Intelligent Overfitting to Environments

Soomin Lee
Department of Mechanical and Process Engineering
ETH Zurich, Switzerland
leesoo@student.ethz.ch

Xiaoao Song
Department of Informatics
The University of Zurich, Switzerland
xiaoao.song@uzh.ch

Abstract: We present a data augmentation method for enhancing visual relocalization capability in environments with substantial changes over time. Our augmentation method focuses on removal of objects from scenes, which enables the network to learn backgrounds which are less subject to changes. We use a relocalization system based on a CNN architecture that generates the 6-DoF camera pose from a RGB image in order to test our method. We train the system with a dataset obtained from an indoor lab environment on a certain day, and we test the system with a dataset obtained from the same place but on a different day with special settings that introduce considerable changes in the environment. We show that the performance of the relocalization system improve with the proposed data augmentation method, and we suggest several amendments that can possibly boost the performance further. The proposed augmentation method can be easily adopted by other visual relocalization systems that use image data to train a model.

Keywords: Camera Relocalization, Data Augmentation, Learning

1 Introduction

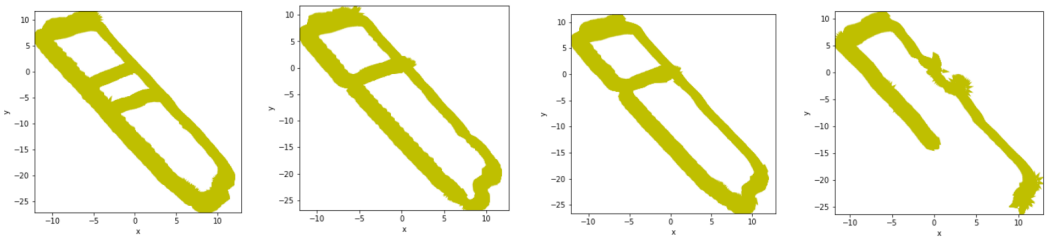
Localization is often an essential component in mobile robotics tasks, and various image-based methods concerning place recognition or relocalization in a previously encountered environment have been developed [1, 2, 3, 4]. Notably, as deep convolutional neural networks (CNNs) have gained a significant amount of attention in the past years in numerous tasks with images such as classification [5, 6, 7], approaches that employ the deep CNN architecture in relocalization or recognition tasks have been proposed and showed promising results [1, 3].

Yet, most of these approaches aim to provide a pose estimation framework that is as general as possible. As a result, such methods are prone to fail whenever the surrounding environment undergoes extreme changes. In this work, we propose to improve the performance of existing estimators and increase their robustness to environments changing over time by intentionally overfitting to a specific scene. This is reasonable as robots often operate in a fixed range of environments, while the environments can differ significantly due to rearrangements of the objects or dynamic objects such as people. We intent on using a custom data augmentation in a way that can accommodate possible changes that can occur in the environment. In particular, we will focus on removing the objects from the scene and revealing the background in order to enable the network to learn the parts of the scene that is less subject to changes.

2 Related Work

One of the typical image-based localization methods is image retrieval [1, 8, 9]. The goal of an image retrieval method is to retrieve an image from database that is most similar to the query image. Although the system can have high performance, it does not provide a precise estimation of the pose since it is unlikely that the query image has the exact pose as one of the images in the database. Another method for localization involves 3D reconstruction [10, 11], which can achieve a high accuracy but is not robust against extreme changes in environments as the feature matching procedure

4rd Conference on Robot Learning (CoRL 2020), —, Japan.



(a) *Both sides* (b) *One side* (c) *Light change* (d) *Open Lab*

Figure 1: The ground truth poses for training sequences ((a) and (b)) and test sequences ((c) and (d)). Each pose is visualized as a coordinate frame and projected on a plane.

can easily fail. The approach we took is end-to-end learning. Kendall et al. [3] showed that a deep CNN can learn a representation which is easily mapped to a 6 DoF pose, allowing regression of pose directly from images. Following this method, we target at estimating poses directly from images while providing appropriate training data that will enable the estimator to tolerate substantial changes in environments.

The relevant field to our data augmentation approach is image inpainting. Image inpainting is a task of synthesizing alternative contents in missing regions such that the modification is visually realistic and semantically correct. In computer vision, two broad approaches to image inpainting exist: patch matching using low-level image features and feed-forward generative models with deep convolutional networks. The former approach [12, 13] can synthesize plausible stationary textures, but usually makes critical failures in non-stationary cases like complicated scenes, faces and objects. The latter approach [14, 15, 16, 17, 18] can exploit semantics learned from large scale datasets to synthesize contents in non-stationary images in an end-to-end fashion. However, deep generative models based on vanilla convolutions apply same filters on all valid, invalid and mixed pixels/features, leading to visual artifacts such as color discrepancy, blurriness and obvious edge responses surrounding holes when tested on any form of masks drawn by users [15, 17].

We adopt [19] which proposed gated convolution for free-form image inpainting. It learns a dynamic feature gating mechanism for each channel and each spatial location such as inside or outside masks, RGB channels, or user-guidance channels.

