A case study on the impact of masking moving objects on the camera pose regression with CNNs

Claudio Cimarelli Dario Cazzato Miguel A. Olivares-Mendez
Holger Voos
Interdisciplinary Center for Security, Reliability and Trust (SnT), University of Luxembourg
29, Avenue J. F. Kennedy, 1855 Luxembourg,
{firstname.lastname}@uni.lu

Abstract

Robot self-localization is essential for operating autonomously in open environments. When cameras are the main source of information for retrieving the pose, numerous challenges are posed by the presence of dynamic objects, due to occlusion and continuous changes in the appearance. Recent research on global localization methods focused on using a single (or multiple) Convolutional Neural Network (CNN) to estimate the 6 Degrees of Freedom (6-DoF) pose directly from a monocular camera image. In contrast with the classical approaches using engineered feature detector, CNNs are usually more robust to environmental changes in light and to occlusions in outdoor scenarios. This paper contains an attempt to empirically demonstrate the ability of CNNs to ignore dynamic elements, such as pedestrians or cars, through learning. For this purpose, we pre-process a dataset for pose localization with an object segmentation network, masking potentially moving objects. Hence, we compare the pose regression CNN trained and/or tested on the set of masked images and the original one. Experimental results show that the performances of the two training approaches are similar, with a slight reduction of the error when hiding occluding objects from the views.

1. Introduction

The estimation of the camera pose from images with respect to the 3D scene is a fundamental task for autonomous systems. Despite tremendous advances obtained with deep learning and end-to-end systems, the possibility of obtaining a complete perception only with one camera is still a very challenging problem due to occlusions, repetitive patterns, sizes of the navigation environment, changes in the environment appearance, and variations of lighting conditions. Thus, it is not surprising that state-of-the-art autonomous systems utilize different sensors and information sources, e.g., IR, GPS, radar, ultrasonic, and/or LiDAR, along with the camera. Anyway, there is a very active research field that tries to operate with visual information only. First of all, other sensors can fail: for example, IR has a problem in case of structured objects, LiDAR could not work properly in case of rain due to the reflectance of raindrops, and GPS cannot be used in certain denied areas and with urban canyons [3]. Secondly, each additional sensor has an impact on the final payload, which is peculiarly sensitive topic in the case of unmanned (UAV) and micro (MAV) aerial vehicles [18]. Finally, the solid theory on global camera pose estimation [8] and the continuous advancements on deep learning based techniques [13, 20] have encouraged the investigation of pure vision-based methods.

Focusing on deep learning, it has produced a single general pattern for solving the more diversified problems related to robotics. For example, on the side of global localization, the use of a single [14] (or multiple [29]) Convolutional Neural Network (CNN) is proposed to estimate the 6 Degrees of Freedom (6-DoF) pose directly from monocular camera images of a confined area. In contrast with the classical approaches using engineered features detector [5], such solutions prove its robustness in outdoor scenarios with the capability of disregarding dynamic elements that are present in the scene, like pedestrians or cars. From another perspective, deep learning is a popular and modern solution for object detection and segmentation [7] in which the image pixels are classified into one of a predefined set of object categories. Therefore, the support of deep learning for inferring essential information from the cameras is not uncommon nowadays in designing an autonomous vehicle for surveillance [28], in which both localization and intrusion detection have to be carried out reliably. Notwithstanding these two tasks apparently move towards distinct directions, it is not clear yet to which extent moving objects in dynamic scenarios could influence the final estimated pose.

