

How to improve CNN-based 6-DoF camera pose estimation

Soroush Seifi Tinne Tuytelaars
 PSI, ESAT, KU Leuven
 Kasteelpark Arenberg 10, 3001 Leuven, Belgium
 {sseifi, tinne.tuytelaars}@esat.kuleuven.be

Abstract

Convolutional neural networks (CNNs) and transfer learning have recently been used for 6 degrees of freedom (6-DoF) camera pose estimation. While they do not reach the same accuracy as visual SLAM-based approaches and are restricted to a specific environment, they excel in robustness and can be applied even to a single image. In this paper, we study PoseNet [1] and investigate modifications based on datasets' characteristics to improve the accuracy of the pose estimates. In particular, we emphasize the importance of field-of-view over image resolution; we present a data augmentation scheme to reduce overfitting; we study the effect of Long-Short-Term-Memory (LSTM) cells. Lastly, we combine these modifications and improve PoseNet's performance for monocular CNN based camera pose regression.

1. Introduction

The performance of many computer vision applications, such as autonomous vehicle navigation, augmented reality, and mobile robotics, heavily depends on good localization of the system with respect to its environment [2, 3, 4]. The recent success of CNNs in related tasks such as image classification and object detection [5, 6, 7] has led researchers to explore learning based solutions for place recognition [8] and camera pose estimation [9]. This seems promising given the ability of CNNs to learn high dimensional representations of the input data, automatically selecting the optimal set of features to accurately regress the camera pose.

One of the main obstacles for training a neural network for a supervised task such as camera pose estimation is the need for abundant labeled data. Fortunately, it has been demonstrated that transfer learning is effective in reducing the need for large labeled datasets [10], [11]. In particular, representations learned by a CNN on a large image classification dataset can be fine-tuned to solve the camera pose estimation problem with much smaller datasets.

The authors of PoseNet [9] leverage CNNs and transfer learning and propose a pure neural network based solution to 6-DoF camera pose estimation (i.e., 3D translation and

3D rotation) for a specific environment, addressing some limitations of traditional vSLAM algorithms [12]. However, the accuracy obtained with this architecture is still significantly below what can be obtained with vSLAM methods, especially if the latter are trained on full sequences. In this study, we investigate how this gap can be reduced. In particular, we explore three possible causes, namely i) cropping of the input images, ii) overfitting to training data and iii) neglecting temporal information. Each time, we propose remedies to alleviate these shortcomings and evaluate their effectiveness according to each dataset's characteristics. In particular, we show the importance of the input image's field-of-view in comparison to its resolution. Second, we reduce the extent to which overfitting affects the performance by introducing a specific scheme for Data Augmentation (DA). Third, we demonstrate the benefits of using Long-Short-Term-Memory cells (LSTMs) over Fully Connected layers (FC). Finally, we incorporate all these techniques to improve PoseNet's performance for camera relocalization.

The remainder of this paper is organized as follows. Section 2 describes related work. Next, in section 3 we give more details on the PoseNet architecture, the loss functions we use and the datasets. Section 4 contains the main contributions of our work. Section 5 concludes the paper.

2. Related Works

Visual SLAM algorithms rely solely on the images coming from a camera, typically with a limited field of view, and do not use any other input such as GPS or inertial sensors. This remains an active area of research in the computer vision community [13], [14]. Although various solutions for vSLAM have been proposed for different environments and applications, many of them share a similar structure and therefore similar limitations [15], [16], [17], [18]: They often lose track due to motion blur, high speed rotations, partial occlusions and presence of dynamic objects in the scene. This makes them unsuitable for demanding applications such as localization of Unmanned Aerial Vehicles (UAVs). Besides, most visual SLAM algorithms rely on

expensive pipelines that require a database of hand-crafted features, the camera’s intrinsic parameters, a good initialization of the algorithm, selecting and storing key-frames and finding feature correspondences among images. In addition, for monocular images, these approaches suffer from a phenomenon known as scale drift where the scale of the objects in the environment cannot be accurately inferred, resulting in inconsistent camera trajectories [19].

To alleviate these problems an end-to-end trainable architecture for camera pose estimation called PoseNet is proposed [9, 20]. The authors of these works modified GoogleNet [21] and leveraged transfer learning from ImageNet [22] classification task to train a network for pose prediction using only monocular images. They further improved its performance in [1] by introducing more sophisticated loss functions for optimization.

[23] extends PoseNet using LSTM cells to better exploit the spatial information in each image. To this end, the last layer’s features from GoogleNet are reshaped in 2D and the rows/columns of the corresponding matrix are fed to LSTM cells, one at each timestep. Finally, the cell’s output for the last timestep is used to predict the 6-DoF pose.

LSTMs have also been used in [24] to exploit the temporal information between consecutive frames for a better localization accuracy. In this case, a bi-directional LSTM is fed with the features and pose information of the frames before and after the current frame to predict the current pose. One limitation of this approach is that using bi-directional LSTMs requires access to future frames at each timestep, which is not possible in real-time online applications. Furthermore, the performance is evaluated only for regressing the position and not the orientation.

