonard. Mobile robot relocation from echolocation constraints. *IEEE Transactions on pattern analysis and machine intelligence*, 22(9):1035–1041, 2000.

[10] Guanci Yang, Zhanjie Chen, Yang Li, and Zhidong Su. Rapid relocation method for mobile robot based on improved orb-slam2 algorithm. *Remote Sensing*, 11(2):149, 2019.

[11] José A Castellanos, Juan D Tardós, and Günther Schmidt. Building a global map of the environment of a mobile robot: The importance of correlations. In *Proceedings of International Conference on Robotics and Automation*, volume 2, pages 1053–1059. IEEE, 1997.

[12] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[13] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[14] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014.

[15] Shichao Yang, Yu Song, Michael Kaess, and Sebastian Scherer. Pop-up slam: Semantic monocular plane slam for low-texture environments. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1222–1229. IEEE, 2016.

[16] Raúl Mur-Artal and Juan D Tardós. Visual-inertial monocular slam with map reuse. *IEEE Robotics and Automation Letters*, 2(2):796–803, 2017.

[17] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[18] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[19] Tim Bailey, Juan Nieto, Jose Guivant, Michael Stevens, and Eduardo Nebot. Consistency of the ekf-slam algorithm. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3562–3568. IEEE, 2006.

[20] Ruben Martinez-Cantin and José A Castellanos. Unscented slam for large-scale outdoor environments. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3427–3432. IEEE, 2005.

[21] Robert Sim, Pantelis Elinas, Matt Griffin, Alex Shyr, and James J Little. Design and analysis of a framework for real-time vision-based slam using rao-blackwellised particle filters. In *The 3rd Canadian Conference on Computer and Robot Vision (CRV'06)*, pages 21–21. IEEE, 2006.

[22] Christian Robert and George Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[23] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017.

[24] Peng Yin, Yuqing He, Lingyun Xu, Yan Peng, Jianda Han, and Weiliang Xu. Synchronous adversarial feature learning for lidar based loop closure detection. In *2018 Annual American Control Conference (ACC)*, pages 234–239. IEEE, 2018.

[25] Zetao Chen, Adam Jacobson, Niko Sünderhauf, Ben Upcroft, Lingqiao Liu, Chunhua Shen, Ian Reid, and Michael Milford. Deep learning features at scale for visual place recognition. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3223–3230. IEEE, 2017.

[26] Jiaxin Li, Huangying Zhan, Ben M Chen, Ian Reid, and Gim Hee Lee. Deep learning for 2d scan matching and loop closure. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 763–768. IEEE, 2017.

[27] Stefan Kohlbrecher, Oskar Von Stryk, Johannes Meyer, and Uwe Klingauf. A flexible and scalable slam system with full 3d motion estimation. In *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, pages 155–160. IEEE, 2011.

[28] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.