Deep Auxiliary Learning for Visual Localization and Odometry

Abhinav Valada* Noha Radwan* Wolfram Burgard

***Abstract*— Localization is an indispensable component of a robot's autonomy stack that enables it to determine where it is in the environment, essentially making it a precursor for any action execution or planning. Although convolutional neural networks have shown promising results for visual localization, they are still grossly outperformed by state-of-the-art local feature-based techniques. In this work, we propose VLocNet, a new convolutional neural network architecture for 6-DoF global pose regression and odometry estimation from consecutive monocular images. Our multitask model incorporates hard parameter sharing, thus being compact and enabling real-time inference, in addition to being end-to-end trainable. We propose a novel loss function that utilizes auxiliary learning to leverage relative pose information during training, thereby constraining the search space to obtain consistent pose estimates. We evaluate our proposed VLocNet on indoor as well as outdoor datasets and show that even our single task model exceeds the performance of state-of-the-art deep architectures for global localization, while achieving competitive performance for visual odometry estimation. Furthermore, we present extensive experimental evaluations utilizing our proposed Geometric Consistency Loss that show the effectiveness of multitask learning and demonstrate that our model is the first deep learning technique to be on par with, and in some cases outperforms state-of-the-art SIFT-based approaches.**

I. Introduction

Visual localization is a fundamental transdisciplinary problem and a crucial enabler for numerous robotics as well as computer vision applications, including autonomous navigation, Simultaneous Localization and Mapping (SLAM), Structure-from-Motion (SfM) and Augmented Reality (AR). More importantly, it plays a vital role when robots lose track of their location, or what is commonly known as the kidnapped robot problem. In order for robots to be safely deployed in the wild, their localization system should be robust to frequent changes in the environment; whether environmental changes such as illumination and seasonal appearance, dynamic changes such as moving vehicles and pedestrians, or structural changes such as constructions.

Visual localization techniques can be broadly classified into two categories; topological and metric methods. Topological localization provides coarse estimates of the position, usually by dividing the map into a discretized set of locations and employing image retrieval techniques [2], [6], [24]. While this approach is well suited for large environments, the resulting location accuracy is bounded by the granularity of the discrete set. Metric localization approaches on the other hand, provide a 6-DoF metric estimate of the pose within the environment. Thus far, local feature-based approaches that utilize SfM information achieve state-of-the-art performance [22], [25]. However, a critical drawback of these approaches is the decrease in speed and increase in complexity of finding feature correspondences as the size of the environment grows. Moreover, most approaches require a minimum number of matches to be able to produce a pose estimate. This in turn causes pose estimation failures when there is large viewpoint changes, motion blur, occlusions or textureless environments.

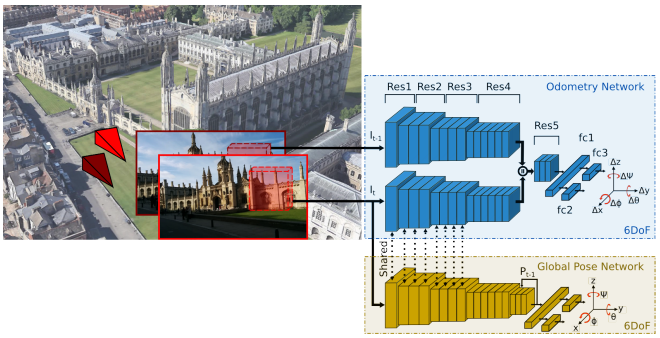

Fig. 1. **VLocNet: Multitask deep convolutional neural network for 6-DoF visual localization and odometry.** Our network takes two consecutive monocular images as input and regresses the 6-DoF global pose and 6-DoF odometry simultaneously. The global pose and odometry subnetworks incorporate hard parameter sharing and utilize our proposed Geometric Consistency Loss function that is robust to environmental aliasing. Online demo: `http://deeploc.cs.uni-freiburg.de/`

Inspired by the outstanding performance of Convolutional Neural Networks (CNNs) in a variety of tasks in various domains and with the goal of eliminating manual engineering of algorithms for feature selection, CNN architectures that directly regress the 6-DoF metric pose have recently been explored [12], [26], [4]. However, despite their ability to handle challenging perceptual conditions and effectively manage large environments, they are still unable to match the performance of state-of-the-art feature-based localization methods. This is partly due to their inability to internally model the 3D structural constraints of the environment while learning from a single monocular image.

As CNN-based approaches become the de-facto standard for more robotics tasks, the need for multitask models becomes increasingly crucial. Moreover, from a robot's learning perspective, it is unlucrative and unscalable to have multiple specialized single-task models as they inhibit both inter-task and auxiliary learning. This has lead to a recent surge in research targeted towards frameworks for learning unified models for a range of tasks across different domains [28], [20], [3]. The goal of these multitask learning methods is to leverage similarities within task-specific features and exploit complementary features learned across different tasks, with the aim of mutual benefit. An evident advantage is the resulting compact model size in comparison to having

*These authors contributed equally. All authors are with the Department of Computer Science, University of Freiburg, Germany. This work has partially been supported by the European Commission under the grant number H2020-ICT-644227-FLOURISH.

multiple task-specific models. Auxiliary learning approaches on the other hand, aim at maximizing the prediction of a primary task by supervising the model to additionally learn a secondary task. For instance, in the context of localization, humans often describe their location to each other with respect to some reference landmark in the scene and giving their position relative to it. Here, the primary task is to localize and the auxiliary task is to be able to identify landmarks. Similarly, we can leverage the complementary relative motion information from odometry to constrict the search space while training the global localization model. However, this problem is non-trivial as we need to first determine how to structure the architecture to ensure the learning of this inter-task correlation and secondly, how to jointly optimize the unified model since different task-specific networks have different attributes and different convergence rates.

