Hierarchical Joint Scene Coordinate Classification and Regression for Visual Localization

Xiaotian Li
Aalto University

Jakob Verbeek
Inria Grenoble

Juho Kannala
Aalto University

Abstract: Visual localization is pivotal to many applications in computer vision and robotics. To address single-image RGB localization, state-of-the-art feature-based methods solve the task by matching local descriptors between a query image and a pre-built 3D model. Recently, deep neural networks have been exploited to directly learn the mapping between raw pixels and 3D coordinates in the scene, and thus the matching is implicitly performed by the forward pass through the network. In this work, we present a new hierarchical joint classification-regression network to predict pixel scene coordinates in a coarse-to-fine manner from a single RGB image. The network consists of a series of output layers with each of them conditioned on the outputs of previous ones, where the final output layer regresses the coordinates and the others produce coarse location labels. Our experiments show that the proposed method outperforms the vanilla scene coordinate regression network and is more scalable to large environments. With data augmentation, it achieves the state-of-the-art single-image RGB localization performance on three benchmark datasets.

Keywords: Computer Vision, Deep Learning, Visual Localization

1 Introduction

Visual localization aims at estimating precise six degree-of-freedom (6-DoF) camera pose with respect to a known environment. It is a fundamental component that enables many intelligent autonomous systems and applications in computer vision and robotics, e.g., augmented reality, autonomous driving, or camera-based indoor localization for personal assistants. Commonly used visual localization methods rely on matching local visual descriptors. Correspondences are typically established between 2D interest points in the query and 3D points in the pre-built structure-from-motion model with nearest neighbor searches, and the 6-DoF camera pose of the query can then be computed from the correspondences [1].

Instead of explicitly establishing 2D-3D correspondences, recent works have proposed to leverage the capability of CNNs to directly regress dense 3D coordinates from an image [2, 3]. In this way, correspondences between 2D points in the image and 3D points in the scene can be obtained without explicit matching, and thus no feature detectors and descriptors are needed. In addition, no 3D maps are required at test time. Therefore, these CNN-based methods have the potential for deployment on mobile devices. Moreover, experimental results have shown that these methods achieve better localization performance on small-scale benchmark datasets compared to the state-of-the-art feature-based methods [3, 4].

Scene coordinate regression networks are typically designed to have a limited receptive field, i.e., only a small local image patch is considered for each scene coordinate prediction. This allows the network to generalize well from limited training data, since local patch appearance is more stable w.r.t. viewpoint change. On the other hand, a limited receptive field size can lead to ambiguous patterns in the scene, especially in large-scale environments. Due to the ambiguities, it is harder for the network to accurately model the regression problem, resulting in inferior performance at test time. Using larger receptive field sizes, up to the full image, to regress the coordinates can mitigate the issues caused by ambiguities. This, however, has been shown to be prone to overfitting the larger input patterns in the case of limited training data, even if data augmentation alleviates this problem to some extent [5].
In contrast, in this work we overcome the ambiguities due to small receptive fields by conditioning on discrete location labels for the pixels. The labels are obtained by a coarse quantization of the 3D space. The pixel location labels are obtained using a classification network, which can more easily deal with the location ambiguity since it is trained using the cross-entropy classification loss which permits a multi-modal prediction in 3D space. Our model allows for several classification layers, using progressively finer location labels, obtained through a hierarchical clustering of the ground-truth 3D point-cloud data. Our hierarchical coarse-to-fine architecture is implemented using conditioning layers that are related to the FiLM architecture [6], resulting in a compact model.

We validate our approach with an ablative study where we compare it to a baseline which uses the same back-bone architecture as our model, but lacks the hierarchical coarse-to-fine joint classification-regression structure. We present results on three datasets used in previous work [3]: 7-Scenes [7], 12-Scenes [8], and Cambridge Landmarks [9]. Our approach shows consistently better performance and achieves state-of-the-art results when trained with data augmentation. Moreover, by compiling the 7-Scenes dataset into a single large virtual scene, we show that our approach scales more robustly to larger environments.

