Faster Visual-Based Localization with Mobile-PoseNet

Claudio Cimarelli, Dario Cazzato, Miguel A. Olivares-Mendez, and Holger Voos

Interdisciplinary Center for Security, Reliability and Trust (SnT), University of Luxembourg, 29, Avenue J. F. Kennedy, 1855 Luxembourg, Luxembourg
{firstname.lastname}@uni.lu

Abstract. Precise and robust localization is of fundamental importance for robots required to carry out autonomous tasks. Above all, in the case of Unmanned Aerial Vehicles (UAVs), efficiency and reliability are critical aspects in developing solutions for localization due to the limited computational capabilities, payload and power constraints. In this work, we leverage novel research in efficient deep neural architectures for the problem of 6 Degrees of Freedom (6-DoF) pose estimation from single RGB camera images. In particular, we introduce an efficient neural network to jointly regress the position and orientation of the camera with respect to the navigation environment. Experimental results show that the proposed network is capable of retaining similar results with respect to the most popular state of the art methods while being smaller and with lower latency, which are fundamental aspects for real-time robotics applications.

Keywords: Deep Learning · Convolutional Neural Networks · 6-DoF Pose Estimation · Visual-Based Localization · UAV

1 Introduction

At the present time, the popularity of Unmanned Aerial Vehicles (UAVs) is rapidly increasing due to their peculiar characteristics. In fact, they are frequently adopted in a broad range of research projects and commercial applications, such as building inspections, rescue operations, and surveillance, which require high mobility and flexible adaptation to complex situations [24]. The ability of a drone to localize itself inside the surrounding environment is crucial for enabling higher degrees of autonomy in the assigned tasks. Global Navigation Satellite System (GNSS) is a common solution to the problem of retrieving a global position, but it often fails due to signal loss in cluttered environments like urban canyons or natural valleys. Moreover, its precision in the localization is correlated with the number of satellites in direct line of sight [3], and the accuracy requirements are often not meet by GPS-like technology since the provided localization comes with an uncertainty up-to some meters.

As an alternative to GPS, Visual-Based Localization (VBL) [25] refers to the set of methods that estimate the 6-Degrees of Freedom (6-DoF) pose of a

camera, that is, its translation and rotation with respect to the map of the navigation environment, solely relying on the information enclosed in the images. In robotics, VBL is commonly used to solve the kidnapped robot problem, whereas in a SLAM pipeline is part of a re-localization module which allows recovering the global position in the map after the tracking is lost or for loop-closing [37]. Visual localization methods can be categorized either as indirect methods, also called topological or appearance-based, or direct methods, sometimes referred to as metric [25]. On the one hand, indirect methods formulate localization as an image-retrieval problem, providing a coarse estimate of the position depending on the granularity of the locations with an image saved in the database [4,40]. On the other hand, direct methods cast localization as a pose regression problem and try to deliver an exact estimate of both position and orientation for each new view [18,31,34]. Thus, direct localization is more appropriate for robot navigation where the operating environment is confined to a well-defined area and we expect to obtain a pose as precise as possible when the tracking is lost.

In this paper, we address the problem of metric localization using as a feature extractor a neural network proposed by the recent research on efficient architectures. In particular, we adopt MobileNetV2 [29] previously trained for the image classification task, as a starting point to build a model for regressing the pose. This choice permits to achieve a trade-off between competitive performance and computation speed. As follows, our contribution is two-fold: from one side, as the best of our knowledge, this is the first attempt to use MobileNetV2 architecture for the localization problem. Moreover, the proposed approach is faster than main state-of-the-art works, while preserving the localization performance. The rest of the manuscript is organized as follows. In Sec. 2, a short review of methods proposed in the recent literature for visual localization is proposed. The methodology, the loss function and the overall structure of the deep learning model are described in Sec. 3. Subsequently, the experimental setup and the obtained results are shown in Sec. 4. Ultimately, Sec. 5 presents the conclusion and future research directions.

