Integrating Recurrent Neural Networks and Loss Function Optimization for Efficient Indoor Camera Positioning

Shamsul Alam, Farhan Mohamed, Bellal Hossain

Abstract

Camera position is essential for many applications, such as monitoring, tracking, and recognizing individuals. This study proposed an integrated design that combines recurrent neural networks (RNNs) and a loss function modification approach to improve the accuracy of indoor camera location. RNNs enable the system to generate accurate estimations based on previous information by extracting temporal dependencies and patterns from the camera information. We optimized the loss function to enhance the indoor camera position's overall performance and convergence speed. This combination technique allows the proposed method to considerably increase the accuracy of camera location prediction in indoor conditions. We validated the effectiveness of the proposed approach and demonstrated its improved accuracy and robustness through extensive evaluation of many indoor datasets. The results show that our combined approach outperforms existing methods and has enormous potential for real-world applications in indoor activity recognition, navigation optimization systems, and safety surveillance.

Keywords: Camera positioning, Indoor navigation, Pose estimation, Pose loss, Recurrent neural network.

MSC 2020: 68T01, 68T07, 68T40.

ACM CCS 2020: I.2.9, I.2.10.

1 Introduction

In recent years, there has been a significant increase in the interest in positioning surveillance systems inside buildings. These systems are

doi:10.56415/csjm.v33.04

necessary to keep security, monitor things, and distribute resources wisely. One of the most critical aspects of these systems that determines their effectiveness is the positioning of indoor cameras. The accurate position of cameras minimizes blind spots, permits comprehensive coverage, and improves monitoring effectiveness in general. However, occlusions, poor visibility, and complex spatial arrangements are some aspects that need to be considered when determining the most appropriate camera position in indoor environments. Combining recurrent neural networks (RNNs) with loss function optimization has been demonstrated to be a helpful approach to these difficulties [1]. RNNs are a particular kind of artificial neural network that shows noteworthy potential in modelling complex temporal movement, making them especially beneficial for understanding video data in indoor monitoring [2]. RNNs perform effectively with sequential data. Temporal dependencies and patterns may be collected using the internal memory and feedback mechanisms of RNNs, making it easier to create reliable camera positioning systems.

Moreover, one of the most significant processes in training deep learning models is optimizing the loss function. The learning process can focus on particular goals by fine-tuning the loss function and improving the model's capacity to estimate the best camera positions in indoor environments accurately. This approach provides a new and comprehensive framework to address the issues associated with indoor camera positioning, achieving enhanced security and surveillance capabilities through the combination of RNNs and loss function optimization.

The current approach explores modifying indoor camera position techniques by integrating RNNs with loss function optimization by investigating the underlying theories, real-world applications, and experimental evaluations. This study aims to understand the effectiveness of this combination approach and its possible consequences for indoor surveillance systems. After an in-depth evaluation of the available research literature, this article presents a brief overview of the most recent approaches, their drawbacks, and the potential for future developments in the field of indoor camera positioning.

The research presents a contribution to enhancing indoor camera

position techniques through the application of innovative deep-learning techniques. An overview of the significant contributions of this study is provided below:

1. To propose the RNN model, which will enhance the indoor camera positioning system. More precise and contextually aware camera positioning has been made possible with this approach.

2. To present novel approaches for modifying the loss functions related to indoor positioning. The error between the projected camera positions and the actual ground truth is effectively minimized using this study.

2 Retated Works

The advancement of deep learning algorithms has revolutionized the field of indoor camera positioning, resulting in surveillance systems that are now more precise and effective. Recurrent neural networks (RNNs) and loss function optimization are two ideas that have garnered significant interest in improving camera monitoring and positioning accuracy. This section shows various methods and methodologies that the researchers employ while examining the field's significant achievements and current developments. Loss function optimization has been a growing area of study for researchers in recent years to improve the performance of indoor camera positioning systems. More specifically, the robustness and generalization of the positioning of camera systems were enhanced using a unique loss function [3] design that effectively decreased the impact of noise and outliers in the training data. The importance of incorporating adaptive loss functions that dynamically modify the relative weights of different data points was emphasized in [4]. This method reduced the effect of incorrect or misleading information in the training process.

Recently, there has been an increasing number of techniques for combining RNNs with ideal loss functions. To improve camera positioning techniques and allow for adaptive optimization of loss functions based on changes in the environment and object movements, re-

search [5] focused on combining RNNs with reinforcement learning approaches. Improved accuracy and reduced computing complexity can be achieved by integrating Long Short-Term Memory (LSTM), a type of recurrent neural network with a loss function optimized for indoor camera position, as demonstrated in a study [6].