In summary, the contributions of this paper are as follows:

- We introduce a new joint classification-regression architecture for scene coordinate prediction.
- We show that the joint coarse-to-fine hierarchy improves the performance and scalability over vanilla scene coordinate regression network.
- We show that our method achieves state-of-the-art results for single-image RGB localization on three benchmark datasets when trained with data augmentation.

2 Related Work

Visual localization aims at predicting 6-DoF camera pose for a given query image. To obtain precise 6-DoF camera pose, visual localization methods are typically structure-based, i.e., they rely on 2D-3D correspondences between 2D image positions and 3D scene coordinates. With the established 2D-3D correspondences, a RANSAC [10] optimization scheme is responsible for producing the final pose estimation. The correspondences are typically obtained by matching local features such as SIFT [11], and many matching and filtering techniques have been proposed, which enable efficient and robust city-scale localization [1, 12, 13].

Image retrieval can also be used for visual localization [14]. The pose of the query image can be directly approximated by the most similar retrieved database image. Since compact image-level descriptors are used for matching, image retrieval methods can scale to very large environments. The retrieval methods can be combined with structure-based methods [15, 16, 17, 18] or relative pose estimation [19, 20] to predict precise poses. Typically, the retrieval step helps restrict the search space and leads to faster and more accurate localization.
In recent years, learning-based localization approaches have been explored. One popular direction is to replace the entire localization pipeline with a single neural network. PoseNet [9] and its variants [21, 22, 23, 24] directly regress the camera pose from a query image. However, Sattler et al. [25] have pointed out that direct pose regression is more closely related to pose approximation via image retrieval than to accurate pose estimation via 3D structure. Therefore, these approaches are still outperformed by structure-based methods. By fusing estimated pose information from the previous frame, [26, 27] achieve significantly better performance. However, they require sequences of images rather than single images.

Instead of learning the entire pipeline, scene coordinate regression based methods learn the first stage of the pipeline in the structure-based approaches. Namely, either a random forest [7, 28, 29, 30, 31, 32, 33, 34, 35] or a neural network [2, 3, 4, 5, 32, 36] is trained to directly predict 3D scene coordinates for the pixels and thus the 2D-3D correspondences are established. These methods do not explicitly rely on feature detection, description and matching, and are able to provide correspondences densely. They are more accurate than the feature-based method at small and medium scale, but do not scale well to larger scenes. In order to generalize well from novel viewpoints, these methods typically rely on only local image patch to produce the scene coordinate predictions. However, this makes the training difficult, especially when the scale of the scene is large. In this work, we introduce the element-wise conditioning layer to modulate the intermediate feature maps of the regression network using coarse location information. We show this leads to better localization performance and makes scene coordinate regression more scalable.

In [30], a joint classification-regression forest is trained to predict scene coordinates. However, the forest predicts scene IDs and coordinates directly rather than doing joint classification-regression in a hierarchical coarse-to-fine manner. Our approach is also related to the work of Rogez et al. [37, 38] which proposed a classification-regression approach for human pose estimation from single images. Similar to our experience, they found an initial classification layer followed by a class-conditioned regression layer to break the complexity of a original complicated regression problem. Our work differs from theirs in the use of our FiLM-like conditioning layers, and the use of a classification hierarchy.

3 Hierarchical Joint Classification-Regression Scene Coordinate Prediction

In this section we describe our hierarchical coarse-to-fine classification-regression approach for scene coordinate prediction. Figure 1 gives a schematic overview of our approach. Note that we address single-image RGB camera localization [3] in this work, rather than using RGB-D information [7, 28, 29, 31, 34, 35] or a sequence of images [26, 27] during inference.

Hierarchical location labels. We hierarchically partition the ground-truth 3D point-cloud data with k-means. In this way, in addition to the ground-truth 3D scene coordinates, each pixel in a training image is also associated with a number of labels, from coarse to fine, obtained at different levels of the clustering tree. For each level, except the root, our network has a corresponding classification output layer which predicts for each pixel these discrete location labels.