2 Related Work

In this section, we review the methods that have been proposed in the recent literature of visual localization techniques. Currently, the approaches to the direct localization problem go into three distinct directions: one is to rely on matching 2D image features with 3D points of a structured model of the environment; another is to use classic machine learning algorithms to learn the 3D coordinates of each the pixels in order to establish the matches; lastly, we can provide an end-to-end differentiable solution to regress the 6-DoF pose using Convolutional Neural Networks (CNNs). Then, we briefly summarize the most efficient neural network architectures for image processing, and the main applications of CNNs to the field of UAVs navigation.

Local feature-based localization is a family of methods usually supported by a 3D reconstruction of the environment created through a Structure-from-Motion

(SfM) pipeline [33]. Hence, they establish correspondences between 2D features extracted from a query image, such as SIFT [23] or ORB [28], and those associated with the 3D points in the model. Finally, the pose of the camera is recovered providing the set of putative matches to a Perspective-n-Point (PnP) algorithm [14] inside a Random Sample Consensus (RANSAC) loop [27].Irschara *et al.* [13] use methods of image retrieval in conjunction with a compressed scene representation composed of real and synthetic views. Li *et al.* [22] propose to invert the search direction using a prioritization scheme. Sattler *et al.* [30] enhance the 2D-3D matching with a Vocabulary-based Prioritized Search (VPS) that estimates the matching cost for each feature to improve the performances, and combine the two opposite search directions [31]. Despite being very precise when correct correspondences are found, the main drawbacks are the computational costs, which does not scale with the extent of the area to cover, and the need to store a 3D model [25].

Scene Coordinates Regression methods use machine learning to speed up the matching phase by directly regressing the scene coordinates of the image pixels. Shotton *et al.* [34] train a random forest on RGB-D images, and formulate the localization problem as an energy function minimization over the possible camera location hypothesis. Hence, they use the Kabsch algorithm [15] inside a RANSAC loop to iteratively refine the hypothesis selection. The downside of these methods is the need for depth maps and of high-resolution images to work well.

Deep Learning has been adopted only recently to solve the direct localization problem. Following the success of neural networks in many computer vision tasks ranging from image classification to object detection [21], PoseNet [18] is the first work in which CNNs are applied to the pose regression task. In particular, they reuse a pre-trained GoogLeNet [36] architecture on the ImageNet dataset [5], demonstrating the ability of the network to generalize to a completely different task thanks to transfer learning [6]. In later works, Walch *et al.* [39] extend PoseNet with LSTM [9] to encode contextual information, and Wu *et al.* [41] generates synthetic pose to augment the training dataset. Subsequently, Kendall *et al.* [16] introduce a novel formulation to remove any weighting hyperparameter from the loss function. Though these single CNN methods for pose regression were not able to surpass the average performance of classical approaches [32,34], they demonstrate themselves capable of handling the most visually difficult frames, being more robust to illumination variance, cluttered areas, and textureless surfaces [39].

Recently, Multi-Task networks [38,26] demonstrate that by leveraging auxiliary task learning, such as Visual Odometry or Semantic Segmentation, the neural network improves on the main task of global localization. As a result, they were able to outperform the state-of-the-art of feature-based and scene coordinate regression methods.

Since the current approaches rely on very deep network architectures, e.g. GoogLeNet, our proposal is to replace them with a more efficient architecture in order to produce a more appealing solution for the deployment on a UAV.

Improving on the previous generation of "mobile" networks [10], MobileNetV2 [29] combine the *depthwise separable convolution* with a *linear bottleneck layer* drastically decreasing the number of operation and weights involved in the computation of the output. In this work, we show that this shallower network is able to run faster than other single CNN solutions without sacrificing the localization accuracy.