Although PoseNet and its family of algorithms are not as accurate as vSLAM algorithms mentioned before, they work on monocular images and are shown to be more robust to motion blur and changes in the lighting conditions. Furthermore, unlike traditional vSLAM solutions, they do not require access to camera parameters, good initialization and hand-crafted features.

Posenet’s successor architectures improve its performance by making its architecture more complex while neglecting the effect of the data on the final performance. In this paper, we try to improve the performance for PoseNet family of algorithms by targeting the information that can still be gained according to datasets’ attributes without imposing a more complex CNN architecture. Such modifications can be applied to the above mentioned works as well as those proposed in [25, 26, 27] to further improve their performance.

3. Background

In this section, we provide some background information on the architecture we used as the baseline for our experi-

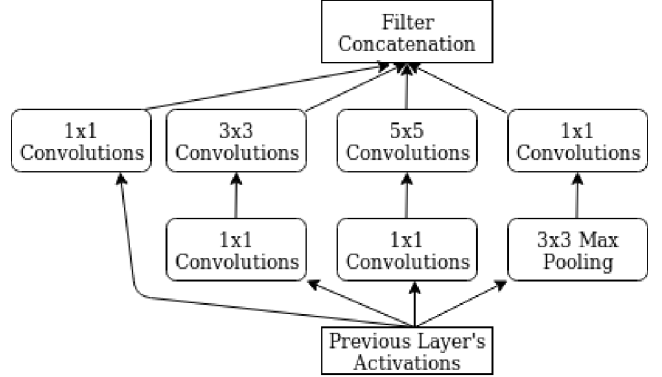


Figure 1. The inception module.

ments, largely building on [1]. We modify this baseline in the next sections to improve the localization accuracy.

3.1. GoogleNet and Posenet

Originally designed for object classification and detection, GoogleNet [21] is a 22 layer deep neural network based on a module known as Inception. An illustration of the inception module can be found in Figure 1. GoogleNet takes as input an image of 224×224 pixels and propagates it through 9 inception modules stacked on top of each other using Rectified Linear Units (ReLU) as the activation function. Each layer in such a network learns a further abstraction of the input data. The highest level abstraction - which resides on the last layer of the network - along with two intermediate abstractions are fed to fully connected and softmax layers to predict the objects’ classes.

Posenet replaces these softmax classification layers with two parallel fully connected layers with 3 and 4 units respectively. These regress to pose (represented by (x, y, z) coordinates) and orientation (represented as a quaternion). Furthermore, a 2048 units fully connected layer is added on top of the last inception module. Figure 2 illustrates PoseNet’s architecture. The yellow blocks represent the pretrained modules that PoseNet inherits from GoogleNet. The green blocks show PoseNet-specific modules that need to be trained from scratch.

3.2. Loss Function and Optimization

The network described above, outputs a vector x' and a quaternion q' to represent the estimated position and orientation respectively. The parameters of the network are optimized for each image I using the loss function:

$$Loss(I) = ||x - x'||_2 + \beta ||q - q'||_2$$

where x and q represent the groundtruth position and orientation. Since quaternions are constrained to the unit manifold, the orientation error is typically much smaller than the position error. Therefore, a constant scale factor β is used for balancing the loss terms.

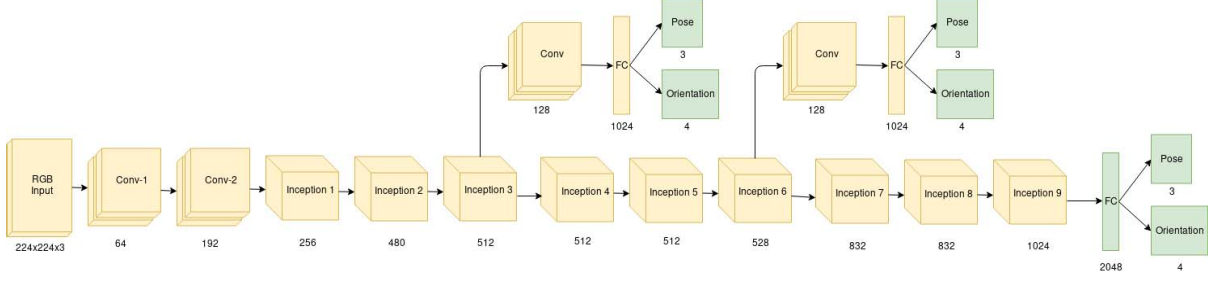


Figure 2. The Posenet architecture. Yellow modules are shared with GoogleNet while green modules are specific to Posenet.