In this work, we address the problem of global pose regression by simultaneously learning to estimate visual odometry as an auxiliary task. We propose the VLocNet architecture consisting of a global pose regression sub-network and a Siamese-type relative pose estimation sub-network. Our network based on the residual learning framework, takes two consecutive monocular images as input and jointly regresses the 6-DoF global pose as well as the 6-DoF relative pose between the images. We incorporate a hard parameter sharing scheme to learn inter-task correlations within the network and present a multitask alternating optimization strategy for learning shared features across the network. Furthermore, we devise a new loss function for global pose regression that incorporates the relative motion information during training and enforces the predicted poses to be geometrically consistent with respect to the true motion model. We present extensive experimental evaluations on both indoor and outdoor datasets comparing the proposed method to state-of-the-art approaches for global pose regression and visual odometry estimation. We empirically show that our proposed VLocNet architecture achieves state-of-the-art performance compared to existing CNN-based techniques. To the best of our knowledge, our presented approach is the first deep learning-based localization method to perform on par with local feature-based techniques. Moreover, our work is the first attempt to show that a joint multitask model can precisely and efficiently outperform its task-specific counterparts for global pose regression and visual odometry estimation.

II. Related Work

There are numerous approaches that have been proposed for localization in the literature. In this section, we review some of the techniques developed thus far for addressing this problem, followed by a brief discussion on approaches for visual odometry estimation.

Sparse feature-based localization approaches learn a set of feature descriptors from training images and build a codebook of 3D descriptors against which a query image is matched [22], [8]. To efficiently find feature correspondences within the codebook, Shotton *et al.* [23] and Valentin *et al.* [25] train regression forests on 3D scene data and use RANSAC to infer the final location of the query image. Donoser *et al.* propose a discriminative classification approach using random ferns and demonstrate improved pose accuracy with faster run times [7]. Despite the accurate pose estimates provided by these methods, the overall run-time depends on the size of the 3D model and the number of feature correspondences found. The approach presented in this paper does not suffer from these scalability issues as the learned model is independent of the size of the environment. Moreover, since it does not involve any expensive matching algorithm, it has a time complexity of $\mathcal{O}(1)$.

Deep learning-based localization: PoseNet [12] was the first approach to utilize DCNNs to address the metric localization problem. The authors further extended this work by using a Bayesian CNN implementation to estimate the uncertainty of the predicted pose [10]. Concurrently, Walch *et al.* [26] and Clark *et al.* [4] propose DCNNs with Long-Short Term Memory (LSTM) units to avoid overfitting while still selecting the most useful feature correlations. Contrary to these approaches and inspired by semantic segmentation architectures, Melekhov *et al.* introduce the HourglassPose [16] network that utilizes a symmetric encoder-decoder architecture followed by a regressor to estimate the camera pose. In order to provide a more robust approach to balance both the translational and rotational components in the loss term, the commonly employed fixed weight regularizer was replaced with learnable parameters in [11]. The authors also introduced a loss function based on the geometric reprojection error that does not require balancing of the pose components, but it often has difficulty in converging. More recently, Laskar *et al.* proposed a learning procedure that decouples feature learning and pose estimation, closely resembling feature-based localization approaches [14]. Unlike most of the aforementioned approaches that utilize the Euclidean loss for pose regression, we propose a novel loss function that incorporates motion information while training to learn poses that are consistent with the previous prediction.

Visual Odometry: Another closely related problem in robotics is estimating the incremental motion of the robot using only sequential camera images. In one of the earlier approaches, Konda *et al.* [13] adopt a classification approach to the problem, where a CNN with a softmax layer is used to infer the relative transformation between two images using a prior set of discretized velocities and directions. Another approach is proposed by Nicholai *et al.* [19], in which they combine both image and LiDAR information to estimate the relative motion between two frames. They project the point cloud on the 2D image and feed this information to a neural network which estimates the visual odometry. Mohanty *et al.* [17] propose a Siamese AlexNet-based architecture called DeepVO, in which the translational and rotational components are regressed through an L2-loss layer with equal weight values. In similar work, Melekhov *et al.* [15] add a weighting term to balance both the translational and rotational components of the loss, which yields an improvement in the predicted pose. Additionally, they use a spatial pyramid pooling layer in their architecture which renders their approach robust to varying input image resolutions. Inspired by the success of residual networks in various visual recognition tasks, we propose a Siamese-type

two stream architecture built upon the ResNet-50 [9] model for visual odometry estimation.

Contrary to the task-specific approaches presented above where individual models are trained for global pose regression and visual odometry estimation, we propose a joint end-to-end trainable architecture that simultaneously regresses the 6-DoF global pose and relative motion as an auxiliary output. By jointly learning both tasks, our approach is robust to environmental aliasing by utilizing previous pose and relative motion information, thereby combining the advantages of both local feature and deep learning-based localization methods. Moreover, by sharing features across different scales, our proposed model significantly outperforms the state-of-the-art in CNN-based localization while achieving competitive performance for visual odometry estimation.

III. Deep Pose Regression

The primary goal of our architecture is to precisely estimate the global pose by minimizing our proposed Geometric Consistency Loss function, which in turn constricts the search space using the relative motion between two consecutive frames. We formulate this problem in the context of auxiliary learning with the secondary goal of estimating the relative motion. The features learned for relative motion estimation are then leveraged by the global pose regression network to learn a more distinct representation of the scene. More specifically, our architecture consists of a three-stream neural network; a global pose regression stream and a Siamese-type double-stream for odometry estimation. An overview of our proposed VLocNet architecture is shown in Fig. 1. Given a pair of consecutive monocular images (I_t, I_{t-1}), our network predicts both the global pose $\mathbf{p}_t = [\mathbf{x}_t, \mathbf{q}_t]$ and the relative pose $\mathbf{p}_{t,t-1} = [\mathbf{x}_{t,t-1}, \mathbf{q}_{t,t-1}]$ between the input frames, where $\mathbf{x} \in \mathbb{R}^3$ denotes the translation and $\mathbf{q} \in \mathbb{R}^4$ denotes the rotation in quaternion representation. For ease of notation, we assume that the quaternion outputs of the network have been normalized a priori. The input to the Siamese streams are the images I_t, I_{t-1}, while the input to the global pose stream is I_t. In the remainder of this section, we present the constituting parts of our VLocNet architecture along with how the joint optimization is carried out.

