

© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Accepted for publication in IEEE Transactions on Instrumentation & Measurement

# A Stacked LSTM based Approach for Reducing Semantic Pose Estimation Error

Rana Azzam<sup>1</sup> , Yusra Alkendi<sup>1</sup> , Tarek Taha<sup>2</sup> , Shoudong Huang<sup>3</sup> , and Yahya Zweiri<sup>1,4</sup> 

**Abstract**—Achieving high estimation accuracy is significant for semantic simultaneous localization and mapping (SLAM) tasks. Yet, the estimation process is vulnerable to several sources of error including limitations of the instruments used to perceive the environment, shortcomings of the employed algorithm, environmental conditions, or other unpredictable noise. In this paper, a novel stacked long short term memory (LSTM) based error reduction approach is developed to enhance the accuracy of semantic SLAM in presence of such error sources. Training and testing datasets were constructed through simulated and real-time experiments. The effectiveness of the proposed approach was demonstrated by its ability to capture and reduce semantic SLAM estimation errors in training and testing datasets. Quantitative performance measurement was carried out using the absolute trajectory error (ATE) metric. The proposed approach was compared to vanilla and bidirectional LSTM networks, shallow and deep neural networks, and to support vector machines. The proposed approach outperforms all other structures and was able to significantly improve the accuracy of semantic SLAM. To further verify the applicability of the proposed approach, it was tested on real-time sequences from the TUM RGB-D dataset, where it was able to improve the estimated trajectories.

**Index Terms**—LSTM, Deep Learning, Semantic SLAM, Localization Error, Measurement Uncertainty, Sensor Noise

## I. INTRODUCTION

**S**IMULTANEOUS localization and mapping (SLAM) is one of the most prevalent research problems in the robotics community. It is defined as the problem of estimating the trajectory of a robotic vehicle and incrementally constructing a map of its surroundings, provided with measurements perceived from the environment [1]. SLAM serves as a key enabler of a wide range of applications in mobile robotics, such as search and rescue [2], [3], [4], autonomous navigation [5], and augmented reality [6]. Semantic SLAM [1] relies on visual measurements obtained by a vision sensor. It exploits understanding of the surrounding structure to build highly expressive maps that are easy for human operators to understand. It started to captivate a tremendous amount of attention, especially after the deep learning breakthrough, which led to advancements in object detection and tracking

techniques [7]. The accuracy of the localization is a critical success factor in robotic tasks, particularly those involving interaction with humans. Examples of such tasks are search and rescue, autonomous driving, and elder care. Owing to its infancy, semantic SLAM is yet to achieve more robustness in the presence of noisy measurements, like those occurring due to inaccurate object pose estimation with respect to the vision sensor.

The uncertainty of SLAM estimates might arise due to measurement errors that differ based on the adopted approach to SLAM. In the case of object-based semantic SLAM, errors mostly occur when post-processing the sensory data to determine the poses of the observed features relative to the sensor in the environment. This process starts with detecting the landmark in the environment and determining its bounding box, then computing its centroid. The centroid of the landmark is then utilized to compute the relative pose between the feature and the vision sensor. Furthermore, occlusions have a significant impact on the accuracy of the estimated object pose [8]. Occlusions happen when part of the object is observed in an image, while the rest is either hidden by other objects in the scene or is out of the field of view of the vision sensor. Due to the advancements in deep learning based object detectors, occluded objects can still be detected and correctly labeled in an image. Hence, if they are not properly accounted for, the estimated pose of an occluded object can be far from the true one and may consequently cause severe accuracy degradation. In addition, the limitations of the sensors used to perceive the environment introduce another primary source of uncertainty.

The approach proposed in this paper aims at reducing the joint effects of several sources of errors on the accuracy of semantic SLAM estimates. These errors might arise from limitations of the software and hardware components used to perform semantic SLAM, from external environmental conditions, or from unpredictable noise. Formulating a noise model that accounts for all such errors is challenging, especially because some errors occur unexpectedly during data collection and/or processing. Hence, a stacked LSTM based neural network is proposed in this paper to learn and capture the error patterns associated with the trajectory estimates of semantic SLAM. By comparing the trajectory estimates to the corresponding ground truth, the network is trained to reduce the error and hence enhance the accuracy of semantic SLAM.

The proposed approach is general; it can be used for any SLAM system since it operates on trajectory estimates rather than raw measurements. It targets trajectories of ground vehicles which are usually expressed using three degrees of freedom in the 2-dimensional (2D) space; the vehicle's

<sup>1</sup>Rana Azzam, Yusra Alkendi, and Yahya Zweiri are with KU Center for Autonomous Robotic Systems (KUCARS), Khalifa University of Science and Technology, Abu Dhabi, UAE, e-mail: {rana.azzam, yusra.alkendi, yahya.zweiri}@ku.ac.ae

<sup>2</sup>Tarek Taha, is with Algorhythm's Autonomous Aerial Lab, Abu Dhabi, UAE, email: tarek.taha@algorhythm.com

<sup>3</sup>Shoudong Huang, is with University of Technology Sydney, Sydney, Australia, email: Shoudong.Huang@uts.edu.au

<sup>4</sup>Yahya Zweiri, Faculty of Science, Engineering and Computing, Kingston University London, London SW15 3DW, UK email:y.zweiri@kingston.ac.uk

position  $(x, y)$  and orientation  $(\vartheta)$ . The approach is applicable to any 2D SLAM problem. However, the employed neural network was trained on data obtained using semantic SLAM, and hence, is intended to improve the accuracy of semantic SLAM trajectory estimates.

The proposed approach can be used in applications that require accurate localization of the robotic vehicle. For example, a more accurate estimate of the trajectory estimated by semantic SLAM will result in a meaningful and more accurate map of the environment. Another practical use-case scenario of the approach presented in this paper is in search and rescue applications. If the robots that are employed as first responders after a particular catastrophe have the ability to accurately pinpoint their location, it will expedite the process of rescuing victims, if any, or locating areas that need immediate help.

The contributions of our paper are listed below.

- We developed a novel stacked-LSTM based approach to identify and reduce pose estimation error in object-based semantic SLAM. The approach alleviates the combined effect of predictable and unpredictable noise on the accuracy of trajectory estimates.
- We developed an automated search approach to select the architecture and hyper-parameters of the proposed stacked-LSTM neural network.
- We extensively tested the proposed approach on simulated and real-time experiments, where its superiority compared to shallow neural networks (SNN), deep neural networks (DNN), support vector machines (SVM), and semantic SLAM was proven.

The rest of the paper is organized as follows. Section II presents recent related research work from the literature. The proposed approach is introduced in Section III, followed by experimental validation in Section IV. Finally, the conclusions of this work are drawn in Section V.

## II. RELATED WORK

### A. Deep Neural Networks:

Neural networks are trained to exhibit a particular behavior, suited for the problem at hand, when fed with data. During training, the internal parameters of the network, referred to as weights, are adjusted to minimize the discrepancy between the network's prediction and the desired output [9]. A SNN is a network with an input layer, one hidden layer, and an output layer. Networks with two or more hidden layers are referred to as deep neural networks (DNNs). DNNs are much more efficient than SNNs with regards to the required number of computational units, especially when modelling a complex problem. This is attributed to the non-linear nature of the activation functions occurring at several layers in the DNN [10].

Furthermore, recurrent neural networks (RNN) are artificial neural networks that are capable of informing knowledge from a context. This is attributed to the use of loops which allows information to be fed back to the network after being processed. However, such networks might suffer from vanishing gradients, which motivated the need for Long Short Term Memory (LSTM) cells [11]. LSTM cells enable

RNNs to retain information that are essential, and discard them otherwise. This functionality cannot be realized when using conventional neural networks. DNNs and LSTMs have been exhibiting state-of-the-art performance in a multitude of various applications, including computer vision [12], [13], [14] and robotics [15], [16], [17].

### B. SLAM and the Intervention of Deep Learning

A rich body of literature has addressed the SLAM problem, and a wide range of algorithms exhibiting varying levels of performance in terms of reliability, accuracy, and efficiency have been proposed [18], [19], [20]. The utilization of deep learning approaches has been witnessed in a substantial share of these approaches in the past few years [1], and their capability to outperform the classical approaches has been demonstrated [21], [22], [23], [24], [17]. In addition, deep learning based object detection techniques [25], [26], [27] promoted the advancement of object-based semantic SLAM, which relies on observations of landmarks that can be semantically labelled in the environment, such as the approaches presented in [28], [29]. Obtaining a reliable observation of a landmark in the environment and accurately pinpointing its position with respect to the sensor remains a challenge. On a different note, much less research effort was made in the area of employing deep learning approaches to improve the accuracy of state estimation as discussed in the next section.