The authors of PoseNet replace this constant scale factor with an adaptive one in [1]. The new loss function is formulated using homoscedastic uncertainty:

$$Loss(I) = ||x - x'||_2 \times e^{(-\hat{s}_x)} + \hat{s}_x + ||q - q'||_2 \times e^{(-\hat{s}_q)} + \hat{s}_q \quad (1)$$

where $\hat{s} := \log \hat{\sigma}^2$ is a free scalar value trained by back propagation and $\hat{\sigma}^2$ denotes the homoscedastic uncertainty. The uncertainty term $\hat{\sigma}_q^2$ is typically smaller than $\hat{\sigma}_x^2$, resulting in a larger weighting factor for the orientation loss term and a well-balanced loss function.

For all experiments in this paper we optimize the above-mentioned adaptive loss function with the Adam optimizer [28]. The learning rate, β_1 , β_2 and ϵ for the Adam optimizer are set to 0.0001, 0.9, 0.999 and 1e-08 respectively.

3.3. Datasets

Following PoseNet and its successor architectures [1][9][20][23][24], we report our results on the Cambridge Landmarks [9] and 7-Scenes [29] datasets with median error values. It is worth mentioning that these two datasets differ both in their scale and data. Cambridge Landmarks consist of 1920×1080 images captured outdoors using a phone camera. The precise temporal information is lost as a result of the frame stream being sampled and many frames being removed. The labels for this dataset are produced using Structure from Motion. The 7-Scenes dataset is recorded indoors using a Kinect RGB-D camera at a lower 640×480 resolution. All frames are kept and the labels are produced by a KinectFusion system [30].

The Cambridge Landmarks dataset covers larger areas in volume with relatively smaller number of frames compared to the sequences in the 7-Scenes dataset. Therefore, pose prediction in the Cambridge Landmarks dataset is more closely related to place recognition while in the 7-Scenes dataset there is potential to exploit the temporal information for continuous localization. In the following sections we sometimes refer to these datasets as outdoor and indoor datasets, respectively. Example frames for all sequences of both datasets are shown in Figure 4.

4. Method and Experiments

4.1. Increased Field-of-View

One drawback to transfer learning and using pretrained networks is that it imposes strong restrictions in terms of the network architecture. As an example, changing the input size is not an option. Consequently, given that GoogleNet takes 224×224 RGB images as its input, PoseNet resizes the smallest dimension of each image to 256 while keeping the original aspect ratio and crops a centered 224×224 window of the resulting image to train the network. In another training scheme, the network is trained on random crops of each image. In both of these cases localization accuracy is affected since the information outside cropping boundaries are lost.

Instead, we propose to use the entire field of view, by simply rescaling the input image to 224×224 pixels, even if that leads to a different aspect ratio. We hypothesize that the loss of aspect ratio should not affect the performance of the network much, since the new aspect ratio is consistent for all images in the dataset. Besides, field-of-view is of more importance compared to image resolution since the pooling layers in the network smooth the high frequency details of a higher-resolution image.

Figure 3 shows an example of the difference in the field-of-view for the input to PoseNet and our network. The left column represents two images which their positions are 12 meters away from each other. The middle column shows the crop selected by PoseNet as the network’s input while the right column illustrates our proposed alternative. As can be seen in the figure, PoseNet’s field-of-view (middle column/the green rectangles in the right column) consists of dynamic objects and roughly similar distant landmarks which might not be very helpful for accurate localization. However, when considering the full field of view, as we suggest, closer and more useful landmarks for pose estimation, as highlighted by the red rectangles, come into view and can be exploited by the network.

Results in Table 1 confirm that keeping the whole field of view in the image performs better than using a higher resolution crop with the original aspect ratio. This modification is specifically more in favor of the outdoor datasets where



Figure 3. From left to right: Original input images, PoseNet’s input and our input. PoseNet (middle/green rectangle) can lose important information for localization (highlighted in red).

Dataset	Centered Crop	Whole Field of View	Improvement
King’s College	1.24m, 1.84°	0.97m, 1.27°	21.7%, 30.9%
Old Hospital	3.07m, 5.13°	3.10m, 4.94°	-0.9%, 3.7%
Shop Facade	1.23m, 5.74°	0.93m, 4.25°	24.3%, 25.9%
St Mary’s Church	2.21m, 5.92°	1.66m, 4.24°	24.8%, 28.3%
Street	21.67m, 32.8°	14.88m, 24.35°	31.3%, 25.7%
Average	5.88m, 10.28°	4.30m, 7.81°	20.2%, 24.7%
Chess	0.18m, 5.92°	0.16m, 4.84°	11.1%, 18.2%
Fire	0.40m, 12.21°	0.35m, 12.10°	12.5%, 0.9%
Heads	0.24m, 14.20°	0.20m, 13.17°	16.6%, 7.2%
Office	0.30m, 7.59°	0.25m, 6.39°	16.6%, 15.8%
Pumpkin	0.34m, 6.04°	0.27m, 5.53°	20.5%, 8.4%
Red Kitchen	0.36m, 7.32°	0.30m, 6.21°	16.6%, 15.1%
Stairs	0.35m, 13.11°	0.43m, 12.86°	-22.8%, 1.9%
Average	0.31m, 9.48°	0.28m, 8.72°	10.1%, 9.6%