3 Methodology

Inspired by previous works on direct visual localization exploiting CNNs [16,41], our aim is to estimate the camera pose from a single RGB image by adding a regressor fed by the output of the network chosen as a base feature extractor. In the following subsections, we describe the representation of the pose vector, the loss function used to learn the task of pose estimation, and the architectural details of the deep learning model.

3.1 Pose Representation

The output for each input image consists of a 7-dimensional vector $\mathbf{p}$, representing both translation and rotation of the camera w.r.t. the navigation environment:

$$\mathbf{p} = [\mathbf{x}, \mathbf{q}] \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^3$, represents the position in the 3D space, and the orientation $\mathbf{q} \in \mathbb{R}^4$ is expressed as a quaternion.

Our choice of using a quaternion over other representations for the orientation is motivated by the fact that any 4-dimensional vector can be mapped to a valid rotation by scaling its norm to unit length. Instead, opting for rotation matrices would require to enforce the orthonormality constraint, since the set of rotation matrices belongs to the special orthogonal Lie group, $SO(3)$ [16]. Other representations, such as Euler angles and axis-angle, suffer from the problem of periodic repetition of the angle values around 2π.

However, Wu *et al.* [41] proposed a variant of the Euler angles, named Euler6, to overcome the issue of periodicity in which they regress a 6-dimensional vector $e = [sin\phi, cos\phi, sin\theta, cos\theta, sin\psi, cos\psi]$. Notwithstanding in [41] the authors showed empirically an improvement over quaternions, we decided not to express the rotation as Euler6 for a closer comparison with the majority of the state-of-the-art approaches. Anyway, in Section 4 we also compare our solution with the aforementioned work.

3.2 Loss Function

In order to train the network for the task of pose estimation, we minimize the difference between the ground truth pose, $[\mathbf{x}, \mathbf{q}]$, associated with an image $\mathcal{I}$

in the training dataset, and the pose predicted by the deep learning model, $[\hat{\mathbf{x}}, \hat{\mathbf{q}}]$. Hence, the loss function aims to optimize the two components of the pose, translation and orientation, denoted by $\mathcal{L}_x$ and $\mathcal{L}_q$ respectively:

$$\mathcal{L}_x(\mathcal{I}) = \left\|\mathbf{x} - \hat{\mathbf{x}}\right\|_p \tag{2}$$

$$\mathcal{L}_q(\mathcal{I}) = \left\|\mathbf{q} - \frac{\hat{\mathbf{q}}}{\|\hat{\mathbf{q}}\|}\right\|_p \tag{3}$$

where with the notation $\|\cdot\|_p$ we refer to the p-norm. In our experiments, we apply $p = 2$, which corresponds to the Euclidean norm. Besides, the predicted quaternion is normalized to unit length to ensure a valid rotation representation.

Even though the Euclidean norm is a valid metric for 3D translation vectors, in the case of quaternions it does not take in consideration that the valid rotations lie on the unit 3-sphere, and that mapping from unit quaternion to the SO(3) group is 2-to-1 [11]. However, Kendall *et al.* [18] argue that, as the difference between the predicted and ground truth quaternions decreases, the Euclidean distance converges to the spherical distance.

Since the two components, $\mathcal{L}_x$ and $\mathcal{L}_q$ of the loss function that we want to minimize is on a different scale, a weight β is added to the quaternion error in order to balance the backpropagated gradient magnitude [18]. In light of this, the loss function is defined as:

$$\mathcal{L}(\mathcal{I}) = \mathcal{L}_x(\mathcal{I}) + \beta \cdot \mathcal{L}_q(\mathcal{I}) \tag{4}$$