3 Methods

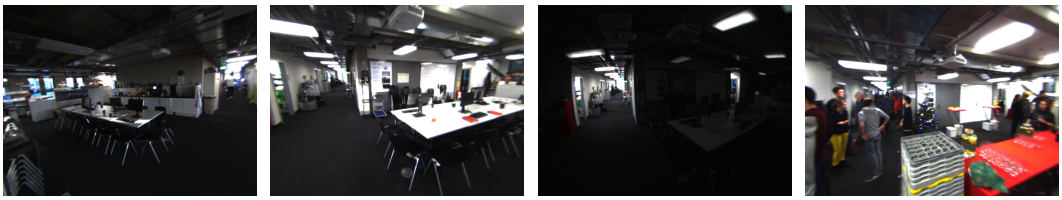
3.1 Dataset

The RGB image dataset we use are collected from an indoor lab environment (Autonomous Systems Lab at ETH Zurich), and training labels, namely camera poses, are obtained using the Maplab framework [20]. The Figure 1 shows the resulting poses of each sequence in the dataset we use in the experiments, and Figure 2 shows the example images from each sequence. Note that the example images have different viewpoints, yet they show the same place of the lab. *Both sides*, *One side* and *Light Change* sequences contain images of the lab on the same day with the same layout, but *Both sides* and *One side* sequences have a medium level of light change while *Light Change* sequence has the maximum level of light change. *Open Lab* sequence contains images of the lab on a different day, with a considerably different layout and many people. It features the lab environment during a special event that involved live demonstrations of research projects from the lab.

3.2 Data Augmentation

3.2.1 Object Detection

For the custom data augmentation, we employ the method proposed by Yu et al. [19], however, this model cannot automatically choose instances. To solve this, we first adopt the Mask R-CNN model [21] to segment instances from the original images, and then randomly remove some instances such as some chairs and sofas. Figure 3 (b) shows the detected objects from the image by Mask R-CNN and tagged by bounding box. Before feeding the mask and the instance-removed image into



(a) *Both sides* (b) *One side* (c) *Light Change* (d) *Open Lab*

Figure 2: Example images from each sequence

inpainting, we propose a dilation for the mask. This happens because the Mask R-CNN may not be able to produce segment masks which accurately reflect the true boundaries of the objects and when the instance is removed, its remaining border will have a big influence on the image inpainting part.

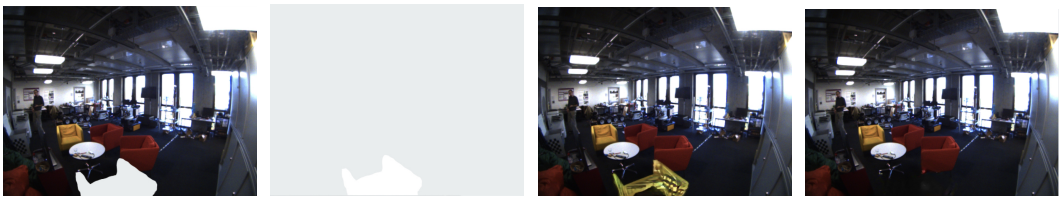
(a) *Input image* (b) *Instance segmentation* (c) *Mask of random removed instances* (d) *Mask dilation* (e) *Inpaint image*

Figure 3: Data augmentation pipeline

3.2.2 Inpainting

Next, we adopt Generative Image Inpainting model [19] to inpaint the images with some objects removed. For both Mask R-CNN and inpainting model, we use pretrained weights. After inpainting, the augmented image revealed the background that previously was covered by the objects. Figure 3 (a) to (c) shows the whole data augmentation pipeline.

An example of inpainting can be seen in the Figure 4. Without mask dilation, the remaining border of the object had a big influence on inpainting and therefore we see a yellow artificial sofa appears in the same position. After using dilation, the sofa is removed and the floor is revealed.



(a) *A sofa is removed* (b) *A hand-drawn mask* (c) *Inpainting without mask dilation* (d) *Inpainting with mask dilation*

Figure 4: Examples of inpainting

3.3 Training

For the model, we adopt the PoseNet model from Kendall et al. [3] and use the implementation of the model provided by Walch et al. [22]. The network is initialized from a pretrained *GoogLeNet* model [23] on the dataset *Places* [24], which is also provided by Walch et al. [22]. Each image is undistorted and resized from 540×720 to 256×341 prior to training. During training, we randomly crop images to the input size 224×224 and use central crops during testing. Also, the image mean of each training sequence is subtracted from images. We use *Adam* optimization method, taking the

same hyperparameters from [22]. The batch size is 75 and is randomly shuffled. Using the same loss function in [3], we set the relative weight of the orientation error with respect to the positional error β as 150.

4 Experiment

4.1 Experimental Setup

We train our model on three datasets: (A) original dataset, (B) augmented dataset with images that have objects covered with black masks without inpainting, (C) augmented dataset with images that have objects deleted and inpainted. The original training dataset consists of two sequences, *Both sides* and *One side*. Then the model is evaluated separately on the other two sequences, *Light Change* and *Open Lab*. We conduct several experiments with different removal ratio such as 20%, 60% and 85%. In all of our experiments, we limit the single instance detection area within the range of 0.2% - 45% of the whole image. Any detected instances that exceed this range will be ignored. We set this constrain because we believe that the instances that are too small are not of substantial help in understanding the environment, and the instances that are too large are not conducive to subsequent image inpainting process. The augmentation related parameters can be seen in Table 1.

