International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (E-ISSN 2250-2459, Scopus Indexed, ISO 9001:2008 Certified Journal, Volume 12, Issue 08, August 2022)
Manuscript Received: 12 June 2022, Received in Revised form: 25 July 2022, Accepted: 12 August 2022 DOI: 10.46338/ijetae0822_15

Performance Evaluation of Recurrent Neural Networks Applied to Indoor Camera Localization

Muhammad S. Alam[1], AKM B. Hossain[2], Farhan B. Mohamed[3]
[1,2,3]*School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, 81310, Johor Baharu, Johor, Malaysia*

***Abstract* — Researchers in robotics and computer vision are experimenting with the image-based localization of indoor cameras. Implementation of indoor camera localization problems using a Convolutional neural network (CNN) or Recurrent neural network (RNN) is more challenging from a large image dataset because of the internal structure of CNN or RNN. We can choose a preferable CNN or RNN variant based on the problem type and size of the dataset. CNN is the most flexible method for implementing indoor localization problems. Despite CNN's suitability for hyper-parameter selection, it requires a lot of training images to achieve high accuracy. In addition, overfitting leads to a decrease in accuracy. Introduce RNN, which accurately keeps input images in internal memory to solve these problems. Long-short-term memory (LSTM), Bi-directional LSTM (BiLSTM), and Gated recurrent unit (GRU) are three variants of RNN. We may choose the most appropriate RNN variation based on the problem type and dataset. In this study, we can recommend which variant is effective for training more speedily and which variant produces more accurate results. Vanishing gradient issues also affect RNNs, making it difficult to learn more data. Overcome the gradient vanishing problem by utilizing LSTM. The BiLSTM is an advanced version of the LSTM and is capable of higher performance than the LSTM. A more advanced RNN variant is GRU which is computationally more efficient than an LSTM. In this study, we explore a variety of recurring units for localizing indoor cameras. Our focus is on more powerful recurrent units like LSTM, BiLSTM, and GRU. Using the Microsoft 7-Scenes and InteriorNet datasets, we evaluate the performance of LSTM, BiLSTM, and GRU. Our experiment has shown that the BiLSTM is more efficient in accuracy than the LSTM and GRU. We also observed that the GRU is faster than LSTM and BiLSTM.**

***Keywords*— Indoor camera localization, Gated recurrent unit, Long-short term memory, PoseNet, Recurrent neural network.**

I. INTRODUCTION

Camera localization refers to estimating the camera pose of an image from a random scene. An image, video, or sequence of images as an input.

The output depends on how the scene is represented and which method to estimate the camera's location. Many vision applications, such as the navigation of mobile robots and autonomous driving vehicles, can benefit from camera localization. Navigation of mobile robots, autonomous driving vehicles, and image-based localization of virtual reality camera localization are essential aspects that have recently attracted significant interest from academics and industry. The most adaptable and cost-effective way localize the camera in indoor environment use deep architecture.

Deep learning has a wide range of applications, with several achievements in the image processing field. Convolutional neural networks are supposed to mimic the activity of the visual cortex. On any visual identification application, CNNs perform exceptionally well. Convolutional layers and pooling layers are individual layers in the CNN architecture. These layers enable the network to encode the attributes of specific pictures. We may use CNN to learn good visual features for localization that seem resistant to motion blur and changes in light. CNN is also suitable for hyperparameter selection or tuning [1]. Convolutional Neural Networks (CNN), a deep learning-based camera localization, performs convolution operations on RGB images to estimate camera poses. The first attempt to use CNNs for direct camera pose regression was PoseNet [2] as shown in Figure 1. PoseNet computes with fully connected layers and uses GoogLeNet as a framework for feature extraction [3]. In Bayesian PoseNet, researchers introduced PoseNet to account for uncertainty in pose estimation [4]. Some other research has focused on frameworks to improve camera localization. The researchers combined global poses with relative poses by predicting comparative poses from the image sequence [5].