Table 1. Effect of increased field of view on localization accuracy

Dataset	Baseline	Baseline-Augmented	Whole view-Augmented	Improvement (Column 2&4)
King’s College	1.24m, 1.84°	1.46m, 3.60°	1.34m, 3.80°	-8.0%, -106.5%
Old Hospital	3.07m, 5.13°	2.56m, 7.54°	2.64m, 6.50°	14.0%, -26.7%
Shop Facade	1.23m, 5.74°	1.31m, 7.14°	1.23m, 7.33°	0.0%, -27.7%
St Mary’s Church	2.21m, 5.92°	2.30m, 7.66°	1.80m, 5.85°	18.5%, 1.1%
Street	21.67m, 32.8°	17.63m, 34.86°	13.77m, 27.09°	36.4%, 17.4%
Average	5.88m, 10.28°	5.05m, 12.16°	4.15m, 10.11°	12.1%, -27.1%
Chess	0.18m, 5.92°	0.20m, 5.08°	0.17m, 4.65°	16.6%, 21.4%
Fire	0.40m, 12.21°	0.37m, 9.86°	0.34m, 8.80°	15.0%, 27.9%
Heads	0.24m, 14.20°	0.19m, 11.57°	0.16m, 9.43°	33.3%, 33.5%
Office	0.30m, 7.59°	0.29m, 7.50°	0.27m, 6.93°	10.0%, 8.6%
Pumpkin	0.34m, 6.04°	0.30m, 5.54°	0.24m, 4.86°	29.4%, 19.5%
Red Kitchen	0.36m, 7.32°	0.32m, 6.73°	0.29m, 5.82°	19.4%, 20.4%
Stairs	0.35m, 13.11°	0.42m, 5.97°	0.38m, 6.36°	-8.5%, 43.8%
Average	0.31m, 9.48°	0.29m, 7.46°	0.26m, 6.69°	16.4%, 25.0%

Table 2. Data augmentation’s effect on localization accuracy

the area is larger and the landmarks close to the border of the image typically move faster than those in a centered crop. Therefore, such landmarks can play an important role for camera localization.

4.2. Data Augmentation

Figure 4 demonstrates the performance of our baseline model-which uses Posenet’s crop of the input image- on training and test data for the different sequences. As can be seen in this figure, the error on training data goes to zero after a small number of epochs while the error on test data tends to be much higher. Besides, the performance on orientation prediction degrades over time for most of the indoor sequences which is a sign of overfitting to the training data.

As shown in Figure 4 for many of sequences there is a limited overlap between the training and test data for the orientation trajectories. This makes it hard for the network to extrapolate from the data it has seen during training given

Dataset	Baseline	Length 1	Length 5	Length 10	Length 20
King’s College	1.24m, 1.84°	1.32m, 2.08°	1.48m, 2.66°	1.79m, 2.77°	2.05m, 3.28°
Old Hospital	3.07m, 5.13°	3.09m, 4.85°	3.38m, 6.35°	3.79m, 5.79°	4.05m, 7.36°
Shop Facade	1.23m, 5.74°	1.07m, 5.56°	1.25m, 6.86°	1.41m, 7.39°	1.28m, 9.68°
St Mary’s Church	2.21m, 5.92°	2.32m, 8.00°	2.69m, 7.71°	3.05m, 9.87°	3.42m, 8.71°
Street	21.67m, 32.8°	25.06m, 37.74°	29.25m, 35.35°	24.23m, 34.95°	26.80m, 35.04°
Average	5.88m, 10.28°	6.57m, 11.64°	7.58m, 11.78°	6.85m, 12.15°	7.52m, 12.81°
Chess	0.18m, 5.92°	0.19m, 6.20°	0.16m, 5.88°	0.17m, 7.07°	0.18m, 6.07°
Fire	0.40m, 12.21°	0.36m, 12.00°	0.35m, 10.95°	0.39m, 12.38°	0.38m, 12.21°
Heads	0.24m, 14.20°	0.18m, 14.74°	0.18m, 14.72°	0.20m, 15.78°	0.22m, 13.37°
Office	0.30m, 7.59°	0.32m, 8.35°	0.28m, 7.46°	0.29m, 8.39°	0.29m, 7.04°
Pumpkin	0.34m, 6.04°	0.34m, 6.70°	0.34m, 6.39°	0.34m, 7.29°	0.32m, 5.88°
Red Kitchen	0.36m, 7.32°	0.35m, 7.49°	0.32m, 7.05°	0.35m, 7.82°	0.35m, 6.84°
Stairs	0.35m, 13.11°	0.34m, 11.85°	0.36m, 10.83°	0.35m, 11.56°	0.33m, 11.30°
Average	0.31m, 9.48°	0.29m, 9.61°	0.28m, 9.04°	0.29m, 10.04°	0.29m, 8.95°