A. Global Pose Regression

In this section, we describe the architecture of our global pose sub-network, which given an input image I_t and a previous predicted pose $\hat{\mathbf{p}}_{t-1}$, predicts the 7-dimensional pose $\hat{\mathbf{p}}_t$. Similar to previous works [12], [26], $\mathbf{p}$ is defined relative to an arbitrary global reference frame.

1) Network Architecture: To estimate the global pose, we build upon the ResNet-50 [9] architecture with the following modifications. The structure of our network is similar to ResNet-50 truncated before the last average pooling layer. The architecture is comprised of five residual blocks with multiple residual units, where each unit has a bottleneck architecture consisting of three convolutional layers in the following order: 1×1 convolution, 3×3 convolution, 1×1 convolution. Each of the convolutions is followed by batch normalization, scale and Rectified Linear Unit (ReLU). We modify the standard residual block structure by replacing ReLUs with Exponential Linear Units (ELUs) [5]. ELUs help in reducing the bias shift in the neurons, in addition to avoiding the vanishing gradient and yield faster convergence. We replace the last average pooling layer with global average pooling and subsequently add three inner-product layers, namely *fc1*, *fc2* and *fc3*. The first inner-product layer *fc1* is of dimension 1024 and the following two inner-product layers are of dimensions 3 and 4, for regressing the translation $\mathbf{x}$ and rotation $\mathbf{q}$ respectively. Our proposed Geometric Consistency Loss, detailed in Sec. III-A.2, ensures that the predicted pose is consistent with that obtained by accumulating the relative motion to the previous pose. Therefore, we feed the previous pose (groundtruth pose during training and predicted pose during evaluation) to the network so that it can better learn about spatial relations of the environment. We do not incorporate recurrent units into our network as our aim in this work is to localize only using consecutive monocular images and not rely on long-term temporal features. We first feed the previous pose to an inner-product layer *fc4* of dimension D and reshape its output to $H \times W \times C$, which corresponds in shape to the output of the last residual unit before the downsampling stage. Both tensors are then concatenated and fed to the subsequent residual unit. In total, there are four downsampling stages in our network and we experiment with fusing at each of these stages in Sec. IV-E.

2) Geometric Consistency Loss: Learning both translational and rotational pose components with the same loss function is inherently challenging due to the difference in scale and units between both the quantities. Eq. (1) and Eq. (2) describe the loss function for regressing the translational and rotational components in the Euclidean space.

$$\mathscr{L}_x(I_t) := \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_\gamma \tag{1}$$

$$\mathscr{L}_q(I_t) := \|\mathbf{q}_t - \hat{\mathbf{q}}_t\|_\gamma, \tag{2}$$

where $\mathbf{x}_t$ and $\mathbf{q}_t$ denote the ground-truth translation and rotation components, $\hat{\mathbf{x}}_t$ and $\hat{\mathbf{q}}_t$ denote their predicted counterparts and γ refers to the L^γ-norm. In this work, we use the L^2 Euclidean norm. Previous work has shown that the performance of a model trained to jointly regress the position and orientation, outperforms two separate models trained for each task [11]. Therefore, as the loss function is required to learn both the position and orientation, a weight regularizer β is used to balance each of the loss terms. We can represent this loss function as:

$$\mathscr{L}_\beta(I_t) := \mathscr{L}_x(I_t) + \beta \mathscr{L}_q(I_t). \tag{3}$$

Although initial work [12], [26], [27], [18] has shown that by minimizing this function, the network is able to learn a valid pose regression model, it suffers from the drawback of having to manually tune the hyperparameter β for each new scene in order to achieve reasonable results. To counteract this problem, recently [11] learnable parameters were introduced to replace β. The resulting loss function is:

$$\mathscr{L}_s(I_t) := \mathscr{L}_x(I_t) \exp(-\hat{s}_x) + \hat{s}_x + \mathscr{L}_q(I_t) \exp(-\hat{s}_q) + \hat{s}_q, \tag{4}$$

where $\hat{s}_x$ and $\hat{s}_q$ are the two learnable variables. Each variable acts as a weighting for the respective component in the loss function. Although this formulation overcomes the problem of having to manually select a β value for each scene, it

does not ensure that the estimated poses are consistent with the previous motion.

As a solution to this problem, we propose a novel loss function that incorporates previous motion information, thereby producing consistent pose estimates. We introduce an additional constraint which bootstraps the loss function by penalizing pose predictions that contradict the relative motion. More precisely, in addition to the loss term shown in Eq. (4), we enforce that the difference between $\hat{\mathbf{p}}_t$ and $\hat{\mathbf{p}}_{t-1}$ be as close to the groundtruth relative motion $\mathbf{p}_{t,t-1}$ as possible. We use $\mathscr{R}_x(I_t)$ and $\mathscr{R}_q(I_t)$ to denote the relative motion between the current image I_t and the previous predicted pose $\hat{\mathbf{p}}_{t-1}$ as:

$$\mathscr{R}_x(I_t) := \hat{\mathbf{x}}_t - \hat{\mathbf{x}}_{t-1} \tag{5}$$

$$\mathscr{R}_q(I_t) := \hat{\mathbf{q}}_{t-1}^{-1}\hat{\mathbf{q}}_t. \tag{6}$$

The components from Eq. (5) and Eq. (6) compute the relative motion in terms of the network's predictions. We integrate these components into an odometry loss term to minimize the variance between the predicted poses. The corresponding odometry loss can be formulated as:

$$\mathscr{L}_{x_{odom}}(I_t) := \|\mathbf{x}_{t,t-1} - \mathscr{R}_x(I_t)\|_\gamma \tag{7}$$

$$\mathscr{L}_{q_{odom}}(I_t) := \left\|\mathbf{q}_{t,t-1} - \mathscr{R}_q(I_t)\right\|_\gamma, \tag{8}$$

where $\mathscr{L}_{x_{odom}}$ computes the difference between the groundtruth relative translational motion and its predicted counterpart, while $\mathscr{L}_{q_{odom}}$ computes a similar difference for the rotational component. We combine both the odometry loss terms with the loss function from Eq. (4), thereby minimizing:

$$\begin{aligned}\mathscr{L}_{Geo}(I_t) := &\left(\mathscr{L}_x(I_t) + \mathscr{L}_{x_{odom}}(I_t)\right)\exp(-\hat{s}_x) \\ &+ \hat{s}_x + \left(\mathscr{L}_q(I_t) + \mathscr{L}_{q_{odom}}(I_t)\right)\exp(-\hat{s}_q) + \hat{s}_q.\end{aligned} \tag{9}$$

We hypothesize that, by utilizing this relative motion in the loss function, the resulting trained model is more robust to perceptual aliasing within the environment.