Use a strategy to focus attention on geometrically significant features [6]. They achieved pose regression through multitasking learning that combines information from associated activities.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (E-ISSN 2250-2459, Scopus Indexed, ISO 9001:2008 Certified Journal, Volume 12, Issue 08, August 2022)

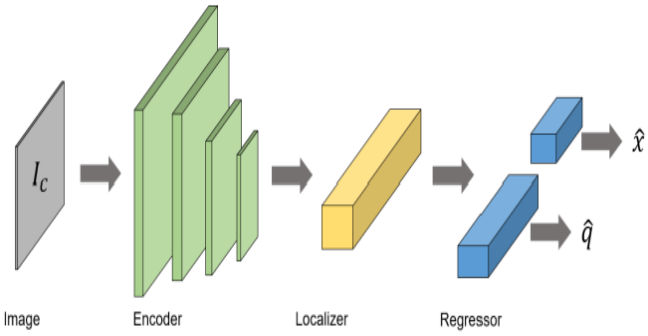

Fig. 1 PoseNet Architecture

A deep learning-based system usually requires large annotated images to achieve high accuracy. To overcome this challenge, use a 3D model to generate synthetic images. To develop a map of benchmarks that approximates the difference between synthetic and original images in pattern representations [7]. To create a geolocation image dataset, compare the synthesized images with synthetic ones in a dataset [8]. With the difficulty of CNN requiring a significant quantity of labeled data for training, several researchers have turned to pre-trained, fine-tuned deep architecture. They use GoogLeNet to extract features, which are subsequently fine-tuned for camera localization. In addition, CNN models create high-dimensional feature vector output, which is prone to overfitting in training data and degrades model accuracy. There are also certain limitations in feature matching for relative camera pose estimation.

RNNs are a type of neural network that is both powerful and reliable, and one of the most intriguing algorithms now in use because with internal memory. RNNs can keep critical data about the input they receive thanks to their internal memory, allowing them to forecast what will happen next with extraordinary accuracy. They are the best option for sequential data, such as time series. Compared to other algorithms, recurrent neural networks can learn more about a series. Because the user can only proceed along a continuous trajectory, the user's present location is associated with its past locations in the situation of indoor localization. As a result, RNN uses the trajectory parameters to improve localization accuracy. In [9], the researchers use CNN-RNN architectures to constrain the network by the temporal smoothness of camera motion.

In [10], the authors create a viewgraph using graph neural networks to share non-consecutive frame information. Short-term memory is a problem for recurrent neural networks.

If the sequence is lengthy enough, it will have problems passing information from earlier point steps to later ones. When attempting to predict anything, the vanishing gradient problem affects recurrent neural networks during backpropagation. Gradients are values used to update the weights of a neural network. When a gradient propagates backward in time, it is called the vanishing gradient issue. When a gradient value falls below a certain threshold, it becomes useless for learning. However, RNNs have an issue with vanishing gradients, making it challenging to learn massive data sequences. Solutions use the LSTM idea because LSTM makes the slope steep enough to keep the training short and high accuracy.

The LSTM [11] is a recurrent neural network that can learn long-term patterns from its data. A conventional LSTM has input, output, and reset gates and a memory cell that allows data to flow in and out of memory cells and is controlled by the input and forget gates. They remove the SoftMax layers and replace them with a 2048-dimensional dense layer in GoogleNet for image feature extraction [3]. The initial weights are GoogleNet pre-trained on the locations image dataset because of their appropriateness for scene classification. The LSTM with CNN architecture achieves structured dimensionality reduction and improves localization accuracy [12]. In [9], the researchers introduced an RCNN model for camera position regression from image or video inputs, which can smooth pose estimation. Creating synthetic images improves the camera pose regression by using a 3D model created from natural images [13]. In a coarse visual localization using images, researchers compare natural and artificial images based on features derived from a CNN using a similarity metric [8]. Researchers use a similarity metric to compare natural and synthetic images based on information extracted from CNN. The researchers classify the actual image based on its similarity to a synthetic image with a known camera pose. The BIM-PoseNet [14] model trains artificial images extracted from a 3D model to predict authentic images' camera position and orientation. They considered the result with an accuracy of fewer than 2m by storing the natural and synthetic images. Then, synthetic images were used to simulate the uncertainty of pose estimation using Bayesian BIM-PoseNet [14].

In [15], the researchers introduced the bidirectional LSTM (BiLSTM). In the frame-wise phoneme regression task, bidirectional LSTMs outperformed unidirectional LSTMs and standard RNNs. According to the findings, a bidirectional LSTM architecture is an excellent solution for camera localization.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (E-ISSN 2250-2459, Scopus Indexed, ISO 9001:2008 Certified Journal, Volume 12, Issue 08, August 2022)

A camera localization based on a hybrid bidirectional LSTM system outperformed unidirectional LSTM. We have tested bidirectional LSTMs on the camera localization issue and beat the state-of-the-art camera localization on online and offline data.

The network performance depends on the distance between the LSTM units. When the network is over-fitted with a more extended number of LSTM units, the network performs well enough for validation data but rather poorly for unknown test data. A gated recurrent unit (GRU) is a more sophisticated RNN version that is more computationally efficient than an LSTM [16]. Since it has fewer training parameters, GRU uses less memory and executes faster than LSTM. It has a more straightforward structure than the LSTM and is better at handling small datasets. It also uses less memory and produces more rapid results.