Our network has a final output layer that regresses the continuous 3D scene coordinates for the pixels, generating putative 2D-3D matches. Instead of directly regressing the absolute coordinates, we regress the relative positions to the cluster centers at the finest level, which accelerates convergence of network training. Conditioned on the preceding label map, the output layers have deceasing sizes of receptive field. Note that for the conditioning information, we use ground truth label maps during training and predicted at test time. Coarser location labels are obtained with larger receptive fields that are more robust to ambiguities, and the final regression layer has a relatively limited receptive field size and is more stable to viewpoint changes. Note that at test time the receptive field size of the regression layer is also large since it is influenced by the predicted location labels.

Conditioning layers. To make use of the discrete location labels information predicted by the network at coarser levels, these predictions should be fed back to the finer levels. Inspired by the Feature-wise Linear Modulation (FiLM) conditioning method [6], we introduce several conditioning layers just before each of the output layers. A conditioning parameter generator takes the predicted label map \( \ell \) as input, outputs a set of scaling and shifting parameters \( \gamma(\ell) \) and \( \beta(\ell) \), and these parameters are fed into the conditioning layer to apply linear transformation to the input feature.
map. Unlike FiLM layers, however, which does channel-wise modulation, our conditioning layer performs linear modulation in a per-element manner, i.e., element-wise multiplication and addition as shown in Figure 2 (right). Therefore, instead of vectors, the output parameters $\gamma(\ell)$ and $\beta(\ell)$ from a generator are feature maps of the same dimensions as the input feature map to the corresponding conditioning layer. More formally, give the input feature map $x$, the scaling and shifting parameters $\gamma(\ell)$ and $\beta(\ell)$ from a generator are feature maps of the same dimensions as the input feature map to the corresponding conditioning layer. More formally, give the input feature map $x$, the scaling and shifting parameters $\gamma(\ell)$ and $\beta(\ell)$ the linear modulation can be written as:

$$f(x, \ell) = \gamma(\ell) \odot x + \beta(\ell),$$

where $\odot$ denotes the Hadamard product. In addition, the generators consist of only 1x1 convolutional layers so that each pixel is only conditioned on the label of its own. As a non-linearity, we add a ELU activation [39] after the feature modulation.

**Network architecture.** Our overall network architecture is shown in Figure 2 (left). In our experiments we use 3-level hierarchy for all the datasets, i.e., our network has two classification output layers and one regression output layer. As mentioned above, the space of the scene is partitioned into regions which are hierarchically organized in a tree structure. The first classification branch outputs the labels of the coarse regions, and the second one predicts the labels of the finer subregions. We use strided convolution, upconvolution and dilated convolution for the two classification branches to enlarge the size of receptive field, while preserving the output resolution. All the layers after the conditioning layers have kernel size of 1 such that the conditioning information of a pixel affects only itself. More details on the architecture are provided in the supplementary material.

**Loss function.** Our network predicts location labels and regresses scene coordinates at same time. Therefore, we need both regression loss and classification loss during training. For the regression task, we minimize the Euclidean distance between the predicted scene coordinates $\hat{y}$ and ground truth scene coordinates $y$,

$$\mathcal{L}_r = \sum_i \|y_i - \hat{y}_i\|,$$

where $i$ ranges over the pixels in the image. For the classification task, we use cross-entropy loss at both levels, i.e.

$$\mathcal{L}^j_c = -\sum_i \ell^j_i \top \log \hat{\ell}^j_i,$$

where $\ell^j_i$ denotes the one-hot coding of the ground-truth label of pixel $i$ at level $j$, and $\hat{\ell}^j_i$ denotes the vector of predicted label probabilities for the same label, and the log is applied element wise. The final loss function is given by