Table 1: Setup of experiments

Data sequence	No. of images	No. of augmented images	Avg. ratio of remove area
Both sides	200	765	7.68%
One side	148	637	7.11%

(a) Experiment 1: instances removal ratio: 20%, no. of frames skipped: 20, max. augmentation per image: 10

Data sequence	No. of images	No. of augmented images	Avg. ratio of remove area
Both sides	200	1284	15.00%
One side	148	1159	14.25%

(b) Experiment 2: instances removal ratio: 60%, no. of frames skipped: 20, max. augmentation per image: 15

Data sequence	No. of images	No. of augmented images	Avg. ratio of remove area
Both sides	667	1069	20.71%
One side	494	1041	20.65%

(c) Experiment 3: instances removal ratio: 85%, no. of frames skipped: 6, max. augmentation per image: 5

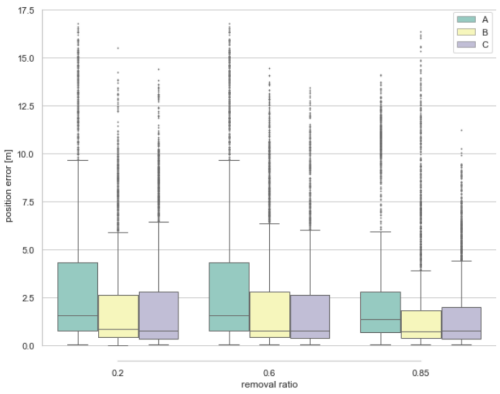
4.2 Experimental Results

The test results of datasets with different settings are shown in Figure 5 and Table 2, and the plots in Appendix A and Appendix B visualize the individual errors locally on the map.

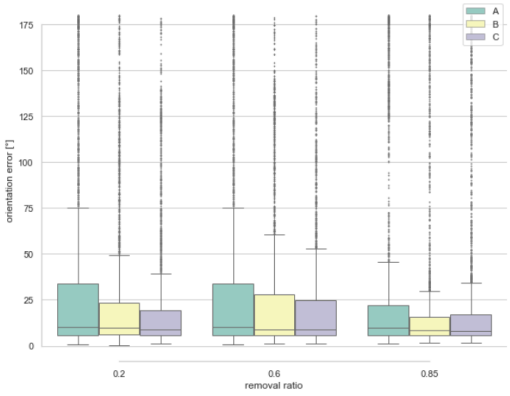
For the tests on *Light Change* sequence, one can observe that both position and orientation errors decrease with the data augmentation in all cases. Some differences between the results of the dataset (B) and (C) might come from the quality of inpainted images, implying that having black masks over objects are helpful than having poorly inpainted backgrounds. One can confirm these remarks with the plots in Appendix A. The red part with the highest error from the results of (A) in the plots is where the low light condition takes part in. This area becomes slightly smaller for orientation prediction than that for position prediction, as the orientation relatively does not change much in the hallway and thus excludes some of the parts of the hallway. The improvements in the area support our claim deduced from the statistics that the augmented dataset (B) and (C) help improve the pose estimation when the lighting condition changes in the environment. Also, the part where the error increases with inpainting such as in Figure A.1(b)-(C) can be explained by misguided inpainting as in Figure 6 and lack of viewpoints in the training sequences.

For the tests on *Open Lab* sequence, the dataset (B) shows improvements in all cases while the dataset (C) doesn't always yield better results compared to the dataset (A). Exploiting the plots in Appendix B, the parts with the highest errors are verified to be the parts that actually have a lot of changes, namely open areas with more people and live demonstrations. These parts are extremely challenging to predict accurately because not only there are a lot of discrepancies between training and test images, but also there are discrepancies between the training trajectories the test trajectory. To be more precise, the camera focuses more on the open area parts with different views of ongoing events while those viewpoints are not included in the training samples.

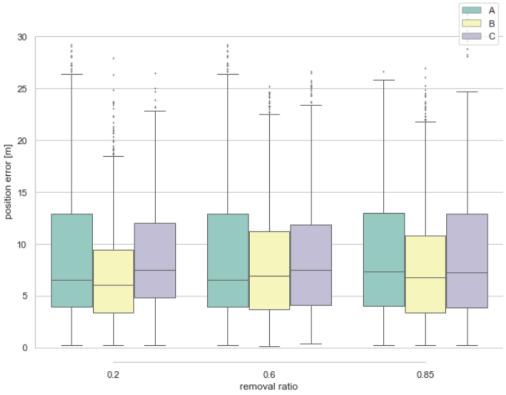
Nevertheless, poor pose estimations for those parts are partially alleviated with the augmentation method applied to the dataset (B). On the other hand, the low quality of inpainting appears to be one of the main reasons behind the low performance of the dataset (C) as in Figure 6. Moreover, bad inpainting of the objects that will never be changed in the environment, such as the door in Figure 6b, is also a disadvantage to learning of the network. Since neither the original images nor the inpainted images look alike to the test images, it can be reasoned that faulty inpainting influences the test result more significantly when testing on *Open Lab* than on *Light Change*. Besides, the results of inpainting change depending on the choice of instances. Since we randomly select a subset of instances to remove among the detected instances, it is possible that the dataset with a high removal ratio is even more prone to erroneous inpainting. For instance, if there exists an instance that inpainting fails, it is more likely that the instance is included in the subset when more instances need to be selected. As the experiments with the dataset (B) show some promising results, more experiments with better inpainted images should be conducted prior to drawing a conclusion on how the proposed data augmentation method help on improving image relocalization capability.



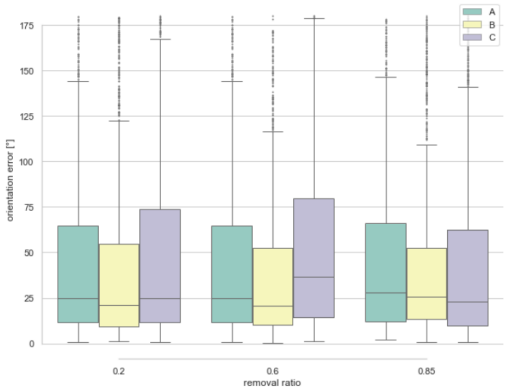
(a) Position error [m] when tested on *Light Change*



(b) Orientation error [°] when tested on *Light Change*



(c) Position error [m] when tested on *Open Lab*



(d) Orientation error [°] when tested on *Open Lab*

Figure 5: Test results in graphs. The box shows the quartiles of the error distribution and the whiskers extend to show the rest of the distribution. Also, the line in the box indicates the median value. The points outside of the whiskers are points that are regarded as outliers using the inter-quartile range.