B. Visual Odometry

In order to integrate motion specific features in our global pose regression network, we train an auxiliary network to regress the 6-DoF relative pose from the images (I_t, I_{t-1}). We do so by constructing a two stream Siamese-type network also based on the ResNet-50 architecture. We concatenate features from the two individual streams of ResNet-50 truncated before the last downsampling stage (*Res5*). We then pass these concatenated feature maps to the last three residual units, followed by three inner-product layers, similar to our global pose regression network. We minimize the following loss function:

$$\begin{aligned}\mathscr{L}_{vo}(I_t, I_{t-1}) := &\mathscr{L}_x(I_t, I_{t-1})\exp(-\hat{s}_x) \\ &+ \hat{s}_x + \mathscr{L}_q(I_t, I_{t-1})\exp(-\hat{s}_q) + \hat{s}_q.\end{aligned} \tag{10}$$

With a slight abuse of notation, we use $\mathscr{L}_x(I_t, I_{t-1})$ to refer to the L^2 Euclidean loss in the translational component of the visual odometry and $\mathscr{L}_q(I_t, I_{t-1})$ for the rotational component. Similar to our approach used for the global pose regression, we additionally learn two weighting parameters to balance the loss between both components. We detail the training procedure in Sec. IV-B.

C. Deep Auxiliary Learning

The idea of jointly learning both the global pose and visual odometry stems from the inherent similarities across both tasks in the feature space. More importantly, sharing features across both networks can enable a competitive and collaborative action as each network updates its own weights during backpropagation in an attempt to minimize the distance to the groundtruth pose. This symbiotic action introduces additional regularization while training, thereby avoiding overfitting. Contrary to the approaches that use a two stream shared Siamese network for visual odometry estimation, we do not share weights between the two temporal streams, rather we share weights between the stream that takes the image I_t from the current timestep as input and the global pose regression stream. By learning separate discriminative features in each timestep before learning the correlation between them, the visual odometry network is able to effectively generalize to challenging corner cases containing motion blur and perceptual aliasing. The global pose regression network also benefits from this feature sharing, as the shared weights are pulled more towards areas of the image from which the relative motion can be easily estimated.

While sharing features across multiple networks can be inferred as a form of regularization, it is not clear a priori for how many layers should we maintain a shared stream. Sharing only a few initial layers does not have any additive benefit to either network, as early layers learn very generic feature representations. On the other hand, maintaining a shared stream too deep into the network can negatively impact the performance of both tasks, since the features learned at the stages towards the end are more task specific. In this work, we studied the impact of sharing features across both sub-networks and experimented with varying the amount of feature sharing. We elaborate on these experiments in Sec. IV-F. Another critical aspect of auxiliary learning is how the optimization is carried out. We detail our optimization procedure in Sec. IV-B. Finally, during inference, the joint model can be deployed as a whole or each sub-network individually, since the relative pose estimates are only used in the loss function and there is no inter-network dependency in terms of concatenating or adding features from either sub-networks. This gives additional flexibility at the time of deployment compared to architectures that have cross-connections or cross-network fusion.

IV. Experimental Evaluation

In this section, we present results using our proposed VLocNet architecture in comparison to the state-of-the-art on both indoor and outdoor datasets, followed by detailed analysis on the architectural decisions and finally, we demonstrate the efficacy of learning visual localization models along with visual odometry as an auxiliary task.

A. Evaluation Datasets

We evaluate VLocNet on two publicly available datasets; Microsoft 7-Scenes [23] and Cambridge Landmarks [12]. We use the original train and test splits provided by all the datasets to facilitate comparison and benchmarking.

TABLE I

COMPARISON OF MEDIAN LOCALIZATION ERROR OF VLOCNET WITH EXISTING CNN MODELS ON THE 7-SCENES DATASET.

Scene	PoseNet [12]	Bayesian PoseNet [10]	LSTM-Pose [26]	VidLoc [4]	Hourglass-Pose [16]	BranchNet [27]	PoseNet2 [11]	NNnet [14]	VLocNet (Ours)
Chess	0.32m, 8.12°	0.37m, 7.24°	0.24m, 5.77°	0.18m, N/A	0.15m, 6.53°	0.18m, 5.17°	0.13m, 4.48°	0.13m, 6.46°	**0.036**m, **1.71**°
Fire	0.47m, 14.4°	0.43m, 13.7°	0.34m, 11.9°	0.26m, N/A	0.27m, 10.84°	0.34m, 8.99°	0.27m, 11.3°	0.26m, 12.72°	**0.039**m, **5.34**°
Heads	0.29m, 12.0°	0.31m, 12.0°	0.21m, 13.7°	0.14m, N/A	0.19m, 11.63°	0.20m, 14.15°	0.17m, 13.0°	0.14m, 12.34°	**0.046**m, **6.64**°
Office	0.48m, 7.68°	0.48m, 8.04°	0.30m, 8.08°	0.26m, N/A	0.21m, 8.48°	0.30m, 7.05°	0.19m, 5.55°	0.21m, 7.35°	**0.039**m, **1.95**°
Pumpkin	0.47m, 8.42°	0.61m, 7.08°	0.33m, 7.00°	0.36m, N/A	0.25m, 7.01°	0.27m, 5.10°	0.26m, 4.75°	0.24m, 6.35°	**0.037**m, **2.28**°
RedKitchen	0.59m, 8.64°	0.58m, 7.54°	0.37m, 8.83°	0.31m, N/A	0.27m, 10.15°	0.33m, 7.40°	0.23m, 5.35°	0.24m, 8.03°	**0.039**m, **2.20**°
Stairs	0.47m, 13.8°	0.48m, 13.1°	0.40m, 13.7°	0.26m, N/A	0.29m, 12.46°	0.38m, 10.26°	0.35m, 12.4°	0.27m, 11.82°	**0.097**m, **6.48**°
Average	0.44m, 10.4°	0.47m, 9.81°	0.31m, 9.85°	0.25m, N/A	0.23m, 9.53°	0.29m, 8.30°	0.23m, 8.12°	0.21m, 9.30°	**0.048**m, **3.80**°