$$\mathcal{L} = w_1 \mathcal{L}^1_c + w_2 \mathcal{L}^2_c + w_3 \mathcal{L}_r$$
| 7-Scenes | DSAC++ [3] | Baseline | Ours | 12-Scenes | DSAC++ [3] | Baseline | Ours |
|---------|------------|----------|------|-----------|------------|----------|------|
| Chess   | 97.1%      | 95.5%    | 97.5%| Kitchen-1 | 100%       | 100%     | 100% |
| Fire    | 89.6%      | 94.7%    | 96.7%| Living-1  | 100%       | 100%     | 100% |
| Heads   | 92.4%      | 97.3%    | 100% | Bed       | 99.5%      | 100%     | 100% |
| Office  | 86.6%      | 81.4%    | 86.6%| Kitchen-2 | 99.5%      | 100%     | 100% |
| Pumpkin | 59%        | 57.8%    | 59.9%| Living-2  | 100%       | 100%     | 100% |
| Kitchen | 66.6%      | 56.3%    | 65.5%| Luke      | 95.5%      | 93.4%    | 96.3%|
| Stairs  | 29.3%      | 69.3%    | 87.2%| Gates 362 | 100%       | 100%     | 100% |
| Complete| 76.1%      | 74.7%    | 80.5%| Gates 381 | 96.8%      | 98.8%    | 99.9%|

Table 1: The percentage of accurately localized test images (error < 5cm, 5°) for 7-Scenes and 12-Scenes, and median errors (cm, °) for Cambridge Landmarks. ‘Complete’ refers to the percentage among all test images of all scenes.

where \( w_1, w_2, w_3 \) are weights for the loss terms. Details on the weights and training procedure are provided in the supplementary material.

4 Experiments

In this section, we present our experimental setup and evaluation results on three standard visual localization datasets.

4.1 Datasets and Experimental Setup

We use three standard benchmark datasets for visual localization. The 7-Scenes [7] dataset is a widely used RGB-D dataset that contains seven indoor scenes. Sequences of RGB-D images of the scenes are recorded by a KinectV1. Ground truth poses and dense 3D models obtained are also provided. 12-Scenes [8] is another indoor RGB-D dataset. It is composed of twelve rooms captured with a Structure.io depth sensor and an iPad color camera, and ground truth poses are provided along with the RGB-D images. The recorded environments are significantly larger compared to 7-Scenes.

Cambridge Landmarks [9] is an outdoor RGB visual localization dataset. It consists of RGB images of six scenes captured using a Google LG Nexus 5 smartphone. Ground truth poses and sparse 3D reconstructions generated with structure from motion are provided. In addition to these three datasets, we synthesize a larger scene based on 7-Scenes by placing all the seven scenes into a single coordinate system. This large integrated 7-Scenes dataset is denoted by i7-Scenes.

For the Cambridge Landmarks, we report median pose accuracy as in the previous works. Following [3, 4], we do not include the Street scene, since the 3D reconstruction of this scene has rather poor quality that hampers performance. For the 7-Scenes and 12-Scenes and i7-Scenes, we report the percentage of the test images with error below 5cm and 5°, which gives more information about the localization performance.

To generate the ground truth coarse location labels, we run hierarchical k-means clustering on dense point cloud models. For all the scenes, unless stated otherwise, the branching factor is set to 25. For the i7-Scenes, we simply use the labels generated for the 7-Scenes. For our network, we use the same VGG-style [40] architecture as DSAC++ [3] as the base regression network, except we use ELU activation [39] instead of ReLU [41]. The output neurons of the regression layer, the first and second classification layer have a receptive field size of 73×73, 185×185, and 409×409, respectively. To show the advantage of the proposed method, we create a baseline network without the classification branches, the conditioning layers and the generators (i.e., with the base regression network only), and train it with the Euclidean loss term only. Unless specified otherwise, we perform affine data augmentation with additive brightness changes during training. For pose estimation,
we follow [3], and use the same PnP-RANSAC algorithm with the same hyperparameter settings. Further details about the architecture, training and other settings can be found in the supplementary material.

