2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

Do We Really Need Scene-specific Pose Encoders?

Yoli Shavit
Toga Networks a Huawei Company
Email: yoli.shavit@huawei.com

Ron Ferens
Toga Networks a Huawei Company
Email: ron.ferens@huawei.com

***Abstract*—Visual pose regression models estimate the camera pose from a query image with a single forward pass. Current models learn pose encoding from an image using deep convolutional networks which are trained per scene. The resulting encoding is typically passed to a multi-layer perceptron in order to regress the pose. In this work, we propose that scene-specific pose encoders are not required for pose regression and that encodings trained for visual similarity can be used instead. In order to test our hypothesis, we take a shallow architecture of several fully connected layers and train it with pre-computed encodings from a generic image retrieval model. We find that these encodings are not only sufficient to regress the camera pose, but that, when provided to a branching fully connected architecture, a trained model can achieve competitive results and even surpass current *state-of-the-art* pose regressors in some cases. Moreover, we show that for outdoor localization, the proposed architecture is the only pose regressor, to date, consistently localizing in under 2 meters and 5 degrees.**

I. Introduction

Solving the pose estimation problem amounts to accurately predicting the position and orientation of an agent within a predefined coordinate system. The agent can be a robot, a handheld device or a vehicle. In visual pose estimation, we aim to solve this challenge solely using visual inputs generated by RGB and/or depth sensors mounted on the agent. That is, we assume that given enough visual clues, one can deduce the pose within the reference scene. State-of-the-art methods for visual pose estimation such as [1]–[3], rely on the availability of a structural model that can map 2D points in an image to 3D points in the reference scene (structure-based methods). Given 2D-2D matches between the query image and one or more train images, 2D-3D matches are obtained using the structural model, and passed to a Perspective-n-Point (PnP) algorithm in conjunction with RANSAC [4]. The resulting estimated pose/s can be further evaluated and re-ranked with a pose verification procedure [3], [5]. In the past few years, deep learning methods for regressing the camera pose from a single image have emerged [6]–[13]. While these absolute pose regressors (APRs) are typically an order of magnitude less accurate than their structure-based counterpart methods and require scene specific training, they offer a significant reduction in runtime (milliseconds versus seconds) and replace the heavy structure-based pipeline with a single forward pass.

A representative architecture for APRs consists of a convolutional pose encoder, based on a common backbone architectures such as GoogLeNet [14] or ResNet [15], followed by a multilayer perceptron (MLP) for separately regressing the position and orientation vectors (Fig. 1a). The pose encoder is commonly initialized using weights trained for object classification or place recognition and then fine tuned as part of the training process for encoding the training set of the scene of interest.

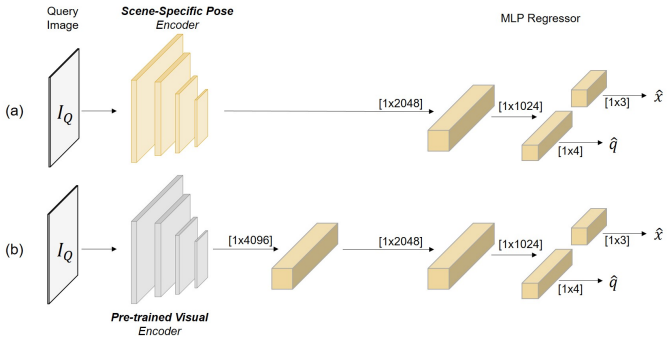

Fig. 1. End-to-end pose regression architectures. (a) A representative architecture of a pose repressor, using a scene-specific pose encoder and an MLP regressor. (b) An MLP architecture using generic scene-independent encodings, learnt for place recognition. Yellow coloring indicates layers that are updated during training.

Recently, a comprehensive study of APRs [16] framed camera pose regression methods as an approximation of image retrieval (IR), where the network learns some interpolation of memorized images and their poses from the training set. The authors further showed that APRs cannot consistently outperform an IR based method, which estimates the pose through interpolation of nearest neighbors' poses. Following this work, we ask whether scene- (or even dataset-) specific pose encoders are actually necessary for performing pose regression? To answer this question, we take an MLP architecture typically used by APRs (Fig. 1a). We train it with encodings generated using an IR model which was pre-trained to capture visual similarity for place recognition. As oppose to other APRs, here the encoder is not involved in the training process and the images' encodings, rather than the images themselves, are used as input for learning the camera pose regression task (Fig. 1b). We find that the resulting model achieves similar performance to a baseline pose regressor [6] (see Table I).