### C. Enhancing SLAM Estimation Accuracy

The accuracy of state estimation in SLAM applications is vulnerable to the effects of several error sources. Such errors occur in one or more stages in the SLAM pipeline, such as data collection, data processing, and optimization. Most of the existing work in the literature assumes that noise models always follow fixed distributions that can be mathematically formulated [30]. Nonetheless, this is not always the case in practical applications and might lead to severe degradation in estimation accuracy.

When visual measurements and dead reckoning are used together to estimate the state of a system, estimation uncertainty may result from visual sensor noise [31], [32], landmark detection and localization accuracy [33], odometry drift [34], or failure to arrive at a globally optimum estimate due to measurement noise. The effect of unpredictable nonuniform noise as well as external environmental conditions is also inevitable [35]. To enhance the accuracy of localization, the solutions found in the literature can be classified into: (1) controlling the environment under investigation [36], (2) sensor data fusion [37], [38], (3) improving measurement covariance estimation [39], [30], [35], [40], or (4) correcting measurement errors, which can be further classified into classical [41], [42], [43], [44] and learning approaches [16], [45], [34], [46], [17], [47].

The work presented in [36] studies the placement of passive tags, used as landmarks, in the environment, to always keep the localization accuracy within a particular range. In another vein, the robustness of indoor localization was supported by accumulating sensory data, which compensated for the

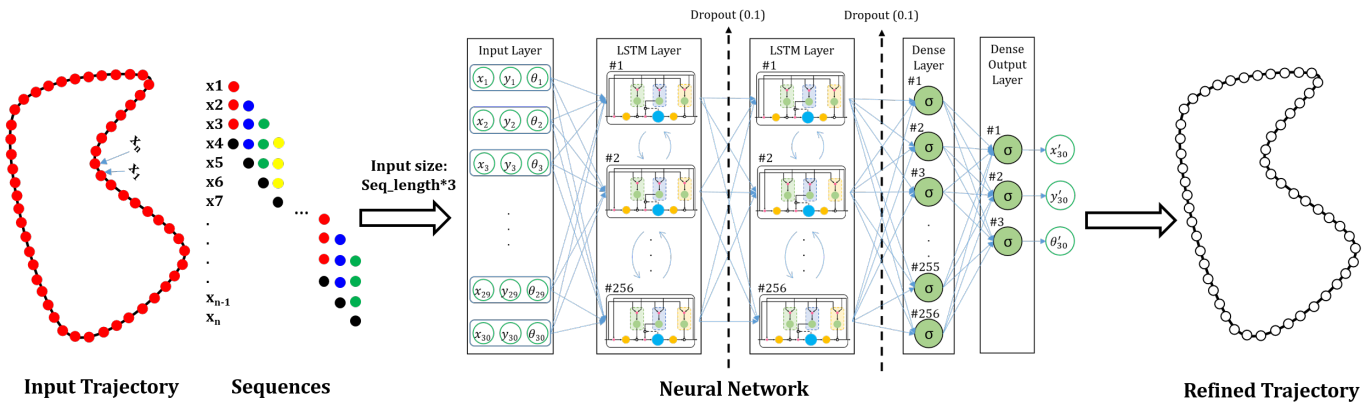


Fig. 1: Proposed deep learning approach

limitations of the employed sonar sensors as presented in [37]. Another example of measurement fusion can be found in [38] where measurements recorded by multiple IMUs along with other exteroceptive sensors were integrated to improve localization accuracy. Instead of assuming a fixed measurement noise model, the work proposed in [39] predicts the noise model based on raw measurements by means of a DNN. The DNN was able to accurately predict the covariance of measurements obtained by light and vision sensors. However, the approach assumes that noise models follow a zero-mean Gaussian distribution which does not hold all the time, putting in doubt the generality of the approach. Similarly, the work presented in [30] relaxes the assumption of a fixed measurement noise model for dead reckoning and QR code detections by employing a tailored extended  $H_\infty$  filter. The approach is general and more computationally expensive than the extended Kalman filter, yet achieves higher accuracy. Similarly, the approach proposed in [35] improves the accuracy of SLAM estimates by employing an adaptive Gaussian particle filter whose job is to compensate for bias in measurements. More particularly, the approach targets unmodeled, unpredictable noise patterns that are experienced in marine environments. A recent noise model learning approach was proposed in [40] where a deep neural network was trained to estimate the covariance of inertial measurements, which are then used in an extended Kalman filter to perform localization. Evaluation results demonstrated the applicability of the approach, yet in its current version, it works for inertial sensors only.

Several previous studies have addressed correction of measurements using classical or learning approaches. In [41], visual sensor limitations were accounted for by superimposing camera oscillations to improve the accuracy of visual SLAM. The work presented in [42] utilizes probabilistic fuzzy logic to reduce measurement uncertainties occurring due to stochastic and non-stochastic disturbances. This approach handles dead-reckoning and range measurements and was proven to outperform ordinary fuzzy logic in terms of improving the accuracy of positioning and mapping estimates. The work presented in [43] was able to cope with occasional failure of inertial sensors during localization, by means of a discrete-time  $H_\infty$  filter. Localization was supported by a reference wireless sensor network. In [44], a novel ultrasonic sensor

with self-configuration abilities was developed to cope with collisions of ultrasonic waves and hence enhance localization accuracy. The developed algorithm is also capable of handling topological changes in the environment in a real-time manner.

In [16], a deep learning based approach is employed to improve the altitude estimation of a flying robotic vehicle. Moreover, in [45], the accuracy of the odometry of a wheeled cart, calculated using its dynamic equations, was improved using an SNN. The network was trained to compute an estimate of the vehicle's traveled distance and orientation. However, since the network is composed of a single hidden layer only, it might not capture all patterns of estimation errors, and hence cannot generalize well. Correction of odometry measurements was also addressed in [34], [46] where Gaussian processes were trained based on the discrepancies between the odometry model and ground truth. The scalability and accuracy of the model proposed in [46] were achieved through deep kernel learning. Furthermore, the approach proposed in [17] improves the accuracy of stereo visual localization. The authors proposed a loss function based on the Lie group  $SE(3)$  geodesic distance and used it to train a DNN to more accurately estimate the relative transformation between subsequent images. The advantages of the proposed approach over classical visual odometry was demonstrated through several experiments. However, the DNN operates on images, which makes the approach computationally expensive and requires a large DNN structure to achieve the sought performance. Additionally, an end-to-end learning approach to visual odometry was presented in [15] where sequential learning was employed to improve the pose certainty. The DNN operates on raw images, from which it infers the uncertainty of the poses. Operating on large amounts of data requires the hardware to have high computational capabilities. The work proposed in [47] utilizes deep learning to improve depth estimation which is then used to perform dense monocular SLAM. In a classical factor graph approach, the authors propose to use multiple objective functions, or factors, addressing several types of errors to further improve estimation accuracy. These factors are the photometric, reprojection, and sparse geometric factors. Combining these factors have resulted in robust motion estimation when tested on several real-time sequences.

The proposed stacked-LSTM based approach has the fol-

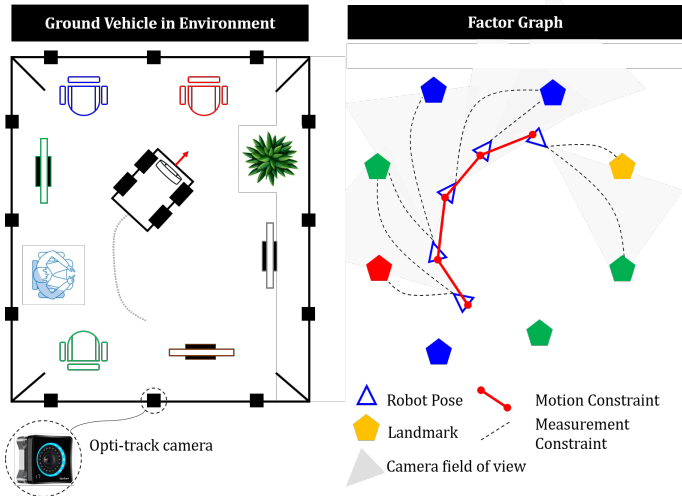


Fig. 2: Semantic SLAM

lowing advantages over the aforementioned methods:

- It alleviates the effects of all the possible disturbances experienced while performing SLAM, including measurement errors, sensor failure, data processing faults, or any other unpredictable noise.
- It operates on a trajectory rather than raw sensor measurements like images. Hence, it can be used to reduce pose estimation errors irrespective of the employed sensors.
- It is efficient since the input to the neural network is a short segment of the trajectory, which takes much less time than images to be processed.
- It does not require any particular arrangement of the environment. More specifically, it does not depend on the number, geometry, or placement of landmarks in the environment.
- The stacked nature of the LSTM and dense layers along with the nonlinear activation functions facilitate identifying complex error patterns that could be challenging to model mathematically.

### III. PROPOSED APPROACH

The deep learning approach proposed in this paper is depicted in Fig. 1. In general, a ground vehicle's trajectory, estimated using semantic SLAM, is passed to a neural network which will identify and reduce possible pose estimation errors. The semantic SLAM algorithm will be described in Section III-A along with the error sources that contribute to reducing the accuracy of pose estimation. Section III-B details the deep learning based pose estimation error reduction approach.