This paper's primary focus is structure feature-based pose estimation techniques and regression-based estimations. PoseNet [7] is the first study to regress camera pose using trained convolutional neural networks (CNNs) on single RGB images. It overcomes the requirement for extra mechanisms or cross-frames and keyframes for pose prediction. A Bayesian CNN using Bernoulli distributions is suggested to enhance localization performance and comprehend model uncertainty [8]. A Surround View System PoseNet (SVS-PoseNet) [9], which also indicates utilizing deep learning for camera localization, is similar to a deep neural network based on a classification network. It uses the same environments for hyperparameters throughout training datasets rather than fine-tuning parameters. It improves performance for indoor datasets. Adaptable weight pose minimization was proposed using Geometric PoseNet (GeoPoseNet) [10] to enhance localization and performance.

The network concentrates on the central area of the input images by AtLoc [11], which adds an attention module before calculating the regression coordinates measurements. Spatial Pyramid max-pooling units, the foundation of the DNN architecture known as SPP-Net [12], share the same loss function as GeoPoseNet [10]. Training data production is unnecessary for this alternative approach to enhancing localization performance. Together with the PoseNet hyperparameters [13], the variational lower bound of two log marginal likelihoods is used to compute the losses for the CNN and SVI GPs in the GoogLeNet-based PoseNet (GPoseNet) [13] loss function. The localization performance was significantly improved by MapNet [14] by using an extra loss term from image pairings as a geometric constraint. In advance of evaluating the coefficients of regression, an attention module is included, allowing the network to concentrate on the central area of the input images.

Furthermore, VlocNet [15] employs two sub-networks to learn visual odometry as an additional task while simultaneously regressing

the global position. The pose error is decreased by adjusting the loss of geometric consistency. VlocNet++ [15] combines the loss of global pose regression, visual odometry loss [16], and cross-entropy loss for semantic segmentation loss. Furthermore, AtLoc+ [11] outperforms AtLoc with a single image input by using temporal constraints to simultaneously learn the absolute pose loss and the relative pose loss. RelocNet [17] builds on NNnet [18] by learning global image features using a camera frustum and a geometric relative pose loss, improving the final result. The separate process CamNet [19] proposed consists of three independent processes. Applications include coarse-retrieval, fine-retrieval, and relative pose regression. A three-branch Siamese architecture is the foundation for each step [20]. Given two images as input, Relative NN [21] provided an end-to-end method to regress the relative pose between two cameras. Regression using the fixed Euclidean loss is performed using a Siamese Hybrid-CNN with a pretrained AlexNet network comprising two branches [22].

In summary, the integration of recurrent neural networks with loss function optimization has significant potential for enhancing the accuracy and efficiency of indoor camera positioning systems. As researchers continue to explore new architectures and techniques, developing flexible solutions for precise camera monitoring in indoor environments will significantly contribute to the progress of surveillance technology and ensure more effective security measures.

3 Method and Materials

3.1 Introduction

Apart from merging RNNs, specific attention is given to optimizing the loss functions associated with recurrent neural network training. The ability of the model to accurately predict camera positions depends on loss function optimization, which effectively penalizes deviations from the ground truth values. This dual emphasis on architectural sophistication and loss function correlation is the core of our proposed approach. Figure 1 shows the overview of camera pose estimation through the recurrent neural network and pose loss optimization.

3.2 RNN Architecture

Recurrent neural networks (RNNs) are utilized for indoor navigation. We additionally examine numerous RNN models available for obtaining, fine-tuning, and training indoor data and error correction concerning different camera characteristics [23], [24]. The structure and loss function of the RNN model are altered before the training phase, and various image size-matching pre-training models are developed. Moreover, a position error correction method is developed in this study to address positional differences among different camera platforms.

PoseNet [7] is a convolutional neural network for real-time camera pose estimation that estimates camera pose from a single RGB image. Another type of deep learning-based camera pose estimation technique is a classification system based only on image position estimates. They represented the camera pose loss function as follows:

$$loss(I) = \left\| c_{est} - c_{gt} \right\|_2 + \beta \left\| r_{est} - \frac{r_{gt}}{\left\| r_{gt} \right\|} \right\|_2. \tag{1}$$

The estimated camera pose is $[c_{est}, r_{est}]$ and the ground truth camera pose is $[r_{gt}, r_{gt}]$, and the relative weight of orientation and position errors are determined by the hyperparameter β, which is contingent on the training dataset. In this part, if the loss function determines the orientation loss, it will affect the position prediction's accuracy. The research focuses on positional accuracy. Hence, the orientation is not considered for prediction. Since the direction of the estimation has been determined, the loss function for Equation (2) is rewritten.

$$loss(I) = \left\| c_{est} - c_{gt} \right\|_2. \tag{2}$$

The predicted camera position is c_{est}, and the predicted camera orientation is r_{est}. The study used a stochastic gradient descent technique to return the camera to its starting position after generating the Euclidean loss through training. In equation (2), the loss function is displayed, with and denoting the expected and exact camera positions on the ground, respectively. A pre-trained model is introduced into additional training models upon training several image recognition datasets to consider attributes beforehand.