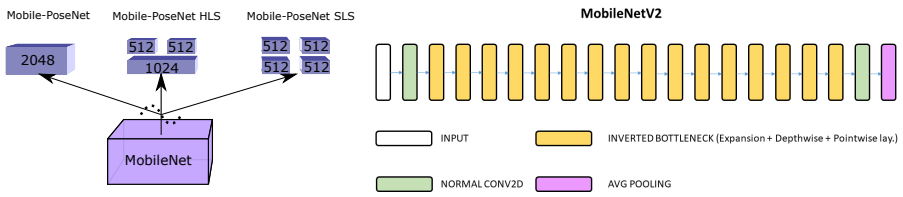
In order to remove any hyperparameter from the loss function, [16] replaced β with two learnable variables, $\hat{s}_x$ and $\hat{s}_q$, in the formulation of the loss with *homoscedastic uncertainty*:

$$\mathcal{L}(\mathcal{I}) = \mathcal{L}_x(\mathcal{I}) \cdot exp(-\hat{s}_x) + \hat{s}_x + \mathcal{L}_q(\mathcal{I}) \cdot exp(-\hat{s}_q) + \hat{s}_q \tag{5}$$

Homoscedastic uncertainty captures the uncertainty of the model relative to a single task, for example, treating the regression of translation and rotation as two separated tasks, while learning multiple objectives at the same time. For this reason, is useful in multitask settings to weight the loss components based on the different measurement units relative to the particular task [17]. In our experiments, we initialized $\hat{s}_x$ and $\hat{s}_q$ to 0.5 and 0.1 respectively.

3.3 Deep Learning Model

In order to build a small network for localization, we decided to adapt the novel MobileNetV2 [29] by adding fulling connected layers to regress the pose; for this reason, we refer to our proposed network as *Mobile-PoseNet*. MobileNetV2 is an architectural design for neural networks that leverages efficient convolution operations, namely the *depthwise separable convolution*, and a novel layer, the *linear bottleneck* with *inverted residual* block, to produce a light weight network with optimized computation time.



(a) Branching of the fully connected layers.

(b) Architecture of MobileNetV2. For more details on its structure refer to the original paper [29].

Fig. 1: Mobile-PoseNet's architecture.

The *depthwise separable convolution* reduces the number of parameters and of Multiply-Adds (MAdds) operations by decomposing the standard convolution operation with N filters of size $D_K \times D_K \times N$ into two steps: *depthwise convolution* and *pointwise convolution*. Having an input with M channels, the *depthwise convolution* is composed of M filters of size $D_K \times D_K \times 1$, operating on each m_{th} input channel separately. Then, the *pointwise convolution* applies N filters of size $1 \times 1 \times M$ to combine the channels into new features [10]. In addition, MobileNetV2's authors reformulate the original *residual block* [8], which is used to support the propagation of the gradient through deep stacked layer. On the one hand, they remove the non-linearity at the shortcut connected layers, where the *residual function* is computed, so that more information is preserved. On the other hand, they apply the shortcut connections directly at the bottleneck instead of the expansion layer; in this way, the authors assert, the memory footprint can be drastically reduced. Ultimately, MobileNetV2 allows tuning a *width multiplier* α in order to choose the preferred trade-off between accuracy and size of the network. We set $\alpha = 1$ to obtain a network with 3.4M parameters and 300M MAdds, resulting in a sensible shrinking compared to GoogLeNet with 6.8M parameters and 1500M MAdds.

Thus, we perform an average pooling on the output of MobileNetV2 last convolutional layer, deriving a vector of 1x1280 dimension that contains an high-dimensional feature representation of the input image. Therefore, we connect a fully connected layer of 2048 neurons followed by a *ReLu6* [20] non-linearity, which maps the features to the desired 7-dimensional pose vector. ReLu6, as stated by the authors, helps to learn a sparse feature representation earlier in the training. More importantly, it can be exploited to optimize fixed-point low-precision calculations [10].

Furthermore, to improve the generalization capability of the network, we add a *Batch Normalization* layer [12] before the non-linearity. Hence, this layer learns how to shift the mean and variance of the input batches after normalizing them. In addition, we adopt *Dropout* [35], which is an alternative form of activation regularization that reduces overfitting and indirectly induces sparsity by dropping random neurons at training time.