#### A. Semantic SLAM

The adopted semantic SLAM is designed for ground vehicles and is performed based on measurements from the vehicle's wheel encoders and an RGB-D camera that is mounted on top. A factor graph is used to model the problem as shown in Fig. 2, where robot poses and map features (landmarks) are represented as nodes. The graph contains two types of edges: solid edges between every two consecutive pose nodes,

to represent a spatial constraint (denoted as  $e_{mot}$ ), and dashed edges between a pose node and a landmark node, to represent an observation of the landmark at that pose (denoted as  $e_{meas}$ ). Each edge models a nonlinear quadratic constraint, that can be mathematically formulated as shown below:

$$e_{mot} = (x_t - g(u_t, x_{t-1}))^T R_t^{-1} (x_t - g(u_t, x_{t-1})) \quad (1)$$

where  $g$  is the motion model,  $u_t$  is the control command and  $x_t$  is the robot pose at time  $t$ , and  $R_t$  is the covariance matrix of the motion noise, which is assumed to be Gaussian.

$$e_{meas} = (z_t^j - h(x_t, m_j))^T Q_t^{-1} (z_t^j - h(x_t, m_j)) \quad (2)$$

where  $z_t^j$  are measurements,  $h$  is the measurement function,  $m_j$  is a landmark, and  $Q_t$  is the covariance matrix of the measurement noise, which is assumed to be Gaussian.

The goal of graph SLAM is then to find a configuration of the nodes that minimizes the error introduced by the constraints. In our approach, this is done by means of the incremental smoothing and mapping algorithm, iSAM2 [48].

1) *Landmark Pose Estimation and Data Association*: To perform semantic SLAM, the vehicle's relative position to the observed landmarks must be computed. This is done using input RGB-D frames. RGB images are passed to the object detector, you only look once (YOLO) [25]. For each detected object in an image, YOLO predicts a label and a bounding box. The relative position between the camera and the detected object is then computed as the distance between the camera and the centroid of the object, as proposed in [29]. Briefly, the input depth image that corresponds to the input RGB image is converted to point cloud. The point cloud is segmented in order to extract the cluster of points that belongs to the detected landmark using kd-tree search. The geometric centroid of the cluster is then computed.

Detected objects are associated to landmarks in the map based on their label, which is predicted by YOLO. Since multiple objects of the same category might exist in the environment, the observation is associated to the closest landmark within a particular distance threshold. If no landmarks exist within that threshold, a new instance is inserted in the map.

2) *Measurement Uncertainty*: There are several factors that contribute to reducing the estimation accuracy when performing semantic slam. Starting from the inputs, the sensors used to perceive the environment suffer from limitations that decrease the accuracy of the obtained measurements. For example, uncertainty of RGB-D measurements might be caused by (1) axial noise [49], which increases when the distance to the detected object increases, (2) lateral noise [49], which increases near the image corners, (3) multi-path interference [50], (4) flying pixels [50], and (5) the scene's characteristics, such as color variations, temperature [50], and illumination conditions. In addition, odometry drift is affected by the accuracy of wheel encoders [34], wheel materials, floor flatness and materials. Fig. 3 describes the model that was used to simulate odometry noise and add it to simulated odometry which is considered to be perfect with no error.

Furthermore, object detection might result in incorrect labels or bounding box predictions. If such false detections are not treated as outliers, the accuracy of pose estimation and



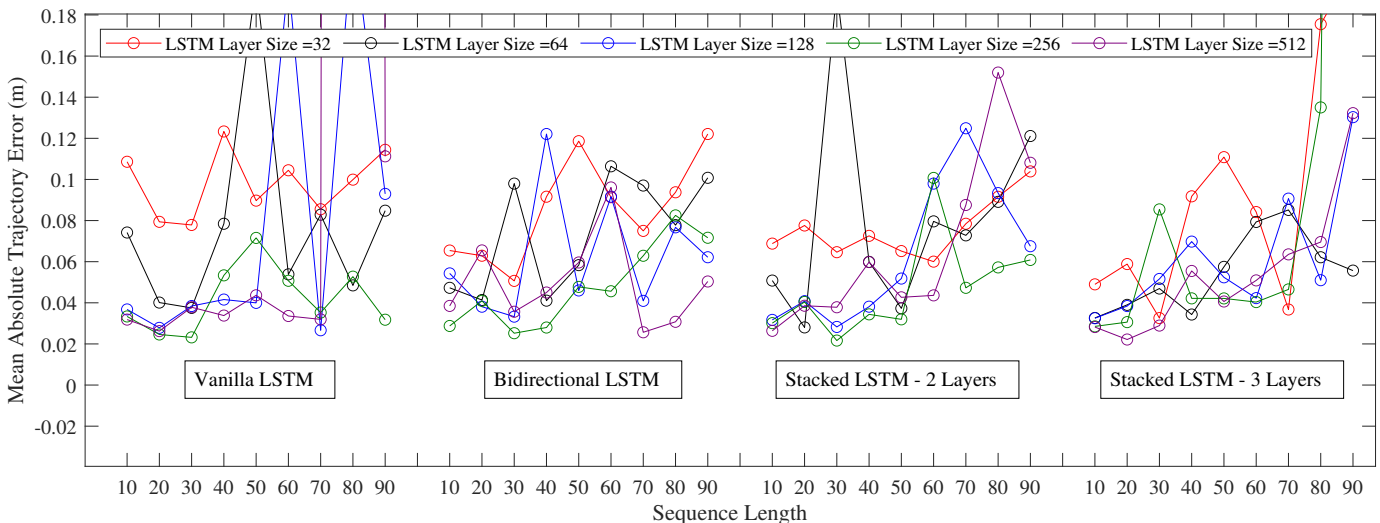


Fig. 5: Mean ATE obtained on the training and validation datasets by different LSTM network architectures

are shown in Fig. 5. Several architectures performed well and were able to improve the accuracy of the estimated trajectory. However, the architecture with two stacked LSTM layers, each with 256 units, with a sequence length of 30 exhibited the highest performance among all the considered LSTM networks in terms of reducing ATE.

In an attempt to further improve the results, one, two, and three dense layers were added after the LSTM layers and various sizes and activation functions were tested, including sigmoid, swish [51], tanh, ReLU, and linear. Adding a dense layer of size 256 with a sigmoid activation function resulted in smoother trajectories with lower mean ATE. The mean ATE achieved by the other hybrid architectures did not improve. To aid generalization and overcome overfitting, dropout layers were added to the architecture.

Fig. 6 depicts the architecture of the adopted stacked LSTM neural network. The network accepts trajectory segments of length 30. Each of the poses in a segment consists of a triple,  $X = [x_m, y_m, \vartheta_m]^T$ , representing the position and orientation of the vehicle at a time instance. The segment is then passed to two stacked LSTM layers, separated by a dropout layer, with dropout probability of 0.1. Each LSTM layer consists of 256 units. Then, after another dropout layer, a fully connected layer with 256 neurons, activated by sigmoid is added. Finally, a dense layer, activated by sigmoid is used to predict the improved pose.

The computational complexity of the proposed model per time step is  $O(W)$  where  $W$  is the size of the weight-space. This is attributed to the fact that the time complexity to update a single weight is  $O(1)$ . The size of the weight-space is a function of the input size, hidden units, and output size [52], which were detailed earlier. The total number of trainable parameters in the proposed model is around 850k parameters. The architecture of the proposed model and its hyper-parameters, such as the batch size and the number of training epochs, are fixed. Hence, for  $N$  training samples, the complexity becomes  $O(N)$  since one training epoch runs in  $O(1)$ .

The proposed neural network will be compared to shallow

and deep fully connected neural networks (SNN and DNN respectively) and to support vector machines (SVM). Hence, a pool of SNNs, DNNs, and SVMs were investigated to search for the best structure from each paradigm for our problem. The parameters that were varied for SNN and DNN are the number of neurons per layer and the activation functions. Different depths of DNNs were also attempted. As for SVMs, a set of variables, like the kernel and its corresponding parameters, were changed and the SVM that resulted in the lowest mean ATE across the training and validation datasets was selected. The search for the most suited SNN, DNN, and SVM was done to ensure the fairness of our comparisons.

#### IV. EXPERIMENTAL VALIDATION

In this section, the proposed approach is validated through a set of simulated and real-time sequences from publicly available datasets. The performance of the stacked LSTM neural network is then compared against other regression techniques, including SVM, SNN, and DNN where it proved to outperform them.

As mentioned earlier, the proposed approach can be applied to trajectories computed using any 2D SLAM. However, the datasets used here were obtained using semantic SLAM.