3.3 Position Prediction Process

The camera positional error is the Euclidean distance between the camera's position estimated value and the camera's position ground truth values ($c_{est} - c_{gt}$), where c_{gt} is the ground truth position value, and c_{est} is the estimated camera position value. The localization accuracy measurement technique calculates the pose error (Xcm) of the suggested camera localization system using the Euclidean distance. The three groups' respective stated thresholds for best, average, and worst pose errors are 0.25 meter, 0.5 meter, and 0.5 meter. The absolute difference between the estimated camera position value and ground truth camera position values measures the accuracy of the camera position prediction. Our proposed recurrent deep learning approach is a regression problem, so its output is camera position error in meters or centimeters.

Table 1. Training and testing images for 7-Scenes dataset

Scene Name	Training Images	Testing Images	Total Images
Chess	4000	2000	6000
Fire	2000	2000	4000
Heads	1000	1000	2000
Office	6000	4000	10000
Pumpkin	4000	2000	6000
RedKitchens	7000	5000	12000
Stairs	2000	1000	3000
Total	26000	17000	43000

3.4 Dataset

The model evaluated by this study used the Microsoft Research dataset, Microsoft 7-Scene [25], [26]. The RGB-D dataset known as 7-Scenes, which includes seven distinct interior scenes, is widely used. Figure 1 illustrates some of the examples. The 640 × 480 RGB-D images were taken with a portable Kinect camera and matched to the ground truth

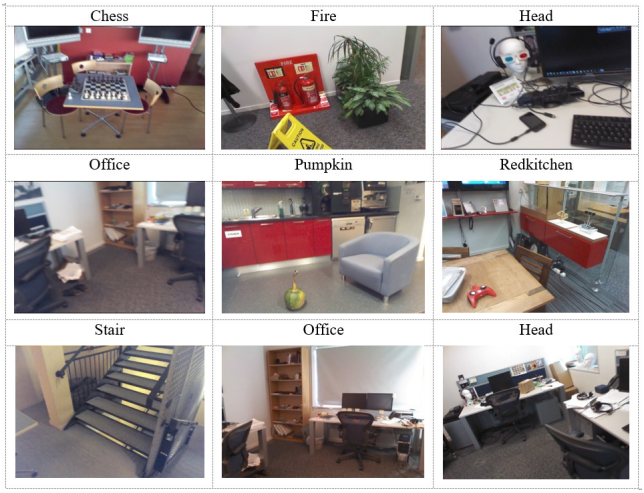

Figure 1. Example images of 7-Scenes dataset

camera positions found using the Kinect fusion technique. Every scene also has a sophisticated 3D model. Each scene comprises 2 thousand to 12 thousand tracked RGB-D camera frames divided into training and testing data. A total of 43 thousand images were used in the proposed approach, as shown in Table 1. Determining each pixel's distance from the camera is known as depth estimation. One can derive depth from a scene by looking at it from one or more perspectives.

3.5 Data Pre-precessing and Hyperparameter Assigned

The position prediction is produced as the result of an RNN framework, which takes images as input. The trained model used the Longshot Term Memory (LSTM) approach. The ADAM optimization technique [27], which requires less fine-tuning and is comparatively flexible in learning rate and other training parameters, is how we trained recurrent neural networks. The network is uniformly trained by scaling the images to 256 pixels on a 7-Scenes dataset. The input image intensity values were scaled to range from -1 to 1. All other network elements were started randomly, except the ResNet34 [28] component, which was pre-trained on the ImageNet dataset. We used a random and cen-

tralized cropping strategy to reduce the size of 256×256 pixel images for the proposed network throughout the training and testing phases. We implement our plans with the 5×10^{-5} learning rate Adam solver, which we employ with PyTorch [30] and Python 3.10. We trained the network on a CPU using the following parameters: epochs = 20, batch_size = 64, train dropout = 0.5, test dropout = 0.0, and weight initializations, $\beta = 3.0$ and $\gamma = 0.0$. The reliability of this study is validated by comparing the outcomes of various RNN networks, such as LSTM.

3.6 Hardware and Programming Language

A well-defined experimental setup is necessary to integrate the RNN and loss function optimization and achieve an efficient camera position. The hardware employed in this research included a high-performance Graphics Processing Unit (GPU) to accelerate the training of deep learning models. An NVIDIA Tesla V100 GPU was used to speed up the processing. Python 3.10 is the primary programming language used in the software framework. It uses popular deep-learning tools, such as PyTorch. These programs provide a flexible and reliable framework for training and implementing neural network models.