Ultimately, we test the branching technique proposed in [41] to regress the translation and rotation vectors separately (see Figure 1a). Hence, we symmetrically split the neurons into two groups of 1024, so that we maintain the same total number intact. Additionally, we experiment with a third version of the network that keeps a common fully connected layer for translation and rotation of 1024 neurons and splits in half the rest forming two groups of 512. Our purpose is to compare the benefits of jointly learning position and orientation, that is, sharing the information enclosed in the common weights, against training two individual branches for each task. Therefore, we distinguish these design choices by referring to the first as *symmetric layer split (SLS)*, and to the latter as *half layer split (HLS)*.

4 Experiments and Results

In this section, we evaluate our proposed solutions on two datasets, *7-Scenes* [34] and *Cambridge Landmarks* [18]. The first includes indoor images, whereas the second one contains pictures captured in an outdoor urban environment. They have been chosen to demonstrate how the proposed method behaves in scenarios showing opposite characteristics.

4.1 Datasets

(a) Chess (b) Fire (c) Heads (d) Old Hospital (e) Street

Fig. 2: *7-Scenes* and *Cambridge Landmarks* sample images

7-Scenes [34] is a dataset for RGB-D designed to benchmark relocalization methods. Thus, it was collected through a Microsoft Kinect camera in seven indoor scenarios, which contains in total more than 40k frames with 640x480 resolution and an associated depth map. The challenging aspects of this dataset are its high variations in the camera pose in a small area generating motion blur, perceptual aliasing, and light reflections. These unique characteristics make the pose estimation particularly difficult for methods relying on handcrafted features, especially in views where textured areas are not clearly distinguishable [39].

Cambridge Landmarks was introduced in [18], and currently provides six outdoor scenarios. It contains more than 10k images, sampled from a high-resolution video captured by a smartphone. The ground truth labels were generated through an SfM reconstruction of the environment. Visual clutter caused

by the presence of pedestrians and vehicle plus a substantial variance in the lighting conditions are the main challenges posed by this dataset.

All the scenes in both datasets are subdivided in sequences, depending on the trajectories from which they were generated. In fact, each of the sequences shows a different perspective of the surrounding environment. Hence, for training and testing our model, we use the same partitioning of the datasets as provided by the respective authors. Thus, we create a separate "dev" set for evaluating the models during the training phase by taking a random sample of 10% of the frames from all the sequences in the training set. Anyway, we prefer to form the "dev" set from trajectories that are unseen in the training set in case we found a number of sequences high enough for a specific dataset scene; the purpose is to estimate more accurately the performance on the test set and choose wisely the parameters and stopping criteria for training.

4.2 Experimental Setup

The network is implemented using the TensorFlow-Slim open-source library [1,2]. We initialized MobileNet with weights pre-trained on the ImageNet dataset, and the fully connected layers using the method proposed by He *et al.* [7]. Before training, we normalize the images by computing an RGB image that represents the standard deviation and the mean of a particular dataset scene. Then, for each image, we remove the mean and divide by the standard deviation in order to center the data and uniformly scale the pixel intensities. Dropout rate is set to 0.1, which means only 10% of the neurons are turned off during training, whereas Batch Normalization momentum is set to 0.99. We optimized the models using Adam [19] with a learning rate $\alpha = 1e^{-4}$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$, on batches of size 128 shuffled at each new epoch, using an NVIDIA Tesla V100 16GB. Thus, we let the training last until the convergence of the loss is reached on the "dev" set.

4.3 Discussion of the results

In Table 1, we compare the results with three other CNN-based localization methods: PoseNet [18], PoseNet2 [16] with learned σ^2 weights in the loss, and BranchNet [41], which represents rotations with Euler6 and splits the network in two branches in order to regress the position and the orientation separately. Whereas we benchmark our result against PoseNet [18] because it pioneered the approach to the direct localization problem using CNNs, we share with the other methods some architectural choices. On the one hand, we adopt the *homoscedastic uncertainty* introduced by [16] to balance different loss components; on the other hand, we split the network layers following the work of [41], who showed significant improvements.