Secondly, we extend the above MLP with a branching architecture (Fig. 2), similarly to [10]. We name this architecture *IRPNet*, as it uses IR encodings for pose regression. In order to teach our model to regress the camera pose, the loss function needs to be optimized for both position

978-1-7281-8808-9/20/$31.00 ©2020 IEEE

2020 25th International Conference on Pattern Recognition (ICPR) | 978-1-7281-8808-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICPR48806.2021.9412225

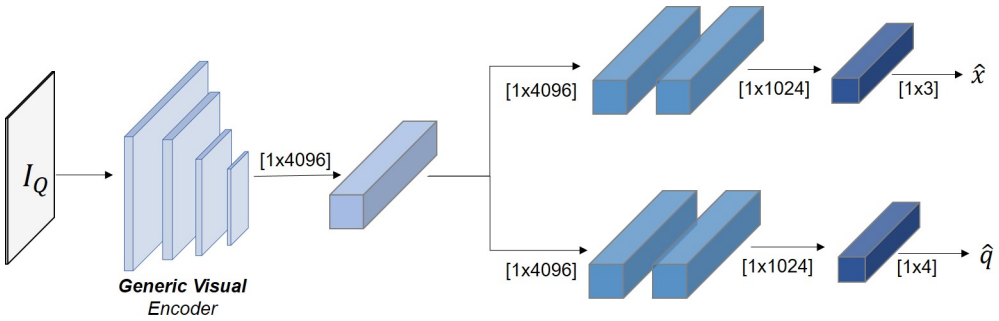

Fig. 2. IRPNet architecture. An MLP with two branches for separately regressing the position and orientation vectors

and orientation objectives, which differ in scale. This multi-objective tradeoff is challenging as the network needs to optimize for related, yet different tasks in nature. A common solution is to introduce a balancing factor, either manually [6] or through learning [8]. However, they still require dataset-specific fine-tuning [6] or initialization [19]. Instead, we train separately for orientation and position.

We evaluate IRPNet on outdoor and indoor datasets (Tables II and IV, respectively), commonly used to benchmark APRs. We find that IRPNet consistently localizes in under $2m$ and 5° on an outdoor benchmark. This lower bound cannot be guaranteed by any other APR, to date, or by an IR based interpolation method proposed by [16]. In addition, together with another *state-of-the-art* APR, IRPNet is the only model able to rank among the top-three APRs on both outdoor and indoor dataets (Tables III and V, respectively). IRPNet also offers several other advantages compared to APRs with pose encoders: it is faster to train (10x) and does not require scaling or cropping of the query image. It also facilitates lighter storage of the train dataset as image encodings can be stored instead of the images themselves. Compared to scene-specific encodings, an independent model allows extending the train set without retraining the encoder and elevates the need to pre-compute encodings more than once.

To summarize, our contributions are as follows:

- We propose a new paradigm for camera pose regression, where scene-specific encoders are not required and scene-agnostic IR encoders can be used instead.
- We test our hypothesis and show that encodings trained for visual similarity are sufficient for learning to regress the camera pose.
- We introduce IRPNet, a branching MLP architecture, which consistently rank among the top-three APRs on both outdoor and indoor datasets and is the only model to provide a consistent lower bound on orientation and position errors on an outdoor benchmark. In addition, the proposed method enables lighter database storage, is faster to train and does not require image pre-processing such as scaling or cropping.

II. Related Work

A. Absolute Pose Regression

Visual pose regression was first suggested by [6]. The proposed architecture, named PoseNet, was based on a GoogLeNet backbone where classifiers were replaced with fully-connected (FC) layers to regress the absolute pose of the input image. The model was trained to minimize a Pose Loss function for an image I:

$$L(I) = s_x L_x(I) + s_q L_q(I) \tag{1}$$

$L_x(I)$, the Position Loss, is the Euclidean distance between the ground truth and estimated position vectors: $L_x(I) = ||\hat{x} - x||_2$. $L_q(I)$, the Orientation Loss, is the Euclidean distance between the ground truth and estimated unit vector quaternions: $L_q(I) = ||\hat{q} - \frac{q}{||q||}||_2$. The hyperparameters s_x and s_q control the tradeoff between the two losses. In the original work, s_x was set to 1 and s_q (referred to as β) was fine-tuned for each scene and differed in scale between outdoor and indoor scenes.