3.7 Evaluation Metrics

A comparison of the expected and ground truth camera positions is made using measures such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE) to assess the performance of the RNN. We used qualitative measures, such as visual inspection of projected camera trajectories superimposed on the indoor environment, to assess the model's effectiveness in capturing complicated spatial dynamics.

Mean Absolute Error (MAE): The mean absolute error represents the average of the absolute differences between the dataset's actual and anticipated values.

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |x_i - y_i|, \tag{3}$$

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

Mean Square Error (MSE): The mean squared error represents the squared average difference between the data set's original and predicted values.

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

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

Root Mean Square Error (RMSE): The square root of the Mean Squared error is called the Root Mean Squared Error.

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

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

4 Results

4.1 Introduction

The objective of the study was to use loss function correction to increase the indoor camera positioning systems' accuracy and robustness. Through implementing cutting-edge error-correcting approaches and improving the loss functions, the research aimed to increase surveillance and positioning accuracy in indoor environments. The section that follows provides a detailed description of the study's outcomes.

4.2 Predicted Pose Error

The 7-Scenes dataset contains seven scenes, each containing from 2 to 12 sequences, and each sequence contains 1000 frames. The proposed model was trained and tested separately for each scene with loss function optimization and without loss function optimization. Figure 2 shows how the recurrent neural network model was trained both with and without loss function optimization to determine the positional error from sequences of images. The positional error in the loss function optimization situation is within the range of 0.15 to 0.18 meters.

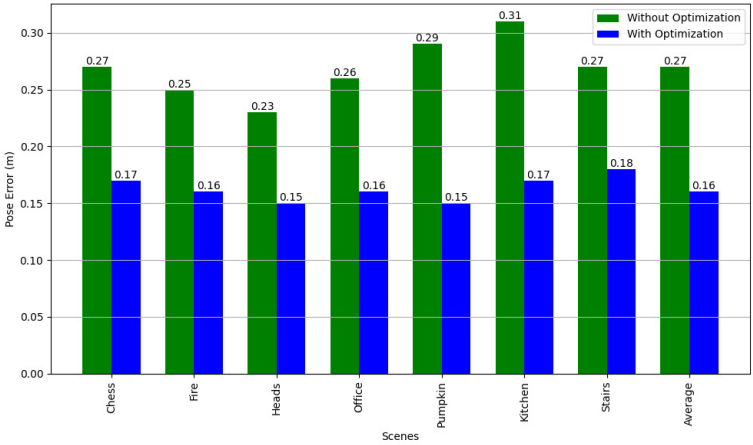

Figure 2. The pose error values obtained with and without the loss function optimization for 7-Scenes dataset

Out of seven scenes, "Head" has a minimum positional error of 0.15 meters, and the "Stairs" has a maximum position error of 0.18 meters. The other five scenes' position errors are rather close together. However, for the scenes without loss function optimization, the positional error ranges from 0.23 to 0.31 meters. In this particular case, the "Head" scene has the lowest positional error (0.23m), whereas the "RedKitchen" scene has the maximum positional error (0.31m). The other five scenes' position errors are relatively close together.

4.3 Evaluation of the RNN Model

We trained the deep learning model using loss function optimization and without loss function optimization for model validation, and we calculated the MAE, MSE, and RMSE for every scene. The resulting error distribution values, both with and without loss function optimization, are displayed in Table 2 and Table 3.

The results in Table 2 show that the error rate differs in various situations without employing loss function optimization. The "Heads" scenario has the lowest error out of the seven scenes, with an MAE

(0.1317), an MSE (0.0699), and an RMSE (0.2344). The maximum error distribution is displayed by "RedKitchen," which has an MAE (0.2232), an MSE (0.0837), and an RMSE (0.3275).

Table 2. Error distribution without loss function optimization

Scene Name	MAE	MSE	RMSE
Chess	0.1546	0.0700	0.2701
Fire	0.1964	0.0738	0.2980
Heads	0.1317	0.0699	0.2344
Office	0.2023	0.0783	0.2896
Pumpkin	0.2076	0.0790	0.3075
RedKitchens	0.2232	0.0837	0.3275
Stairs	0.1867	0.0746	0.2841

Again, looking at Table 3 more closely, the results of loss function optimization indicate that the error rate varies with the scene. The scenes with the lowest error among the seven are the "Heads" scene (RMSE (0.1931) and the MAE (0.1117)) and the "Chess" scene (MSE (0.0402)). "RedKitchen" exhibits the most significant error distribution, with an MAE (0.2132); "Office" scene gives an MSE (0.0672); and "RedKitchen" scene gives an RMSE (0.2782).