Table 3. Localization Accuracy Using LSTM Cells

Dataset	Baseline	Length 1	Length 5	Length 10	Length 20
King’s College	1.24m, 1.84°	0.93m, 1.59°	1.26m, 2.28°	1.40m, 2.47°	1.54m, 3.47°
Old Hospital	3.07m, 5.13°	3.02m, 4.94°	3.37m, 5.20°	3.49m, 5.68°	3.49m, 6.33°
Shop Facade	1.23m, 5.74°	0.82m, 4.15°	1.31m, 7.87°	1.60m, 6.95°	1.44m, 11.84°
St Mary’s Church	2.21m, 5.92°	1.95m, 5.51°	2.07m, 5.56°	2.28m, 7.03°	2.53m, 8.10°
Street	21.67m, 32.8°	16.48m, 30.00°	18.31m, 33.61°	20.93m, 34.42°	26.80m, 35.04°
Average	5.88m, 10.28°	4.64m, 8.52°	5.26m, 10.20°	5.94m, 11.27°	7.16m, 12.95°
Chess	0.18m, 5.92°	0.17m, 6.57°	0.15m, 5.84°	0.16m, 5.71°	0.15m, 6.02°
Fire	0.40m, 12.21°	0.32m, 12.53°	0.32m, 12.65°	0.36m, 13.00°	0.36m, 13.43°
Heads	0.24m, 14.20°	0.17m, 14.64°	0.19m, 13.85°	0.18m, 13.67°	0.20m, 14.53°
Office	0.30m, 7.59°	0.28m, 8.65°	0.27m, 8.96°	0.24m, 8.30°	0.27m, 9.15°
Pumpkin	0.34m, 6.04°	0.32m, 7.06°	0.31m, 6.89°	0.36m, 8.08°	0.31m, 6.79°
Red Kitchen	0.36m, 7.32°	0.29m, 7.07°	0.28m, 6.91°	0.26m, 5.94°	0.27m, 6.00°
Stairs	0.35m, 13.11°	0.30m, 11.46°	0.36m, 10.19°	0.33m, 11.69°	0.31m, 12.34°
Average	0.31m, 9.48°	0.26m, 9.71°	0.26m, 9.32°	0.27m, 9.48°	0.26m, 9.75°

Table 4. Localization Accuracy Using LSTM Cells+Whole View (Green: Improves PoseNet based architectures, Refer to Table 5.)

the limited number of training examples for each sequence.

Therefore, on each training epoch, we double the number of training examples by rotating each image with a random number in the range $[-20, 20]$ degrees. We manipulate the quaternion part of the label of the image to account for the new orientation. This way, we introduce new examples to the network with the same position but a different orientation. Therefore, this has the potential to improve the performance both for position and orientation. Table 2 shows the performance of the proposed method. A closer look at Figure 4 and Table 2 suggests that generating new orientation labels is mostly useful for the indoor datasets where there is limited overlap between training and test orientation trajectories. However, for the outdoor sequences with enough overlap it can have a negative effect on the performance.

It is worth mentioning that generating more training examples using a range other than $[-20, 20]$ or other data augmentation schemes such as horizontal, vertical flipping of the images can still further improve the results. However, the purpose of this section is to show overfitting as a problem for PoseNet and data augmentation as a remedy. Therefore further evaluation of the possible augmentation schemes are omitted.

4.3. LSTM experiments

In order to exploit the temporal information between consecutive frames, we replaced the 2048 units fully connected layer in PoseNet with two parallel LSTM cells with 64 units for pose and orientation regression. This architecture is different from [23] where LSTMs are used for extracting spatial information and also different from [24] where stacked bi-directional LSTMs are used for pose-only regression. Figure 5 illustrates the architecture we used for our experiments.

