IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 31, 2020, accepted May 26, 2020, date of publication June 9, 2020, date of current version June 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3001177

A Prediction Method of Localizability Based on Deep Learning

YANG GAO[1,2], (Member, IEEE), SHU QI WANG[1], JING HANG LI[2], MENG QI HU[2], (Member, IEEE), HONG YAO XIA[1], HUI HU[1], AND LAI JUN WANG[1]

[1]School of Automobile, Chang'an University, Xi'an 710064, China
[2]Department of Mechanical and Industrial Engineering, University of Illinois at Chicago, Chicago, IL 60607, USA

Corresponding authors: Yang Gao (nchygy@126.com) and Shu Qi Wang (w18209261604@163.com)

ABSTRACT As a basis of many missions, the accuracy of localization is highly important for mobile robots. For the generally used map matching based localization algorithms, the accuracy of localization, which is described by localizability, is greatly impacted by the environment. Consequently, this paper proposed a novel method to predict the localizability for the map matching based localization algorithms, based on the environment map. Firstly, the uncertainty of localization in map matching and dead-reckoning is analyzed based on which entropy of localization is chosen to describe the localizability instead of the generally used covariance. Next, based upon the flow chart of the map-based localization algorithm, a localizability predictor, which is composed of three different models, is designed to predict the entropy. Here a Convolutional Neural Network (CNN) is designed for the first model to predict the entropy of localization that comes from map matching. A Long Short-Term Memory (LSTM) neural network is designed for the second model to predict the entropy that comes from the dead-reckoning. Finally, a Multilayer fully connected Neural Network (MNN) is designed for the last model to predict the entropy after fusing the entropy results that come from the two models described above. Both simulation results and experimental results have proven that the proposed predictor can offer a better estimator of localizability compared to other existing approaches.

INDEX TERMS Localizability, map matching, mobile robot, deep learning, neural network.

I. INTRODUCTION

As one of the most fundamental functions of mobile robots, localization algorithms are used to estimate the localization of mobile robots. Based upon that, missions like navigation, mapping, *et al.* are capable of being carried out by knowing its position and orientation (localizations) at any time. Generally, localization is determined by using the sensor systems installed on the robot. In this case, the task is called the self-localization problem, which is mostly discussed in this paper. Throughout this paper we define the localization algorithms to be the self-localization algorithms. Map matching based algorithms are a group of widely used localization algorithms which localize the robot by matching the map with the perceptions of the robot [1].

There are many kinds of map matching based localization algorithms and a good review can be found in [1]. Among them, Monte Carlo based localization algorithms are widely adopted due to their good balance between the localization accuracy and the computation complexity [2]. Unfortunately, the map details of the environment and the imperfect motion estimation for the robot greatly impact the accuracy of the localization results of the map matching based localization algorithms. This would further impact the performance of other dependent missions like path following, *et al.* Consequently, this paper focused on the localizability of Monte Carlo based localization algorithms.

Many researchers have focused on the localizability and the uncertainty of map matching based localization algorithms. The early research of localizability can be found in the field of coastal navigation by Roy, Burgard, Fox, and Thrun [3]. They employed the entropy of the probability distribution of the localization result during navigation to model the navigation uncertainty. Rohde, Stellet, Mielenz, and Zöllner have proposed a robot localizability estimation method for landmark-based localization in a dynamic environment. There, an analytical model of upper bounds on localization uncertainty is estimated while detectable landmark types and minimum detection rates are estimated based on specified upper bounds on pose estimation uncertainty [4]. Moreover,

The associate editor coordinating the review of this manuscript and approving it for publication was Zhonglai Wang.

localizability has also been studied extensively in the field of localization in the wireless network [5]–[7]. A good survey can be found by Zhen Yang *et al.* in [7].

The localizability of map matching based localization algorithms has been researched by two fields. One is the field of active Simultaneous Localization and Mapping (SLAM), where the map-based localization algorithm has been integrated to localize the robot. In this field, actions are generally determined based on the balance of the uncertainty of localization and the requirements of the environment exploration task. The exploitation of pose uncertainty for exploration phases has been extensively studied by many researchers, where the uncertainty has been believed to keep increasing until the loop is closed [8]. Barfoot, T. D. and Furgale, P. T. have proposed a practical method to associate the uncertainty with transformation matrices on Lie groups [9]. In the research proposed by Youngji Kim and Ayoung Kim [10], a hypothesis is that the monotonicity of pose uncertainty is preserved when the uncertainty is propagated on Lie groups rather than on Euclidean vector space. Additionally, Rodriguez-Arevalo, M. L., *et al.* have also proposed a study of the different representations of the uncertainty of localization. They concluded that only D-opt and Shannon's entropy may preserve the monotonicity of uncertainty [8]. In the research by Papachristos, C., *et al.* the belief of the robot state and the tracked landmarks has propagated over the branches of a random tree to predict the uncertainty of the robot state and the landmarks, so that the path that minimizes the expected localization and mapping uncertainty has been selected [11]. In the research by Maurović *et al.* [12], the localization uncertainty was considered in an active SLAM algorithm to improve the localization accuracy.

The other field is the field of the uncertainty of localization algorithms or the localizability of that. Early research of this topic was proposed by Censi, which first offered a method to estimate the lower bound of the covariance of localization based on the Cramér–Rao Bound, while binary map and differentiable obstacles were assumed [13]. Based on that, a series of researches involving localizability were then developed. Qian, *et al.* proposed an improved version of Censi's method by adding a factor to represent the influence of dynamic obstacles [14], while the localizability matrices in use were derived from the work of Wang *et al.* [15]. A more recent effort on localizability was proposed by Ruiz-Mayor *et al.* [16]. Instead of directly estimating the covariance of the localization uncertainty, they offered a probabilistic model of the indistinguishability for perception, assuming that the perceptual ambiguity can represent the localizability. Weikun Zhen, *et al.* also proposed a new method to evaluate the localizability of a given 3D map [17]. Arvanitakis *et al.* have proposed a navigation algorithm, which has taken the pose uncertainty into account [18]. Yang Gao, *et al.* have proposed a new evaluation function for mobile robot path planning algorithms for better balancing between the localizability and the traditional requirements for optimal path [19]. Carvalho Filho *et al.* proposed a theoretical analysis of the impact of parametric uncertainties of mobile robot kinematic model on velocity and pose estimation [20]. There, they also proposed a model to estimate the uncertainty of localization through dead-reckoning for differential driven mobile robots.

As it can be found, many researchers have offered many results on localizability or uncertainty of localization. However, to the best of our knowledge, most of the existing researches like the method by Censi [13] and Wang *et al.* [15] and the method by Ruiz-Mayor *et al.* [16] can only offer an indirect prediction of the localizability, while the influence of dead-reckoning is ignored. Other methods like the method by Papachristos *et al.* [11] and the method by Carvalho Filho *et al.* [20] can only offer a prediction of localizability through an inducement or propagation of the uncertainty. These indirect predictions may lead to a loss of accuracy in the prediction.

Consequently, in this paper, we focused on the localizability prediction for the map matching based localization algorithms while a differential driven robot, equipped with LIDAR and encoder, is assumed. Firstly, we will analyze the generally adopted framework of map matching based localization so that the propagation of uncertainty in localizing would be analyzed. Secondly, a deep learning neural network will be proposed based on this analysis. Consequently, this paper is organized as follows: Section 2 proposed an analysis of the framework of generally adopted map matching based localization and the propagation of uncertainty. Section 3 proposed our localizability predictor. Section 4 proposed a description of the datasets and several experimental studies. Finally, conclusions are drawn in Section 5. Table 1 shows the symbols used in this paper.

II. THE ANALYSIS OF MAP MATCHING BASED LOCALIZATION AND THE PROPAGATION OF THE UNCERTAINTY

A. THE FRAMEWORK OF MAP MATCHING BASED LOCALIZATION

Applications that involve robot localization generally use data fusion techniques to merge information from both odometers and other sensors, decreasing the uncertainty in localization [20]. The Kalman filter and particle filter are often employed as the data fusion algorithm, which applies information about the laws of the robot's motion at the forecast stage and information from the sensors at the update stage or the weight calculation stage. Assume the

$$\begin{aligned} \tilde{S}_k &= F(\tilde{S}_{k-1}, \tilde{u}_k) + \tilde{Q}_k \\ \tilde{O}_k &= H(\tilde{S}_k) + \varepsilon_k \end{aligned} \tag{1}$$

As it is shown in table I, let $S_k\ (x_k, y_k, \theta_k)$ be the current 2D pose of the robot and as well as the state of the Kalman filter. Let $\tilde{S}_k$ denote the estimation of S_k. Assume $\tilde{S}_k$ conforms to a Gaussian distribution and described as $\tilde{S}_k \sim N\left(S_k, \tilde{\delta}_{sk}\right)$, which means the mean value and the covariance of $\tilde{S}_k$ are S_k and $\tilde{\delta}_{sk}$ respectively. Then formula (1) shows two state

TABLE 1. Symbols in use.