Table 3. Error distribution with loss function optimization

Scene Name	MAE	MSE	RMSE
Chess	0.1346	0.0402	0.2522
Fire	0.1564	0.0462	0.2589
Heads	0.1117	0.0571	0.1931
Office	0.1723	0.0672	0.2461
Pumpkin	0.1876	0.0583	0.2575
RedKitchens	0.2132	0.0573	0.2782
Stairs	0.1467	0.0651	0.2431

Each of these two scenarios has comparatively more minor errors in the loss function optimization. Therefore, it is apparent in the five remaining scenes that minimizing the loss function lowers errors. It is

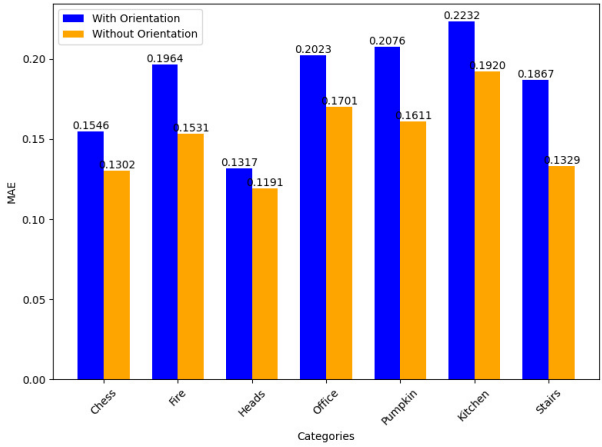

Figure 3. MAE for the with and without loss function optimization for the 7-Scenes dataset

clear from comparing the error findings in Table 2 and Table 3 that loss function optimization improves the model's accuracy.

We can compare the error distribution of the seven scenes with and without loss function optimization. As seen in Figures 3–5, there are comparatively smaller errors in the loss function optimization for each of the seven scenarios.

Assume, for instance, that we contrast the errors in the "Chess" scene. In that instance, we can observe that the error for MAE(0.1546) without loss function optimization and MAE(0.1346) with loss function optimization; for MSE(0.0700) with loss function optimization and without loss function optimization error MSE(0.0402); and for RMSE (0.2701) without loss function optimization error and RMSE(0.2522) with loss function optimization error.

Comparing the errors from Figures 3–5 of the "Stairs" scene, we can observe that, for MAE(0.1867) without loss function optimization and MAE(0.1467) with loss function optimization; for MSE(0.0746) without loss function optimization and MSE(0.0651) with loss function optimization; RMSE(0.2841) without loss function optimization, and RMSE(0.2431) with loss function optimization.

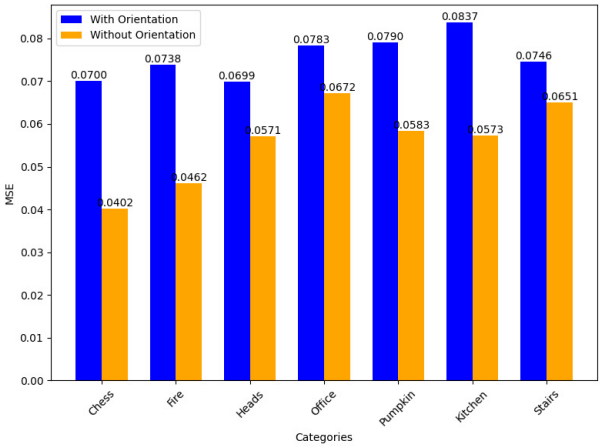

Figure 4. MSE for the with and without loss function optimization for the 7-Scenes dataset

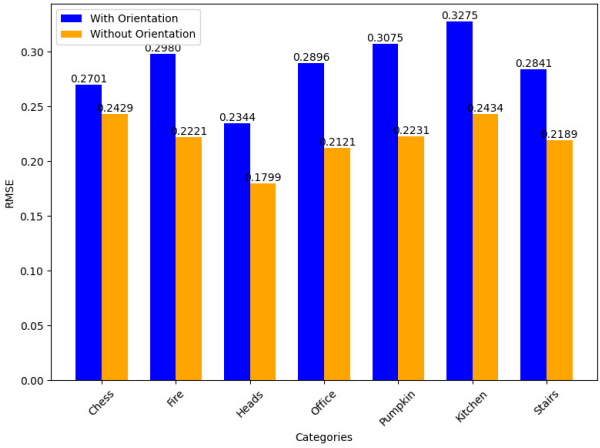

Figure 5. RMSE for the with and without loss function optimization for the 7-Scenes dataset

These two scenarios have comparatively smaller errors with the loss function optimization. Therefore, it is apparent in the five remaining scenes that loss function optimization lowers the errors. It is clear from comparing the error findings in Figures 3–5 that loss function optimization improves the model's accuracy.