In this paper, we measured the performance evaluation of LSTM, BiLSTM, and GRU for indoor camera localization using the reduced InteriorNet [17] and 7-Scenes [18] dataset. End-to-end learning is used to train different parameterized models for labeling temporal data.

II. Related Works

In [2], the authors developed the PoseNet architecture, which uses a single RGB image to predict the camera pose. This algorithm comprises a CNN trained end-to-end in camera position and orientation. The main contribution is a deep learning regression model for camera localization. It focused PoseNet solely on GoogLeNet. It is also presented a new camera rotation parameter suited for deep learning-based camera pose regression. The results significantly improve over earlier attempts in the indoor 7-Scenes dataset [18] and the outdoor Oxford Robot-Car [19] dataset. In [1], the researchers proposed a GPoseNet model that regresses the camera pose from a single RGB image. They did this model in Bayesian PoseNet, a probabilistic version of the camera relocalization methods. It extracts features from an RGB image and uses linear regression to estimate the 6DoF pose of a moving camera. One difficulty considered in computer vision is how to estimate people's locations in an interior space as precisely as workable. Changing the weight-of-loss function in a 23-layer CNN architecture [20]. Resize the pictures before the training step to keep the entire image as the CNN input value [21] suggested using a CNN model to generate important features and estimate camera settings for 3D reconstruction.

The LSA minimized its processed features using four convolution layers and max-pooling to simplify the network. GeoPoseNet [22] and GPoseNet [1] explore different modules to improve localization instead of using loss functions with fixed parameters or learnable loss functions. GeoPoseNet [22] proposed the reprojection loss, which characterizes the error in reprojecting the scene geometry. The VidLoc architecture uses CNN-RNN networks to constrain the network by temporal smoothness of camera motion [9]. This model uses CNN to analyze video image frames and a bidirectional LSTM to incorporate temporal information. The LSTM is a technique that allows ordinary RNNs to learn long-term temporal dependencies. In [9], the researchers proposed a recurrent model that employs several frames for pose prediction to decrease pose estimate error. A CNN repeat neural network (RNN) model for effective global localization from a monocular image sequence is presented. Using a texture-less 3D model of the indoor space in BIM-PoseNet [14] and Bayesian BIM-PoseNet [14] avoids 3D image-based reconstruction. In [12], the researchers introduce a neural network-based PoseNet and LSTM for single image regression. The researchers performed their model on the Cambridge Landmarks datasets for content-based image retrieval, where a Siamese network was trained on pairs of images taken from a nearby location. The performance is not good enough compared to the most modern method. The LSTM approach is used [12]. Based on this research, a deep architecture that uses syntactic images for training and recurrent neural network-based PoseNet directly estimates camera localization [23]. A BIM-PoseNet [14] uses synthetic image sequences to estimate the camera pose to improve localization performance. This process reduced localization performance caused by accounting for range changes between synthetic and original images. Domain matching approach to solving the localization performance degradation problem [24]. The proposed network includes a deep Bayesian CNN and an LSTM component to capture the spatial-temporal interactions between subsequent frames. The LSTM [11] is a recurrent neural network that can learn long-term information from its data. In [25], the researchers proposed a deep learning strategy for UWB localization to address these UWB system problems for indoor localization. Long-term and short-term memory (LSTM) networks predict the user's position in the proposed deep learning model. Based on the TOA-distance model of the UWB system, they suggest an LSTM model estimate the current user location.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (E-ISSN 2250-2459, Scopus Indexed, ISO 9001:2008 Certified Journal, Volume 12, Issue 08, August 2022)

In [26], the researchers proposed a novel deep ConvNet training architecture for image-based camera localization in urban streets. The VNLSTM-PoseNet network employs an LSTM structure to decrease structural dimensionality and chooses the most relevant features for real-time camera pose regression on the fully connected layer. Although transfer has developed learning approaches to minimize the amount of training data necessary for RNNs, reducing deployment costs, this has yet to be investigated in LSTM-based indoor localization. In [27], the authors offer a fingerprint localization architecture based on LSTMs that uses transfer learning methods to deliver excellent accuracy and low deployment costs. It lowers the cost of indoor localization and makes it easier to use, making it more widely available.

III. RNN Architecture

This section presents the three models used in the experiments, followed by a description of the chosen architecture.

A. Recurrent Neural Network

RNN is a neural network-based memory space and loops for dealing with sequence data. The RNN architecture is at the heart of LSTMs. The basic structure of the RNN is shown in Figure 2.