The rest of this section is organized as follows: The experimental set-up used to record the training dataset is presented in Section IV-A. In Section IV-B, the structure of the training datasets is described, followed by details about the training process. After that, the performance of the proposed approach is analyzed then compared against that of SNN, DNN, and SVM in Section IV-C. Finally, in Section IV-D, the proposed approach is tested on a set of simulated and real-time experiments, including three SLAM sequences from the TUM RGB-D dataset [53].

##### A. Experimental Set-up

A simulated pioneer 3AT robot with an RGB-D camera mounted in a front-forward position was used to navigate in

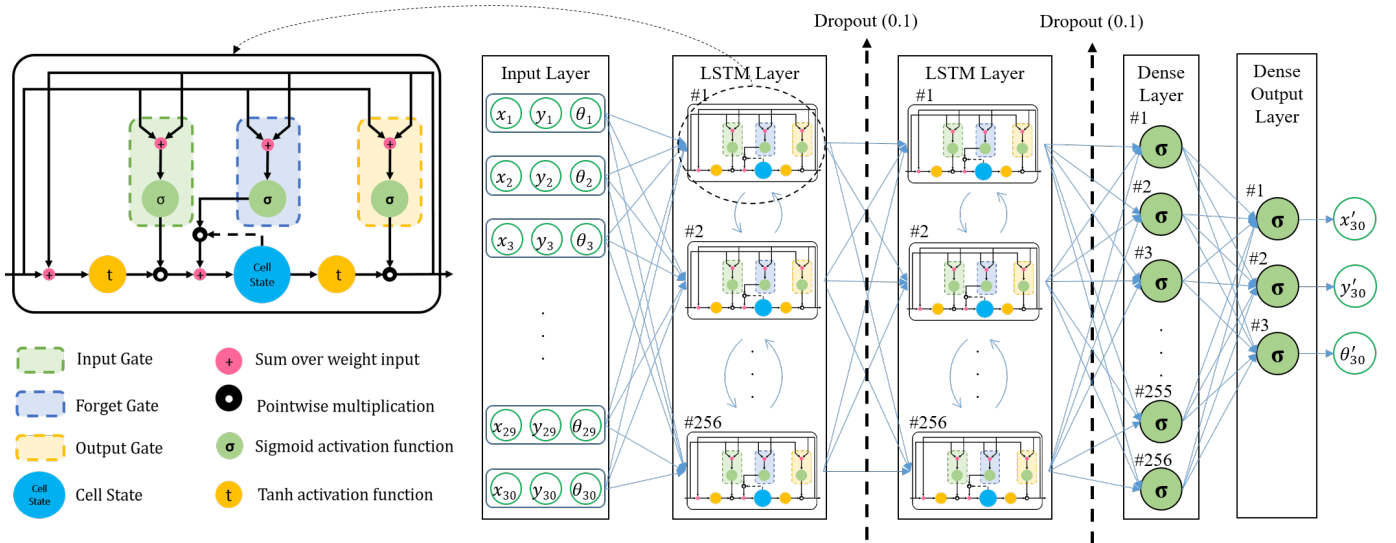


Fig. 6: Proposed Neural Network Architecture

several simulated environments and collect the sensory data required to perform semantic SLAM. A sample of the simulated environments is shown in Figure 7 with the simulated robotic vehicle at its starting point. This environment is  $13 \times 10m^2$  and is populated with 37 object instances of two different categories; 19 TV monitors and 18 bottles. Other simulated environments were also used where the object categories include potted plant, table, and person (who is assumed to be static while recording the experiment).

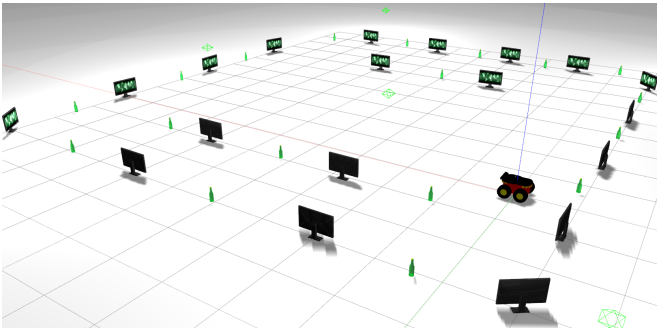


Fig. 7: Simulated environment setup with ground vehicle at its starting point

Several simulated trajectories were recorded in these simulated environments then passed to the semantic SLAM algorithm to generate training data. Since the odometry measurements obtained from simulations are perfect, the odometry noise model described in Figure 3, was used to simulate noise and add it to the recorded measurements. Another source of error was observed when passing the RGB frames to the object detector, YOLO [25], then performing segmentation to determine the centroid of the observed object. The centroids of the detected objects were seen to deviate from their true positions. This was mainly due to object occlusions since YOLO was able to detect an object even if part of it is occluded. Consequently, only the visible part of the object was used to compute the centroid of that instance, causing an error in the measurement. The error varies depending on

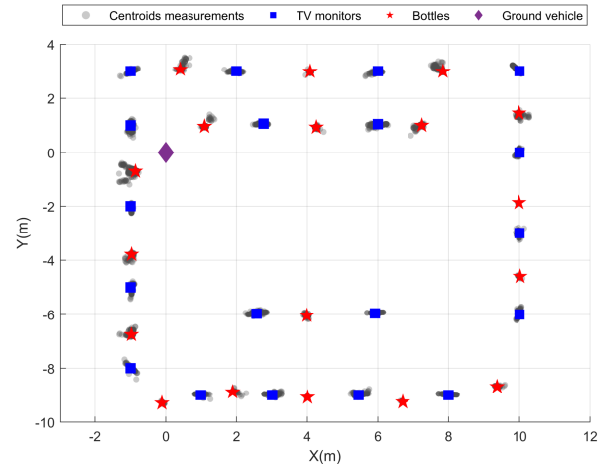


Fig. 8: Distribution of object measurements from some recorded simulated experiments in the environment shown in Fig. 7

the size of the object and hence the standard deviation of the noise associated with each object observation was set based on the object's dimensions. Object measurements from some recorded experiments in the previously described simulated environment are shown in Figure 8.

Real-time experiments were taken from the TUM RGB-D dataset [53], where a Pioneer 3AT robot with a Kinect RGB-D sensor mounted in a front forward position was joysticked in a large hole. Multiple instances of chairs and tables appeared in the environment and were used as observations to perform semantic SLAM. Recordings of experiments are provided with the corresponding ground truth which was used during the training process. Table I summarizes details about the three trajectories taken from the dataset.

The specifications of the computer used to conduct the semantic SLAM experiments are listed in Table II.

The semantic SLAM algorithm was implemented using the Robot Operating System (ROS) [54] Kinetic on Ubuntu 16.04.



TABLE I: Summary of TUM RGB-D sequences used for evaluating the proposed approach

Name	Tajectory Length	Trajctory Dimensions
Freiburg_pioneer_slam	40.38m	5.50m x 5.94m
Freiburg_pioneer_slam2	21.735m	4.98m x 5.34m
Freiburg_pioneer_slam3	18.135m	5.29m x 5.25m

TABLE II: Hardware specifications

Computer type:	ASUS STRIX laptop
Processor:	with Intel core i7-6700HQ @ 2.60GHz × 8
System type:	64-bit operating system
Operating System:	Linux – Kubuntu 16.04 distribution

The communication between the simulated/real hardware and ROS was performed through RosAria and OpenNI for the ground vehicle and the RGB-D sensor, respectively. The system software was implemented in C++, where gtsam [55] and its iSAM2 [48] implementation was used to perform incremental smoothing and mapping, YOLO [25] was used for object detection, openCV [56] and depth\_image\_proc were used for processing RGB and depth images respectively, and point cloud library (PCL) [57] was used to process point clouds.

### B. Dataset Preparation

A total of 18 different simulated trajectories were generated and used to construct the dataset. Every estimated trajectory was divided into smaller overlapping segments of length 30 as described in Fig. 4. The ground truth corresponding to the last pose in each segment, is set as the target for that input segment and is referred to as  $T = [x, y, \vartheta]^T$ . In simulated experiments, the recorded odometry measurements, before adding simulated noise, is set as the target. In real-time experiments, ground truth data was obtained from the dataset online.

The output of the network is an improved estimate of the robot’s 2D poses along the trajectory, where each pose is denoted as  $Y = [x_e, y_e, \vartheta_e]^T$ . It is worth mentioning that in the current version of the system, reduction of pose estimation error is done offline.

For the neural network to perform well, the datasets need to be re-scaled to a common range. To that end, all the collected data were normalized to the range [0.05, 0.95]. The reason why this particular range is selected is to avoid the problem of vanishing gradients that occurs when the neurons saturate, i.e. reach the minimum or maximum value of the activation function (0 and 1 respectively for sigmoid), and hence the derivative of the function at that point drops to values close to zero. Using the same normalization parameters, predictions were re-scaled to the original data range.