Table 2: Median and RMSE test results

Removal ratio		A	B	C
0.2	Median	1.57 m, 10.13 °	0.83 m, 9.66 °	0.74 m, 8.79 °
	RMSE	5.24 m, 60.03 °	3.68 m, 45.15 °	3.97 m, 41.89 °
0.6	Median	1.57 m, 10.13 °	0.76 m, 9.08 °	0.74 m, 8.92 °
	RMSE	5.24 m, 60.03 °	4.04 m, 46.15 °	3.48 m, 40.75 °
0.85	Median	1.13 m, 8.34 °	0.73 m, 8.33 °	0.74 m, 8.19 °
	RMSE	5.04 m, 59.68 °	3.60 m, 46.94 °	2.71 m, 49.98 °

(a) Median and RMSE test results on *Light Change*

Removal ratio		A	B	C
0.2	Median	6.47 m, 24.89 °	6.01 m, 21.02 °	7.46 m, 24.78 °
	RMSE	10.94 m, 60.41 °	8.40 m, 61.61 °	9.59 m, 68.88 °
0.6	Median	6.47 m, 24.89 °	6.87 m, 20.65 °	7.46 m, 36.59 °
	RMSE	10.94 m, 60.41 °	9.58 m, 54.42 °	10.20 m, 73.09 °
0.85	Median	6.67 m, 27.15 °	6.77 m, 25.79 °	7.27 m, 23.07 °
	RMSE	9.24 m, 62.08 °	9.66 m, 62.13 °	10.69 m, 64.56 °

(b) Median and RMSE test results *Open Lab*.

(a) The original image (left) and the poorly inpainted image (right) of the lobby.

(b) The original image (left) and the poorly inpainted image (right) of the hallway.

Figure 6: Examples of inaccurate inpainting and suboptimal instance selection

5 Conclusion

In this project we aim to improve relocalization capability in changing environments by overfitting to a specific scenario. We use a custom data augmentation method to enable positioning a robot in a changed environment. Specifically, we focused on augmenting the data by removing the objects

from the scene. The proposed procedure can be applied to any image dataset in order to increase robustness to changing environments.

In order to improve the results, we believe the next step would be improving the inpainting quality. In this work, we use a pretrained model provided by Yu et al. [19] that might not be optimal for inpainting images of indoor office environments. Therefore, retraining the model with indoor dataset might influence the results to a large extent. Also, we only have two training sequences that do not contain various poses, orientations in particular, compared to the indoor datasets used in [3]. Moreover, *Open Lab* sequence have different viewpoints as mentioned earlier. Accordingly, it seems reasonable to add more training sequences for the relocalization system to work better. In addition, the parameter β from the PoseNet model [3] has not been explored sufficiently to find the optimal value. Lastly, tuning and adjusting the way of creating the augmented dataset can potentially enhance the performance. For instance, we can make a dataset that includes several incrementally increasing removal ratios. Removing all the detected objects instead of randomly choosing a subset of them is also possible. Aside from it, it will be beneficial to have a better instance segmentation method as our results depend heavily on it as well.

Acknowledgments

We'd like to express our appreciation to Andrei Cramariuc, Florian Tschopp and Margarita Grinvald for providing guidance and feedback throughout the project. Also, we thank Dr. Cesar Dario Cadena Lerma and Dr. Jen Jen Chung for their support and sharing valuable insights.

References

[1] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[2] M. Cummins and P. Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.

[3] 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*, pages 2938–2946, 2015.

[4] 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*, pages 2930–2937, 2013.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[6] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[8] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2911–2918. IEEE, 2012.

[9] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3304–3311. IEEE, 2010.

[10] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2599–2606. IEEE, 2009.

[11] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *2011 International Conference on Computer Vision*, pages 667–674. IEEE, 2011.

[12] A. F. Connelly Barnes, Eli Shechtman and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (TOG) (Proceedings of SIGGRAPH 2009)*, 2009.

[13] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 76, pages 1033–1038. IEEE, 1999.

[14] Y. S. Chao Yang, Q. T. Xiaofeng Liu, and C.-C. J. Kuo. Image inpainting using block-wise procedural training with annealed adversarial counterpart. 2018.

[15] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and Locally Consistent Image Completion. *ACM Transactions on Graphics (Proc. of SIGGRAPH)*, 36(4):107, 2017.

[16] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi. Edgeconnect: Generative image inpainting with adversarial edge learning. 2019.

[17] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Generative image inpainting with contextual attention. *arXiv preprint arXiv:1801.07892*, 2018.

[18] C. Zheng, T.-J. Cham, and J. Cai. Pluralistic image completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1438–1447, 2019.

[19] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589*, 2018.

[20] T. Schneider, M. T. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart. maplab: An open framework for research in visual-inertial mapping and localization. *IEEE Robotics and Automation Letters*, 2018. doi:10.1109/LRA.2018.2800113.