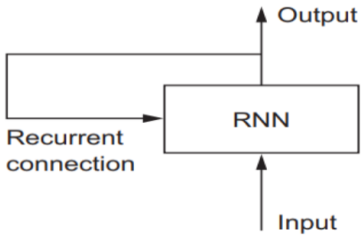

Fig. 2 Fully connected RNN architecture

Each time iteration t the hidden state h_t is:

$$h_t = \sigma_h(W_{xh}h_t + W_{hh}h_{t-1} + b_h) \quad (1)$$

Where σ_h is the activation function, W_{xh} is the weight matrix between the input and hidden layer, W_{hh} is the weight matrix between the two hidden layers, and b_h is a bias vector of hidden layers. The network output y_t is:

$$y_t = \sigma_y(W_{hy}h_t + b_y) \quad (2)$$

Where σ_y is the output layer activation function, W_{hy} is the weight matrix between the hidden layer and output, while b_y is the output layer bias vector.

B. Long Short-Term Memory

The Long Short-Term Memory (LSTM) is a particular type of RNN that prevents gradients from disappearing. LSTMs using a technique known as gates may learn long-term dependencies. These gates can tell us whether data in a sequence should be kept or discarded. The three gates of LSTM are input, forget, and output. Several advanced, recurrent architectures, including LSTM and GRU [28], have addressed the RNN. LSTMs were good at solving sequence-based problems with long-term constraints, while GRU, a much simpler LSTM architecture, was recently developed and implemented in machine learning [29]. An LSTM's control flow is like a recurrent neural network. As it travels, it receives input and relays information. The mechanisms within the LSTM cells differ, as shown in Figure 3. The first gate of the LSTM is the forget gate. The procedure will determine if the data is kept or discarded. The sigmoid function transports data from the previously hidden layer and current input data. Next, we will look at the output gate. The output gate decides the hidden state's next concealed state. It is important to remember that the hidden state includes information from previous inputs. The concealed state is also used to make predictions.

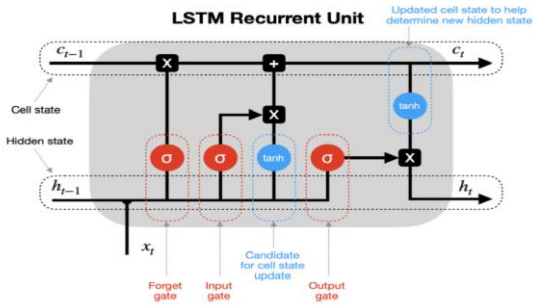

Fig. 3 LSTM Architecture

C. Bidirectional LSTM

Bidirectional LSTMs are LSTM models that use existing data and the future of a single time step as input. At each moment, we can preserve knowledge from the past and the future in BiLSTM. Bidirectional RNN [30] is a BiLSTM idea that analyzes sequence inputs in front and rear directions using two separate hidden layers, as shown in Figure 4.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (E-ISSN 2250-2459, Scopus Indexed, ISO 9001:2008 Certified Journal, Volume 12, Issue 08, August 2022)

BiLSTM connect the two hidden layers to a virtually identical output layer. Bidirectional long-term memory (BiLSTM) seems to be the approach to storing forward and backward sequence information in each neural network. A bidirectional LSTM is a sequential processing system of two LSTMs, one processing the input and the other processing it backward. Bidirectional LSTMs (BiLSTM) are LSTM systems that incorporate input data from a single time step's past and future. In BiLSTM, we may store information from the past and the future.

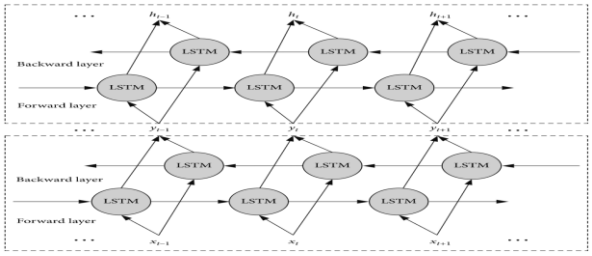

Fig. 4 BiLSTM architecture

D. Gated Recurrent Unit

The GRU is a relatively new recurrent neural network that follows an LSTM. GRUs have rejected cell states in favor of data transfer via the hidden state. There are two gates: one is for reset, and the other is for an update. The update gate works the same way as an LSTM forgets, and input gates operate, as shown in Figure 5. It chooses which data should be deleted and which can be re-entered. The reset gate is another gate used to determine how much past information should be lost. GRUs are relatively faster than LSTMs because they contain fewer tensor operations.