Backpropagation [58] was used to train the network in a supervised manner. The Adam optimizer was employed, with a learning rate of 0.0001, to minimize the mean absolute error, over 1000 training epochs. The batch size was set to 100. Building, training, and testing the different neural networks was done using Keras [59] with a Tensorflow backend (version 2.0).

### C. Performance Evaluation

In this section, the performance of the proposed stacked LSTM will be compared to DNN, SNN, and SVM using a set of simulated trajectories. The datasets were randomly split into two parts; 80% for training, and 20% for validation to aid the model’s regularization.

To ensure fairness when comparing the proposed approach to DNN, SNN, and SVM a similar search strategy was adopted to find the most suited architecture using the same training dataset. Fully connected SNNs with varying activation functions, including sigmoid, swish, ReLU, linear and tanh, and layer sizes set to  $2^m$  for  $m \in [5, 9]$ , were examined. Along the same lines, fully connected DNNs with depths varying from two to six layers, layer sizes set to  $2^m$  for  $m \in [5, 9]$ , and activation functions including sigmoid, swish, ReLU, linear and tanh, were investigated. The SNN and DNN that achieved the lowest mean ATE were selected to be compared to the proposed approach. The SNN that performed the best in terms of reducing pose estimation error had 512 neurons in its single hidden layer and was activated using ReLU. The DNN, on the other hand, had 6 hidden layers, each of size 256 neurons, and activated using sigmoid. The third regression technique that will be compared against the proposed approach is SVM. More particularly, varying structures of the epsilon support vector regression model [60] were explored. Several kernels including linear, sigmoid, and polynomial with varying regularization and epsilon parameters were tested. The best performing SVM out of the tested pool was of a 5-degree polynomial kernel. The predictions of these three models were compared to that of the proposed stacked-LSTM approach as will be presented next.

Fig. 9 depicts nine different trajectories, along which the performance of the proposed approach is evaluated and compared to the other alternatives. It is evident that the proposed stacked LSTM was capable of identifying error patterns in the input trajectories and significantly improving them along all the depicted trajectories. DNN predictions have also shown substantial improvements to the trajectories, yet there are segments of the trajectories where DNN predictions still suffered from errors. SNN predictions demonstrated improvements at times, but especially after turns in the trajectory, predictions exhibited high fluctuations and hence large ATE. SVM predictions were mostly less accurate than the input trajectories generated by semantic SLAM and hence, no improvement to the ATE was observed.

Table III lists the mean ATE achieved by the stacked LSTM network, SNN, DNN, and SVM along each trajectory. SVM predictions have not shown any improvement to the mean ATE along any of the nine trajectories. Stacked LSTM, SNN, and DNN, on the other hand, were able to reduce the mean ATE along all the trajectories. However, the proposed approach has clearly outperformed all the other alternatives, by achieving the lowest mean ATE along all the trajectories.

### D. Performance Analysis on Publicly Available Datasets

To further verify the applicability of the proposed approach, the training dataset was extended to include more simulated

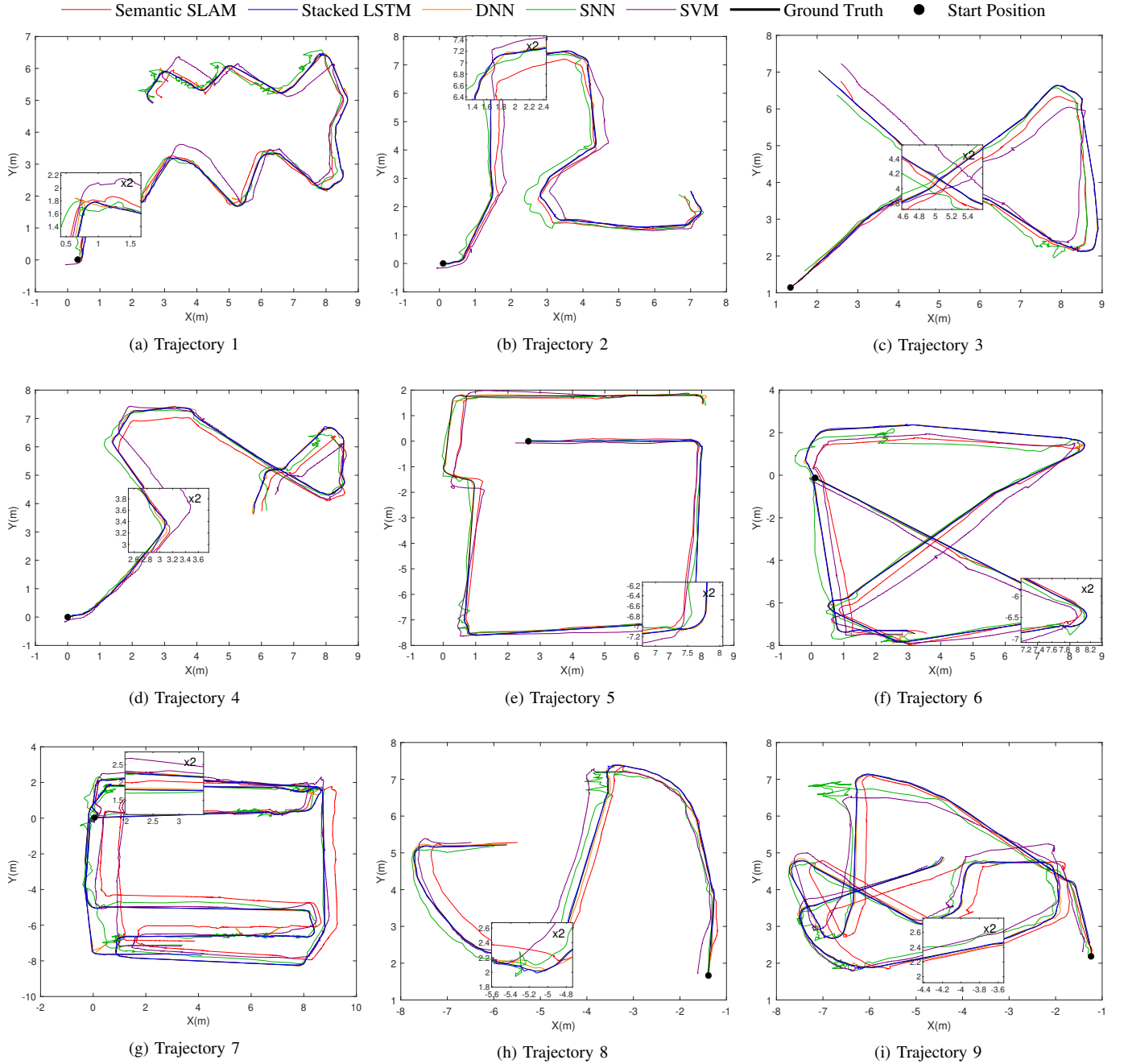


Fig. 9: Training and validation results on simulated trajectories

TABLE III: Comparison of mean ATE (m) achieved along the trajectories in Fig. 9

	Semantic SLAM	Stacked LSTM	SNN	DNN	SVM
Trajectory 1	0.20	0.025	0.19	0.048	0.68
Trajectory 2	0.20	0.021	0.16	0.035	0.59
Trajectory 3	0.30	0.025	0.16	0.034	0.68
Trajectory 4	0.26	0.024	0.15	0.040	0.58
Trajectory 5	0.17	0.019	0.095	0.034	0.57
Trajectory 6	0.32	0.024	0.26	0.037	0.75
Trajectory 7	0.71	0.022	0.22	0.042	0.86
Trajectory 8	0.24	0.024	0.15	0.047	0.42
Trajectory 9	0.32	0.025	0.17	0.042	0.48

and real-time experiments. The latter were taken from publicly available datasets that are used as a benchmark by the

robotics community, particularly, Freiburg2\_Pioneer\_SLAM, Freiburg2\_Pioneer\_SLAM2, and Freiburg2\_Pioneer\_SLAM3 from the TUM RGB-D dataset [53]. These public datasets resemble a practical use-case scenario where a ground vehicle performs a maneuver in an indoor environment, populated with objects. The vehicle is equipped with wheel encoders and an RGB-D sensor mounted in a front-forward position, as described in Section IV-A. Vision measurements and the concurrent odometry are passed to the semantic SLAM system, which estimates the vehicle's trajectory in the environment. The process of acquiring and processing data is vulnerable to several sources of error that hinder the accuracy of the esti-