We fed sequences with 1, 5, 10 and 20 consecutive

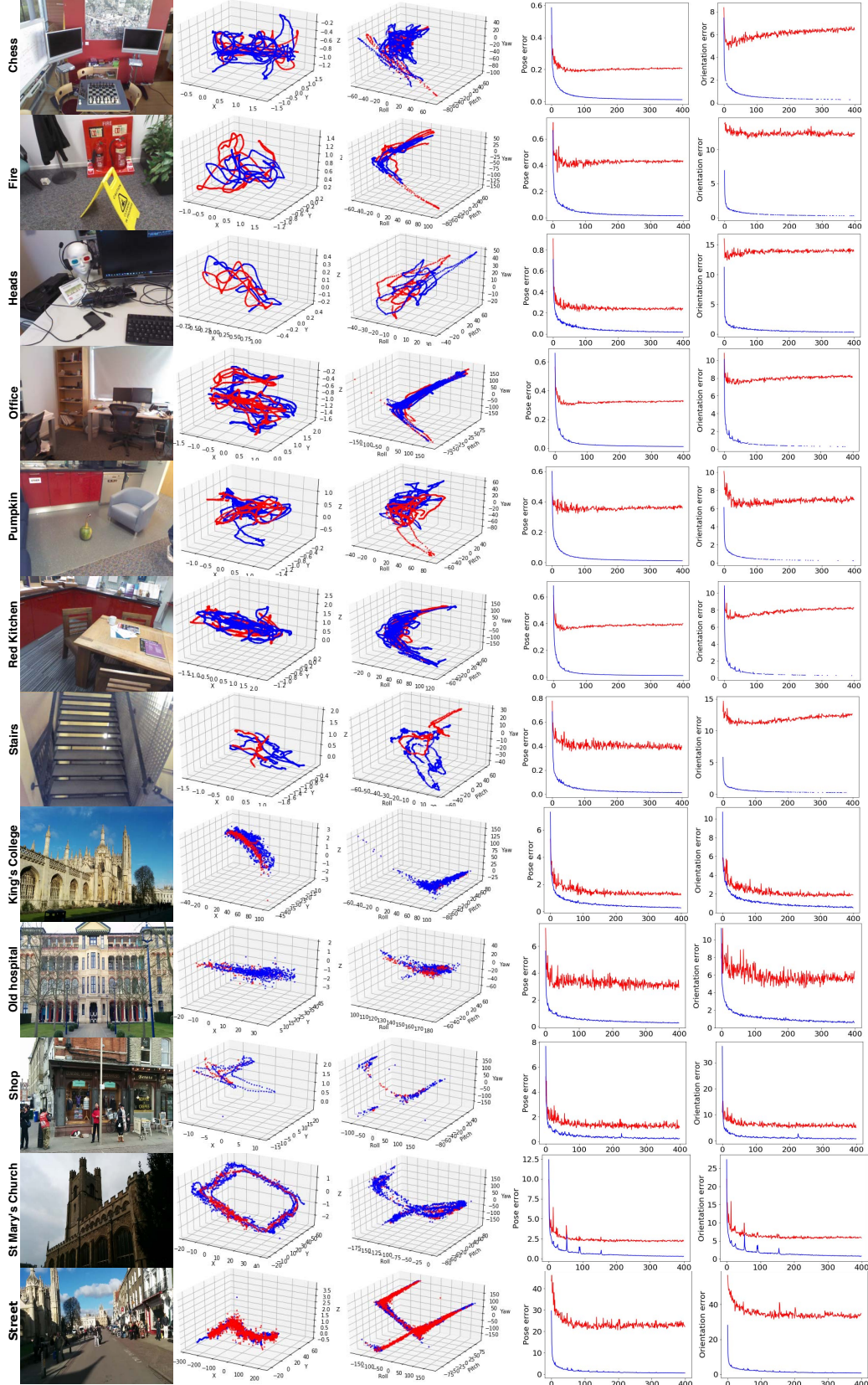


Figure 4. Left to right: an example image taken from each sequence, position trajectories for test (red) and training (blue) data, orientation trajectories, performance of the baseline model on test (red) and training data (blue) over different epochs, for pose and orientation.

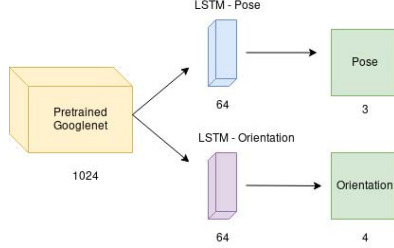


Figure 5. Proposed architecture for localization using LSTMs

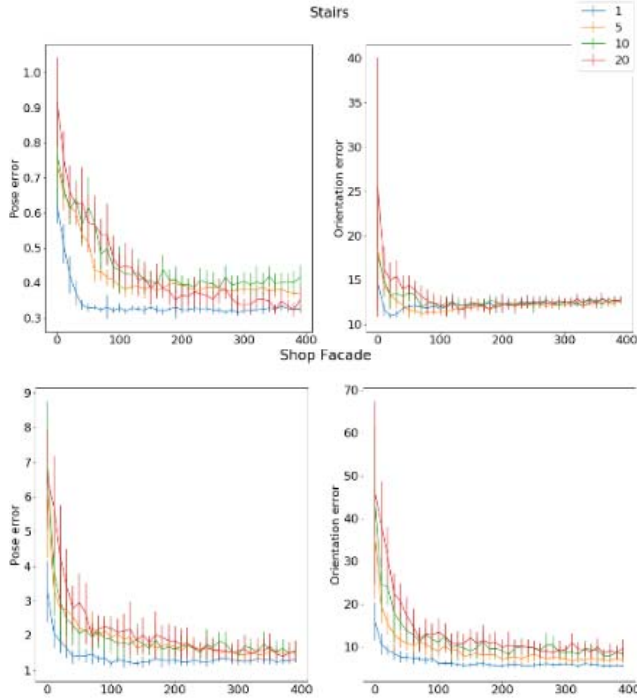


Figure 6. Results on Localization datasets using LSTM cells with different input sequence lengths

frames as input to our LSTM cells in separate experiments. Figure 6 illustrates the performance of our model on the test data, for one indoor and one outdoor localization scene. Experiments with different LSTM sequence lengths are shown in different colors in this figure. The results are averaged for every 10 epochs and the variance on every 10 epochs is shown with error bars. Tables 3 and 4 show results on all localization datasets for different LSTM sequence lengths. The green colored result in the table improves the performance for PoseNet based architectures on the corresponding outdoor localization sequence (refer to Table 5 for results on those architectures).