The proposed layout of a deep convolutional encoder followed by an MLP regressor, was soon adopted as a baseline for further improvements. Different variations to the encoder and MLP architectures were proposed in order to reduce overfitting [7]–[9], [11], [12]. For example, replacing the GoogLeNet encoder with a ResNet encoder [11] or duplicating the MLP components to form a branching architecture [10], [12]. In addition, extensions to the loss function were proposed in order to better learn the balance between the orientation and position objectives [8] or to incorporate other modalities [17]. For a detailed review of architecture families and losses, used for camera pose regression, see [18].

In terms of accuracy, APRs are currently inferior to structure-based methods [16]. However, they only require an RGB input, have a small memory signature and an order of magnitude shorter inference time [6]. More recently, researchers have proposed to jointly learn absolute pose regression with other tasks such as relative pose regression (RPR, finding the relative motion between two images) and semantic segmentation [19], [20]. Learning these additional auxiliary tasks led to a significant improvement in performance over other pose regressors, matching classical *state-of-the-art*

pipelines in some cases. However, the learning was designed for a sequential workflow (relying on the previous pose to predict the current pose), making the proposed approach less suitable for re-localization scenarios.

In order to avoid the need to train a model per scene, some methods focused on improving the generalization of pose regressors [21]–[23]. These models were first trained offline for RPR on a reference dataset. The trained encoder was then used to fetch the nearest neighbor/s in order to estimate the relative motion to the query. Using the known pose of the neighbor/s and the estimated relative motion, the pose of the query was finally estimated. While RPR-based methods improved generalization of absolute pose regression, their pose encoders were still specific to a given dataset and the generated encodings could not be transfered between different datasets. In addition, generalization either came on the expense of accuracy [21], [22], or when high accuracy was achieved, it was only evaluated on a small scale indoor dataset [23].

B. Image Retrieval

Image retrieval methods rely on global image representations to efficiently index and fetch images. These global descriptors can be computed by aggregating local features using methods such as VLAD [24] and Fisher Vector [25]. More recently, deep convolutional neural networks (CNNs) were proposed as a surrogate for aggregation methods [26]–[28]. For example, NetVLAD [26], fine-tunes a pretrained CNN encoder, followed by a learnt VLAD component (convolution and soft assignment), PCA and whitening. Learned global descriptors were shown to be useful for place recognition, achieving *state-of-the-art* localization [26]–[28].

In the context of visual pose estimation, image retrieval methods can provide a rough estimate by using the known poses of the query's nearest neighbors. For example, by taking the pose of the closest neighbor. As part of studying the properties of APRs, a recent work [16] proposed an IR based baseline, where the pose of a query image is estimated with a weighted sum of its nearest neighbors's poses. The weighting is found at inference time, by minimizing the distance between the visual encodings of the neighbors' and the query image. The performance of this baseline was not consistently surpassed on indoor and outdoor benchmarks by any APR, to date.

Visual pose estimation pipelines [2], [3], achieving *state-of-the-art* accuracy on large-scale datasets with challenging conditions, also leverage on IR encodings. Pretrained models for place recognition are used to fetch a relatively small set of nearest neighbors, providing a reduced search space for subsequent feature matching. In addition, latent feature maps, generated by IR models, were shown to effectively replace classical features in the matching stage [3], [27], [28].

III. Regressing camera pose without a pose encoder

A. Image retrieval encoder for camera pose estimation

Although IR visual encodings have proven to be useful for place recognition and pose estimation pipelines, they have not been exploited by pose regression methods. Here, we propose to remove the pose encoder and use instead global descriptors learned for place recognition. In order to test this paradigm we take an MLP architecture, commonly used by APRs (Fig. 1a), and add one FC layer to match the input dimension of the MLP. The resulting architecture (Fig. 1b), consists of a sequence of 4096-, 2048- and 1024-dimensional FC layers, followed by a 3-dimensional FC layer for regressing the position and a 4-dimensional FC layer for regressing the orientation. In order to generate the input for our model, we apply a NetVLAD model with a VGG16 [29] backbone, pretrained on the Pittsburgh 250K dataset [30]. We generate m augmented encodings for each image by randomly jittering its color, hue and saturation m times and then encoding the resulting augmentations. At train time, we randomly select an augmentation and feed it to the network (Fig. 3a). At test time, the query image is loaded and its encoding is computed on-the-fly (Fig. 3b). We normalize images using the mean and standard deviation used in [26], but no further range scaling, resizing or cropping is applied at train or at test time. We train our model with a commonly used pose loss [8] where the scaling between the position and orientation objectives is learned as part of the training process.