### 4.2 Evaluation Results

To validate our model, we compare it in Table 1 with the baseline model, as well as DSAC++ [3] which currently has the state-of-the-art results for single-image RGB localization on all three datasets. We report the numbers given in the original DSAC++ paper. A more complete comparison to other recent methods (with worse results) is provided in the supplementary material. In general, methods that exploits extra depth information or sequences of images provide better localization performance. However, they are not directly comparable to our method, and thus we do not compare to those methods in this work. Overall, our approach yields excellent results. Compared to the baseline, our approach provides consistently better localization performance on all the scenes across the three datasets. During training, we also observed consistently lower regression training error. This indicates that the additional information from the conditioning layers makes the training of the regression branch easier, thus leading to better performance at test time. Although trained with data augmentation, the baseline method is still outperformed by DSAC++ on 7-Scenes and Cambridge Landmarks, which is trained without data augmentation. This is not surprising, given that the two additional training steps proposed in DSAC++ (optimizing the reprojection error and end-to-end training) boost the accuracy (from 72.4% to 76.1% on 7-Scenes [3]). In fact, despite some minor implementation differences, the baseline without data augmentation is supposed to be identical to DSAC++ trained without the two additional training steps. Below, in Section 4.3 and Table 3 we provide additional results to assess the impact of data augmentation in more detail. As we can see, without data augmentation, the baseline method provides similar performance on 7-Scenes (72.6%). Remarkably, with data augmentation, our approach can already outperform DSAC++ on all three datasets. Potentially our approach could also benefit from the two additional training steps, but we leave this for future work. Since the additional branches and layers do not add significant overhead, our approach has nearly the same run-time as DSAC++ at test time.

The individual scenes from 7-Scenes dataset all have very limited physical extent. To go beyond such small environments we use our i7-Scenes dataset, which integrates all the scenes from 7-Scenes in a single coordinate system. In Table 2 we compare our method to the baseline as well as DSAC++ in this setting. We use the publicly available implementation to produce the numbers for DSAC++. Here, DSAC++$_{init}$ denotes that the network is trained with only the first step of the full DSAC++ 3-step training procedure (optimizing the Euclidean error). We see that the localization performance of the baseline decreases dramatically when trained on all the scenes together compared to trained and tested on each of the scenes individually, cf. Table 1. Our method is much more robust to this increase in the environment size and significantly outperforms the baseline, underlining the importance of the proposed approach when the environment is large and potentially contains more ambiguities.

| i7-Scenes | Baseline | DSAC++$_{init}$ [3] | DSAC++ [3] | Ours | Ours-512 | Ours-512+ |
|-----------|---------|---------------------|-----------|------|---------|---------|
| Chess     | 87.4%   | 89.5%               | 95.8%     | 95.5%| 90.8%   | 94%     |
| Fire      | 53.9%   | 45.5%               | 68.1%     | 87.6%| 74.6%   | 83.6%   |
| Heads     | 18.1%   | 14.4%               | 33.2%     | 90.7%| 84.3%   | 91%     |
| Office    | 50.3%   | 59.4%               | 78.4%     | 82.3%| 74.6%   | 82.5%   |
| Pumpkin   | 44.8%   | 40.2%               | 53.1%     | 57.6%| 54.9%   | 58.4%   |
| Kitchen   | 23.3%   | 43.5%               | 61.2%     | 56.7%| 43%     | 52%     |
| Stairs    | 4%      | 2.2%                | 2.7%      | 54.1%| 36.3%   | 62.7%   |
| Complete  | 41.9%   | 48.4%               | 64.1%     | 72.9%| 63.2%   | 71.5%   |

Table 2: The percentage of accurately localized test images on our integrated 7-Scenes dataset, where the 7 scenes are placed into a single coordinate system and a single model is trained for localization. Ours-512 and Ours-512+ refer to compact variants of our full model, see text for details.
DSAC++ also significantly outperforms the baseline on i7-Scenes. According to the results, the two additional DSAC++ training steps are even more helpful to improving the localization performance when the environment is large, as DSAC++ exceeds the accuracy of DSAC++init by 15.7%. By inspecting the results, it seems that the two additional training steps proposed in DSAC++ can make the already good scene coordinate predictions even better, and deteriorate the predictions for local image patches which are difficult. This makes it easier for the RANSAC algorithm to filter out the outliers and find the correct predictions to fit the model, thus leading to better performance. However, this does not change the fact that the receptive field of the network is limited. Thus, the hardest cases remain hard since there are not enough good correspondences due to the ambiguities. Our method still has overall better performance than DSAC++, and clearly outperforms it on the hardest scenes. For example, in Figure 3, we show that the Heads scene is hard since it is difficult to distinguish its test frames from the training frames of the Office scene. Stairs is also hard since it already contains many repetitive patterns. The baseline and DSAC++ can easily fail in these cases, while our method is able to provide reasonable performance, since without global image context, it is extremely difficult for the network to produce enough precise scene coordinate predictions. As mentioned above, our method could also benefit from the two additional training steps. This could be addressed in the future work.