While we expected to have high localization gains using longer sequence lengths, our results show that LSTM cells with sequence length 1 seem to perform on average as good as longer sequence lengths. This behaviour is expected on the outdoor datasets where the frames are downsampled and temporal information cannot be exploited.

Dataset	PoseNet	Spatial LSTMs [23]	Vidloc[24]	Posenet Adaptive Loss	This Paper
King's College	1.66m, 4.86°	0.99m, 3.65°	-,-°	0.99m, 1.06°	1.45m, 4.75°
Old Hospital	2.62m, 4.90°	1.51m, 4.29°	-,-°	2.17m, 2.94°	2.47m, 5.64°
Shop Facade	1.41m, 7.18°	1.18m, 7.44°	-,-°	1.05m, 3.97°	1.13m, 7.35°
St-Mary's Church	2.45m, 7.96°	1.52m, 6.68°	-,-°	1.49m, 3.43°	2.10m, 8.46°
Street	-,-°	-,-°	-,-°	20.7m, 25.7°	14.55m, 36.04°
Average	-,-°	-,-°	-,-°	5.28, 7.42°	4.34, 12.44°
Chess	0.32m, 6.60°	0.24m, 5.77°	0.18m, -°	0.14m, 4.50°	0.17m, 5.34°
Fire	0.47m, 14.00°	0.34m, 11.9°	0.21m, -°	0.27m, 11.8°	0.30m, 10.36°
Heads	0.30m, 12.2°	0.21m, 13.7°	0.14m, -°	0.18m, 12.1°	0.15m, 11.73°
Office	0.48m, 7.24°	0.30m, 8.08°	0.26m, -°	0.20m, 5.77°	0.27m, 7.10°
Pumpkin	0.49m, 8.12°	0.33m, 7.00°	0.36m, -°	0.25m, 4.82°	0.23m, 5.83°
Red Kitchen	0.58m, 8.34°	0.37m, 8.83°	0.31m, -°	0.24m, 5.52°	0.29m, 6.95°
Stairs	0.48m, 13.1°	0.40m, 13.7°	0.26m, -°	0.37m, 10.6°	0.30m, 8.30°
Average	0.44m, 9.94°	0.31m, 9.85°	0.24m, -°	0.23m, 7.87°	0.24, 7.94°

Table 5. Whole Field-of-View + Data Augmentation + LSTM

For indoor datasets, this behaviour is justified by the fact that consecutive frames differ from each other with very small translations. CNNs trained on classification datasets such as ImageNet are designed to be translation-invariant and treat pose as a nuisance variable; therefore, most of the temporal information is lost in such CNN. The extracted features for consecutive frames are always more than 98% similar according to our experiments. This makes it hard for the LSTMs to differentiate consecutive frames.

Besides, with sequence length one, results using the LSTM architecture seem consistently better than the CNN architecture, especially for the whole field-of-view setting. Therefore, we conclude LSTM cells can improve localization performance independently from their sequence length due to their more complex architecture compared to fully connected layers. It is worth mentioning that we reproduced the same results with different settings such as using stacked LSTMs, higher number of units for LSTMs and same LSTM to regress both pose and orientation.

4.4. Putting it All Together

Table 5 illustrates our localization performance compared to other works in the literature when we combine all the proposed modifications to the PoseNet baseline. The green colored numbers in the table suggest improvement over the performance of the PoseNet based works on the corresponding localization sequences. This table and the results discussed before suggest that for sequences where the frames are downsampled and there is enough overlap between training and test data, the best performance is achieved by using the whole field-of-view as input. Otherwise, data augmentation can help to reduce the margin between test and training data. Besides, LSTM cells are generally helpful due to their relatively more complex architecture compared to fully connected layers, even when applied to sequences of length one, i.e. a single input image.

5. Conclusion

In this paper we study three different ways to improve camera localization accuracy of PoseNet targeting its training data's characteristics rather than network's architecture.

Our experiments show that the field-of-view is typically more important than the input image's resolution. Furthermore, depending on the training labels' abundance one can benefit from data augmentation to cover more areas during training time to gain accuracy on test time. Besides, while LSTM cells do not seem to be able to exploit the temporal information due to the feature extraction network being invariant to small translations, they can perform better due to their more complex nature compared to fully connected layers.

Acknowledgment This work was supported by the FWO SBO project Omnidrone ¹.