Table I shows the results when training on an outdoor dataset (Cambridge Landmarks [6]). Further details about the training procedure and the datasets used for evaluation are provided in Section IV. Although we do not perform any fine-tuning and use a pre-trained encoder, the shallow MLP model is able to outperform the baseline APR in 62.5% of the error measurements and consistently achieves a significantly smaller orientation error (1.8x on average).

TABLE I
Median position error (in meters) and orientation error (in degrees) of a baseline APR [6] and a representative MLP architecture trained with visual encodings.

Method	K. College	Old Hospital	Shop Facade	St. Mary
PoseNet	**1.92**/5.4	**2.31**/5.38	1.46/8.08	**2.65**/8.48
MLP	1.96/**2.53**	2.62/**4.00**	**1.35/3.70**	2.84/**5.79**

B. Network Architecture

Encouraged by the initial proof of concept demonstrated in section III-A, we propose IRPNet, an MLP architecture for regressing camera pose from IR encodings. We follow previous works [10], [12] and split the MLP into two branches (one per objective) and add one 4096-dimensional shared layer. Each branch consists of a sequence of a 4096-, 2048-, 1024- and p-dimensional FC layers. One branch regresses the position vector ($p = 3$) and one branch regresses the orientation vector ($p = 4$). The architecture of IRPNet is shown is Fig 2.

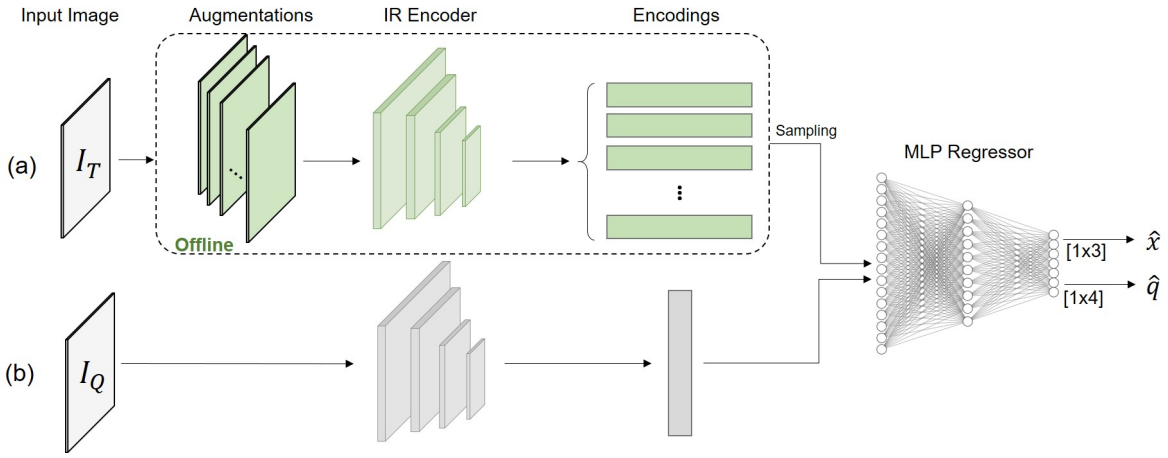

Fig. 3. Our proposed training (a) and testing (b) procedure for camera pose regression using IR encoding. (a) At train time we sample a single entry from pre-computed IR encodings of the input image (I_T) augmentations and feed it to an MLP regressor. (b) At test time, given a query image (I_Q), we generate its IR encoding and infer the camera pose by passing it to through an MLP regressor.

TABLE II
MEDIAN POSITION ERROR (IN METERS) AND ORIENTATION ERROR (IN DEGREES) OF APRS, WHEN TESTED ON THE CAMBRIDGE DATASET.