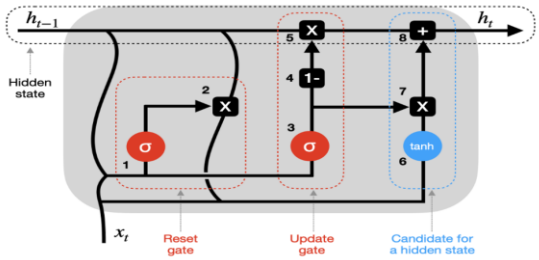

Fig. 5 GRU architecture

E. Experimental Setup

As the foundation for the RNN architecture, we are employing an RCNN framework that accepts an image sequence as input and outputs positional and orientational errors. Our experiments used three models: Long-shot Term Memory (LSTM), another with bidirectional LSTM, and the third with a GRU layer.

Bidirectional LSTM or GRU has the same formula as LSTM. We performed backpropagation training using the ADAM optimization method [31], which is relatively tolerant of learning rate and other training parameters and requires less fine-tuning. It trained the network uniformly on two different datasets, InteriorNet [17], and 7-Scenes [18], by resizing the images to 256 pixels. After that, we adjusted the input images to have intensity values between -1 and 1. The ResNet34 component of the network is pre-trained on the ImageNet dataset while we randomly initialize the other elements. We resize 256x256 pixels images for the network using a random and central cropping method throughout the training and testing process. The augmentation phase is necessary to increase the model's generalization capabilities under various meteorological scenarios. We implement our plans in Python 3.10 with PyTorch, using the Adam solver with a learning rate of 5 $x10^{-5}$. We trained the network with the hyperparameters on a CPU: epoch 20, batch size 64, train dropout 0.5, test dropout 0.0, and weight initializations of β is 0.0 and γ is 3.0. This research is validated by comparing the performance of many RNN networks, such as LSTM, BiLSTM, and GRU. For the 7-Scenes [18] dataset and reduced InteriorNet [17] dataset, we use the LSTM, BiLSTM, and GRU networks.

IV. Experimental Results

The dataset is provided first, followed by a list of the measurement metrics used in the tests, and finally, the findings and discussion.

A. Datasets

This study will evaluate the model using the InteriorNet [17] dataset. Imperial College London released this data set in 2018. The data set comprises 10K scenes, 1.7M rooms, and 5M frames. RGB, depth, and semantic instances are available in this dataset. The image resolution of the dataset is 640x480. It included the camera and pose information in the file cam0.info. It had all the intrinsic and extrinsic parameters in the cam0.ccam file. This research also uses the Microsoft 7-Scenes [18] dataset developed by Microsoft Research. The 7-Scenes dataset comprises seven separate interior worlds and is a commonly used RGB-D dataset. The RGB-D images were taken with a 640x480 resolution handheld Kinect camera and linked to the ground truth camera positions captured using the Kinect fusion method. A detailed 3D model also accompanies each scene. Each scene comprises multiple sequences of tracked RGB-D camera frames split into training and testing data.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (E-ISSN 2250-2459, Scopus Indexed, ISO 9001:2008 Certified Journal, Volume 12, Issue 08, August 2022)

B. Evaluation Metrics

Some error calculation techniques are used to evaluate the performance of the deep learning regression model, such as Mean Absolute Error (MAE), Mean Square Error (MSE), and Root Mean Square Error (RMSE) (RMSE).

Mean Absolute Error (MAE):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i| \quad (3)$$

Where x_i is the predicted, and y_i is the mean value.

Mean Square Error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2 \quad (4)$$

Where x_i is the predicted, and y_i is the mean value.

Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2} \quad (5)$$

Where x_i is the predicted, and y_i is the mean value.

C. Hyperparameters Setup

The following parameters and hyperparameters have been used for the training and testing of the model. Batch size is 64, β is -3.0, crop size is 256, learning rate is $5e^{-0.5}$, Testing dropout is 0%, training dropout is 5%, validation frequency is 5, activation function is ReLU, and weight decay is 0.0005.

D. Experimental Result for 7-Scenes Dataset

We used two separate datasets to test the three models BiLSTM, LSTM, and GRU. The first one is the 7-Scenes dataset with seven different scenes.

Accuracy: We exhibit the experimental results of the 7-Scenes dataset in Table I. The median positional and orientational errors of three architectures are BiLSTM (0.26m, 9.65^0), LSTM (0.31m, 9.85^0), and GRU (0.35m, 10.23^0), with BiLSTM scoring the highest in terms of precision.