Table 4. Average pose errors of existing algorithms and our proposed architecture

Camera pose error in meter(m)								
Methods	Chess	Fire	Head	Office	Pumpkin	RedKitchen	Stairs	Avg. Pose Error
PoseNet [7]	0.32	0.47	0.29	0.48	0.47	0.59	0.47	0.45
LSTM-PoseNet [29]	0.24	0.34	0.21	0.31	0.33	0.37	0.41	0.31
MapNet [14]	0.08	0.27	0.18	0.17	0.22	0.23	0.30	0.21
AtLoc [11]	0.10	0.25	0.16	0.17	0.21	0.23	0.26	0.20
EpiLoc [31]	0.07	0.24	0.14	0.18	0.18	0.23	0.24	0.18
CGAPoseNet [32]	0.19	0.20	0.18	0.19	0.19	0.20	0.21	0.19
Proposed (without optimization)	0.27	0.25	0.23	0.26	0.29	0.31	0.27	0.27
Proposed (with optimization)	0.17	0.16	0.15	0.16	0.15	0.17	0.18	0.16

4.4 Comparison with Existing Researches

Table 4 compares the results of our recurrent deep architecture with state-of-the-art research. It shows seven scenes' individual and average pose errors from our study and state-of-the-art research using the 7-Scene dataset. Recent research results include PoseNet, LSTM-PoseNet, MapNet, AtLoc, EpiLoc, and CGAPoseNet. First, we will examine the effects of camera pose error in loss function optimization on simulation results and compare the results obtained with deep architecture with state-of-the-art. We have trained the recurrent deep

architecture twice, once when the loss function was optimized and the other time when the loss function wasn't optimized. It can be seen that the pose error increases when the loss function is not optimized. Here, it can be seen that the average positional error is 0.16 m when the loss function is optimized, and the average positional error is 0.27 m when the loss function is not optimized. The positional error decreases when the loss function is optimized.

According to Table 4, the average positional error among the available investigations ranges from 0.16 m to 0.45 m; PoseNet has the highest average positional error at 0.45 m, while EpiLoc has the lowest average positional error at 0.18 m. When the loss function is optimized, the average positional error obtained in the study is 0.16 m, which is less than in all the research indicated in the table. It is apparent from the analysis of the results above that, after loss function optimization, our recurrent deep architecture's (0.16 m) results have a small position error, which is essential for using more accurate camera localization.

5 Discussion

The position of indoor cameras through loss function optimization is essential in monitoring and surveillance systems. Loss function optimization has significantly increased the indoor positioning of the camera system's performance. The proposed approach has successfully minimized the discrepancy between the actual and predicted camera positions by carefully modifying loss functions, including cross-entropy loss and mean squared error (MSE). It allows observing and localizing targets more precisely inside restricted environments. This optimization process has dramatically increased the surveillance system's ability to handle complicated data patterns and changes, increasing its dependability and effectiveness. The research procedure has shown several challenges and limitations despite the findings indicating positive advancements in indoor camera positioning. The computational complexity of implementing complicated optimization algorithms is a significant limitation, necessitating the development of resource-saving and more straightforward processes to ensure real-time application. In conclusion, recurrent neural networks and loss function optimization

work well together to significantly improve indoor camera location systems' accuracy, robustness, and efficacy. This research opens up new avenues for developing sophisticated surveillance systems, which might be used for everything from security and safety monitoring to automated indoor navigation and human-machine interaction.

6 Conclusion

In this study, we have used RNN and error correction to examine the complex process of advanced indoor camera positioning. We've demonstrated how applying sophisticated algorithms and state-of-the-art techniques may significantly increase the accuracy and dependability of indoor camera positioning systems. By appropriately addressing various error sources and optimizing the loss function, we have successfully decreased the camera pose error. Our research results highlight the importance of loss function optimization and robust position prediction algorithms in achieving precise indoor camera positioning. We have developed a comprehensive framework using recurrent neural network integration that improves indoor camera position performance and lowers errors, leading to more accurate and reliable monitoring and tracking capabilities.

7 Future Works

Real-time camera position changes in response to changing environmental dynamics can be achieved by extending the current method to optimize the loss function. Examining how numerous cameras coordinate indoors is an exciting avenue for future research. Analyzing techniques that provide seamless collaboration across several cameras, such as integrating intelligent information-sharing protocols and distributed optimisation algorithms, can significantly enhance the overall precision and robustness of the camera positioning system. It is necessary to extend its implementation to other indoor circumstances and environments to determine whether the suggested methodology is effective. Future research should focus on conducting extensive cross-validation

tests in various indoor contexts to evaluate the generalizability of the suggested system. Creating diverse datasets that show various room designs, furniture arrangements, and lighting conditions may be necessary to maintain the system's dependability and flexibility in real-world situations.

8 Acknowledgement

The authors would like to thank the Faculty of Computing, University Technology Malaysia.

References

[1] M. S. Alam, F. B. Mohamed, A. Selamat, and A. B. Hossain, "A Review of Recurrent Neural Network Based Camera Localization for Indoor Environments," *IEEE Access*, vol. 11, pp. 43985–44009, 2023. [Online]. Available: https://doi: 10.1109/ACCESS.2023.3272479.