In the last two columns of Table 2 we consider the effect of reducing the number of channels in the last two convolutional layers before the regression output layer from 4,096 to 512, the model size decreases significantly but the regression branch becomes less discriminative and harder to train. This leads to significantly decreased accuracy in column “Ours-512”. We find that by adding more conditioning layers at earlier stage of the network and using less shared layers between the regression and classification branches, denoted “Ours-512+”, the model size can be reduced without much loss in accuracy.

### 4.3 Detailed Analysis

**Data augmentation.** We perform affine transformations to the images with additive brightness changes as data augmentation during training. In general, this improves the generalization capability of the network and makes it more robust to lighting and viewpoint changes. However, this could also increase the ambiguity level of the training data and make the network training more difficult. Moreover, since the ground truth coordinates and poses are not perfect, data augmentation may also increase the error at test time in some rare cases. Our results on 7-Scenes without using data augmentation are given in Table 3. Our method still has better results than the baseline without data augmentation. According to the numbers in Table 1, our method benefits more from data augmentation (+4.2%) than the baseline (+2.1%).

| 7-Scenes | Chess | Fire | Heads | Office | Pumpkin | Kitchen | Stairs | Complete |
|----------|-------|------|-------|--------|---------|---------|--------|----------|
| Baseline | 94.9% | 81.5%| 84.7% | 83.9%  | 56.7%   | 62.9%   | 33.6%  | 72.6%    |
| Ours     | 96.6% | 84.9%| 95.5% | 86.3%  | 61.8%   | 66.5%   | 37.6%  | 76.3%    |

Table 3: Results without using the data augmentation.

**Hierarchy.** For all the previous experiments, we use a network with two levels of classification layers for predicting gradually finer location labels. Here we experiment with only one classification branch. We use either the first (Ours-C1) or the second (Ours-C2) branch to predict the finest labels.
Note that we have different size of receptive field at different levels and typically we want to use more global context. The results in Table 4 show that the hierarchical architecture achieves overall better results than the single branch alternatives. We did not explore more hierarchical levels in this work, although it might be helpful for localization in larger environments.

|                      | Ours | Ours-C2 | Ours-C1 |
|----------------------|------|---------|---------|
| 7-Scenes             | 80.5%| 80.3%   | 80.2%   |
| i7-Scenes            | 72.9%| 71.8%   | 71.8%   |

Table 4: Results with (Ours) and without hierarchical classification.

Directly using global information. In Table 5 we show that direct regression of scene coordinates with large receptive field size is also problematic. We use the same baseline network (without classification branches), but now with dilated convolution such that the receptive field size of the output neurons is much larger (409×409). Although the training loss decreases, the network does not generalize well at test time, which results in inferior localization performance.

|                      | Chess | Fire  | Heads | Office | Pumpkin | Redkitchen | Stairs | Complete |
|----------------------|-------|-------|-------|--------|---------|------------|--------|----------|
| 7-Scenes             | 81.7% | 65.5% | 85.7% | 63%    | 48.6%   | 55.1%      | 15.5%  | 60%      |
| i7-Scenes            | 69.5% | 38.5% | 32.6% | 39.7%  | 39.1%   | 43.8%      | 0%     | 41.4%    |

Table 5: Baseline results when dilated convolution is used to increase the receptive field.