TABLE I
MEDIAN POSE ERROR FOR 7-SCENES DATASET

Network	Median Pose Error
BiLSTM	0.26m, 9.65^0
LSTM	0.31m, 9.85^0
GRU	0.35m, 10.23^0

Speed: In terms of execution time, GRU outperforms both LSTM and BiLSTM. GRU is not comparable to LSTM and BiLSTM because its accuracy uses are significantly superior. For the 7-Scenes dataset, GRU ran 8.43 ms, whereas LSTM and BiLSTM ran 9.2 ms and 9.01 ms. Table II shows the running time of BiLSTM, LSTM, and GRU.

TABLE II
RUNNING TIME FOR 7-SCENES DATASET

Network	Running Time
BiLSTM	9.01ms
LSTM	9.2ms
GRU	8.43ms

Evaluation Metric: Evaluate the model through LSTM, BiLSTM, and GRU. The evaluation results of the three distinct networks, as shown in Table III, for the 7-Scenes dataset.

TABLE III
EVOLUTION METRIC FOR 7-SCENES DATASET

Network	MAE	MSE	RMSE
BiLSTM	0.1131	0.0365	0.1911
LSTM	0.1182	0.0353	0.1880
GRU	0.1155	0.0364	0.1908

Losses: Figure 6, 7, & 8 demonstrate the training loss versus the validation loss of BiLSTM, LSTM, and GRU. We applied the test set after each epoch, and the model learned during training was evaluated on the test data right after each epoch.

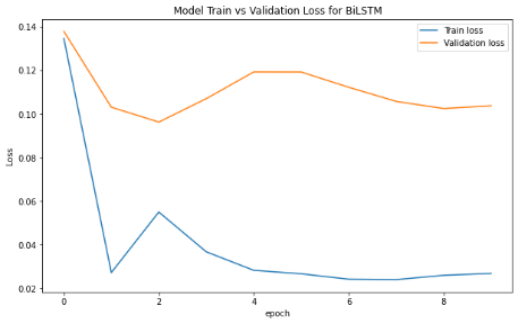

Fig. 6 Train loss vs. validation loss of BiLSTM for the 7-Scenes dataset

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (E-ISSN 2250-2459, Scopus Indexed, ISO 9001:2008 Certified Journal, Volume 12, Issue 08, August 2022)

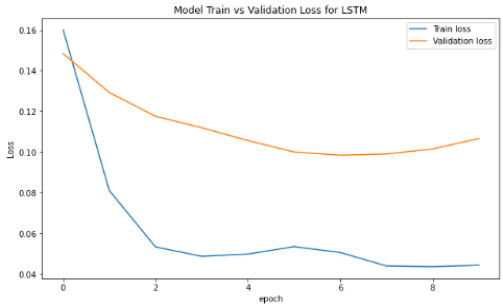

Fig. 7 Train loss vs. validation loss of LSTM for the 7-Scenes dataset

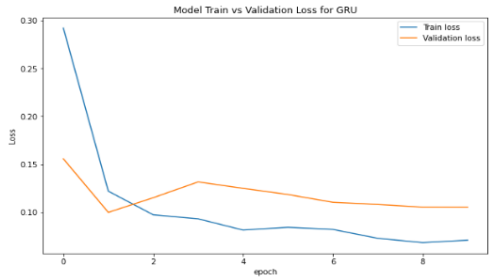

Fig. 8 Train loss vs. validation loss of GRU for the 7-Scenes dataset

E. Experimental Result for InteriorNet Dataset

We used two separate datasets to test the three models BiLSTM, LSTM, and GRU. The second one is the InteriorNet dataset, which contains ten thousand scenes.

Accuracy: We exhibits the experimental results of the InteriorNet dataset experiments in Table IV. The median positional and orientational errors of three architectures are BiLSTM (0.26m, 9.65^0), LSTM (0.35m, 9.97^0), and GRU (0.43m, 10.35^0), with BiLSTM scoring the highest in terms of precision.

TABLE IV
MEDIAN POSE ERROR FOR INTERIORNET DATASET

Network	Median Pose Error
BiLSTM	0.31m, 9.72^0
LSTM	0.35m, 9.97^0
GRU	0.43m, 10.35^0

Speed: In terms of execution time, GRU outperforms both LSTM and BiLSTM, as shown in Table V. GRU is not comparable to LSTM and BiLSTM because its accuracy uses are significantly superior. For the InteriorNet dataset, GRU ran 8.9 ms, whereas LSTM and BiLSTM ran 9.8 ms and 9.6 ms.