Symbol	The mean of the symbol
S	2D Pose of the robot
$\widetilde{S}_k$	Estimation of the S at time k
$\widetilde{u}_k$	The motion of the robot at time k
$\widetilde{Q}_k$	The Gaussian noise that comes with the dead-reckoning.
x, y, θ	X , Y coordinates and the orientation of the robot
k	Current time step
$A\sim N(\overline{A},\delta_A)$	A conforms a Gaussian distribution whose mean is $\overline{A}$ and its covariance is δ_A
$\widetilde{\delta}_{sk}$	Covariance of $\widetilde{S}_k$
δ_{sm}	The covariance of the $\widetilde{S}$ based on map matching
δ_{si}	The covariance of $\widetilde{S}$ conducted by the i-th laser beam
$F()$	Motion model of the robot
$H()$	Function of perception
X_{k1}	the Lidar perception data at time k
X_{k2}	the transitional displacement Δd at time k and the angular displacement $\Delta\theta$ between time k-1 to k
$\widetilde{O}_k$	The current perception data from the sensor
o_i	The range value of the i-th laser beam
φ_i	The angle of the i-th laser beam in the robot coordinate system
ε_k	0 mean Gaussian observation noise.
δ_r	The variance of the observation noise
δ_q	Covariance of $\tilde{S}_k$ comes with the dead-reckoning.
R_d	Turning radius of the motion of the robot
var ()	Function to take covariance of its input random variables
N	Number of the laser beams in one lidar scan
i	The index of the laser beam
Δd	The transitional displacement
$\Delta\theta$	Angular displacement
$E()$	Function of calculating the entropy
$E_d(\widetilde{S})$	The entropy of $\widetilde{S}$ based on dead-reckoning
$E_{Mi}(\widetilde{S})$	The entropy of $\widetilde{S}$ obtained from the i-th laser beam based on map matching
$E(\widetilde{S})$	The entropy of $\widetilde{S}$ after information fusion

equations in use. The first equation shows the prediction stage where $F()$ is the motion model of the robot as shown in formula(2)-(3) [21], $\tilde{Q}_k \sim N(0, \delta_q)$ is the 0 mean Gaussian noise and $\tilde{u}_k$ is the motion of the robot. The second equation shows the update stage where $H()$ is the function of perception, $\tilde{O}_k$ is the current perception data from the sensor and $\varepsilon_k \sim N(0, \delta_r)$ is the 0 mean Gaussian noise in perception. Consequently, the Kalman filter allows the evaluation of the state of the system based on a noisy prediction of its evolution and noisy measurements of its state.

B. THE UNCERTAINTY IN LOCALIZATION

As shown in formula (1) the uncertainty of localization propagates in two stages. The first equation in formula (1) shows the propagation in the prediction stage, which is, in fact, a dead-reckoning process. The second equation in formula (1) shows the propagation in the update stage, which includes a map matching process and an information fusion process.

1) THE UNCERTAINTY IN DEAD-RECKONING

The research in [21], [22] has offered a theoretical analysis for the uncertainty which comes from the dead-reckoning. But here, we offer a simplified version for better efficiency. As shown in Table 1, according to formula (1), $\tilde{\delta}_{sk}$ can be obtained as shown in formula (4) if we apply a first-order Taylor expansion to the first equation of formula (1). Moreover, based on the formula (2) - (3), $\frac{\partial F}{\partial S}$ can be determined by formula (5) where Δd and $\Delta\theta$ are the transitional displacement and the angular displacement respectively. Here, R_d is the turning radius of the robot at pose$(x_{k-1}, y_{k-1}, \theta_{k-1})$. Let $E_d(\tilde{S})$ denotes the entropy, obtained in dead-reckoning, then $E_d(\tilde{S})$ can be obtained accordingly as shown in formula (6).

$$\begin{aligned} x_k &= x_{k-1} - R_d \sin\theta_{k-1} + R_d \sin(\theta_k) \\ y_k &= y_{k-1} + R_d \cos\theta_{k-1} - R_d \cos(\theta_k) \\ \theta_k &= \theta_{k-1} + \Delta\theta \end{aligned} \tag{2}$$

$$R_d = \frac{\Delta d}{\Delta\theta} \tag{3}$$

$$\tilde{\delta}_{sk} = \frac{\partial F}{\partial S}\tilde{\delta}_{sk-1}(\frac{\partial F}{\partial S})^T + \delta_q \tag{4}$$

$$\frac{\partial F}{\partial S} = \begin{pmatrix} \frac{\partial F}{\partial x} \\ \frac{\partial F}{\partial y} \\ \frac{\partial F}{\partial \theta} \end{pmatrix} = \begin{pmatrix} 1 - \frac{\partial R_d}{\partial x_{k-1}}(\sin\theta_{k-1} - \sin(\theta_k)) \\ 1 + \frac{\partial R_d}{\partial y_{k-1}}(\cos\theta_{k-1} - \cos(\theta_k)) \\ 1 + \frac{\partial \Delta\theta}{\partial \theta_{k-1}} \end{pmatrix} \tag{5}$$

$$E_d(\tilde{S}) = \frac{1}{2}\ln\left(\left|2\pi e(\frac{\partial H}{\partial \tilde{S}}\delta_s\frac{\partial H}{\partial \tilde{S}}^T + \delta_q)\right|\right) \tag{6}$$

2) THE UNCERTAINTY IN MAP MATCHING

Assume the sensor in use is a Lidar and let the perception data by the i-th laser beam be denoted by (o_i, φ_i) where o_i is the range value and φ_i denotes the angle of the laser beam in the robot coordinate system. Then the laser observation model can be obtained as formula (7) shows. Among them, the ε is one dimensional 0 mean Gaussian observation noise.

$$o_i = H(\tilde{S}, \varphi_i) + \varepsilon \tag{7}$$

$$o_i - \varepsilon = H(\tilde{S}, \varphi_i) + \frac{\partial H}{\partial \tilde{S}}(\tilde{S} - S) \tag{8}$$

The equation (8) can be obtained by applying the first-order Taylor expansion at $\tilde{S}$ to equation (7). Let var () denote the function to take covariance of its input random variables, then equation (9) can be obtained by calculating the covariance on the equation (8). Here δ_{si} denotes the covariance of $\tilde{S}$ conducted by the i-th laser beam. As $\left(\frac{\partial H(\tilde{S},\varphi_i)}{\partial \tilde{S}}\right)$ is a 1×3 vector rather than a matrix, it is impossible to be obtained directly by (9). Considering that the right side of (9)

shows a compression of δ_{si}, which is compression from a 3×3 matrix to 1 dimension, we choose entropy to represent the localizability instead of the covariance δ_{si}. Then equation (10) can be obtained by calculating the entropy on both sides of equation (9), where E () is the function of calculating the entropy. Consequently, the entropy of $\tilde{S}$ obtained from the i-th laser beam, denoted by $E_{\mathrm{M}i}(\tilde{S})$, can be obtained as shown in (11).

$$var(o_i-\varepsilon)=\left(\frac{\partial H(\tilde{S},\varphi_i)}{\partial\tilde{S}}\right)\delta_{si}\left(\frac{\partial H(\tilde{S},\varphi_i)}{\partial\tilde{S}}\right)^T \tag{9}$$

$$\begin{aligned}E(r_i)+E(\varepsilon)&=\frac{1}{2}\ln(|2\pi e\delta_s|\left|\frac{\partial H}{\partial\tilde{S}}\frac{\partial H}{\partial\tilde{S}}^T\right|)\\&=\frac{1}{2}\ln(|2\pi e\delta_s|)+\frac{1}{2}\ln(\left|\frac{\partial H}{\partial\tilde{S}}\frac{\partial H}{\partial\tilde{S}}^T\right|)\end{aligned} \tag{10}$$

$$E_{\mathrm{M}i}(\tilde{S})=E(r_i)+E(\varepsilon)-\frac{1}{2}\ln(\left|\frac{\partial H}{\partial\tilde{S}}\frac{\partial H}{\partial\tilde{S}}^T\right|) \tag{11}$$