Method	K. College	Old Hospital	Shop Facade	St. Mary	Average
PoseNet (Learnable) [8]	0.99/1.06	2.17/2.94	1.05/3.97	1.49/3.43	1.43/2.85
GeoPoseNet [8]	0.88/1.04	3.20/3.29	0.88/3.78	1.57/3.32	1.63/2.86
IPRNet (Ours)	1.18/2.19	1.87/3.38	0.72/3.47	1.87/4.94	1.42/3.45
MapNet [17]	1.07/1.89	1.94/3.91	1.49/4.22	2.0/4.53	1.63/3.64
LSTM-PN [9]	0.99/3.65	1.51/4.29	1.18/7.44	1.52/6.68	1.30/5.52
GPoseNet	1.61/2.29	2.62/3.89	1.14/5.73	2.93/6.46	2.08/4.59
BayesianPN [7]	1.74/4.06	2.57/5.14	1.25/7.54	2.11/8.38	1.92/6.28
PoseNet [6]	1.92/5.4	2.31/5.38	1.46/8.08	2.65/8.48	2.09/6.84
SVS-Pose [10]	1.06/2.81	1.50/4.03	0.63/5.73	2.11/8.11	1.33/5.17
IR Baseline [16]	1.48/4.45	2.68/4.63	0.9/4.32	1.62/6.06	1.67/4.87

TABLE III
POSITION, ORIENTATION RANKINGS OF APRS, WHEN TESTED ON THE CAMBRIDGE DATASET.

Method	K. College	Old Hospital	Shop Facade	St. Mary	Average	Final Rank
PoseNet (Learnable)	2, 2	5, 1	4, 3	1, 2	3, 2	1
GeoPoseNet	1, 1	9, 2	3, 2	3, 1	4, 2	2
IPRNet (Ours)	6, 4	3, 3	2, **1**	4, 4	4, 3	**3**
MapNet	5, 3	4, 5	9, 4	5, 3	6, 4	6
LSTM-PN	2, 7	2, 7	6, 7	2, 6	1, 7	5
GPoseNet	7, 5	8, 4	5, 5	9, 5	7, 5	7
BayesianPN	8, 8	7, 8	7, 8	6, 8	7, 8	8
PoseNet	9, 9	6, 9	8, 9	8, 9	8, 9	9
SVS-Pose	4, 6	1, 6	1, 5	6, 7	3, 6	3

IV. EXPERIMENTS

A. Datasets

For evaluation purposes we chose an indoor and an outdoor dataset, commonly used for benchmarking APRs: 7 Scenes [31] and Cambridge Landmarks [6]. These two datasets encompass common challenges and scenarios typical of visual localization applications: large, medium and small scales, indoor/outdoor, repeating elements, textureless features, significant viewpoint changes and trajectory variations between train and test sets. The 7 Scenes dataset includes seven small-scale indoor scenes, with a spatial extent of a few squared meters. The Cambridge Landmarks dataset is a mid-scale urban outdoor dataset. We report results on four of its six scenes as the other two scenes were either not consistently covered in the literature or reported as possibly faulty [9], [10].

B. Training procedure

The input to our network are encodings of augmented train images, generated as described in section III-A, with ($m =$) 10 augmentations per image. We train our network with a mini-batch of size 8 and optimze using Adam, with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-10}$. We use an initial learning rate of $\lambda = 10^{-2}$ ($\lambda = 10^{-3}$) and a weight decay of $\frac{\lambda}{10^2}$ for indoor (outdoor) localization. Dropout is applied after

TABLE IV
MEDIAN POSITION ERROR (IN METERS) AND ORIENTATION ERROR (IN DEGREES) OF APRS, WHEN TESTED ON THE 7 SCENES DATASET.

Method	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Average
PoseNet (Learnable)	0.14/4.50	0.27/11.8	0.18/12.1	0.2/5.77	0.25/4.82	0.24/5.52	0.37/10.6	0.24/7.87
GeoPoseNet	0.13/4.48	0.27/11.3	0.17/13.0	0.19/5.55	0.26/4.75	0.23/5.35	0.35/12.4	0.23/8.12
IPRNet (Ours)	0.13/5.64	0.25/9.67	0.15/13.10	0.24/6.33	0.22/5.78	0.30/7.29	0.34/11.6	0.23/8.49
MapNet	0.08/3.25	0.27/11.7	0.18/13.3	0.17/5.15	0.22/4.02	0.23/4.93	0.3/12.1	0.21/7.79
LSTM-PN	0.24/5.77	0.34/11.9	0.21/13.7	0.30/8.08	0.33/7.0	0.37/8.83	0.40/13.7	0.31/9.86
GPoseNet	0.20/7.11	0.38/12.3	0.21/13.8	0.28/8.83	0.37/6.94	0.35/8.15	0.37/12.5	0.31/9.95
BayesianPN	0.37/7.24	0.43/13.7	0.31/12.0	0.48/8.04	0.61/7.08	0.58/7.54	0.48/13.1	0.47/9.81
PoseNet	0.32/8.12	0.47/14.4	0.29/12.0	0.48/7.68	0.47/8.42	0.59/8.64	0.47/13.8	0.44/10.43
IR Baseline	0.18/10.0	0.33/12.4	0.15/14.3	0.25/10.1	0.26/9.42	0.27/11.1	0.24/14.7	0.24, 11.72

TABLE V
POSITION, ORIENTATION RANKINGS OF APRS, WHEN TESTED ON THE 7 SCENES DATASET.