Microsoft 7-Scenes: is a dataset comprised of RGB-D images collected from seven different scenes in an indoor office environment [23]. The images were collected with a handheld Kinect RGB-D camera and the groundtruth poses were extracted using KinectFusion [23]. The images were captured at resolution of 640×480 pixels and each scene contains multiple sequences recorded in a room. Each sequence was recorded with different camera motions in the presence of motion blur, perceptual aliasing and textureless features in the room, thereby making it a popular dataset for relocalization and tracking.

Cambridge Landmarks: provides images collected from five different outdoor scenes around the Cambridge University [12]. The images were captured using a smartphone at a resolution of 1920×1080 pixels while walking in different trajectories and pose labels were computed using an SfM method. The dataset exhibits substantial clutter caused by pedestrians, cyclists and moving vehicles, making it challenging for urban relocalization.

B. Network Training

In order to train our network on different datasets, we rescale the images maintaining the aspect ratio such that the shorter side is of length 256 pixels. We calculate the pixel-wise mean for each of the scenes in the datasets and subtract them with the input images. We experimented with augmenting the images using pose synthesis [27] and synthetic view synthesis [18], however they did not yield any performance gains, rather in some cases they negatively affected the pose accuracy. We found that using random crops of 224×224 pixels acts as a better regularizer helping the network generalize better in comparison to synthetic augmentation techniques while saving preprocessing time. For evaluations, we use the center crop of the images.

We use the Adam solver for optimization with $\beta_1 = 0.9, \beta_2 = 0.999$ and $\varepsilon = 10^{-10}$. We train the network with an initial learning rate of $\lambda_0 = 10^{-4}$ with a mini-batch size of 32 and a dropout probability of 0.2. Details regarding the specific $\hat{s}_x$ and $\hat{s}_q$ values used for our Geometric Consistency Loss function are covered in Sec. IV-E. In order to learn a unified model and to facilitate auxiliary learning, we employ different optimization strategies that allow for efficient learning of shared features as well as task-specific features, namely alternate training and joint training. In alternate training we use a separate optimizer for each task and alternatively execute each task optimizer on the task-specific loss function, thereby allowing synchronized transfer of information from one task to the other. This instills a form of hierarchy into the tasks, as the odometry sub-network improves the estimate of its relative poses, the global pose network in turn uses this estimate to improve its prediction. It is often theorized that this enforces commonality between the tasks. The disadvantage of this approach is that a bias in the parameters is introduced by the task that is optimized second. In joint training on the other hand, we add each of the task-specific loss functions and use a single optimizer to train the sub-networks at the same time. The advantage of this approach is that the tasks are trained in a way that they maintain the individuality of their functions, but as each of our tasks is of different units and scale, the task with the larger scale often dominates the training.

We experiment with bootstrapping the training of VLocNet with different weight initializations for each of the aforementioned optimization schemes. Results from this experiment are discussed in Sec. IV-F. Using the principle of transfer learning, we trained the individual models by initializing all the layers up to the global pooling layer with the weights of ResNet-50 pretrained on ImageNet [21] and we used Gaussian initialization for the remaining layers. We use the TensorFlow [1] deep learning library and all the models were trained on a NVIDIA Titan X GPU for a maximum of 120,000 iterations, which approximately took 15 hours.

C. Comparison with the State-of-the-art

We compare the performance of our VLocNet architecture with current state-of-the-art deep learning-based localization methods namely PoseNet [12], Bayesian PoseNet [10], LSTM-Pose [26], VidLoc [4], Hourglass-Pose [16], BranchNet [27], PoseNet2 [11], SVS-Pose [18], and NNnet [14]. We report the performance in terms of the median translation and orientation errors for each scene in the datasets. On the 7-Scenes dataset, we initialized the $\hat{s}_x$ and $\hat{s}_q$ for our loss function with values between -3 to 0 and -4.8 to -3 respectively. Tab. I reports the comparative results on this dataset. Our VLocNet architecture consistently outperforms the state-of-the-art methods for all the scenes by 77.14% in translation and 59.14% in rotation. On the Cambridge Landmarks dataset, we report the results using $\hat{s}_x = -3$ and $\hat{s}_q = -6.5$ for all the scenes. Using our VLocNet architecture with the proposed Geometric Consistency Loss, we improve upon the current state-of-the-art results by 51.6%

TABLE II

COMPARISON OF MEDIAN LOCALIZATION ERROR OF VLOCNET WITH EXISTING CNN MODELS ON THE CAMBRIDGE LANDMARKS DATASET.