As there are N laser beams that lead to the localization result, the uncertainty of $\tilde{S}$ comes from all the localizations using all the laser beams. Let $\tilde{\delta}_{\mathrm{sm}}$ denote the estimation of covariance of $\tilde{S}$ by map matching. Assuming that the perception of each laser beam is independent, then according to the information fusion theory, the estimation of $\tilde{\delta}_s$ after the fusion of all the laser beams can be estimated as shown in (12). Furthermore, the entropy after fusion, denoted by $E_{\mathrm{M}}(\tilde{S})$, is shown in (13).

$$\tilde{\delta}_{\mathrm{sm}}=(\frac{1}{N}\sum_{i=1\cdots N}\delta_{si}^{-1})^{-1} \tag{12}$$

$$E_{\mathrm{M}}(\tilde{S})=\frac{1}{2}\ln(2\pi e\left|\frac{1}{N}\sum_{i=1\cdots N}\delta_{si}^{-1})^{-1}\right|) \tag{13}$$

3) THE UNCERTAINTY IN INFORMATION FUSION

The fusion process happens in the update process which constrains the rising of the covariance according to the map matching residual in Kalman filter and the weights in the Monte Carlo localization. Fortunately, according to information fusion theory, the optimal estimation of the covariance would be obtained by formula (14), where $\tilde{\delta}_{\mathrm{sd}}$ is the estimation of $\tilde{\delta}_s$ by dead-reckoning and $\tilde{\delta}_{\mathrm{sm}}$ is the estimation of $\tilde{\delta}_s$ by map matching. Accordingly, the entropy can be obtained by formula (15).

$$\tilde{\delta}_s=(\frac{1}{2}(\tilde{\delta}_{\mathrm{sd}}^{-1}+\tilde{\delta}_{\mathrm{sm}}^{-1}))^{-1} \tag{14}$$

$$E(\tilde{S})=\frac{1}{2}\ln(2\pi e(\left|\frac{1}{2}(\tilde{\delta}_{\mathrm{sd}}^{-1}+\tilde{\delta}_{\mathrm{sm}}^{-1}))^{-1}\right|) \tag{15}$$

III. THE PREDICTOR OF LOCALIZABILITY BASED ON DEEP LEARNING

There are many possible measurements of localizability, in which covariance $\tilde{\delta}_s$ is the most direct one. However, as it was analyzed above, δ_{si} is difficult to be predicted directly, which makes $\tilde{\delta}_{\mathrm{sm}}$ and $\tilde{\delta}_s$ also difficult to be predicted. Fortunately, the entropy $E(\tilde{S})$ can be obtained by formula (15), while $E_{\mathrm{d}}(\tilde{S})$ can be obtained by formula (6). Consequently, we choose $E(\tilde{S})$ to represent the localizability. However, $E(\tilde{S})$ is still difficult to be calculated from formula (6) and formula (15). As a result, an entropy predictor constructed by a modularized Deep learning neural network M is proposed.

Consider the uncertainty propagating in three processes, the dead-reckoning, the map matching, and the information fusion. This predictor is composed of three different neural network modules accordingly. As shown in Fig. 1, these three components include a Convolutional Neural Network(CNN) module $M1$ modeling the propagation of uncertainty in map matching, a recurrent neural network(RNN) module $M2$ modeling the propagation of uncertainty in dead-reckoning and a multi-layer perceptron module $M3$ modeling the propagation of uncertainty in data fusion. Here we choose CNN structure for $M1$ because according to formula (11) $\sim$ formula (13), the entropy $E_{\mathrm{M}}(\tilde{S})$ is mainly decided by $\left(\frac{\partial H(\tilde{S},\varphi_i)}{\partial\tilde{S}}\right)$, which is a local feature suitable to be extracted by CNN. Because according to formula(4)$\sim$ formula(6), $E_{\mathrm{d}}(\tilde{S})$ is influenced by the prior covariance $\tilde{\delta}_{sk-1}$ and this kind of influence would be properly modeled by RNN. So we choose RNN structure for $M2$.

This predictor takes three inputs, which include X_{k1} is the Lidar perception data at time k, whose size is a 1×1080 vector, X_{k2} includes the transitional displacement Δd at time k and the angular displacement $\Delta\theta$ between time k-1 to k. The output of the predictor is $E_k(\tilde{S})$, which is the entropy of $\tilde{S}$ at time k.

A. THE STRUCTURE OF MODULE M1

As shown in formula (10) $\sim$ (13), the entropy $E_{\mathrm{M}}(\tilde{S})$ is greatly impacted by $\frac{\partial H}{\partial\tilde{S}}$ which reflects the magnitude of the slope at the contact point on the obstacles confronting the Lidar, and such slope may exist in many places in a single frame of perception data. Therefore, the entropy would be predicted by a Convolutional Neural Networks (CNN) module $M1$ where the convolution layers look for the combination of different $\frac{\partial H}{\partial\tilde{S}}$ and the fully connected layers model the data fusion in formula (11) $\sim$ (12). As a well-known CNN structure, Visual Geometry Group(VGG) [24] has shown a good performance on feature extraction, so that it has been adopted in $M1$. As shown in Fig. 1, the input of $M1$ is X_{k1} and the output is an estimation of the entropy, which is denoted as $E_{k1}(\tilde{S})$.

Fig.2 shows the architecture of $M1$. Let $X_{k1}^i(o_i,\varphi_i)$ be the perception data of the i-th laser beam at time k then the perception data at time k would be denoted by $X_{k1}^i(x_{k1}^1,x_{k1}^2\cdots x_{k1}^{1080})$. In this paper a Hokuyo UTM-30LX Lidar is employed, consequently, we have $N=1080$. As shown in Fig. 2 the input layer of $M1$ is directly connected to X_{k1} which is also drawn as a group of black dots at the left end. There are 23 layers of hidden layers, which include 16 convolutional layers, 2 fully connected layers and 5 layers of max pooling. The activation function of the convolutional layer is a Rectified Linear Unit. In the first 12 layers of the network, a small convolution kernel of 3×1 is used to

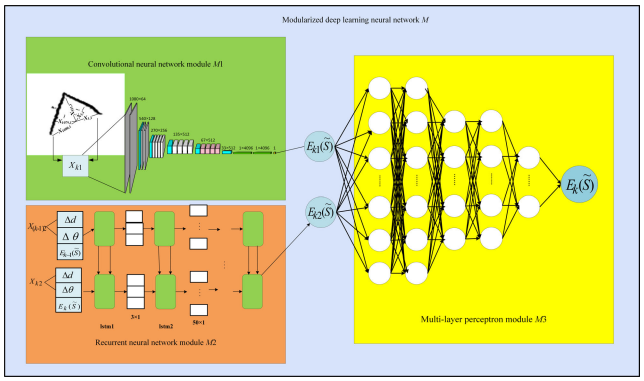


FIGURE 1. Structure of the predictor.

extract the local features, while in the last 4 layers, a large convolution kernel of 10×1 is used to extract macro features. The activation functions are linear rectifier functions.

B. THE STRUCTURE OF MODULE M2 AND MODULE M3

According to formula (4) $\sim$ (5), $\tilde{\delta}_{sk}$ is greatly impacted by $\tilde{\delta}_{sk-1}$, Δd and $\Delta\theta$, as is the entropy calculated from $\tilde{\delta}_{sk}$. As a result, a Recurrent Neural Network (RNN) is employed in module $M2$ to deal with this time correlation problem. As shown in Fig. 3, $M2$ has 10 Long Short-Term Memory (LSTM) layers as its hidden layer while it has only 1 neuron in the output layer to output an estimation of the entropy which is denoted as $E_{k2}(\tilde{S})$. The first layer of $M2$ is set to 3 LSTM units while next 9 layers are set to 50. The output layer employed a linear function as its activation function.

As it was analyzed, module $M1$ and module $M2$ output two estimations, $E_{k1}(\tilde{S})$ and $E_{k2}(\tilde{S})$, to the same entropy $E_k(\tilde{S})$. Then a multi-layer perceptron $M3$ is employed to model the data fusion applied on $E_{k1}(\tilde{S})$ and $E_{k2}(\tilde{S})$. Consequently, $M3$ takes $E_{k1}(\tilde{S})$ and $E_{k2}(\tilde{S})$ as input and outputs the final estimation of $E_k(\tilde{S})$. As shown in Fig. 4, $M3$ has 5 hidden layers, where the first two have 256 neurons, the latter two have 128 neurons and the last hidden layer has 64 neurons. The linear rectification function (Rectified Linear Unit) is chosen as the activation function for all the neurons in hidden layers while a linear function is chosen for the output layer.