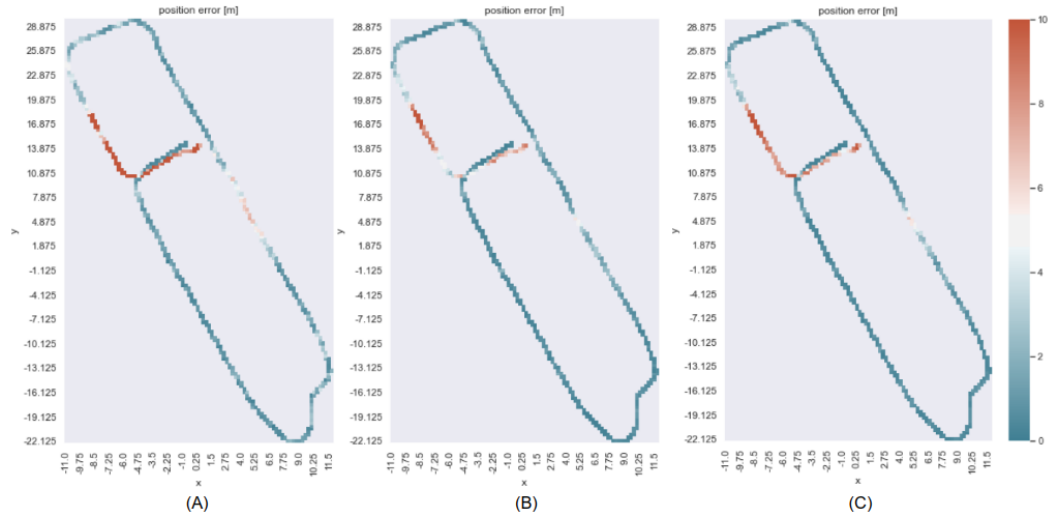
[21] W. Abdulla. Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. https://github.com/matterport/Mask{_}RCNN, 2017.

[22] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, and D. Cremers. Image-based localization using lstms for structured feature correlation. In *ICCV*, October 2017. URL https://github.com/NavVisResearch/NavVis-Indoor-Dataset.

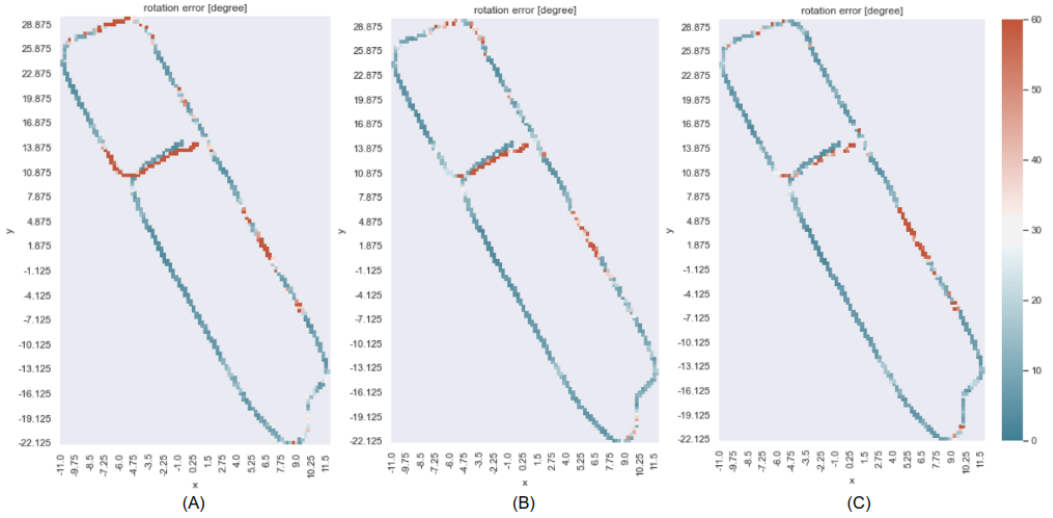
[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015. URL http://arxiv.org/abs/1409.4842.

[24] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.

Appendix A. Pose Error Heatmaps for tests on *Light Change*

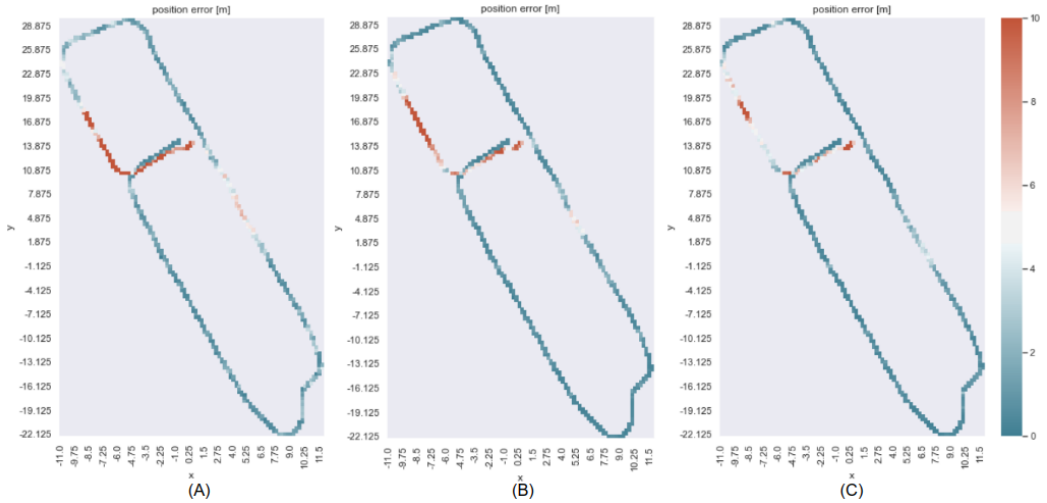


(a) Position error [m]