Scene	PoseNet [12]	Bayesian PoseNet [10]	SVS-Pose [18]	LSTM-Pose [26]	PoseNet2 [11]	VLocNet (Ours)
King's College	1.92m, 5.40°	1.74m, 4.06°	1.06m, 2.81°	0.99m, 3.65°	0.88m, 1.04°	**0.836**m, 1.419°
Old Hospital	2.31m, 5.38°	2.57m, 5.14°	1.50m, 4.03°	1.51m, 4.29°	3.20m, 3.29°	**1.075**m, **2.411**°
Shop Facade	1.46m, 8.08°	1.25m, 7.54°	0.63m, 5.73°	1.18m, 7.44°	0.88m, 3.78°	**0.593**m, **3.529**°
St Mary's Church	2.65m, 8.46°	2.11m, 8.38°	2.11m, 8.11°	1.52m, 6.68°	1.57m, 3.32°	**0.631**m, 3.906°
Average	2.08m, 6.83°	1.92m, 6.28°	1.33m, 5.17°	1.30m, 5.52°	1.62m, 2.86°	**0.784**m, **2.817**°

TABLE III

COMPARISON OF 6DOF VISUAL ODOMETRY ON THE 7-SCENES DATASET.

Scene	LBO [19]	DeepVO [17]	cnnBspp [15]	VLocNet (Ours)
Chess	1.69, 1.13	2.10, 1.15	1.38, 1.12	**1.14**, **0.75**
Fire	3.56, 1.42	5.08, 1.56	2.08, 1.76	**1.81**, 1.92
Heads	14.43, 2.39	13.91, 2.44	3.89, 2.70	**1.82**, **2.28**
Office	3.12, 1.92	4.49, 1.74	1.98, 1.52	**1.71**, **1.09**
Pumpkin	3.12, 1.60	3.91, 1.61	1.29, 1.62	**1.26**, **1.11**
RedKitchen	3.71, 1.47	3.98, 1.50	1.53, 1.62	**1.46**, **1.28**
Stairs	3.64, 2.62	5.99, 1.66	2.34, 1.86	**1.28**, **1.17**
Average	4.75, 1.79	5.64, 1.67	2.07, 1.74	**1.51**, **1.45**

translation [%], orientation [deg/m]

TABLE IV

BENCHMARKING MEDIAN ERRORS ON THE 7-SCENES DATASET.

Scene	Nearest Neighbor [12]	Active Search [22]	VLocNet (Ours)
Chess	0.41m, 11.2°(0)	0.04m, 1.96°(0)	**0.036**m, **1.707**°
Fire	0.54m, 15.5°(1)	0.03m, 1.53°(1)	0.039m, 5.338°
Heads	0.28m, 14.0°(1)	0.02m, 1.45°(1)	0.046m, 6.645°
Office	0.49m, 12.0°(34)	0.09m, 3.61°(34)	**0.039**m, **1.953**°
Pumpkin	0.58m, 12.1°(68)	0.08m, 3.10°(68)	**0.037**m, **2.280**°
RedKitchen	0.58m, 11.3°(0)	0.07m, 3.37°(0)	**0.039**m, **2.205**°
Stairs	0.56m, 15.4°(0)	0.03m, 2.22°(0)	0.097m, 6.476°
Average	0.49m, 13.1°	0.05m, 2.46°	**0.048**m, 3.801°

(n) localization failures

in translation and 1.5% in orientation. Note that we did not perform any hyperparameter optimization, we expect further improvements to the results presented here by tuning the parameters. The results demonstrate that our network substantially improves upon the state-of-the-art on both indoor as well as outdoor datasets.

In order to evaluate the performance of VLocNet on visual odometry estimation, we show quantitative comparison against three state-of-the-art CNN approaches, namely DeepVO [17], cnnBspp [15] and LBO [19]. Tab. III shows comprehensive results from this experiment on the 7-Scenes dataset. For each scene, we report the average translational and rotational error as a function of sequence length. As illustrated in Tab. III, our network achieves an improvement of 27.0% in translation and 16.67% in orientation outperforming the aforementioned approaches and thus reinforcing its suitability for visual odometry estimation.

D. Benchmarking

As mentioned in previous works [26], no deep learning-based localization method thus far has been able to match the performance of state-of-the-art local feature-based approaches. In order to gain insights on the performance of VLocNet, we present results on the 7-Scenes dataset, in comparison with Active Search (without prioritization) [22], which is a state-of-the-art SIFT-based localization method. Moreover, as a proof of validation that our trained network is able to regress poses beyond those shown in the training images, we also compare with Nearest Neighbor localization [12]. Tab. IV shows the comparative results of VLocNet against the aforementioned methods.

Local feature-based approaches often fail to localize in textureless scenes due to the inadequate number of correspondences found. In Tab. IV, we denote the number of images for which the localization fails in parenthesis and for a fair comparison we report the average accuracy of Nearest Neighbor and Active Search for only the images that the localization succeeded. The results show that VLocNet outperforms the Nearest Neighbor approach by 90.20% in translation and 70.98% in orientation, in addition to having no localization failures. This is by far the largest improvement achieved by any CNN-based approach. Moreover, VLocNet achieves state-of-the-art performance in comparison to Active Search on four out of the seven scenes, in addition to achieving overall lower average translation error. Thus far, it was believed that CNN-based methods could be used complementary to SIFT-based approaches as they performed better in challenging perceptual conditions but in other cases they were outperformed by SIFT-based methods. We believe that these results have demonstrated the contrary and have shown that CNN-based approaches are not only more robust but also have the potential to outperform local feature-based methods.

E. Architectural Analysis

In this section, we quantitatively analyze the effect of the various architectural decisions made while designing VLocNet. Specifically, we show the performance improvements for the following:

- VLocNet-M1: ResNet-50 base architecture with ReLUs, L^2 Euclidean loss for translation and rotation with $\beta = 1$
- VLocNet-M2: ResNet-50 base architecture with ELUs, L^2 Euclidean loss for translation and rotation with $\beta = 1$
- VLocNet-M3: ResNet-50 base architecture with ELUs and previous pose fusion using $\mathscr{L}_{Geo}$ loss with $\beta = 1$
- VLocNet-M4: ResNet-50 base architecture with ELUs and previous pose fusion using $\mathscr{L}_{Geo}$ loss with $\hat{s}_x$, $\hat{s}_q$

Tab. V shows the median error in pose estimation as an average of all the scenes in the 7-Scenes dataset. We

TABLE V

COMPARATIVE ANALYSIS OF VLOCNET ON THE 7-SCENES DATASET.