IV. DATASET COLLECTION AND TRAINING

In this paper, the dataset is collected in common indoor and outdoor environments using Pioneer P3DX robot as shown in Fig. 5. The scanning angle range of the UTM-30LX Lidar is 270°, the angular resolution is 0.25°, and the maximum detection distance is set to 3 meters. As shown in Table 2, the dataset is collected from 6 types of environments, while the total number of groups of the dataset is 12451. Some of the typical environments are shown in Fig. 6. The Extensive outdoor scenes are generally empty except for some areas with objects haphazardly arranged. Examples include playgrounds and stadiums, *et al.* Each group of data includes the time stamp k, Lidar perception data X_{k1}, the transitional displacement Δd, the angular displacement $\Delta\theta$, and the localization

TABLE 2. Detail of the data.

Site type	VOLUME (GROUP)	TOTAL GROUPS
Square ground	854	12451
Outdoor scenes	854	
Indoor long corridor	3871	
Extensive outdoor scenes	4013	
Round ground	1069	
Triangular ground	1790	

entropy $E_k(\tilde{S})$. The localization entropy comes from the localization covariance, which is from an AMCL (Augmented Monte Carlo Localization) [21] localization package in ROS (Robot Operating System).

A. TRAINING AND TESTING SETTINGS

The dataset has been divided into 2 subsets, which are subset A and subset B. All the modules are trained separately for each module was designed to model a different source of estimation with different inputs. Module $M1$ models the estimation from map matching taking X_{k1} as input. Consequently, $M1$ is trained with subset A, which includes X_{k1} as input and $E_k(\tilde{S})$ as the label. $M2$ is trained with subset B, which includes X_{k2} as input and $E_k(\tilde{S})$ as the label. As it's stated, module $M1$ is designed to model the propagation of uncertainty in the map matching process. However, the AMCL package only applies the map matching process while the travel distance and the displacement of the orientation exceed a threshold value. Thus, we need to extract the entropies conducted by the map matching. Because dead-reckoning results in a rising of the entropies, while the map matching generally results in a decreasing of the entropies, subset B is constructed by extracting all of the rising entropies, while subset A is constructed by extracting all of the other entropies. Fig. 7a shows a segment of entropies acquired from an experiment where the red curve with a decreasing trend is used as subset A, and the blue curve with an increasing trend is used as subset B.

The histogram of all the entropies in dataset has been shown in Fig. 7b where the horizontal axis shows the value of the entropy, and the vertical axis shows the number of that value. Please note that the width of each column in Fig.7b is set to 1 for convenience. It can be found that the entropy is ranged between -11.57 and 10.46, while the ratio of the entropy with extremely big (value inside [3, 10.46]) or small value (value inside $[-11.57, -10]$) is only 3.2%. This implies that most of the data were acquired in a common case. In the training process, the whole dataset, subset A, subset B were all partitioned into a training set, a validation set, and a testing set. In this paper, the training set occupied 64% of the dataset while the validation set occupied 16% and the testing set occupied 20%, respectively. $M1$, $M2$, and $M3$ are trained and tested separately using subset A, subset B and all the data set respectively. We use a NVIDIA RTX2080TI graphics card in

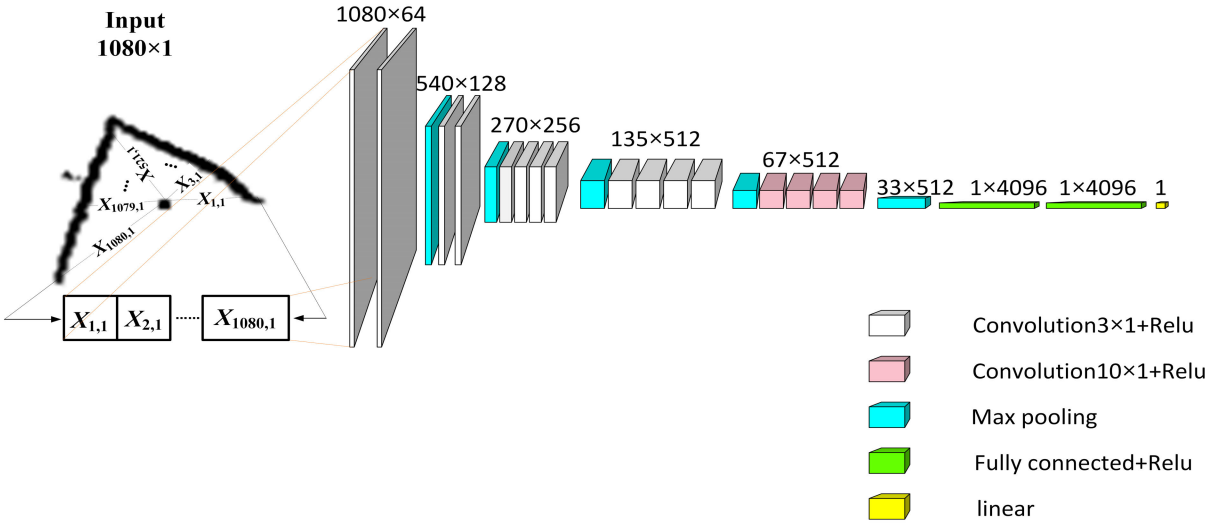


FIGURE 2. Structure of convolutional neural network *M*1.

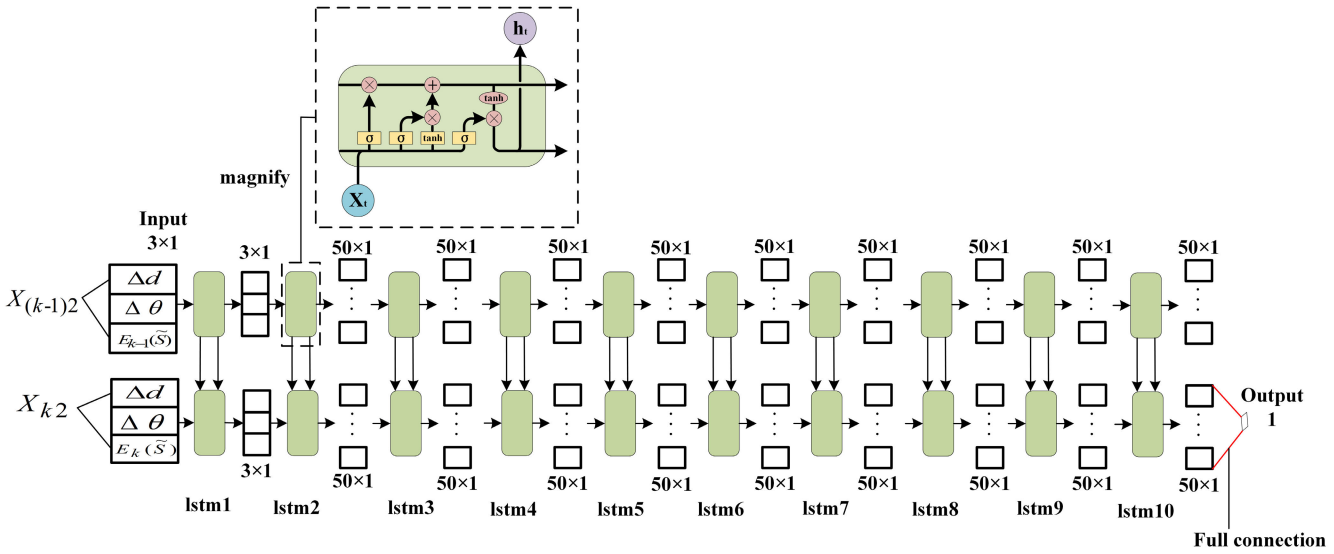


FIGURE 3. Structure of recurrent neural network *M*2.

training and testing under ubuntu16.04 and keras2.0.9. In the following training, the back propagation learning algorithm is adopted while the batch size is set to 20, the epoch is set to 3000, the learning efficiency coefficient is set to 0.002 and the mean square error is adopted as the loss function.

B. EXPERIMENT STUDY FOR THE STRUCTURE OF M1 AND M2

In order to find the performance of $M1$ with different number of layers, we have designed a series of experiments. Some of the results have been shown in Table 2. Here, $M1\_n$ refers to $M1$ with n layers of neuron network. Please be noted that when $n = 22$, 1 convolution layer has been added in front of each of the last three max pooling layers of VGG 19, while $n = 25$, 2 convolution layers have been added similarly. err_t, err_v, err_s represent the error percentages of $M1$ on training set, validation set and testing set respectively.