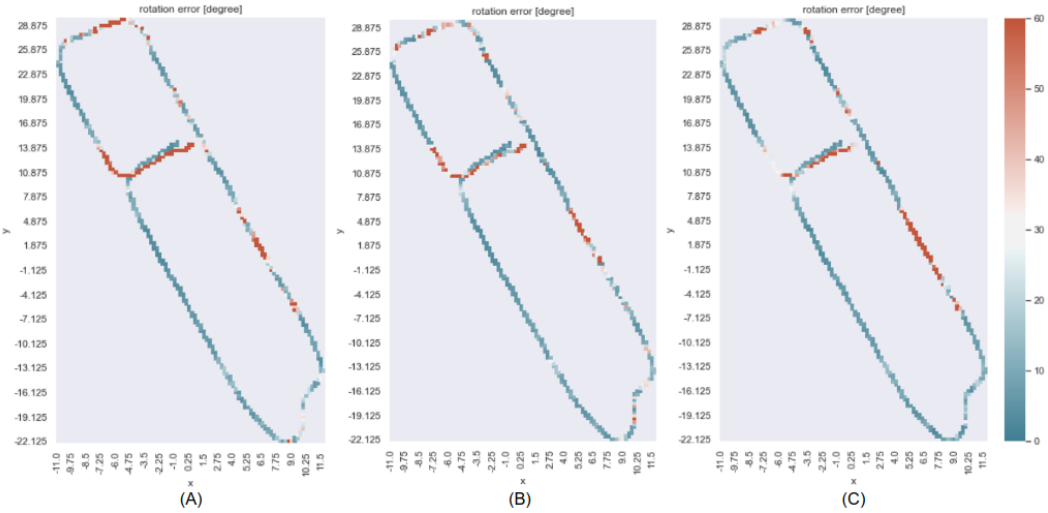


(b) Orientation error [°]

Figure A.1: Visualization of local errors when tested on *Light Change*, in case of the dataset with removal ratio 0.2

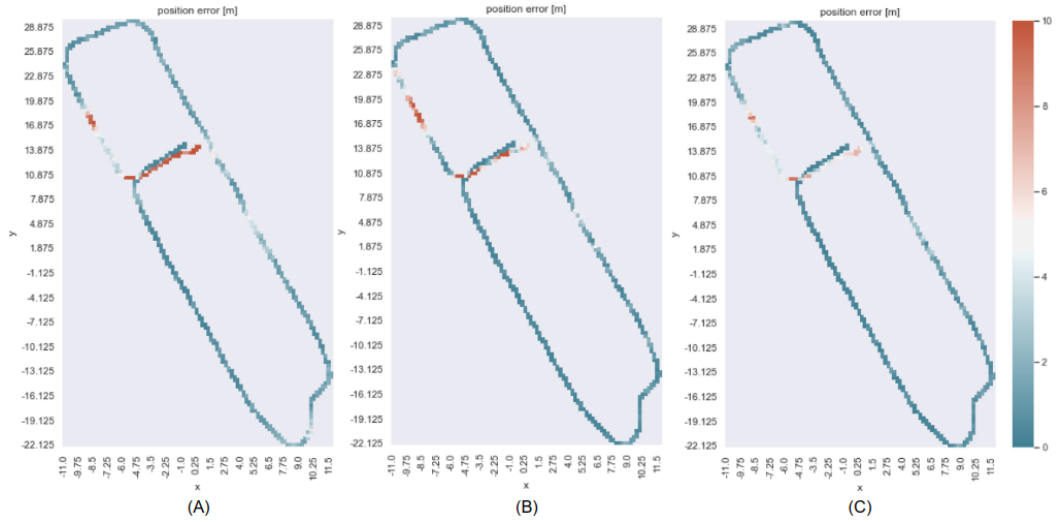


(a) Position error [m]

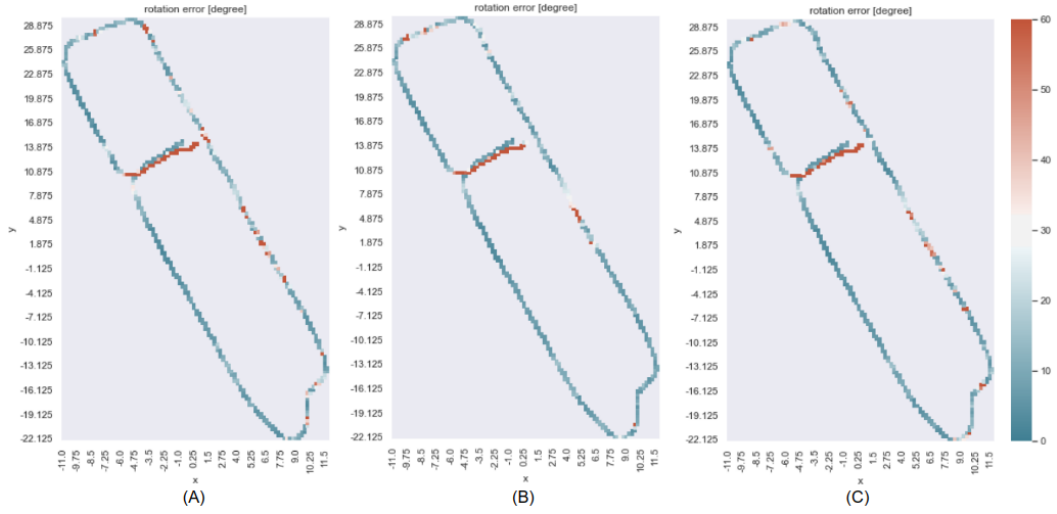


(b) Orientation error [°]

Figure A.2: Visualization of local errors when tested on *Light Change*, in case of the dataset with removal ratio 0.6