[2] M. S. Alam, A. K. M. B. Hossain, and F. B. Mohamed, "Performance Evaluation of Recurrent Neural Networks Applied to Indoor Camera Localization," *International Journal of Emerging Technology and Advanced Engineering*, vol. 12, no. 8, pp. 116–124, 2022. [Online]. Available: https://doi: 10.46338/ijetae0822_15.

[3] J. Zhang et al., "RenderNet: Visual Relocalization Using Virtual Viewpoints in Large-Scale Indoor Environments," *arXiv preprint*, arXiv:2207.12579, 2022. https://doi.org/10.48550/arXiv.2207.12579.

[4] S. Chen, Z. Wang, and V. Prisacariu, "Direct-PoseNet: Absolute pose regression with photometric consistency," in *2021 International Conference on 3D Vision (3DV)*, IEEE, 2021, pp. 1175–1185. https://doi.org/10.48550/arXiv.2104.04073.

[5] S. Zhong, and Y. Liu, "Deep residual learning for image steganalysis," *Multimed Tools Appl*, vol. 77, pp. 10437–10453, 2018. https://doi.org/10.1007/s11042-017-4440-4.

[6] S. Kim, I. Kim, L. F. Vecchietti, and D. Har, "Pose estimation utilizing a gated recurrent unit network for visual localization," *Applied Sciences*, vol. 10, no. 24, Article No. 8876, 2020. [Online]. Available: https://doi.org/10.3390/app10248876.

[7] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2938–2946.

[8] A. Kendall and R. Cipolla, "Modelling Uncertainty in Deep Learning for Camera Relocalization," Sep. 2015, [Online]. Available: http://arxiv.org/abs/1509.05909.

[9] T. Naseer and W. Burgard, "Deep regression for monocular camera-based 6-DoF global localization in outdoor environments," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Vancouver, BC, Canada), 2017, pp. 1525–1530, DOI: 10.1109/IROS.2017.8205957. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8205957/.

[10] A. Kendall and R. Cipolla, "Geometric Loss Functions for Camera Pose Regression with Deep Learning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Honolulu, HI, USA), 2017, pp. 6555–6564, DOI: 10.1109/CVPR.2017.694.

[11] B. Wang, C. Chen, C. X. Lu, P. Zhao, N. Trigoni, and A. Markham, "AtLoc: Attention Guided Camera Localization," 2020. [Online]. Available: https://github.com/BingCS/AtLoc.

[12] P. Purkait, C. Zhao, and C. Zach, "Synthetic View Generation for Absolute Pose Regression and Image Synthesis," in *BMVC 2018*, 2019, Article ID: 69.

[13] M. Cai, C. Shen, and I. Reid, "A hybrid probabilistic model for camera relocalization," in *BMVC 2018*, 2019, Article ID: 238. Accessed: Mar. 10, 2022. [Online]. Available: https://digital.library.adelaide.edu.au/dspace/handle/2440/124684.

[14] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz, "Mapnet: Geometry-aware learning of maps for camera localization," 2017, Accessed: Mar. 10, 2022. [Online]. Available: https://openreview.net/forum?id=lTawICyy5HP.

[15] N. Radwan, A. Valada, and W. Burgard, "VLocNet++: Deep Multitask Learning for Semantic Visual Localization and Odometry," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4407–4414, Oct. 2018, DOI: 10.1109/LRA.2018.2869640. Accessed: Mar. 10, 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8458420/.

[16] R. Clark, S. Wang, A. Markham, N. Trigoni, and H. Wen, "Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6856–6864, openaccess.thecvf.com, Accessed: Mar. 09, 2022. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2017/html/Clark_VidLoc_A_Deep_CVPR_2017_paper.html.

[17] V. Balntas, S. Li, V. Prisacariu, "Relocnet: Continuous metric learning relocalisation using neural nets," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 751–767, openaccess.thecvf.com, Accessed: Mar. 10, 2022. [Online]. Available: http://openaccess.thecvf.com/content_ECCV_2018/html/Vassileios_Balntas_RelocNet_Continous_Metric_ECCV_2018_paper.html.

[18] Z. Laskar, I. Melekhov, S. Kalia, and J. Kannala, "Camera Relocalization by Computing Pairwise Relative Poses Using Convolutional Neural Network," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 929–938. [Online]. Available: https://openaccess.thecvf.com/content_ICCV

_2017_workshops/w17/html/Laskar_Camera_Relocalization_by_ICCV_2017_paper.html.

[19] M. Ding, Z. Wang, J. Sun, J. Shi, and P. Luo, "CamNet: Coarse-to-fine retrieval for camera re-localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 2871–2880, openaccess.thecvf.com, Accessed: Mar. 10, 2022. [Online]. Available: http://openaccess.thecvf.com/content_ICCV_2019/html/Ding_CamNet_Coarse-to-Fine_Retrieval_for_Camera_Re-Localization_ICCV_2019_paper.html.