It can be seen from table 3 that $M1\_19$ has the best performance whose error in training, validation and testing are lowest. Consequently, $M1\_19$ has been chosen for $M1$. As it was proved in [24], many layers with smaller convolution kernels instead of layers with larger convolution kernels can reduce the parameters while improving the feature extraction capability. Consequently, small convolution kernel size in 3×1 was chosen for most layers of $M1$. However, this situation is a little different in our problem, in that not only local features like slope may impact the localizability, but also some macro features like two perpendicular long walls may impact as well. Because of that, we have chosen a larger convolution kernel, size in 10×1, for the last 4 layers of $M1$ for a

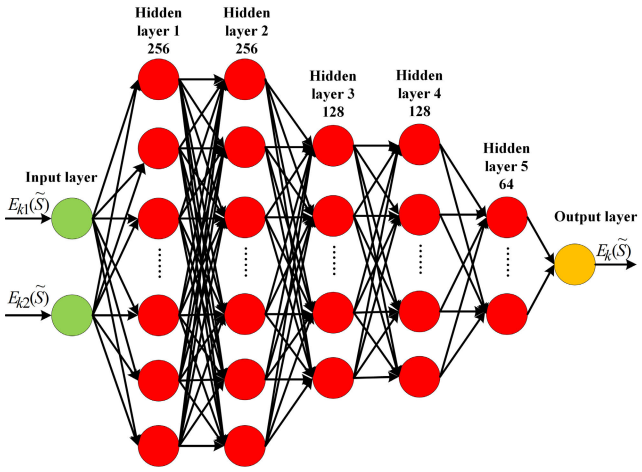


FIGURE 4. Structure of multi-layer perceptron *M*3.

FIGURE 5. The platform in use.

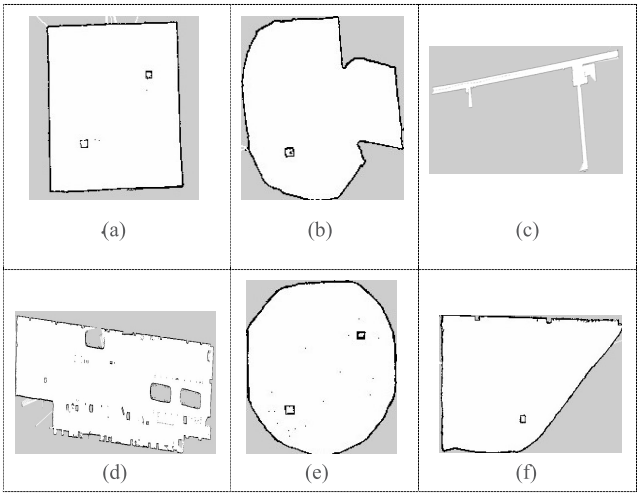


FIGURE 6. (a) Square ground. (b) Outdoor scenes. (c) Indoor long corridor. (d) Extensive outdoor scenes. (e) Round ground. (f) Triangular ground.

larger receptive field. Table 4 has shown the error percentages of $M1\_19$ with different size of convolution kernel on training set, validation set and testing set respectively. Please note

TABLE 3. Error rate of *M*1 with diffent layers.

MODEL	err_t (%)	err_v (%)	err_s (%)
$M1\_11$	3.67	10.21	10.92
$M1\_13$	1.08	7.54	8.97
$M1\_16$	3.49	9.36	9.47
$M1\_19$	0.73	7.71	8.46
$M1\_22$	1.32	8.08	8.65
$M1\_25$	7.98	12.74	13.49

TABLE 4. Error rate of *M*1 with diffent kernel size.

MODEL	err_t (%)	err_v (%)	err_s (%)
$M1\_19\_3$	0.73	7.71	8.46
$M1\_19\_5$	1.20	6.42	7.17
$M1\_19\_7$	1.08	6.87	7.64
$M1\_19\_10$	0.55	6.36	7.15
$M1\_19\_13$	1.12	6.81	7.33
$M1\_19\_67$	1.36	7.95	8.85

that $M1\_19\_m$ in the table denotes the $M1\_19$ structure with the kernel size in $m\times1$. It can be found that $M1\_19\_10$ has the lowest error rates which are, 0.55%, 6.36% 7.15% and respectively. Consequently, we have chosen the structure $M1\_19\_10$ for $M1$.

In order to find the performance of $M2$ with different number of layers, we have also designed a series of experiments. Some of the results have been shown in Table 2. Here, $M2\_n$ refers to $M2$ with n layers of LSTM neuron unit.

It can be found that $M2\_11$ has the lowest error rates which are 2.15%, 5.09% and 1.72% respectively. Consequently, 11 layers have been chosen for $M2$.

TABLE 5. Error rate of $M2$ with diffent layers.

MODEL	err_t (%)	err_v (%)	err_s (%)
$M2\_5$	5.23	9.07	5.69
$M2\_7$	10.12	13.30	11.22
$M2\_9$	14.32	18.05	16.13
$M2\_11$	5.57	6.43	4.14
$M2\_13$	34.78	76.45	29.2
$M2\_15$	35.08	77.00	29.23

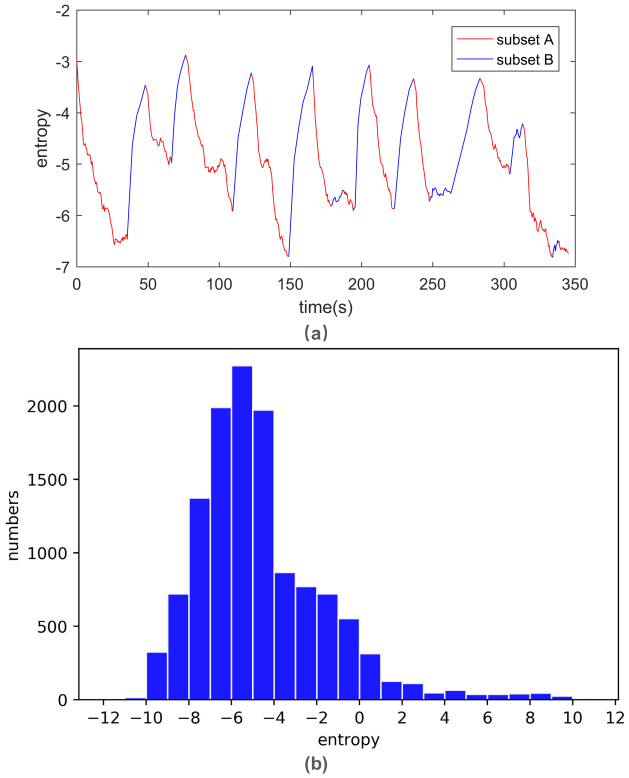


FIGURE 7. (a) An example of segment of entropies in the training data set. (b) The histogram of Entropy of the dataset.

C. PERFORMANCE OF THE PREDICTOR ON DATASET

As our predictor is composed of three modules and $M1$ was trained separately, we show the performance separately too. On subset A, the loss curve of $M1$ on the training set and the validation set was shown in Fig. 8a, where the horizontal axis is the number of epoch and the vertical axis is the value of loss. Please note that only the loss in 1600 epochs were included in the figure because the training has already converged. It can be found that $M1$'s loss decreased rapidly since the 40th epoch. For better understanding of the performance, we have shown the prediction values by $M1$ and the label values on both the training set and the validation set in Fig. 8b, where the horizontal axis is the label value and the vertical axis is the prediction values by $M1$. The black dash line in the figure shows where the prediction values are equal to the label values. Similarly, Fig. 8c showed $M1$'s prediction and the label values on the testing set. It can be found that from 1000 epochs on, the training has been converged. It can be found that from Fig. 8a most of the predictions are close to the corresponding label values.

On subset B, the loss curve of $M2$ on the training set and validation set were also shown in Fig. 9a. It can be found that $M2$'s loss decreased rapidly since the 15th epoch. Again, we have also shown the prediction values by $M2$ and the label values on both the training set and the validation set in Fig. 9b, where the horizontal axis is the label value and the vertical axis is the prediction values by $M2$. Similarly, Fig. 9c showed $M2$'s prediction and the label values on the testing set. It can be found that from 300 epochs on, the training has been converged. It can be found that from Fig. 9a most of the predictions are close to the corresponding label values. Moreover, $M2$ shows better accuracy then that of $M1$. This is because AMCL package used in the data collection highly relies on the result of dead-reckoning which is modelled by $M2$.