Partition granularity. We show that the granularity of the clustering matters in Table 6. We experiment on different branching factors such that the numbers of leaf clusters differ. Although the accuracy varies with the number of leaf clusters, for all tested branching factors we observe overall better results than obtained with the baseline.

|                      | Chess | Fire  | Heads | Office | Pumpkin | Redkitchen | Kitchen | Stairs | Complete |
|----------------------|-------|-------|-------|--------|---------|------------|---------|--------|----------|
| 7-Scenes             |       |       |       |        |         |            |         |        |          |
| Ours-k25             | 97.5% | 96.7% | 100%  | 86.6%  | 59.9%   | 65.5%      | 87.2%   |        | 80.5%    |
| Ours-k9              | 97%   | 96.1% | 99.6% | 86.1%  | 59.5%   | 64.9%      | 77.1%   |        | 79.5%    |
| Ours-k49             | 97.7% | 96.9% | 100%  | 86.4%  | 60.5%   | 65.3%      | 89.2%   |        | 80.7%    |

Table 6: Results of our model with default branching factor 25, and with branching factors 9 and 49. In all cases we use two classification branches.

5 Conclusion

We have proposed a novel hierarchical coarse-to-fine classification-regression approach to scene coordinate prediction. Coarser localization labels are predicted with classification networks on larger image patches trained via cross-entropy, which allows the model to deal with ambiguous local appearances during both training and inference. Accurate regression of the continuous coordinate regression is facilitated by conditioning on the coarse labels. Our experiments on the 7-Scenes, 12-Scenes and Cambridge Landmarks datasets show that the hierarchical classification-regression approach leads to more accurate camera re-localization predictions than the previous regression-only approach, achieving state-of-the-art results for single-image RGB localization when trained with data augmentation. More over, our experiments with our “integrated 7-Scenes” dataset (which combines the 7-Scenes dataset into a single large space), shows that our approach scales more robustly to larger environments compared to the previous scene coordinate regression based approaches.

Despite our encouraging results, our approach has some limitations. First, it requires high quality depth maps or a dense 3D scene reconstruction, which might be difficult to obtain. To address this we want to explore the use of self-supervised reprojection losses [3, 4]. Second, we train our model in a “teacher forced” manner, i.e. during training we use the ground-truth location labels as input to the conditioning layers. At test time, however, we use the predicted location labels as input instead.
Note that during training the model has never seen predicted labels, and the error patterns therein. Scheduled sampling techniques used to train language generation models [42] might prove useful also in our context to remedy this exposure bias. Lastly, for robust localization in challenging large-scale environments with significant appearance changes between training and test images and global ambiguities, pure scene specific training of the network might not be sufficient. One could resort to trainable descriptors [43, 44] and modern image retrieval techniques [14] in future work.

References

[1] T. Sattler, B. Leibe, and L. Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *PAMI*, 39(9):1744–1756, 2017.

[2] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother. DSAC - Differentiable RANSAC for camera localization. In *CVPR*, 2017.

[3] E. Brachmann and C. Rother. Learning less is more - 6D camera localization via 3D surface regression. In *CVPR*, 2018.

[4] X. Li, J. Ylioinas, J. Verbeek, and J. Kannala. Scene coordinate regression with angle-based reprojection loss for camera relocalization. In *ECCV Workshops*, 2018.

[5] X. Li, J. Ylioinas, and J. Kannala. Full-frame scene coordinate regression for image-based localization. In *RSS*, 2018.

[6] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville. FiLM: Visual reasoning with a general conditioning layer. In *AAAI*, 2018.

[7] 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 *CVPR*, 2013.

[8] J. Valentin, A. Dai, M. Nießner, P. Kohli, P. Torr, S. Izadi, and C. Keskin. Learning to navigate the energy landscape. In *3DV*, 2016.

[9] A. Kendall, M. Grimes, and R. Cipolla. PoseNet: A convolutional network for real-time 6-DOF camera relocalization. In *ICCV*, 2015.

[10] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *CACM*, 24(6):381–395, 1981.

[11] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[12] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt. Scalable 6-DOF localization on mobile devices. In *ECCV*, 2014.

[13] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson. City-scale localization for cameras with known vertical direction. *PAMI*, 39(7):1455–1461, 2017.

[14] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*, 2016.

[15] T. Sattler, A. Torii, J. Sivic, M. Pollefeys, H. Taira, M. Okutomi, and T. Pajdla. Are large-scale 3D models really necessary for accurate visual localization? In *CVPR*, 2017.