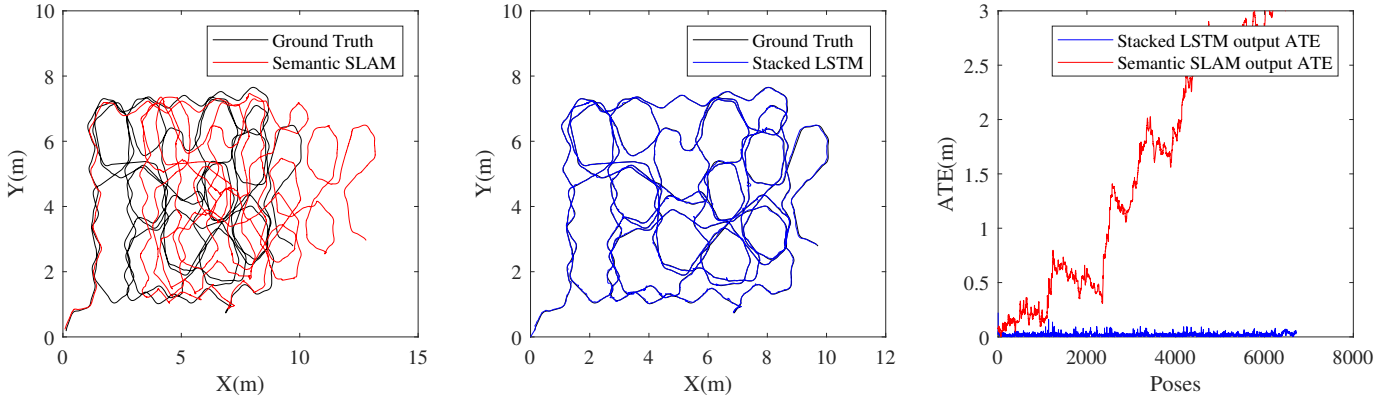


Fig. 10: Simulated experiment - sample 1

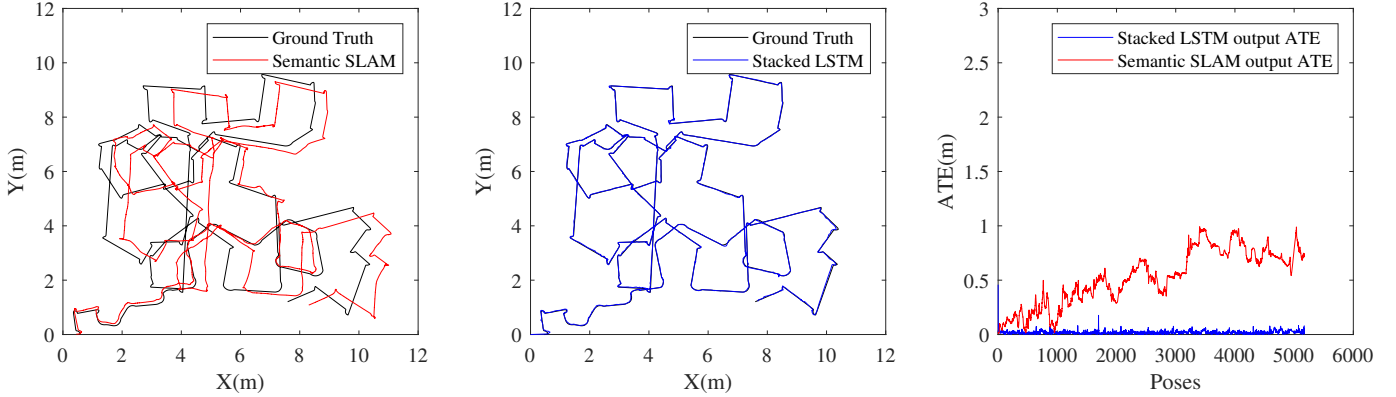


Fig. 11: Simulated experiment - sample 2

mated trajectory. To reduce such inaccuracies, the trajectory is passed to the stacked-LSTM neural network which in turn identifies and reduces pose estimation errors. The trajectory is divided into small overlapping segments as depicted in Fig. 4 and described in Section III-B. Each pose along the trajectory is corrected based on the vehicle's preceding poses. In the current version, the correction is done offline. The same process is applicable to any 2D SLAM estimate.

The training dataset, including all the simulated and real-time trajectories, was divided into three parts, 70% for training, 15% for validation, and 15% for testing. The training set will be used to optimize the weights of the network during the training process. The performance of the neural network on the validation set will be used to further update the network's weights after every training epochs. Finally, an unbiased evaluation of the network will be obtained using the testing set. Table IV lists the mean ATE that the proposed approach achieved on the training, validation and testing sets. It is evident that the stacked LSTM was able to identify and significantly reduce the error patterns along the trajectories in the dataset. The mean ATE on the testing dataset, which was not exposed to the network during training, dropped from 65cm to 2cm. This proves the validity of the proposed approach on simulated and real-time experiments. Examples of trajectories from simulated and real-time experiments are depicted in Figures 11-15. The leftmost plot in each figure shows the ground truth trajectory and the trajectory estimated by semantic SLAM. The middle plot shows the output of the

proposed approach compared to the ground truth trajectory. The rightmost plot depicts the ATE along the trajectory for both the input to the network and its output. Fig. 15 depicts the regression plot of the variable  $\vartheta$  for the sequence depicted in Fig. 14. The Pearson's regression coefficient,  $R$ , is equal to 0.99996 as shown in the plot. The regression plot for  $\vartheta$  in other sequences is very similar.

The proposed stacked-LSTM based approach can generalize well and is robust to input perturbations. However, the network may be fine-tuned to learn new noise models, which were never exposed to the network during training, and hence be able to recognize a wider variety of pose estimation errors. A portion of the data obtained from the new environment can be used to fine-tune the network, which will then be able to reduce pose estimation error along trajectories recorded under the same conditions. If the training dataset consists of a wide range of error patterns, the neural network will have more potential to perform error reduction along previously unseen trajectories.

A possible use-case scenario of the proposed approach is in search and rescue applications. A robot's mission could be to find victims in a collapsed structure, then notify the rescue teams of the victim's location. While performing semantic SLAM, the robot can navigate in the environment and once a victim is found, the robot's trajectory is communicated to the rescue team. This trajectory is first refined, using the proposed stacked-LSTM based noise reduction approach, to pinpoint the robot's position accurately. This will help the

TABLE IV: Mean ATE (m) achieved by the proposed approach on the training, validation, and testing datasets

	Training set	Validation set	Testing set
Input mean ATE (m)	0.65	0.66	0.66
Output mean ATE (m)	0.021	0.026	0.025

TABLE V: Input and output mean ATE (m)

	Input mean ATE (m)	Output mean ATE (m)
Simulated Trajectory 1 (Fig. 10)	1.59	0.051
Simulated Trajectory 2 (Fig. 11)	0.53	0.041
Freiburg2_Pioneer_SLAM Trajectory (Fig. 12)	1.11	0.057
Freiburg2_Pioneer_SLAM2 Trajectory (Fig. 13)	0.69	0.048
Freiburg2_Pioneer_SLAM3 Trajectory (Fig. 14)	0.52	0.042

human responders to arrive at the location in a shorter time.

Another use-case scenario of the proposed approach can be seen in applications that require the robot to map its surrounding environment. After the robot gathers the required visual information from the environment, its trajectory is estimated using semantic SLAM then passed to the proposed stacked-LSTM based neural network for possible error identification and reduction. Reprojecting visual observations along the corrected trajectory will result in a more accurate and reliable map of the environment compared to that obtained directly from the solver.

## V. CONCLUSION

The error in estimating vehicle and landmark poses significantly hinders the success of semantic SLAM and its usability in high-accuracy critical applications. Several predictable and unpredictable sources of uncertainties contribute to forming such error, including landmark local pose estimation, object detection, incorrect data association, visual sensor noise, and odometry drift. The work proposed in this paper employs a novel, general, and efficient deep learning approach to enhance the robustness of semantic SLAM, by reducing the combined effect of such errors on the trajectory estimation. A stacked LSTM based neural network was developed after conducting an extensive search among different neural network types and hyperparameters. The architecture was adopted based on the network's ability to capture various error patterns and significantly decrease the estimation error. Simulated and real-time experiments, including three sequences from the TUM RGB-D dataset, were used to measure the performance of the proposed approach. The results have proven the ability of the proposed approach to successfully identify and reduce pose estimation errors resulting from multiple factors in the semantic SLAM pipeline. The performance of neural network was quantified using the mean absolute trajectory error. It was compared to that of SNN, DNN, and SVM on several test sets, and the maximum estimation error reduction was evidently achieved by the proposed approach.

The capability of the proposed approach to generalize well to new semantic SLAM trajectory estimates relies heavily on the training dataset. The dataset should include samples collected from different environments, under varying conditions, and using various sets of sensors. If the network is exposed to a wide range of error patterns during training, it will have a higher potential to perform error reduction in previously unseen trajectories.

The work presented in this paper can be extended to 3D SLAM which applies to a wider range of robotic vehicles like aerial vehicles. In such case, the neural network will be expected to predict more variables and account for various noise models. It can also be integrated into the online SLAM algorithm and hence error reduction will be performed right after computing a new SLAM estimate, making it more efficient.