On the whole data set, the loss curve of M on the training set and validation set were also shown in Fig. 10a. It can be found that M's loss decreased rapidly since the 10th epoch. Again, we have also shown the prediction values by M and the label values on both the training set and the validation set in Fig. 10b, where the horizontal axis is the label value and the vertical axis is the prediction values by M. The average prediction error rates on the training set and the validation set are 3.09% and 6.43% respectively. Similarly, Fig. 10c showed M's prediction and the label values on the testing set where the average error rate is 2.94%. It can be found that from 1000 epochs on, the training has been converged. It can be found that from Fig. 10a most of the predictions are close to the corresponding label values.

V. EXPERIMENTAL STUDY

To further verify the performance of our predictor, we employed two localizability prediction methods by Censi and by Ruiz-Mayor, respectively for comparison. In this section the robot was deployed in two environments that have never been seen by the robot. In each experiment, the robot was forced to moving along a given path to collect the real data. Then the entropy calculated by the covariance from AMCL package was treated as the ground truth. All

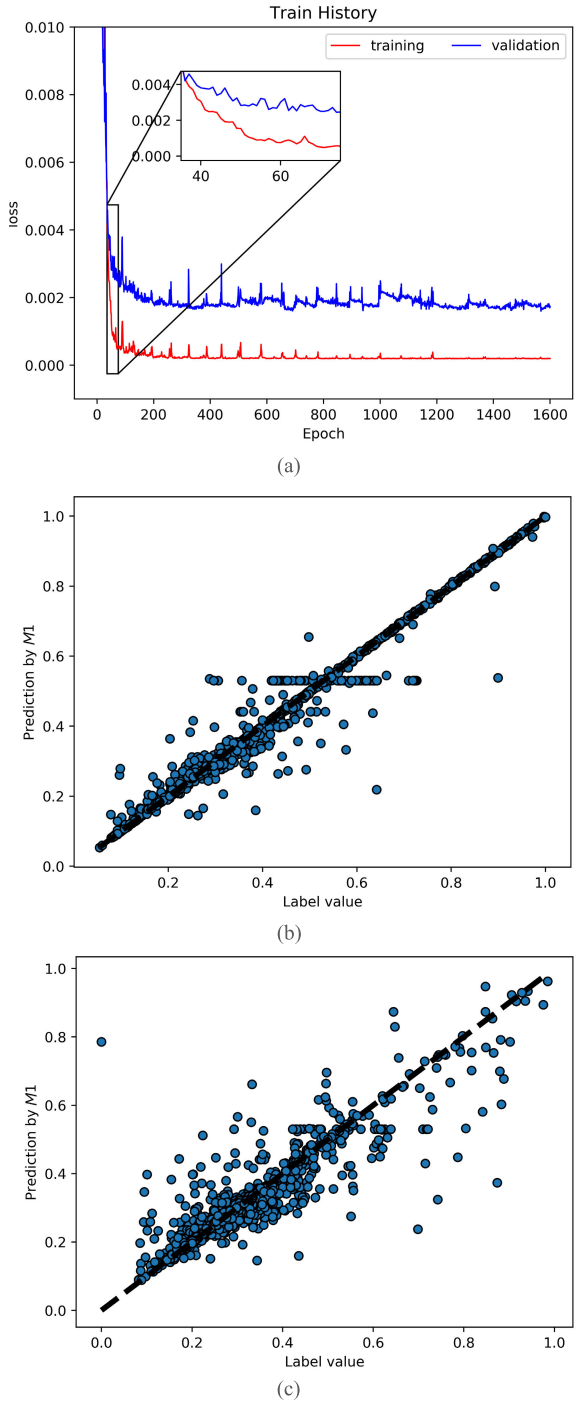


FIGURE 8. (a) The loss curves of $M1$ on the training set and the validation set. (b) $M1$'s predictions and the label values on the training set and validation set. (c) $M1$'s predictions and the label values on the testing set.

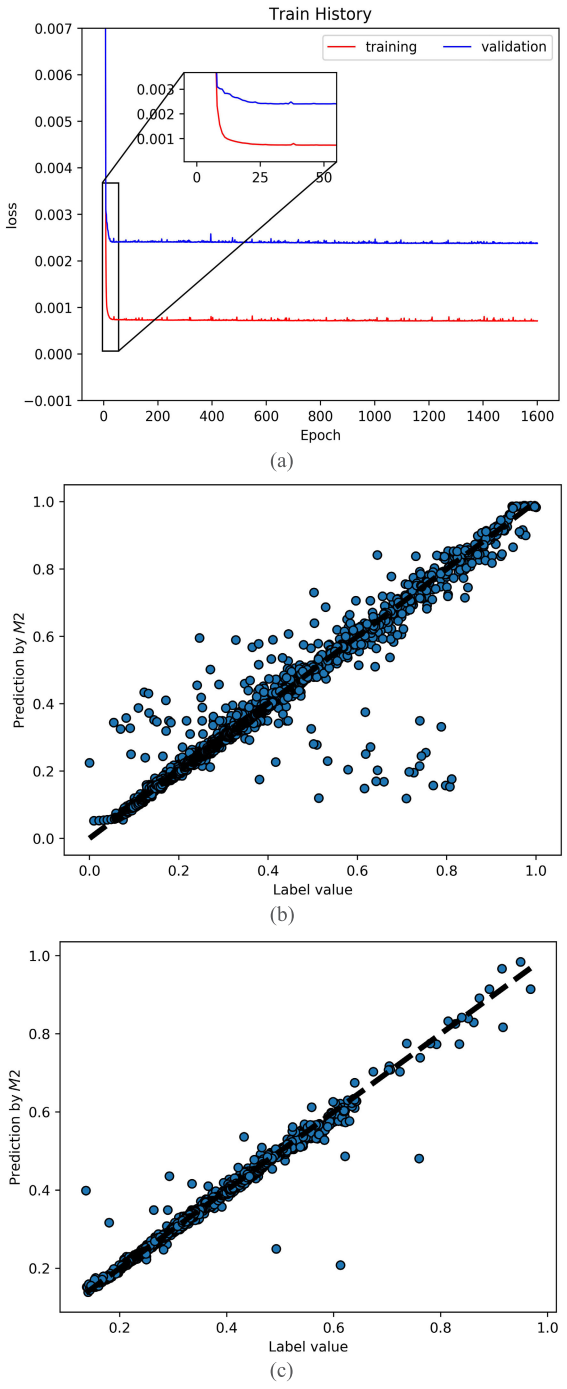


FIGURE 9. (a) The loss curves of $M2$ on the training set and the validation set. (b) $M2$'s predictions and the label values on the training set and validation set. (c) $M2$'s predictions and the label values on testing set.

the prediction methods were then employed to predict the localizability for comparison.

A. COMPARISON IN AN INDOOR ENVIRONMENT

The first comparison was carried out in an indoor environment shown in Fig.11 where the red line showed the track of the robot in experiment. Fig. 12 showed M's prediction and the corresponding label values, which is calculated from the AMCL package. The average error rate of our predictor in this experiment is 6.21%.

Similarly, we also calculated the localizability through Censi's method and Ruiz-mayor's method. In Fig. 13. The red curve showed the ground truth value, the black curve showed the value by our predictor, the blue curve showed the value by Censi's method and the green curve showed the value by