In general, it is evident that Mobile-PoseNet is able to outperform PoseNet and BranchNet in most of the scenarios. Interestingly, the more complex loss function is the main factor that gives us an advantage over these methods and is able to fill the initial gap between the two base networks. In fact, we noted

Table 1: Median localization error on the 7-Scenes and Cambridge Landamarks datasets. The large error reported in the Street dataset is possibly due to the repeated structure of the buildings and to the wide covered area.

	Area or Volume	BranchNet[41] Euler6	PoseNet[18] β Weight	PoseNet2[16] Learn σ^2 Weights	Mobile-PoseNet (proposed)	Mobile-PoseNet *HLS* (proposed)	Mobile-Posenet *SLS* (proposed)
7-Scenes							
Chess	$6m^3$	0.20m, 6.55°	0.32m, 8.12°	0.14m, 4.50°	0.17m, 6.78°	0.18m, 7.27°	0.19m, 8.22°
Fire	$2.5m^3$	0.35m, 11.7°	0.47m, 14.4°	0.27m, 11.8°	0.36m, 13.0°	0.36m, 13.6°	0.37m, 13.2°
Heads	$1m^3$	0.21m, 15.5°	0.29m, 12.0°	0.18m, 12.1°	0.19m, 15.3°	0.18m, 14.3°	0.18m, 15.5°
Office	$7.5m^3$	0.31m, 8.43°	0.48m, 7.68°	0.20m, 5.77°	0.26m, 8.50°	0.28m, 8.98°	0.27m, 8.54°
Pumpkin	$5m^3$	0.24m, 6.03°	0.47m, 8.42°	0.25m, 4.82°	0.31m, 7.53°	0.38m, 9.30°	0.34m, 8.46°
Red Kitchen	$18m^3$	0.35m, 9.50°	0.59m, 8.64°	0.24m, 5.52°	0.33m, 7.72°	0.33m, 9.19°	0.31m, 8.05°
Stairs	$7.5m^3$	0.45m, 10.9°	0.47m, 13.8°	0.37m, 10.6°	0.41m, 13.6°	0.48m, 14.4°	0.45m, 13.6°
Cambridge Landmarks							
Great Court	$8000m^2$	—	—	7.00m, 3.65°	8.68m, 6.03°	8.12m, 5.60°	8.60m, 5.58°
King's College	$5600m^2$	—	1.92m, 5.40°	0.99m, 1.06°	1.13m, 1.57°	1.20m, 1.79°	1.14m, 1.53°
Old Hospital	$2000m^2$	—	2.31m, 5.38°	2.17m, 2.94°	3.11m, 4.11°	2.13m, 3.73°	2.62m, 4.21°
Shop Façade	$875m^2$	—	1.46m, 8.08°	1.05m, 3.97°	1.39m, 6.37°	1.55m, 5.64°	1.73m, 6.19°
St. Mary's Church	$4800m^2$	—	2.65m, 8.48°	1.49m, 3.43°	2.34m, 6.23°	2.16m, 5.97°	2.18m, 6.01°
Street	$50000m^2$	—	—	20.7m, 25.7°	22.9m, 36.3°	22.6m, 32.6°	22.9m, 36.2°