[16] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii. InLoc: Indoor visual localization with dense matching and view synthesis. In *CVPR*, 2018.

[17] P.-E. Sarlin, F. Debraine, M. Dymczyk, and R. Siegwart. Leveraging deep visual descriptors for hierarchical efficient localization. In *CoRL*, 2018.

[18] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019.

[19] Z. Laskar, I. Melekhov, S. Kalia, and J. Kannala. Camera relocalization by computing pairwise relative poses using convolutional neural network. In *ICCV Workshops*, 2017.
[20] V. Balntas, S. Li, and V. A. Prisacariu. RelocNet: Continuous metric learning relocalisation using neural nets. In *ECCV*, 2018.

[21] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu. Image-based localization using hourglass networks. In *ICCV Workshops*, 2017.

[22] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers. Image-based localization using LSTMs for structured feature correlation. In *ICCV*, 2017.

[23] A. Kendall and R. Cipolla. Geometric loss functions for camera pose regression with deep learning. In *CVPR*, 2017.

[24] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-aware learning of maps for camera localization. In *CVPR*, 2018.

[25] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixe. Understanding the limitations of CNN-based absolute camera pose regression. In *CVPR*, 2019.

[26] A. Valada, N. Radwan, and W. Burgard. Deep auxiliary learning for visual localization and odometry. In *ICRA*, 2018.

[27] N. Radwan, A. Valada, and W. Burgard. VLocNet++: Deep multitask learning for semantic visual localization and odometry. *RA-L*, 3(4):4407–4414, 2018.

[28] A. Guzmán-Rivera, P. Kohli, B. Glocker, J. Shotton, T. Sharp, A. W. Fitzgibbon, and S. Izadi. Multi-output learning for camera relocalization. In *CVPR*, 2014.

[29] J. P. C. Valentin, M. Nießner, J. Shotton, A. W. Fitzgibbon, S. Izadi, and P. H. S. Torr. Exploiting uncertainty in regression forests for accurate camera relocalization. In *CVPR*, 2015.

[30] E. Brachmann, F. Michel, A. Krull, M. Y. Yang, S. Gumhold, and C. Rother. Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image. In *CVPR*, 2016.

[31] T. Cavallari, S. Golodetz, N. A. Lord, J. P. C. Valentin, L. di Stefano, and P. H. S. Torr. On-the-fly adaptation of regression forests for online camera relocalisation. In *CVPR*, 2017.

[32] D. Massiceti, A. Krull, E. Brachmann, C. Rother, and P. Torr. Random forests versus neural networks - What’s best for camera localization? In *ICRA*, 2017.

[33] L. Meng, J. Chen, F. Tung, J. Little, J. Valentin, and C. de Silva. Backtracking regression forests for accurate camera relocalization. In *IROS*, 2017.

[34] L. Meng, F. Tung, J. J. Little, J. Valentin, and C. W. de Silva. Exploiting points and lines in regression forests for RGB-D camera relocalization. In *IROS*, 2018.

[35] T. Cavallari, S. Golodetz, N. Lord, J. Valentin, V. Prisacariu, L. Di Stefano, and P. H. Torr. Real-time RGB-D camera pose estimation in novel scenes using a relocalisation cascade. *PAMI*, 2019.

[36] M. Bui, S. Albarqouni, S. Ilic, and N. Navab. Scene coordinate and correspondence learning for image-based localization. In *BMVC*, 2018.

[37] G. Rogez, P. Weinaepfel, and C. Schmid. LCR-Net: Localization-classification-regression for human pose. In *CVPR*, 2017.

[38] G. Rogez, P. Weinaepfel, and C. Schmid. LCR-Net++: Multi-person 2D and 3D pose detection in natural images. *PAMI*, 2019.

[39] D. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). In *ICLR*, 2016.

[40] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, 2014.

[41] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.
[42] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In NeurIPS, 2015.

[43] D. DeTone, T. Malisiewicz, and A. Rabinovich. SuperPoint: Self-supervised interest point detection and description. In CVPR Workshops, 2018.

[44] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. D2-Net: A trainable CNN for joint detection and description of local features. In CVPR, 2019.