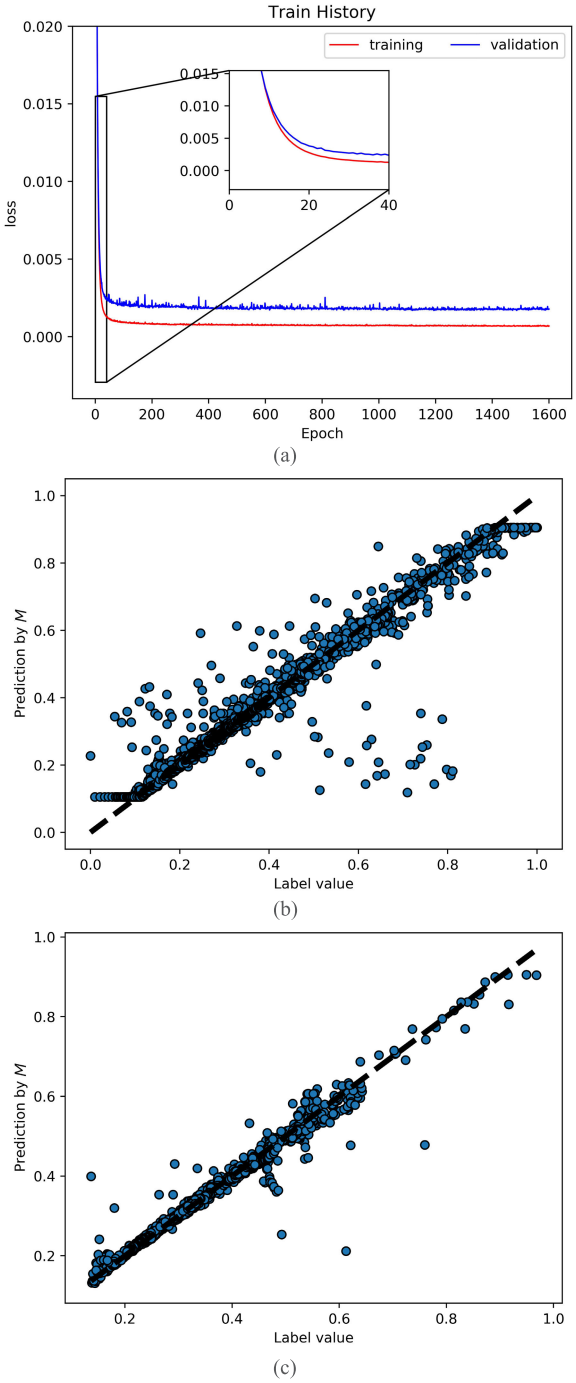


FIGURE 10. (a) The loss curves of M on the training set and the validation set. (b) M's predictions and the label values on the training set and validation set. (c) M's predictions and the label values on the testing set.

Ruiz-mayor's method. Please note that Ruiz- mayor's method only predict the localizability using ambiguity instead of entropy, which are quite different, but should show similar trend.

B. COMPARISON IN AN OUTDOOR ENVIRONMENT

The second comparison was carried out in an outdoor environment shown in Fig.14 where the red line showed the given track of the robot in experiment. Fig. 15 showed M's prediction and the corresponding label values. The average error rate of our predictor in this experiment was 5.14%.

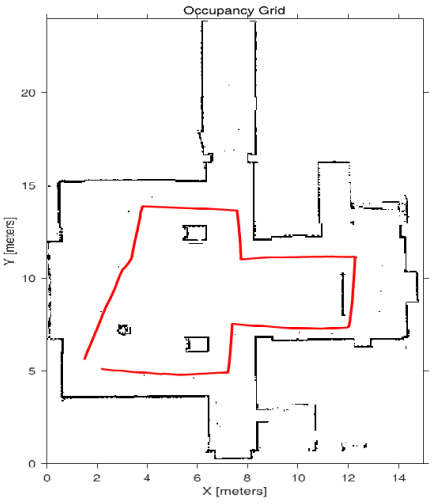


FIGURE 11. Map of indoor environment and the track of the robot in experiment.

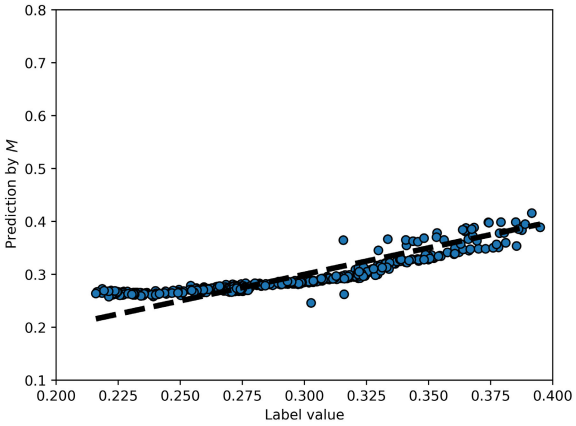


FIGURE 12. M's predictions and the label values in indoor environment.

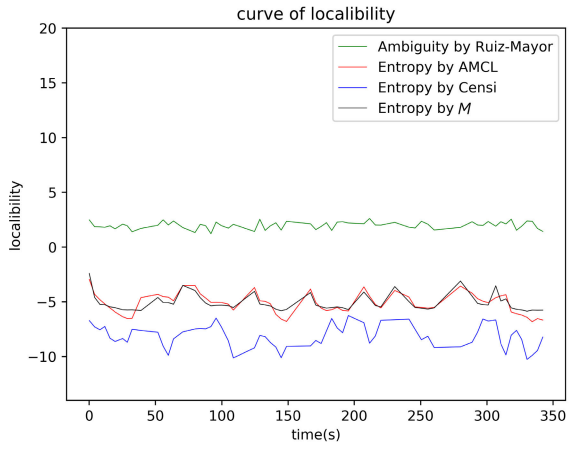


FIGURE 13. Performance comparison in indoor environment.

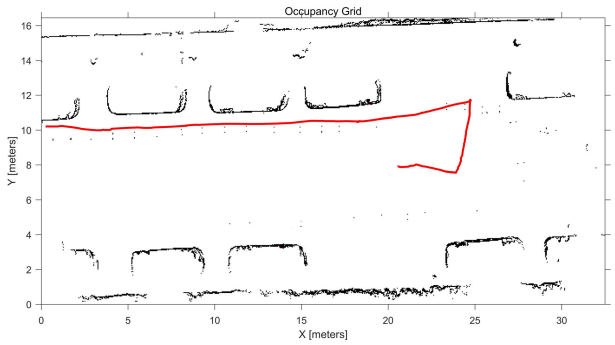


FIGURE 14. Map of outdoor environment and the track of the robot in experiment.

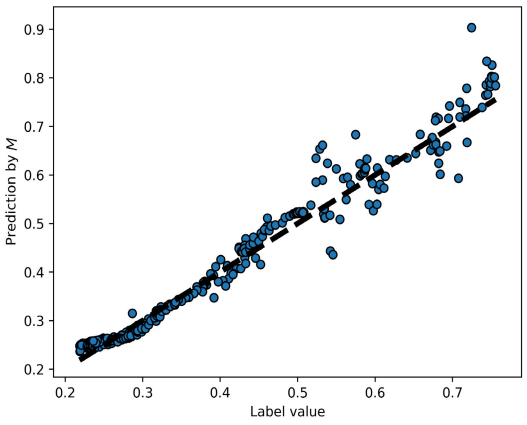


FIGURE 15. *M*'s predictions and the label values in outdoor experiment.

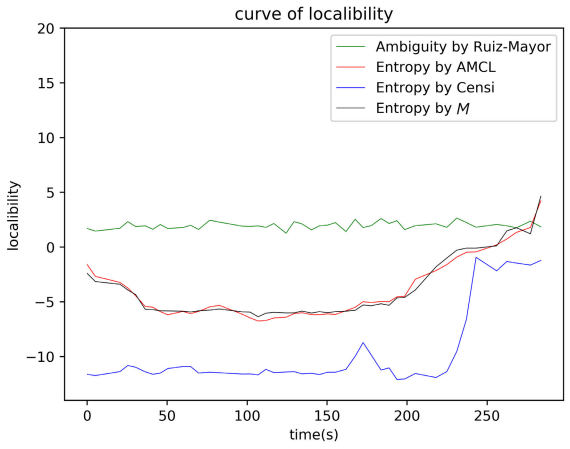


FIGURE 16. Performance comparison in outdoor environment.

Similarly, Fig. 16 showed the four curves of values by our predictor, by Censi's method, by Ruiz-mayor's method and by AMCL. It can be found that our method offer a better prediction compare to the other two method too.

It can be found that compared to the other two methods, our method offers a better prediction which is much closer to the ground truth. This is because of two reasons, (1) the other two methods only predict the localizability of map matching, while in contrast our approach has incorporated the dead-reckoning result in to the localizability prediction. (2) The method by Censi predicts the lower bound of the covariance of localization, while the method by Ruiz-mayor predicts the ambiguity of perception. In contrast our predictor predicts the localizability based on the dataset directly while the other two methods made the prediction in an indirect way.

TABLE 6. Error rate of our predictor and the models.

MODEL	Indoor error (%)	Outdoor error (%)
$M1$	13.06	23.55
$M2$	5.31	8.67
M	6.21	5.14

Table 6 showed the error rates $M1, M2, M$ in the indoor and outdoor environments respectively. It can be found that both $M1, M2$ showed lower error rate in indoor environment than that in outdoor environment. This is because the vibration of the localization entropy in outdoor environment is bigger than that in indoor environment which makes bigger error. Moreover, the result of $M2$ in indoor environment is even better than that of M. This is because the AMCL package highly relies on the dead-reckoning, which may offer good localization result in indoor environment and be well modelled by $M2$. On the other hand, in outdoor environment, the map matching may offer more contribution in the localization result compared to the dead-reckoning. As a result, M performs better than $M2$ in outdoor.