## ACKNOWLEDGEMENTS

This publication is based upon research work supported by the Khalifa University of Science and Technology under Award No. RC1-2018-KUCARS

## REFERENCES

- [1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.
- [2] X. Chen, H. Zhang, H. Lu, J. Xiao, Q. Qiu, and Y. Li, "Robust SLAM system based on monocular vision and LiDAR for robotic urban search and rescue," *SSRR 2017 - 15th IEEE International Symposium on Safety, Security and Rescue Robotics, Conference*, pp. 41–47, 2017.
- [3] A. Denker and M. C. Işeri, "Design and implementation of a semi-autonomous mobile search and rescue robot: SALVOR," *IDAP 2017 - International Artificial Intelligence and Data Processing Symposium*, 2017.
- [4] J. Casper and R. R. Murphy, "Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 33, no. 3, pp. 367–385, 2003.
- [5] A. Pfunder, P. V. Borges, A. R. Romero, G. Catt, and A. Elfes, "Real-time autonomous ground vehicle navigation in heterogeneous environments using a 3D LiDAR," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-September, pp. 2601–2608, 2017.
- [6] D. Ramadasan, M. Chevaldonne, and T. Chateau, "Real-time slam for static multi-objects learning and tracking applied to augmented reality applications," in *2015 IEEE Virtual Reality (VR)*, March 2015, pp. 267–268.
- [7] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object Detection in 20 Years: A Survey," pp. 1–39, 2019. [Online]. Available: <http://arxiv.org/abs/1905.05055>
- [8] S. Soetens, A. Sarris, and K. Vansteenhuyse, "Pose Estimation Errors, the Ultimate Diagnosis," *European Space Agency, (Special Publication) ESA SP*, vol. 1, no. 515, pp. 181–184, 2002.
- [9] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [10] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," *Advances in Neural Information Processing Systems*, no. 1, pp. 153–160, 2007.
- [11] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [13] C. Farabet, C. Couprie, L. Najman, and Y. Lecun, "Learning Hierarchical Features for Scene Labeling," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1915–1929, 2013.
- [14] A. Garcia-Perez, F. Gheriss, D. Bedford, A. Garcia-Perez, F. Gheriss, and D. Bedford, "Going deeper with convolutions," *Designing and Tracking Knowledge Management Metrics*, pp. 163–182, 2019.
- [15] S. Wang, R. Clark, H. Wen, and N. Trigoni, "End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks," *International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 513–542, 2018.
- [16] M. K. Al-Sharman, Y. Zweiri, M. A. K. Jaradat, R. Al-Husari, D. Gan, and L. D. Seneviratne, "Deep-learning-based neural network training for state estimation enhancement: Application to attitude estimation," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 1, pp. 24–34, 2019.

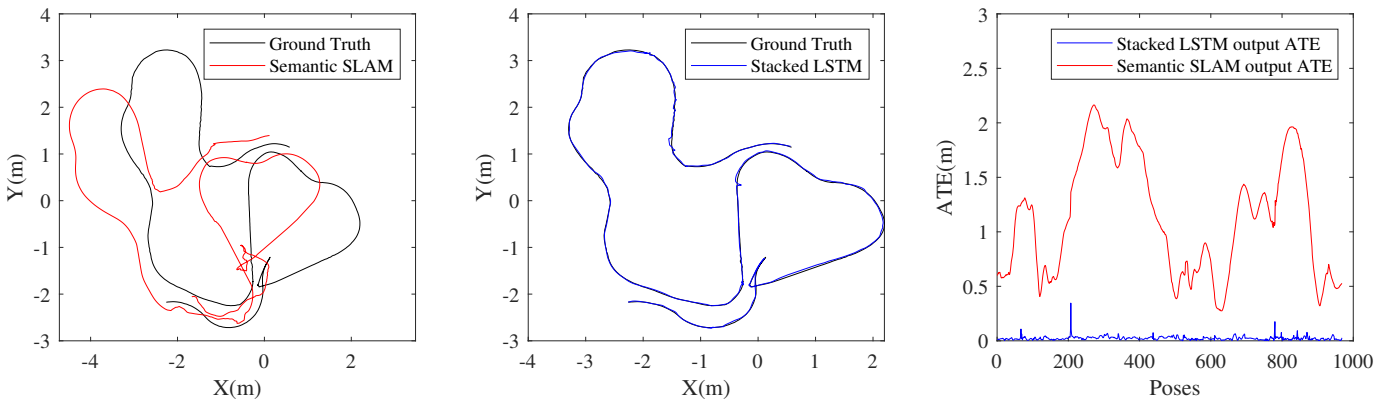


Fig. 12: Real-time experiment - Freiburg2\_Pioneer\_SLAM sequence

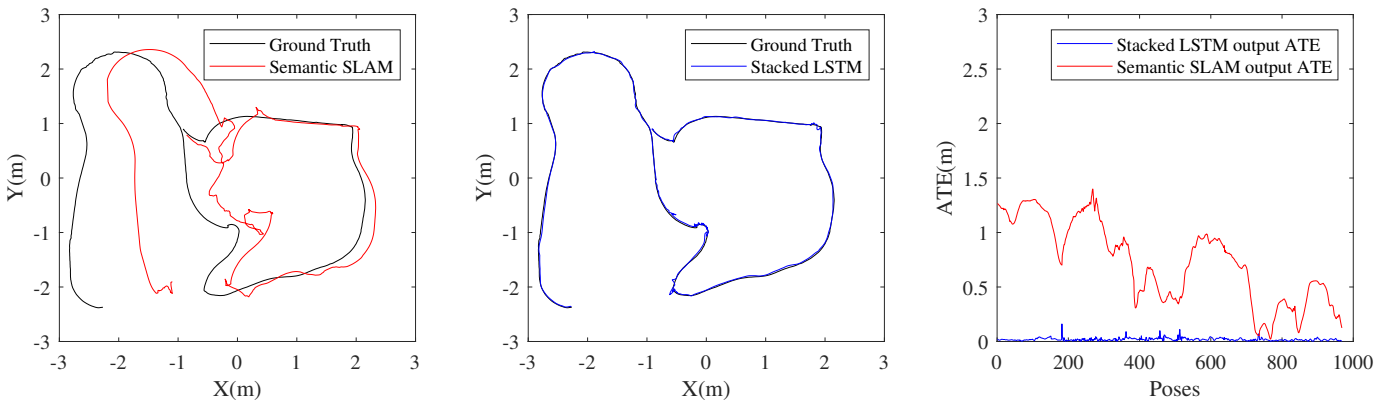


Fig. 13: Real-time experiment - Freiburg2\_Pioneer\_SLAM2 sequence

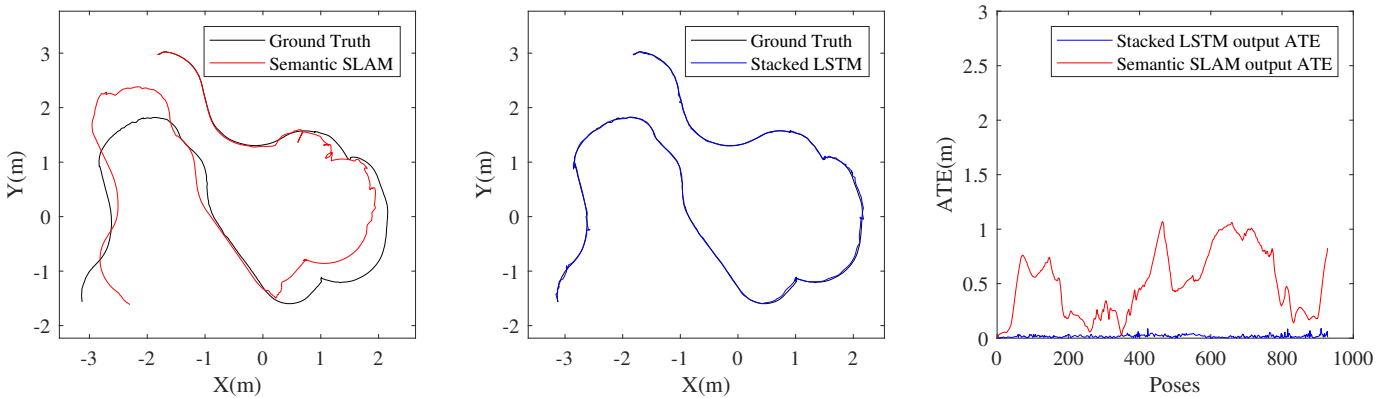


Fig. 14: Real-time experiment - Freiburg2\_Pioneer\_SLAM3 sequence

- [17] V. Peretroukhin and J. Kelly, "DPC-Net: Deep Pose Correction for Visual Localization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2424–2431, 2018.
- [18] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.
- [19] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-Scale Direct monocular SLAM," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8690 LNCS, no. PART 2, pp. 834–849, 2014.
- [20] R. Gomez-Ojeda, F. A. Moreno, D. Zuñiga-Noël, D. Scaramuzza, and J. Gonzalez-Jimenez, "PL-SLAM: A Stereo SLAM System Through the Combination of Points and Line Segments," *IEEE Transactions on Robotics*, vol. 35, no. 3, pp. 734–746, 2019.
- [21] G. Costante, M. Mancini, P. Valigi, and T. A. Ciarfuglia, "Exploring Representation Learning With CNNs for Frame-to-Frame Ego-Motion Estimation," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 18–25, 2016.
- [22] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 International Conference on Computer Vision, ICCV 2015, pp. 2650–2658, 2015.
- [23] C. Cadena, A. Dick, and I. D. Reid, "Multi-modal auto-encoders as joint estimators for robotics scene understanding," *Robotics: Science and Systems*, vol. 12, 2016.
- [24] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-dof camera relocalization," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 International Conference on Computer Vision, ICCV 2015, pp. 2938–2946, 2015.
- [25] J. S. D. R. G. A. F. Redmon, "(YOLO) You Only Look Once: Unified, Real-Time Object Detection," *Cvpr*, 2016.
- [26] Y. Konishi, Y. Hanzawa, M. Kawade, and M. Hashimoto, "SSD: Single