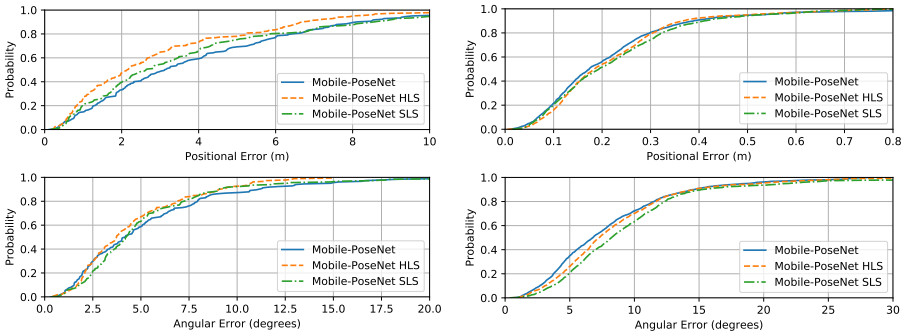
poor performances applying the β weighted loss to MobileNet during experiments. Instead, PoseNet2 obtains the best results in all the benchmark scenes apart from *Old Hospital* in which Mobile-PoseNet *HLS* is able to surpass the translation error by a small margin. However, we note that PoseNet2 uses frames with a resolution of 256x256, whereas our models require an input of 224x224 pixel images. Moreover, we do not augment the dataset trough random crops of the original images as in the competing approaches. Performing such operation would add an additional regularization effect, thus helping the generalization capabilities of the model and resulting in better performances overall. Besides, we observe that Mobile-PoseNet perform better on scenes spread on smaller areas overall. In contrast, Mobile-PoseNet *HLS* competitively gains higher scores in the scenarios of Cambridge Landmarks with an elevated spatial extent.

Finally, we run the network on a TegraTX2 to test the latency, that is, the time interleaving from the submission of one frame into the network to the moment of receiving the estimated pose. Hence, using the integrated TensorFlow tool for run-time statistics, we note that MobileNet-PoseNet takes on average 17.5ms of run time, while the classic PoseNet 24ms.

At last, we want to remark that the proposed solution employs a base feature extractor that carries half the number of parameters, in contrast to the aforementioned state-of-the-art methods with which we compare. This factor contributes to the lower accuracy of the output.

5 Conclusion

In this paper, we introduce an efficient Convolutional Neural Network to solve the localization problem. In particular, we adapt MobileNetV2 with regressor layers to estimate the 6-DoF pose and propose a double modification of the



(a) Old Hospital (b) Chess

Fig. 3: Cumulative probability distribution of the localization error.

architectural design by symmetrically splitting the neurons in the fully connected layer for learning independently the orientation and rotation. Comparison with state-of-the-art methods using a single CNN for direct pose regression shows that our method achieves competitive results, in spite of using a shallower network for feature extraction. In fact, contrary to the other approaches that make use of GoogLeNet, we employ MobileNetV2, which results in a faster and more suitable localization solution for being deployed on-board of a UAV.

Notwithstanding the empirical results in favor of using the Euclidean norm to compute the quaternion error, for future works, we will investigate the combination of quaternion with a different metric in the loss function or to adopt a totally different representation for the rotation.

References

1. MobileNetV2 source code, `https://github.com/tensorflow/models/tree/master/research/slim/nets/mobilenet`
2. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), `https://www.tensorflow.org/`, software available from tensorflow.org
3. Araar, O., Aouf, N.: A new hybrid approach for the visual servoing of vtol uavs from unknown geometries. In: 22nd Mediterranean Conference on Control and Automation. pp. 1425–1432. IEEE (2014)
4. Cummins, M., Newman, P.: Fab-map: Probabilistic localization and mapping in the space of appearance. The International Journal of Robotics Research **27**(6), 647–665 (2008)