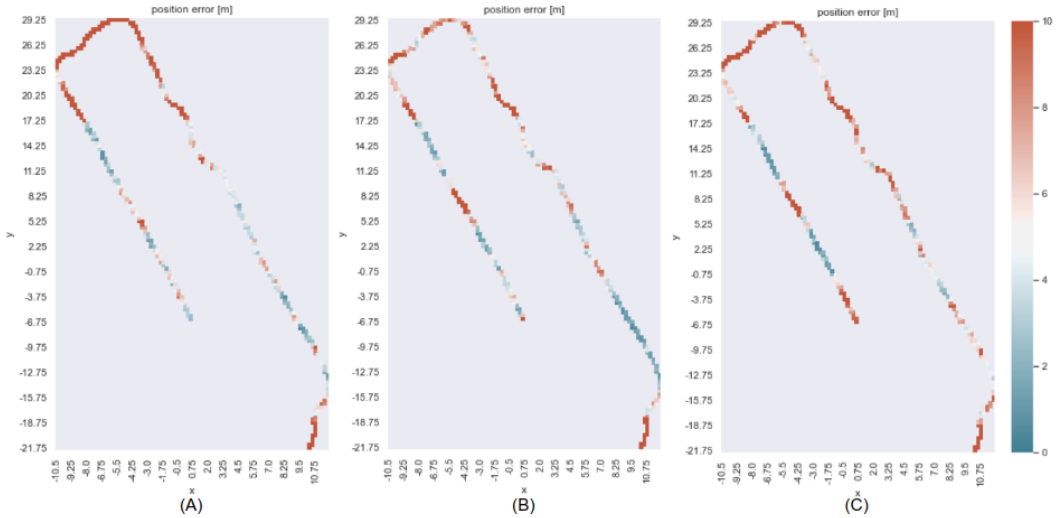
(a) Position error [m]



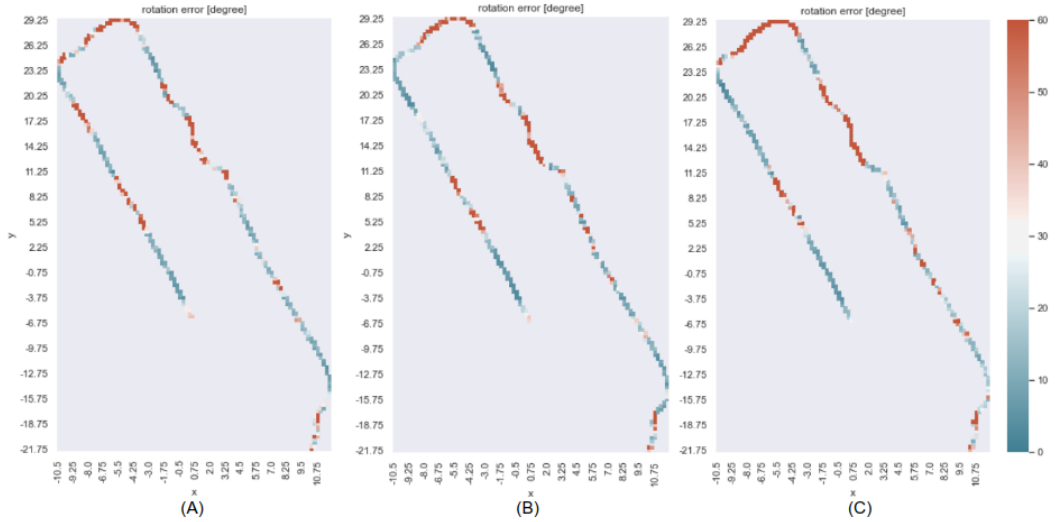
(b) Orientation error [°]

Figure A.3: Visualization of local errors when tested on *Light Change*, in case of the dataset with removal ratio 0.85

Appendix B. Pose Error Heatmaps for tests on *Open Lab*

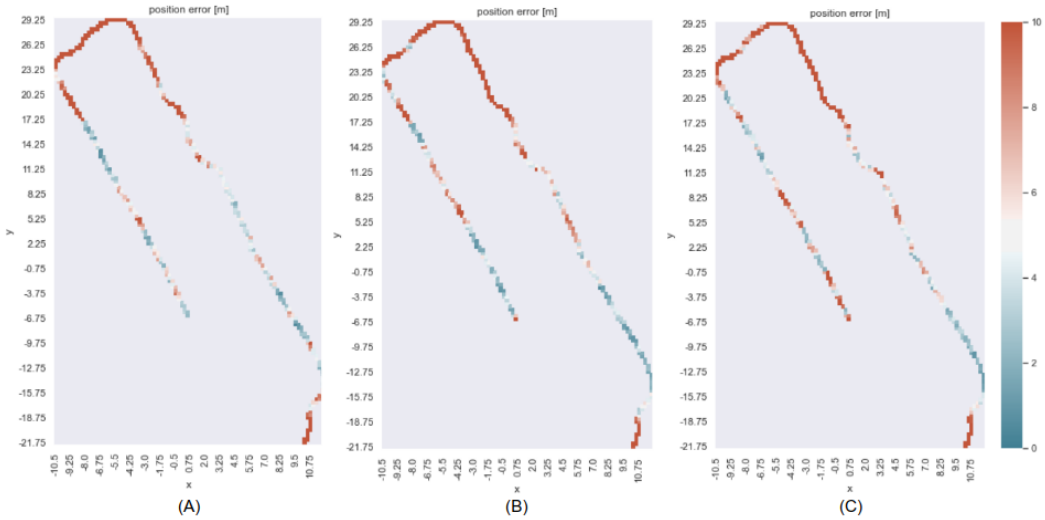


(a) Position error [m]