VI. CONCLUSION

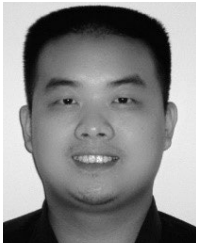
This paper mainly focused on the localizability prediction for mobile robots. The proposed predictor composed of a modularized Deep learning neural network which includes three different neural network modules with the intuition that the dataset in practice would lead to a more direct and more accurate prediction of the localizability. The design of each module is based on the analysis of the propagation of uncertainty in map matching based localization. The experiment result proved that our predictor offered a better prediction compared to the exiting research.

A decrease of the accuracy in the experiment compared to the accuracy of the testing data set has been found, which may be caused by the inconsistency of the structure between the network and the architecture of the localization algorithms. Consequently, a better structure of the neural network would be of interest in our future research. Moreover, better performance from RNN neuron network module had been observed which may come from the localization algorithm adopted in

data collection, further research about this phenomenon is in our future interest. Finally, in the proposed research, 2D localization and Lidar are assumed which may not include all the situations in practice. Consequently, more general research about the localizability prediction would also be in our future interest.

REFERENCES

[1] K. Okawa, "Self-localization estimation for mobile robot based on map-matching using downhill simplex method," *J. Robot. Mechtron.*, vol. 31, no. 2, pp. 212–220, 2019.
[2] Y. Wang, W. Chen, and J. Wang, "Map-based localization for mobile robots in high-occluded and dynamic environments," *Ind. Robot, Int. J.*, vol. 41, no. 3, pp. 241–252, 2014.
[3] N. Roy, W. Burgard, D. Fox, and S. Thrun, "Coastal navigation-mobile robot navigation with uncertainty in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, May 1999, pp. 35–40.
[4] J. Rohde, J. E. Stellet, H. Mielenz, and J. M. Zöllner, "Localization accuracy estimation with application to perception design," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 4777–4783.
[5] A. Alomari, F. Comeau, W. Phillips, and N. Aslam, "New path planning model for mobile anchor-assisted localization in wireless sensor networks," *Wireless Netw.*, vol. 24, no. 7, pp. 2589–2607, Oct. 2018.
[6] H. Wu, A. Ding, W. Liu, L. Li, and Z. Yang, "Triangle extension: Efficient localizability detection in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7419–7431, Nov. 2017.
[7] Z. Yang, C. Wu, Z. Zhou, X. Zhang, X. Wang, and Y. Liu, "Mobility increases localizability: A survey on wireless indoor localization using inertial sensors," *ACM Comput. Surveys*, vol. 47, no. 3, pp. 1–34, Apr. 2015.
[8] M. L. Rodriguez-Arevalo, J. Neira, and J. A. Castellanos, "On the importance of uncertainty representation in active SLAM," *IEEE Trans. Robot.*, vol. 34, no. 3, pp. 829–834, Jun. 2018.
[9] T. D. Barfoot and P. T. Furgale, "Associating uncertainty with three-dimensional poses for use in estimation problems," *IEEE Trans. Robot.*, vol. 30, no. 3, pp. 679–693, Jun. 2014.
[10] Y. Kim and A. Kim, "On the uncertainty propagation: Why uncertainty on lie groups preserves monotonicity?" in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 3425–3432.
[11] C. Papachristos, F. Mascarich, S. Khattak, T. Dang, and K. Alexis, "Localization uncertainty-aware autonomous exploration and mapping with aerial robots using receding horizon path-planning," *Auto. Robots*, vol. 43, no. 8, pp. 2131–2161, Dec. 2019.
[12] I. Maurović, M. Seder, K. Lenac, and I. Petrovic, "Path planning for active SLAM based on the D* algorithm with negative edge weights," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 8, pp. 1321–1331, Aug. 2018.
[13] A. Censi, "On achievable accuracy for range-finder localization," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2007, pp. 4170–4175.
[14] K. Qian, X. Ma, F. Fang, X. Dai, and B. Zhou, "Mobile robot self-localization in unstructured environments based on observation localizability estimation with low-cost laser range-finder and RGB-D sensors," *Int. J. Adv. Robot. Syst.*, vol. 13, no. 5, Aug. 2016, Art. no. 1729881416670902.
[15] Y. Wang, W. Chen, J. Wang, and H. Wang, "Active global localization based on localizability for mobile robots," *Robotica*, vol. 33, no. 08, pp. 1609–1627, Oct. 2015.
[16] A. Ruiz-Mayor, J.-C. Crespo, and G. Trivino, "Perceptual ambiguity maps for robot localizability with range perception," *Expert Syst. Appl.*, vol. 85, no. 1, pp. 33–45, Nov. 2017.
[17] W. Zhen, S. Zeng, and S. Soberer, "Robust localization and localizability estimation with a rotating laser scanner," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Singapore, May 2017, pp. 6240–6245.
[18] I. Arvanitakis, A. Tzes, and K. Giannousakis, "Mobile robot navigation under pose uncertainty in unknown environments," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 12710–12714, 2017.
[19] Y. Gao, J. Liu, M. Q. Hu, H. Xu, K. P. Li, and H. Hu, "A new path evaluation method for path planning with localizability," *IEEE Access*, vol. 7, pp. 162583–162597, 2019.
[20] J. G. N. D. Carvalho Filho, E. A. N. Carvalho, L. Molina, and E. O. Freire, "The impact of parametric uncertainties on mobile robots velocities and pose estimation," *IEEE Access*, vol. 7, pp. 69070–69086, 2019.
[21] T. Sebastian, B. Wolfram, and F. Dieter, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.
[22] M. Golfarelli, D. Maio, and S. Rizzi, "Correction of dead-reckoning errors in map building for mobile robots," *IEEE Trans. Robot. Autom.*, vol. 17, no. 1, pp. 37–47, Feb. 2001.
[23] B. Irani, J. Wang, and W. Chen, "A localizability constraint-based path planning method for autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 7, pp. 2593–2604, Jul. 2019.
[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556



YANG GAO (Member, IEEE) received the Ph.D. degree in mechatronic engineering from Northwestern Polytechnic University, Xi'an, China, in 2011. He is currently an Associate Professor with the School of Automobile, Chang'an University. His research interests include robotics, navigation of mobile robot, simultaneous localization and mapping (SLAM), and control system design of dc motor.

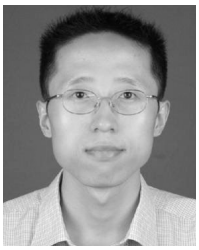
SHU QI WANG received the bachelor's degree in mechatronic engineering from Chang'an University, Xi'an, Shaanxi, China, in 2017, where he is currently pursuing the master's degree with the School of Automobile. His research interest includes deep learning.

JING HANG LI received the bachelor's degree in mechanical engineering from Tongji University, Shanghai, China, in 2016, and the master's degree in mechanical and industrial engineering from the University of Illinois at Chicago, where he is currently pursuing the Ph.D. degree. His current research interests include deep learning, meta-learning, and black-box optimization.

MENG QI HU (Member, IEEE) received the Ph.D. degree in industrial engineering from Arizona State University, USA, in 2012. He is currently an Assistant Professor with the Department of Mechanical and Industrial Engineering, University of Illinois at Chicago. His main research interests include complex system design and optimization, distributed decision support and analysis, swarm intelligence and evolutionary computation, and meta-learning and meta-modeling.

HONG YAO XIA received the bachelor's degree in vehicle engineering from the Xihua University of Science and Technology, Chengdu, China, in 2018. He is currently pursuing the master's degree with the School of Automobile, Chang'an University. His research interests include SLAM technology and mobile robot path planning and navigation.

HUI HU received the Ph.D. degree in systems engineering from Beijing Jiaotong University, Beijing, China, in 2008. She is currently an Associate Professor with the School of Automobile, Chang'an University. Her research interests include robot system optimize and artificial intelligence.



LAI JUN WANG received the Ph.D. degree in systems engineering from Northwestern Polytechnic University, Xi'an, China, in 2004. He is currently an Associate Professor with the School of Automobile, Chang'an University. His research interests include robot manufacturing technology and artificial intelligence.