5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. IEEE Conference on Computer Vision and Pattern Recognition pp. 248–255 (2009)
6. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. In: International conference on machine learning. pp. 647–655 (2014)
7. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1026–1034 (2015)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
10. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
11. Huynh, D.Q.: Metrics for 3d rotations: Comparison and analysis. Journal of Mathematical Imaging and Vision **35**(2), 155–164 (2009)
12. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
13. Irschara, A., Zach, C., Frahm, J.M., Bischof, H.: From structure-from-motion point clouds to fast location recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2599–2606. IEEE (2009)
14. Josephson, K., Byrod, M.: Pose estimation with radial distortion and unknown focal length. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2419–2426. IEEE (2009)
15. Kabsch, W.: A solution for the best rotation to relate two sets of vectors. Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography **32**(5), 922–923 (1976)
16. Kendall, A., Cipolla, R.: Geometric loss functions for camera pose regression with deep learning. In: Camera relocalization by computing pairwise relative poses using convolutional neural network. pp. 5974–5983 (2017)
17. Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 7482–7491 (2018)
18. Kendall, A., Grimes, M., Cipolla, R.: Posenet: A convolutional network for real-time 6-dof camera relocalization. In: IEEE International Conference on Computer Vision (December 2015)
19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
20. Krizhevsky, A., Hinton, G.: Convolutional deep belief networks on cifar-10. Unpublished manuscript **40**(7) (2010)
21. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436 (2015)
22. Li, Y., Snavely, N., Huttenlocher, D.P.: Location recognition using prioritized feature matching. In: European conference on computer vision. pp. 791–804. Springer (2010)
23. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)
24. Lu, Y., Xue, Z., Xia, G.S., Zhang, L.: A survey on vision-based uav navigation. Geo-spatial information science **21**(1), 21–32 (2018)

25. Piasco, N., Sidibé, D., Demonceaux, C., Gouet-Brunet, V.: A survey on visual-based localization: On the benefit of heterogeneous data. Pattern Recognition **74**, 90–109 (2018)
26. Radwan, N., Valada, A., Burgard, W.: Vlocnet++: Deep multitask learning for semantic visual localization and odometry. IEEE Robotics and Automation Letters **3**(4), 4407–4414 (2018)
27. Raguram, R., Frahm, J.M., Pollefeys, M.: A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus. In: European Conference on Computer Vision. pp. 500–513. Springer (2008)
28. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE (2011)
29. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 4510–4520 (2018)
30. Sattler, T., Leibe, B., Kobbelt, L.: Fast image-based localization using direct 2d-to-3d matching. In: International Conference on Computer Vision. pp. 667–674. IEEE (2011)
31. Sattler, T., Leibe, B., Kobbelt, L.: Improving image-based localization by active correspondence search. In: European conference on computer vision. pp. 752–765. Springer (2012)
32. Sattler, T., Leibe, B., Kobbelt, L.: Efficient & effective prioritized matching for large-scale image-based localization. IEEE transactions on pattern analysis and machine intelligence **39**(9), 1744–1756 (2017)
33. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 4104–4113 (2016)
34. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in rgb-d images. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2930–2937 (2013)
35. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research **15**(1), 1929–1958 (2014)
36. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition (2015)
37. Taketomi, T., Uchiyama, H., Ikeda, S.: Visual slam algorithms: A survey from 2010 to 2016. IPSJ Transactions on Computer Vision and Applications **9**(1), 16 (2017)
38. Valada, A., Radwan, N., Burgard, W.: Deep auxiliary learning for visual localization and odometry. In: IEEE International Conference on Robotics and Automation. pp. 6939–6946. IEEE (2018)
39. Walch, F., Hazirbas, C., Leal-Taixe, L., Sattler, T., Hilsenbeck, S., Cremers, D.: Image-based localization using lstms for structured feature correlation. In: Camera relocalization by computing pairwise relative poses using convolutional neural network. pp. 627–637 (2017)
40. Weyand, T., Kostrikov, I., Philbin, J.: Planet-photo geolocation with convolutional neural networks. In: European Conference on Computer Vision. pp. 37–55. Springer (2016)
41. Wu, J., Ma, L., Hu, X.: Delving deeper into convolutional neural networks for camera relocalization. In: IEEE International Conference on Robotics and Automation. pp. 5644–5651. IEEE (2017)