Method	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Average	Final Rank
PoseNet (Learnable)	4, 3	2, 4	3, 3	3, 3	3, 3	3, 3	3, 4	4, 1	4
GeoPoseNet	2, 2	2, 2	2, 4	2, 2	4, 2	1, 2	3, 4	3, 2	2
IPRNet (Ours)	2, 4	**1, 1**	**1**, 5	4, 4	**1**, 4	4, 4	2, 2	2, 4	**3**
MapNet	1, 1	2, 3	3, 6	1, 1	1, 1	1, 1	1, 3	1, 1	1
LSTM-PN	6, 5	5, 5	5, 7	6, 7	5, 6	6, 6	8, 6	6, 7	6
GPoseNet	5, 6	6, 6	5, 8	5, 8	6, 5	5, 6	4, 5	5, 6	5
BayesianPN	8, 7	7, 7	8, 1	7, 6	8, 7	7, 5	8, 6	8, 5	6
PoseNet	7, 8	8, 8	7, 1	7, 5	7, 8	8, 7	7, 8	7, 7	8

the second layer in both branches, with probability 0.2. We train the network once for position and once for orientation, using the the loss function defined in Eq. 1. For indoor localization, we train once with $s_x = 1, s_q = 0$ and once with $s_x = 0, s_q = 1$, for learning to regress the position and orientation, respectively. For outdoor localization, we observe that positional constraints are required when learning to regress the orientation and thus train with $s_x = 1, s_q = 250$. For regressing the position, we follow the same strategy as for indoor localization. As oppose to other APRs using this type of loss, we do not fine tune s_x and s_q per scene. We initialize the network with He initialization [32] and train it for 300 epochs on indoor scenes and for 1000 epochs on outdoor scenes. We implement our model using PyTorch [33] and train on a single NVIDIA Tesla V100 GPU.

C. Results

Tables II and IV show the median position error (in meters) and orientation error (in degrees) of IRPNet and seven other APRs on the Cambridge Landmarks and the 7 Scenes dataset, respectively. Tables III and V give the average ranking for position and orientation, separately and combined, for each scene. The final ranking is taken as the average between the average position rank and average orientation rank per dataset. In this way, we avoid the bias associated with the different scales of position and orientation errors.

For outdoor localization (Tables II and III), we rank at the third place in terms of accuracy. The first and second most accurate APRs are variants of PoseNet, using the same architecture with different loss formulations. We note that IRPNet is the only model that consistently localizes in under $2m$ and 5°. Table VI shows the percentage of pose error measurements under this threshold, which is closely related to medium accuracy bound. We achieve the same ranking (third) also for indoor localization (Tables IV and V). In addition, IRPNet surpasses all other APRs in 28.6% of the error measurements.

As shown in Tables III and V, together with PoseNet+Geometrical Loss, IRPNet is the only APR to consistently rank at the top-three on both benchmarks. Interestingly, the APRs ranked first for indoor (MapNet) and outdoor (PoseNet+Learnable), are not as accurate on both tasks. MapNet is ranked sixth for outdoor localization, while PoseNet+Learnable Loss is ranked at the fourth place.

In addition to accuracy, we also compare the training time of our network to a baseline APR (PoseNet with a ResNet-152 encoder). For a mini-batch of size 8, it took $140ms$ to train a batch with the baseline APR, and $12ms$ to train with IRPNet (over 10 times faster). In addition, the input to our model is significantly smaller and constant in size. For example, when training on the 7 Scenes dataset, a typical APR is trained with batches of $150K$ sized tensors (after image cropping), while IRPNet only requires a $4K$ sized input, without any initial image cropping.