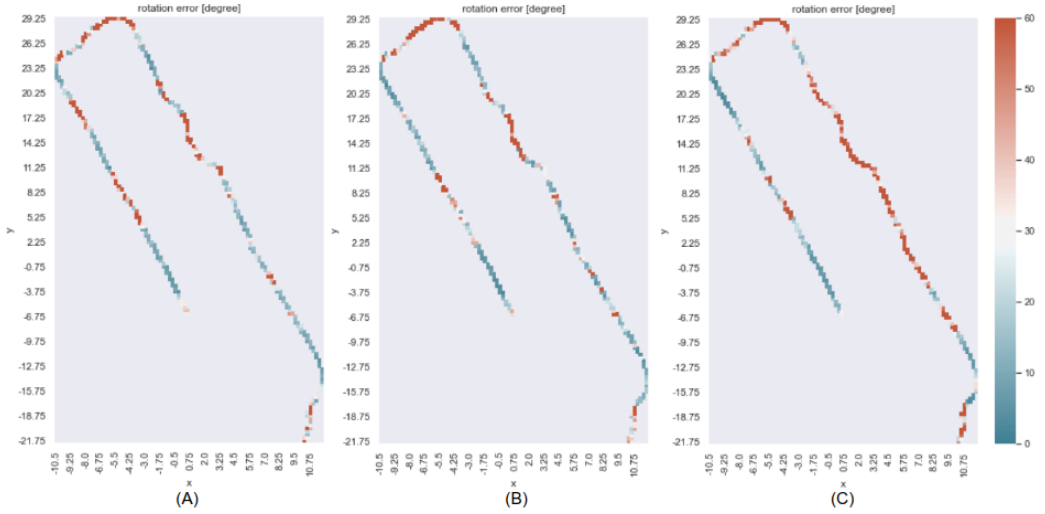


(b) Orientation error [°]

Figure B.1: Visualization of local errors when tested on *Open Lab*, in case of the dataset with removal ratio 0.2

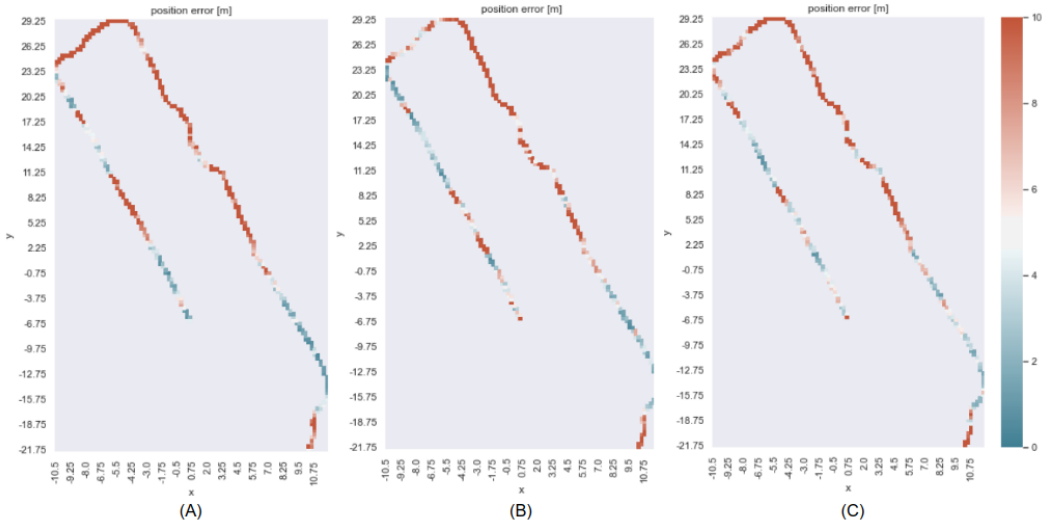


(a) Position error [m]

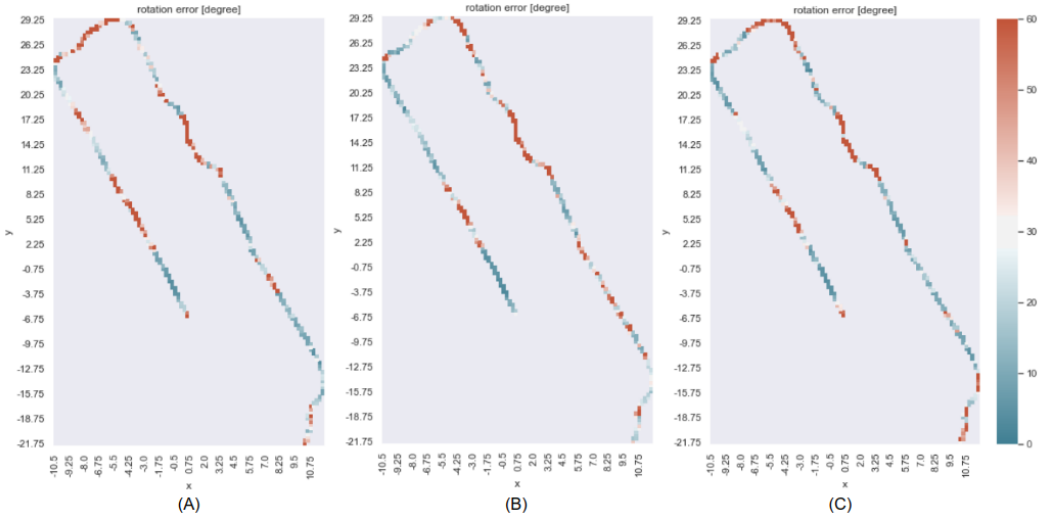


(b) Orientation error [°]

Figure B.2: Visualization of local errors when tested on *Open Lab*, in case of the dataset with removal ratio 0.6



(a) Position error [m]



(b) Orientation error [°]

Figure B.3: Visualization of local errors when tested on *Open Lab*, in case of the dataset with removal ratio 0.85