Model	Position	Orientation
PoseNet [12]	0.44m	10.4°
VLocNet-M1	0.202m	8.873°
VLocNet-M2	0.197m	8.209°
VLocNet-M3	0.081m	7.860°
VLocNet-M4	**0.048**m	**3.801**°

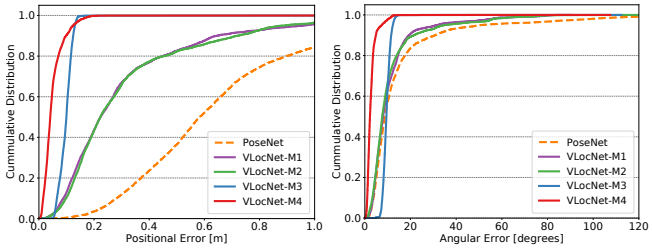

Fig. 2. Qualitative analysis of the localization performance for our proposed VLocNet architecture against PoseNet presented as a cumulative histogram of normalized errors for the Red Kitchen scene.

observe that incorporating residual units in our architecture yields an improvement of 54.09% and 14.68% for the translation and orientation components respectively in comparison to PoseNet. However, the most notable improvement is achieved by fusing the previous pose information using our Geometric Consistency Loss, which can be seen in the improvement of the translational error between VLocNet-M2 and VLocNet-M3. This clearly shows that constricting the search space with the relative pose information while training substantially increases the performance. Furthermore, by utilizing learnable parameters for weighting the translational and rotational loss terms, our network yields a further improvement in performance compared to manually tuning the weighting. In Fig. 2 we show the cumulative histogram error of the aforementioned models trained on the RedKitchen scene. It can be seen that even our base VLocNet model (VLocNet-M1) shows a significant improvement over the baseline method for the translational error. Moreover our final architecture (VLocNet-M4) achieves a rotational error below 10° for 100% of the poses.

We additionally performed experiments to determine the downsampling stage to fuse the previous pose while using our Geometric Consistency Loss function. Fig. 3 shows the median error while fusing the pose at *Res3*, *Res4* and *Res5* in our architecture. It can be seen that fusing at *Res5* where the feature maps are of size 7×7, yields the lowest localization error, while fusing at earlier stages produces varying results for different scenes; either lower translational error at the cost of the orientation or vice versa.

F. Evaluation of Deep Auxiliary Learning

In this section, we evaluate the performance of our jointly trained model using auxiliary learning, along with different optimization strategies that we employed. We explored using both joint and alternating optimization to minimize the loss. We found that the average localization error using an alternate optimization strategy was 28.99% and 18.47% lower in translation and rotation respectively, when compared to a joint optimization. This can be attributed to the difference in scales of the loss values for each task, resulting in the optimizer becoming more biased towards minimizing the global pose regression error at the cost of having suboptimal relative pose estimates. This inadvertently results in worse accuracy for both tasks.

When both global pose regression and visual odometry networks are trained independently, each of them alter the weights of their convolution layers in different ways. Therefore, we evaluated strategies that enable efficient sharing of features between both networks to facilitate the learning of inter-task correlations. Using the model trained on our single-task global pose sub-network (ST) as a baseline, we evaluate the effect of different initializations of the joint model on the localization accuracy. More precisely, we compare the effect of initializing our VLocNet model using weights from: the pretrained task-specific global pose network (MT-GLoc), the pretrained task-specific visual odometry network (MT-VO), and the combined weights from both networks (MT-Dual). Fig. 4 shows the results from this experiment. It can be seen that our joint models that regress relative poses as an auxiliary task, outperform each of the task-specific models, demonstrating the efficacy of our approach. The improvement is most apparent in the Stairs scene which is the most challenging scene in the 7-Scenes dataset as it contains repetitive structures and textureless surfaces. Furthermore, on closer examination, we find that dual initialization of both sub-networks with weights from their task-specific models achieves the best performance, contrary to initializing only one of the sub-networks and learning the other from scratch. Another interesting observation worth noting is that initializing only the global localization stream in the joint network with pretrained weights yields the lowest improvement in pose accuracy compared to the single-task model. This is to be expected as the visual odometry stream does not provide reasonable estimates when the training begins, therefore the localization stream cannot benefit from the motion specific features from the odometry stream.

We summarize the localization performance achieved by our multitask VLocNet incorporating the Geometric Consistency Loss while simultaneously training the auxiliary odometry network in Tab. VI, where we vary the number of shared layers between the global localization and the visual odometry streams. The table shows the median pose error as an average over all the 7-scenes. We experimented with maintaining a shared stream up to the end of *Res2*, end of *Res3* and end of *Res4* in our architecture. The results indicate that the lowest average error is achieved by sharing the streams up to *Res3*, which shows that features learned after *Res3* are highly task-specific and features learned before *Res2* are too generic. In comparison to the single-task VLocNet model, the multitask variant achieves an improvement of 12.5% in translational and 18.49% in rotational components of the pose. We believe that these results demonstrate the utility of learning joint multitask models for visual localization and odometry. A live online demo can be viewed at `http://deeploc.cs.uni-freiburg.de/`.

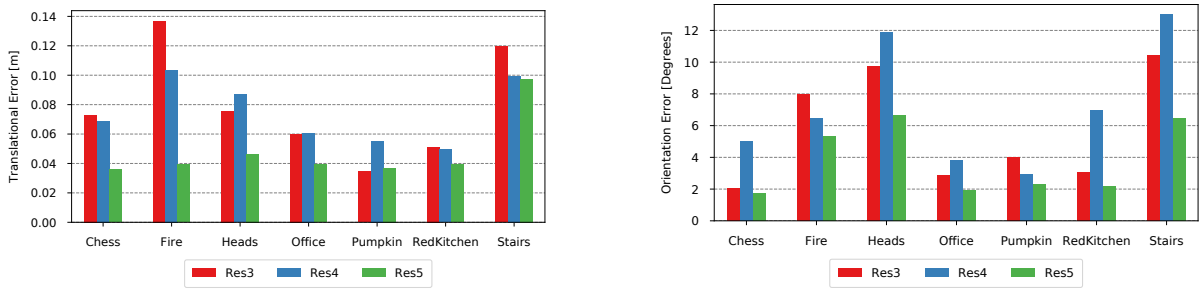

Fig. 3. Comparison of median localization error from fusing previous pose information at various stages in our VLocNet architecture with our proposed Geometric Consistency Loss. The results consistently show that the highest localization accuracy is achieved by fusing the previous predicted pose at *Res5*.