References

- [1] Kendall A, Cipolla R. Geometric loss functions for camera pose regression with deep learning. In *Proc. CVPR 2017 Jul 1 (Vol. 3, p. 8)*.
- [2] Chatila R, Laumond JP. Position referencing and consistent world modeling for mobile robots. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on 1985 Mar (Vol. 2, pp. 138-145)*. IEEE.
- [3] Billinghurst M, Clark A, Lee G. A survey of augmented reality. *Foundations and Trends in Human-Computer Interaction*. 2015 Mar 31;8(2-3):73-272.
- [4] Engel J, Sturm J, Cremers D. Camera-based navigation of a low-cost quadcopter. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on 2012 Oct 7 (pp. 2815-2821)*. IEEE.
- [5] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition 2016 (pp. 770-778)*.
- [6] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems 2012 (pp. 1097-1105)*.
- [7] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. 2014 Sep 4.
- [8] Weyand T, Kostrikov I, Philbin J. Planet-photo geolocation with convolutional neural networks. In *European Conference on Computer Vision 2016 Oct 8 (pp. 37-55)*. Springer, Cham.
- [9] Kendall A, Grimes M, Cipolla R. 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*.
- [10] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*. 2013 Aug;35(8):1798-828.
- [11] Hoo-Chang S, Roth HR, Gao M, Lu L, Xu Z, Nogueira I, Yao J, Mollura D, Summers RM. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*. 2016 May;35(5):1285.
- [12] Taketomi T, Uchiyama H, Ikeda S. Visual SLAM algorithms: a survey from 2010 to 2016. *IPSN Transactions on Computer Vision and Applications*. 2017 Dec;9(1):16.
- [13] Kerl C, Sturm J, Cremers D. Dense visual SLAM for RGB-D cameras. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on 2013 Nov 3 (pp. 2100-2106)*. IEEE.
- [14] Engelhard N, Endres F, Hess J, Sturm J, Burgard W. Real-time 3D visual SLAM with a hand-held RGB-D camera. In *Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum, Vasteras, Sweden 2011 Apr 8 (Vol. 180, pp. 1-15)*.
- [15] Engel J, Schops T, Cremers D. LSD-SLAM: Large-scale direct monocular SLAM. In *European Conference on Computer Vision 2014 Sep 6 (pp. 834-849)*. Springer, Cham.
- [16] Mur-Artal R, Montiel JM, Tardos JD. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*. 2015 Oct;31(5):1147-63.
- [17] Klein G, Murray D. Parallel tracking and mapping for small AR workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on 2007 Nov 13 (pp. 225-234)*. IEEE.
- [18] Newcombe RA, Lovegrove SJ, Davison AJ. DTAM: Dense tracking and mapping in real-time. In *Computer Vision (ICCV), 2011 IEEE International Conference on 2011 Nov 6 (pp. 2320-2327)*. IEEE.
- [19] Cadena C, Carlone L, Carrillo H, Latif Y, Scaramuzza D, Neira J, Reid I, Leonard JJ. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*. 2016 Dec;32(6):1309-32.

¹<https://www.omnidrone720.com/>

- [20] Kendall A, Cipolla R. Modelling uncertainty in deep learning for camera relocalization. In 2016 IEEE international conference on Robotics and Automation (ICRA) 2016 May 16 (pp. 4762-4769). IEEE.
- [21] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition 2015 (pp. 1-9).
- [22] Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on 2009 Jun 20 (pp. 248-255). Ieee.
- [23] Walch F, Hazirbas C, Leal-Taixe L, Sattler T, Hilsenbeck S, Cremers D. Image-based localization using lstms for structured feature correlation. In Int. Conf. Comput. Vis. (ICCV) 2017 Oct 1 (pp. 627-637).
- [24] Clark R, Wang S, Markham A, Trigoni N, Wen H. Vid-loc: A deep spatio-temporal model for 6-dof video-clip relocalization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017 Feb 21 (Vol. 3).
- [25] Zhou T, Brown M, Snavely N, Lowe DG. Unsupervised learning of depth and ego-motion from video. In CVPR 2017 Jul 1 (Vol. 2, No. 6, p. 7).
- [26] Brahmbhatt S, Gu J, Kim K, Hays J, Kautz J. Geometry-aware learning of maps for camera localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018 (pp. 2616-2625).
- [27] Valada A, Radwan N, Burgard W. Deep auxiliary learning for visual localization and odometry. In 2018 IEEE International Conference on Robotics and Automation (ICRA) 2018 May 21 (pp. 6939-6946). IEEE.
- [28] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. 2014 Dec 22.
- [29] Shotton J, Glocker B, Zach C, Izadi S, Criminisi A, Fitzgibbon A. Scene coordinate regression forests for camera relocalization in RGB-D images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2013 (pp. 2930-2937).
- [30] Newcombe RA, Izadi S, Hilliges O, Molyneaux D, Kim D, Davison AJ, Kohi P, Shotton J, Hodges S, Fitzgibbon A. KinectFusion: Real-time dense surface mapping and tracking. In Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on 2011 Oct 26 (pp. 127-136). IEEE.