Using encodings trained for visual similarity also allows an easy integration into existing localization pipelines, such as [3]. The trained model can provide a first rough estimate, serving as another supporting route. As oppose to encodings learned with a pose encoder, IR encodings carry visual similarity information. This can explain, why we were able to independently train our model for orientation and position, while other APRs were not able to successfully adopt this strategy [6], [8].

TABLE VI
PERCENTAGE OF POSE ERROR MEASUREMENTS UNDER 2 METERS AND 5 DEGREES, FOR POSITION AND ORIENTATION RESPECTIVELY, ON THE CAMBRIDGE LANDMARKS DATASET.

Method	Cambridge
PoseNet (Learnable)	87.5%
GeoPoseNet	87.5%
IPRNet (Ours)	100%
MapNet	87.5%
SVS-Pose	62.5%
LSTM-PN	75%
GPoseNet	50%
BayesianPN	37.5%
PoseNet	25%
IR Baseline	75%

V. CONCLUSION

Current APRs rely on a scene-specific encoder to generate a global pose feature vector, which is mapped to an estimated pose vector using an MLP regressor. In this work, we explore an alternative paradigm, where instead of training both an encoder and a pose regressor per scene, we train only the regressor and use a pre-trained visual IR encoder. This paradigm allows for shorter training time, lighter storage which does not depend on a scene-specific model and for integration into existing localization pipelines which are already using IR encodings. We propose an MLP architecture, named IRPNet, for evaluating this paradigm and show that it is able to achieve consistent localization (third place on both outdoor and indoor benchmarks) with minimal tuning of the loss function. Despite its simplicity, this architecture is also the only APR to consistently achieve medium accuracy on an outdoor benchmark. The fact that (scene/dataset) agnostic visual encodings were sufficient to competitively regress the camera pose, compared to scene-specific pose encodings, suggests that APRs could benefit from other independent representations, coming from other modalities. For example, using pre-trained models for image segmentation or for extracting local visual representations. More research into regressor architectures is also required to allow for integrating multi-modality information about the underlying map, capturing both visual and spatial constraints. Finally, given the multi-objective nature of the problem, separately training for orientation and position could take advantage of different modalities that are specific or more informative for a given objective.

REFERENCES

[1] T. Sattler, B. Leibe, and L. Kobbelt, "Efficient & effective prioritized matching for large-scale image-based localization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 9, pp. 1744–1756, 2016.

[2] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 716–12 725.

[3] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7199–7209.

[4] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[5] H. Taira, I. Rocco, J. Sedlar, M. Okutomi, J. Sivic, T. Pajdla, T. Sattler, and A. Torii, "Is this the right place? geometric-semantic pose verification for indoor visual localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4373–4383.

[6] 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.

[7] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," in *2016 IEEE international conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4762–4769.

[8] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5974–5983.

[9] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using lstms for structured feature correlation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 627–637.

[10] 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)*. IEEE, 2017, pp. 1525–1530.

[11] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu, "Image-based localization using hourglass networks," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 879–886.

[12] J. Wu, L. Ma, and X. Hu, "Delving deeper into convolutional neural networks for camera relocalization," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5644–5651.

[13] M. Cai, C. Shen, and I. D. Reid, "A hybrid probabilistic model for camera relocalization," in *BMVC*, 2018.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[15] 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.

[16] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixe, "Understanding the limitations of cnn-based absolute camera pose regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3302–3312.

[17] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz, "Geometry-aware learning of maps for camera localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2616–2625.

[18] Y. Shavit and R. Ferens, "Introduction to camera pose estimation with deep learning," *arXiv preprint arXiv:1907.05272*, 2019.

[19] A. Valada, N. Radwan, and W. Burgard, "Deep auxiliary learning for visual localization and odometry," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 6939–6946.

[20] 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, 2018.

[21] 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.

[22] V. Balntas, S. Li, and V. Prisacariu, "Relocnet: Continuous metric learning relocalisation using neural nets," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 751–767.

[23] 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 International Conference on Computer Vision*, 2019, pp. 2871–2880.

[24] 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*. IEEE, 2010, pp. 3304–3311.

[25] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE trans-*

actions on pattern analysis and machine intelligence, vol. 34, no. 9, pp. 1704–1716, 2011.

[26] 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*, 2016, pp. 5297–5307.

[27] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3456–3465.

[28] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-net: A trainable cnn for joint description and detection of local features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8092–8101.

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

[30] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi, "Visual place recognition with repetitive structures," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 883–890.

[31] 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)*. IEEE, 2013, pp. 173–179.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8024–8035.