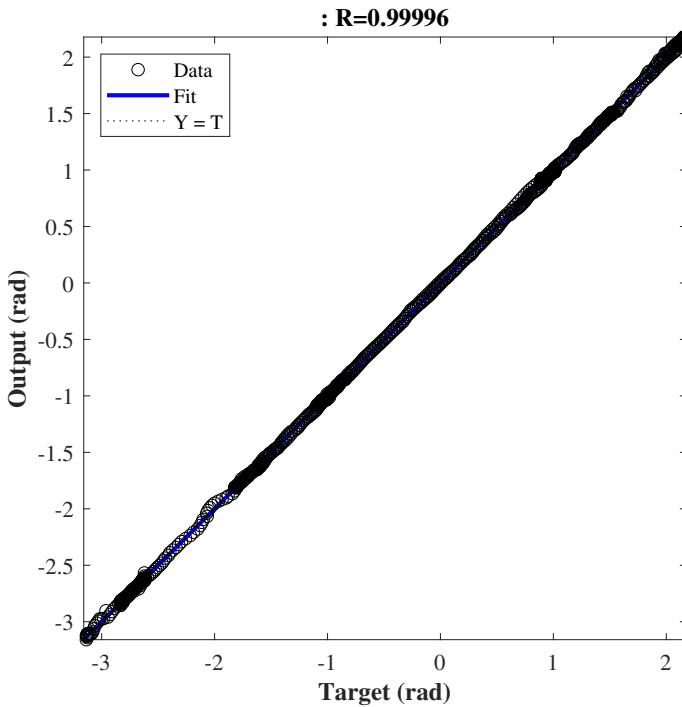


Fig. 15: Regression plot for the orientation variable  $\vartheta$  for the Freiburg\_Pioneer\_SLAM3 sequence

- Shot MultiBox Detector,” *Springer International Publishing AG*, vol. 1, pp. 398–413, 2016.
- [27] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, “Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects,” no. CoRL, pp. 1–11, 2018. [Online]. Available: <http://arxiv.org/abs/1809.10790>
- [28] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, “Probabilistic data association for semantic SLAM,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1722–1729, 2017.
- [29] B. Mu, S. Y. Liu, L. Paull, J. Leonard, and J. P. How, “SLAM with objects using a nonparametric pose graph,” *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016–November, pp. 4602–4609, 2016.
- [30] P. Nazemzadeh, D. Fontanelli, D. Macii, and L. Palopoli, “Indoor localization of mobile robots through QR code detection and dead reckoning data fusion,” *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 6, pp. 2588–2599, 2017.
- [31] P. Ozog and R. M. Eustice, “On the importance of modeling camera calibration uncertainty in visual SLAM,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3777–3784, 2013.
- [32] J. H. Park, Y. D. Shin, J. H. Bae, and M. H. Baeg, “Spatial uncertainty model for visual features using a KinectTM sensor,” *Sensors (Switzerland)*, vol. 12, no. 7, pp. 8640–8662, 2012.
- [33] N. S underhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford, and P. Corke, “The limits and potentials of deep learning for robotics,” *International Journal of Robotics Research*, vol. 37, no. 4–5, pp. 405–420, 2018.
- [34] J. Hidalgo-Carrio, D. Hennes, J. Schwendner, and F. Kirchner, “Gaussian process estimation of odometry errors for localization and mapping,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 5696–5701, 2017.
- [35] A. Rao and W. Han, “An Adaptive Gaussian Particle Filter based Simultaneous Localization and Mapping with dynamic process model noise bias compensation,” *Proceedings of the 2015 7th IEEE International Conference on Cybernetics and Intelligent Systems, CIS 2015 and Robotics, Automation and Mechatronics, RAM 2015*, pp. 210–215, 2015.
- [36] V. Magnago, L. Palopoli, R. Passerone, D. Fontanelli, and D. Macii, “Effective Landmark Placement for Robot Indoor Localization with Position Uncertainty Constraints,” *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 11, pp. 4443–4455, 2019.
- [37] H. Liu, F. Sun, B. Fang, and X. Zhang, “Robotic Room-Level Localization Using Multiple Sets of Sonar Measurements,” *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 1, pp. 2–13, 2017.
- [38] M. Zhang, X. Xu, Y. Chen, and M. Li, “A Lightweight and Accurate Localization Algorithm Using Multiple Inertial Measurement Units,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1508–1515, 2020.
- [39] K. Liu, K. Ok, W. Vega-Brown, and N. Roy, “Deep inference for covariance estimation: Learning Gaussian noise models for state estimation,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1436–1443, 2018.
- [40] M. Brossard, A. Barrau, and S. Bonnabel, “AI-IMU Dead-Reckoning,” *IEEE Transactions on Intelligent Vehicles*, pp. 1–1, 2020.
- [41] M. Heshmat, M. Abdellatif, and H. Abbas, “Improving visual SLAM accuracy through deliberate camera oscillations,” *ROSE 2013 - 2013 IEEE International Symposium on Robotic and Sensors Environments, Proceedings*, pp. 154–159, 2013.
- [42] S. Chen and C. Chen, “Probabilistic fuzzy system for uncertain localization and map building of mobile robots,” *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 6, pp. 1546–1560, 2012.
- [43] H. Hur and H. S. Ahn, “Unknown Input  $H_\infty$  observer-based localization of a mobile robot with sensor failure,” *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 6, pp. 1830–1838, 2014.
- [44] J. W. Yoon and T. Park, “Maximizing localization accuracy via self-configurable ultrasonic sensor grouping using genetic approach,” *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 7, pp. 1518–1529, 2016.
- [45] J. Toledo, J. D. Pi neiro, R. Arnay, D. Acosta, and L. Acosta, “Improving odometric accuracy for an autonomous electric cart,” *Sensors (Switzerland)*, vol. 18, no. 1, 2018.
- [46] M. Brossard and S. Bonnabel, “Learning wheel odometry and imu errors for localization,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019–May, pp. 291–297, 2019.
- [47] J. Czarnowski, T. Laidlow, R. Clark, and A. J. Davison, “DeepFactors: Real-Time Probabilistic Dense Monocular SLAM,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 721–728, 2020.
- [48] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, “ISAM2: Incremental smoothing and mapping using the Bayes tree,” *International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.
- [49] C. V. Nguyen, S. Izadi, and D. Lovell, “Modeling kinect sensor noise for improved 3D reconstruction and tracking,” *Proceedings - 2nd Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization and Transmission, 3DIMPVT 2012*, pp. 524–530, 2012.
- [50] T. S. B. Z. Zhou, G. Zhao, and M. Pietik, “Comparison of Kinect V1 and V2 Depth Images in Terms of Accuracy and Precision,” vol. 1, no. March, pp. 277–289, 2017.
- [51] B. Zoph and Q. V. Le, “Searching for activation functions,” *6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings*, pp. 1–13, 2018.
- [52] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, p. 1735–1780, Nov. 1997. [Online]. Available: <https://doi.org/10.1162/neco.1997.9.8.1735>
- [53] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of RGB-D SLAM systems,” *IEEE International Conference on Intelligent Robots and Systems*, pp. 573–580, 2012.
- [54] A. Koubaa, Ed., *Robot Operating System (ROS): The Complete Reference (Volume 3)*, ser. Studies in Computational Intelligence. Springer, 2018, vol. 778.
- [55] F. Dellaert, “Factor graphs and GTSAM: A hands-on introduction,” no. September, pp. 1–26, 2012.
- [56] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.
- [57] R. B. Rusu and S. Cousins, “3D is here: Point Cloud Library (PCL),” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1–4, 2011.
- [58] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006. [Online]. Available: <https://science.sciencemag.org/content/313/5786/504>
- [59] F. Chollet *et al.*, “Keras,” <https://keras.io>, 2015.
- [60] M. H. Law and J. T. Kwok, “Bayesian support vector regression,” in *In Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics*. Key West, 2001, pp. 239–244.