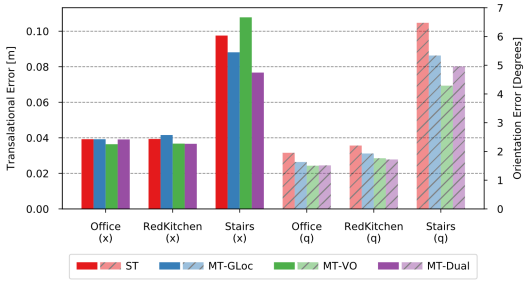

Fig. 4. Performance of our single-task model in comparison to the multitask VLocNet with different weight initializations, on the 7-Scenes dataset. (x) and (q) denote the translation and orientation components.

TABLE VI

SUMMARY OF THE LOCALIZATION PERFORMANCE ACHIEVED BY VLOCNET WITH VARYING AMOUNTS OF SHARING.

	Res2	Res3	Res4
7-Scenes Avg.	0.055m, **2.989**°	**0.042**m, 3.098°	0.053m, 3.174°

V. CONCLUSION

In this paper, we proposed a novel end-to-end trainable multitask DCNN architecture for 6-DoF visual localization and odometry estimation from subsequent monocular images. We present a framework for learning inter-task correlations in our network using an efficient sharing scheme and a joint optimization strategy. We show that our jointly trained localization model outperforms task-specific networks, demonstrating the efficacy of learning visual odometry as an auxiliary task. Furthermore, we introduced the Geometric Consistency Loss function for regressing 6-DoF poses consistent with the true motion model.

Using extensive evaluations on standard indoor and outdoor benchmark datasets, we show that both our single-task and multitask models achieve state-of-the-art performance compared to existing CNN-based approaches, which accounts for an improvement of 80% and 66.69% in translation and rotation respectively. More importantly, our approach is the first to close the performance gap between local feature-based and CNN-based localization methods, even outperforming them in some cases. Overall, our findings are an encouraging sign that utilizing multitask DCNNs for localization and odometry is a promising research direction. As future work, we plan to investigate joint training with more auxiliary tasks such as semantic segmentation and image similarity learning that can further improve the performance.

REFERENCES

[1] M. Abadi, A. Agarwal, P. Barham, *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015.
[2] R. Arandjelovic, P. Gronat, *et al.*, "Netvlad: Cnn architecture for weakly supervised place recognition," in *CVPR*, 2016.
[3] H. Bilen *et al.*, "The missing link between faces, text, planktons, and cat breeds," *arXiv preprint arXiv:1701.07275*, 2017.
[4] R. Clark, S. Wang, *et al.*, "Vidloc: 6-dof video-clip relocalization," *arXiv preprint arXiv:1702.06521*, 2017.
[5] D. Clevert *et al.*, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.
[6] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *IJRR*, vol. 27, no. 6, 2008.
[7] M. Donoser and D. Schmalstieg, "Discriminative feature-to-point matching in image-based localization," in *CVPR*, 2014.
[8] Q. Hao, R. Cai, Z. Li, L. Zhang, Y. Pang, and F. Wu, "3d visual phrases for landmark recognition," in *CVPR*, 2012.
[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2015.
[10] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," *ICRA*, 2016.
[11] ——, "Geometric loss functions for camera pose regression with deep learning," *CVPR*, 2017.
[12] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *ICCV*, 2015.
[13] K. Konda and R. Memisevic, "Learning visual odometry with a convolutional network," in *VISAPP*, 2015.
[14] Z. Laskar, I. Melekhov, S. Kalia, and J. Kannala, "Camera relocalization by computing pairwise relative poses," *arXiv preprint arXiv:1707.09733*, 2017.
[15] I. Melekhov *et al.*, "Relative camera pose estimation using convolutional neural networks," *arXiv preprint arXiv:1702.01381*, 2017.
[16] I. Melekhov, J. Ylioinas, *et al.*, "Image-based localization using hourglass networks," *arXiv preprint arXiv:1703.07971*, 2017.
[17] V. Mohanty *et al.*, "Deepvo: A deep learning approach for monocular visual odometry," *arXiv preprint arXiv:1611.06069*, 2016.
[18] T. Naseer *et al.*, "Deep regression for monocular camera-based 6-dof global localization in outdoor environments," in *IROS*, 2017.
[19] A. Nicolai *et al.*, "Deep learning for laser based odometry estimation," in *RSSws Limits and Potentials of Deep Learning in Robotics*, 2016.
[20] R. Rahmatizadeh *et al.*, "Vision-based multi-task manipulation for inexpensive robots using end-to-end learning," *arXiv:1707.02920*, 2017.
[21] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.
[22] T. Sattler *et al.*, "Efficient effective prioritized matching for large-scale image-based localization," *TPAMI*, vol. 39, pp. 1744–1756, 2017.
[23] 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*, June 2013.
[24] N. Sünderhauf, F. Dayoub, S. Shirazi, *et al.*, "On the performance of convnet features for place recognition," in *IROS*, 2015.
[25] J. Valentin, M. Niener, J. Shotton, A. Fitzgibbon, S. Izadi, and P. Torr, "Exploiting uncertainty in regression forests for accurate camera relocalization," in *CVPR*, 2015.
[26] 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*, 2017.
[27] J. Wu, L. Ma, and X. Hu, "Delving deeper into convolutional neural networks for camera relocalization," in *ICRA*, May 2017.
[28] B. Yu and I. Lane, "Multi-task deep learning for image understanding," in *SoCPaR*, 2014, pp. 37–42.