TABLE V
RUNNING TIME FOR INTERIORNET DATASET

Network	Running Time
BiLSTM	9.6ms
LSTM	9.8ms
GRU	8.9ms

Evaluation Metric: Evaluate the model through LSTM, BiLSTM, and GRU. The evaluation result of the three distinct networks as shown in Table VI. Compared to the LSTM and BiLSTM, GRU generates more minor errors.

TABLE VI
EVALUATION METRIC FOR INTERIORNET DATASET

Network	MAE	MSE	RMSE
BiLSTM	0.1546	0.0730	0.2701
LSTM	0.1564	0.0718	0.2680
GRU	0.1617	0.0699	0.2644

Losses: Figure 9,10, &11 demonstrate the training loss versus the validation loss of BiLSTM, LSTM, and GRU. We applied the test set after each epoch of the model learned during training was evaluated on the test data right after each epoch. We can see that LSTM provides the best value.

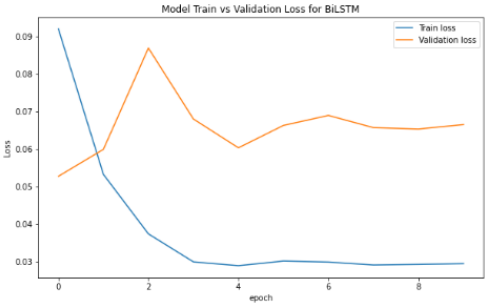

Fig. 9 Train loss vs. validation loss of BiLSTM for the InteriorNet dataset

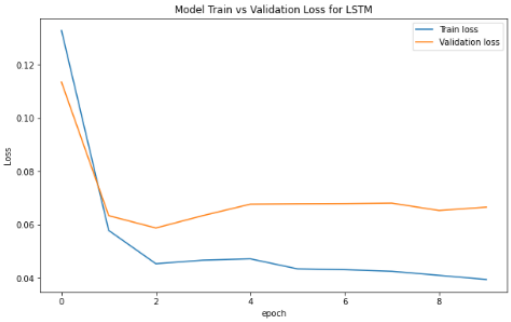

Fig. 10 Train loss vs. validation loss of LSTM for the InteriorNet dataset

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (E-ISSN 2250-2459, Scopus Indexed, ISO 9001:2008 Certified Journal, Volume 12, Issue 08, August 2022)

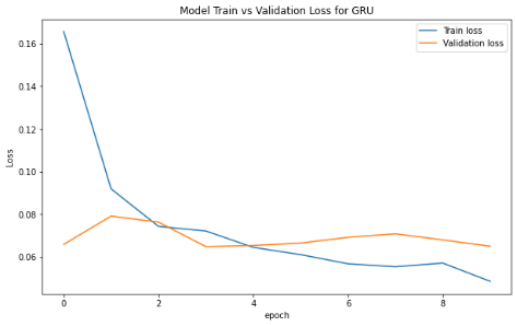

Fig. 11 Train loss vs. validation loss of GRU for the InteriorNet dataset

V. Conclusion And Future Works

The output of CNN models is a high-dimensional feature vector, which is prone to overfitting in training data and reduces model accuracy. Then introduce an RNN to address the difficulties mentioned above by storing essential input data in internal memory. RNNs also have vanishing gradient difficulties, making learning big data sequences difficult. To solve the problem of vanishing gradients, employ LSTM, which makes the gradient steep enough to maintain the training rapid and accurate. A bidirectional LSTM outperforms a single-direction LSTM. A GRU is a more sophisticated RNN variation that is more computationally efficient than a BiLSTM or LSTM. GRU uses less memory and operates quicker than LSTM and BiLSTM since it has fewer training parameters. This paper assessed the performance of LSTM, BiLSTM, and GRU using a reduced InteriorNet [17] and 7-Scenes [18] dataset. As assessment metrics, median pose errors, loss, and running time were calculated. The findings reveal that the BiLSTM and LSTM are similar (BiLSTM scores are slightly higher than LSTM); however, the BiLSTM and LSTM have a longer running duration than GRU. As a result, we recommend GRU with the smaller InteriorNet [17] and 7-Scenes [18] data sets since they reasonably generated good accuracy values.

In future, parameter optimization will examine the impact of various parameter values. We will examine the learning rate, dropout rate, and an increased number of neurons in the hidden layers.

REFERENCES

[1] Cai, M., Shen, C., & Reid, I. (2019). A hybrid probabilistic model for camera relocalization.

[2] Kendall, A., Grimes, M., & Cipolla, R. (2015). Posenet: A convolutional network for real-time 6-dof camera relocalization. In Proceedings of the IEEE international conference on computer vision (pp. 2938-2946).