[20] A. Doumanoglou, V. Balntas, R. Kouskouridas, and T.-K. Kim, "Siamese Regression Networks with Efficient mid-level Feature Extraction for 3D Object Pose Estimation," *arXiv preprint*, arXiv:1607.02257, 2016. https://doi.org/10.48550/arXiv.1607.02257.

[21] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu, "Relative camera pose estimation using convolutional neural networks," in *International Conference on Advanced Concepts for Intelligent Vision Systems* (Lecture Notes in Computer Science, vol. 10617), Springer, 2017, pp. 675–687. https://doi.org/10.1007/978-3-319-70353-4_57.

[22] Z. Cui, L. Pan, and S. Liu, "Hybrid BiLSTM-Siamese Network for Relation Extraction", in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 2019, pp. 1907–1909.

[23] M. S. Alam, F. B. Mohamed, A. Selamat, F. Ahmed, and AKM. Hossain, "Analyzing the Impact of Scene Transitions on Indoor Camera Localization through Scene Change Detection in Real-Time," *Intelligent Automation & Soft Computing*, vol. 39, no. 3, pp. 1–10, 2024, DOI: 10.32604/iasc.2024.051999.

[24] M. S. Alam, F. B., Mohamed, A. Selamat, and AKM. B. Hossain, "A Recurrent Deep Architecture for Enhancing Indoor Camera Localization Using Motion Blur Elimination," *Journal of Robotics*

and Control (JRC), vol. 5, no. 4, pp. 1028–1040, 2024, DOI: https://doi.org/10.18196/jrc.v5i4.21930.

[25] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in RGB-D images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 2930–2937, openaccess.thecvf.com, 2013. Accessed: Mar. 09, 2022. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2013/html/Shotton_Scene_Coordinate_Regression_2013_CVPR_paper.html.

[26] B. Glocker, S. Izadi, J. Shotton, and A. Criminisi, "Real-time RGB-D camera relocalization," in *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, (Adelaide, SA, Australia), 2013, pp. 173–179, DOI: 10.1109/ISMAR.2013.6671777.

[27] Z. Zhang, "Improved Adam Optimizer for Deep Neural Networks," in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, (Banff, AB, Canada), 2018, pp. 1–2, DOI: 10.1109/IWQoS.2018.8624183.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778. https://openaccess.thecvf.com/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf.

[29] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using LSTMs for structured feature correlation," 2017, [Online]. Available: https://openaccess.thecvf.com/content_ICCV_2017/papers/Walch_Image-Based_Localization_Using_ICCV_2017_paper.pdf.

[30] S. Imambi, K. B. Prakash, and G. R. Kanagachidambaresan, "PyTorch," in *Programming with TensorFlow: Solution for Edge Computing Applications*, pp. 87–104, 2021.

[31] L. Xu, T. Guan, Y. Luo, Y. Wang, Z. Chen, and W. Liu, "Epi-Loc: Deep Camera Localization Under Epipolar Constraint," *KSII Transactions on Internet & Information Systems*, vol. 16, no. 6, pp. 2044–2059, 2022.

[32] A. Pepe and J. Lasenby, "Cga-posenet: Camera pose regression via a 1d-up approach to conformal geometric algebra," *arXiv preprint*, arXiv:2302.05211, 2023.

Shamsul Alam, Farhan Mohamed,
Bellal Hossain

Received April 04, 2024
Revised August 16, 2024
Accepted August 26, 2024

Shamsul Alam
ORCID: https://orcid.org/0000-0002-9419-3928
Department of Emergent Computing, Faculty of Computing,
Universiti Teknologi Malaysia, 81310, Johor Bahru, Malaysia
and
Department of Computer Science and Artificial Intelligence,
College of Computing and Information Technology,
University of Bisha, Bisha, 61922, Saudi Arabia
E–mail: shamsul20@graduate.utm.my

Farhan Mohamed
ORCID: https://orcid.org/0000-0002-5298-8642
Department of Emergent Computing, Faculty of Computing,
Universiti Teknologi Malaysia, 81310, Johor Bahru, Malaysia
and
Media and Game Innovation Centre of Excellence (MaGICX),
University Technology Malaysia, Johor Bahru, 81310, Malaysia
E–mail: farhan@utm.my

Bellal Hossain
ORCID: https://orcid.org/0000-0003-3877-7037
Department of Computer Science, Faculty of Computing,
Universiti Teknologi Malaysia, 81310, Johor Bahru, Malaysia
and
Department of Information Systems and Cyber Security,
College of Computing and Information Technology,
University of Bisha, Bisha, 61922, Saudi Arabia
E–mail: k.m.a@graduate.utm.my