[3] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

[4] [4] Kendall, A., & Cipolla, R. (2016, May). Modelling uncertainty in deep learning for camera relocalization. In 2016 IEEE international conference on Robotics and Automation (ICRA) (pp. 4762-4769). IEEE

[5] Laskar, Z., Melekhov, I., Kalia, S., & Kannala, J. (2017). Camera relocalization by computing pairwise relative poses using convolutional neural network. In Proceedings of the IEEE International Conference on Computer Vision Workshops (pp. 929-938).

[6] Wang, B., Chen, C., Lu, C. X., Zhao, P., Trigoni, N., & Markham, A. (2020, April). Atloc: Attention guided camera localization. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 06, pp. 10393-10401).

[7] Rad, M., Oberweger, M., & Lepetit, V. (2018). Feature mapping for learning fast and accurate 3d pose inference from synthetic images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4663-4672).

[8] Ha, I., Kim, H., Park, S., & Kim, H. (2018). Image retrieval using BIM and features from pretrained VGG network for indoor localization. Building and Environment, 140, 23-31.

[9] Clark, R., Wang, S., Markham, A., Trigoni, N., & Wen, H. (2017). Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6856-6864).

[10] Xue, F., Wu, X., Cai, S., & Wang, J. (2020, June). Learning multi-view camera relocalization with graph neural networks. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 11372-11381). IEEE.

[11] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[12] Walch, F., Hazirbas, C., Leal-Taixe, L., Sattler, T., Hilsenbeck, S., & Cremers, D. (2017). Image-based localization using lstms for structured feature correlation. In Proceedings of the IEEE International Conference on Computer Vision (pp. 627-637).

[13] Wu, Y., Tang, F., & Li, H. (2018). Image-based camera localization: an overview. Visual Computing for Industry, Biomedicine, and Art, 1(1), 1-13.

[14] Acharya, D., Khoshelham, K., & Winter, S. (2019). BIM-PoseNet: Indoor camera localisation using a 3D indoor model and deep learning from synthetic images. ISPRS Journal of Photogrammetry and Remote Sensing, 150, 245-258.

[15] Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural networks, 18(5-6), 602-610.

[16] Yang, L., Bai, Z., Tang, C., Li, H., Furukawa, Y., & Tan, P. (2019). SANet: Scene agnostic network for camera localization. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 42-51).

[17] Wenbin, Li, Saeedi, S., McCormac, J., Clark, R., Tzoumanikas, D., Ye, Q., ... & Leutenegger, S. (2018). Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. arXiv preprint arXiv:1809.00716.

[18] Glocker, B., Izadi, S., Shotton, J., & Criminisi, A. (2013, October). Real-time RGB-D camera relocalization. In 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR) (pp. 173-179). IEEE.

[19] Maddern, W., Pascoe, G., Linegar, C., & Newman, P. (2017). 1 year, 1000 km: The Oxford RobotCar dataset. The International Journal of Robotics Research, 36(1), 3-15.

[20] Lu, E. H. C., & Ciou, J. M. (2020). Integration of Convolutional Neural Network and Error Correction for Indoor Positioning. ISPRS International Journal of Geo-Information, 9(2), 74.

[21] Wattanacheep, B., & Chitsobhuk, O. (2020, August). Camera Pose Estimation using CNN. In 2020 the 3rd International Conference on Control and Computer Vision (pp. 84-88).

[22] Kendall, A., & Cipolla, R. (2017). Geometric loss functions for camera pose regression with deep learning. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5974-5983).

[23] Acharya, D., Singha Roy, S., Khoshelham, K., & Winter, S. (2020). A Recurrent Deep Network for Estimating the Pose of Real Indoor Images from Synthetic Image Sequences. Sensors, 20(19), 5492.

[24] Li, Q., Cao, R., Zhu, J., Hou, X., Liu, J., Jia, S., ... & Qiu, G. (2022). Improving synthetic 3D model-aided indoor image localization via domain adaptation. ISPRS Journal of Photogrammetry and Remote Sensing, 183, 66-78.

[25] Poulose, A., & Han, D. S. (2020). UWB indoor localization using deep learning LSTM networks. Applied Sciences, 10(18), 6290.

[26] Li, M., Qin, J., Li, D., Chen, R., Liao, X., & Guo, B. (2021). VNLSTM-PoseNet: A novel deep ConvNet for real-time 6-DOF camera relocalization in urban streets. Geo-spatial Information Science, 24(3), 422-437.

[27] Brattinga, M. (2022). LSTM-based indoor localization with Transfer Learning (Bachelor's thesis, University of Twente).

[28] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

[29] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

[30] Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. IEEE transactions on Signal Processing, 45(11), 2673-2681.

[31] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.