Hence, we propose a preliminary study on the effect of such occlusions by applying an object segmentation network on the input of the localization network. In fact, since both systems would be already naturally present in an autonomous robot designed for surveillance, this is an opportunity to test how one network output could influence the other so that a conscious decision on their coupling can be made. This work contributes to this exciting research line by introducing an attempt to empirically demonstrate if a deep network trained for localization intrinsically incorporates the ability to ignore the moving objects through learning. A convolutional neural network for pose regression is trained on a set of images in which the pixels corresponding to dynamic objects are masked; this neural network is compared with the training on the plain dataset by analyzing the error statistics of the two approaches. The masking passage is achieved by applying a CNN for object detection and segmentation, i.e., Mask R-CNN [9], setting to zero the part of the images that are classified as moving elements. Therefore, it is shown that, on an unseen testing set of images, the performances of the two training approaches are statistically similar, with no significant gain in hiding occluding objects from the views. The rest of the manuscript is organized as follows: Section 2 contains the related work; in Section 3, the problem of Visual Based Localization and the methodology for pose regression with neural networks is described; Section 4 expands on the challenges faced by localization methods in dynamic scenes and lays out the methodology to demonstrate the robustness of neural networks in such situations; Section 6 has the conclusions.

2. Related Work

Works to deal with segmented dynamics object in the simultaneous localization and mapping (SLAM) have been proposed in the past. In [31], a 3D object tracker is used to prevent a SLAM algorithm to rely on features belonging to moving parts in its map, as well as to remove features occluded by moving objects. Riazuelo *et al.* [22] introduced a human tracker to remove certain regions from the SLAM pipeline instead, showing that such strategy improves the performance of camera tracking and relocation. A solution that addresses the same problem for both movable and moving objects has been proposed by Bescos *et al.* [4]. Authors employ a CNN to segment dynamic objects in the images in order to avoid the extraction of features, made by SLAM algorithms, on those parts. They also propose a reconstruction of occluded parts, but for the RGB-D case only. Mask R-CNN [9] is used to segment a-priori dynamics objects, while RGB-D information is combined to strengthen the segmentation and to label moving objects not detected from the CNN. Instead, in [30], the object detection network YOLOv3 [21] has been used to propose a semantic SLAM in real-time.

If benefits of a-priori knowledge and of adding a segmentation step have been shown in classic SLAM scenarios, very few works have been provided for the case of neural networks. In fact, state-of-the-art architectures automatically learn to extract the relevant information, i.e., the "important" parts of an image for the different task under consideration.An attempt in the state of the art between masking parts of the images and neural network performance has been provided in [6], where random region masking has been used as a way to get regularization on the input layer (*cutout*). Furthermore, saliency maps can visually provide empirical evidence of the ability of neural networks to recognize relevant data. In [23], a milestone work has been proposed in order to visualize models for the image classification tasks of CNNs through the visual saliency maps, a topographical representation of unique features in the visual processing. Furthermore, [24, 26] advanced in the methods for quantifying the input pixels contribution to the final prediction. Hence, they propose different techniques to visualize saliency maps describing the magnitude of the back-propagated gradient. Sundararajan *et al.* [26] introduce two axiomatic principles that saliency methods should enforce in order to be reliable in their evaluation. In [24], it is proposed a smoothing technique in which Gaussian noise is added to the input creating multiple intermediate saliency maps that are averaged together.

3. Visual Based Localization

The problem of visual localization can be formulated as the estimation of the position of a camera (represented, in general, by a rotation matrix R and a translation vector T) in a finite area. Differing from the set of *indirect* methods, which return a coarse estimate but are often used in wide areas, *direct* pose estimation methods pursue a precise metric solution for restricted environments [19]. Hence, their outcome is the 6 Degrees of Freedom (6-DoF) Pose that uniquely identifies the translation and rotation of the camera in the navigation environment. Herein, we refer to the pose as the 7-dimensional vector $\mathbf{p}$ defined as:

$$\mathbf{p} = [\mathbf{x}, \mathbf{q}] \quad \mathbf{x} \in \mathbb{R}^3, \mathbf{q} \in \mathbb{R}^4 \tag{1}$$

where $\mathbf{x}$ expresses the translation in the 3D environment in meters, whereas, $\mathbf{q}$, the rotation as a quaternion. The advantage of choosing the quaternion is twofold. First, it avoids the problem of gimbal lock that is instead frequent with the classic Euler angles counterpart. Secondly, it is always possible to derive a valid rotation just by normalizing the quaternion vector to the unit length.

3.1. Neural Network for Pose Regression

In this work, we cast the pose estimation problem as a regression modeled through a neural network. Remarkably,

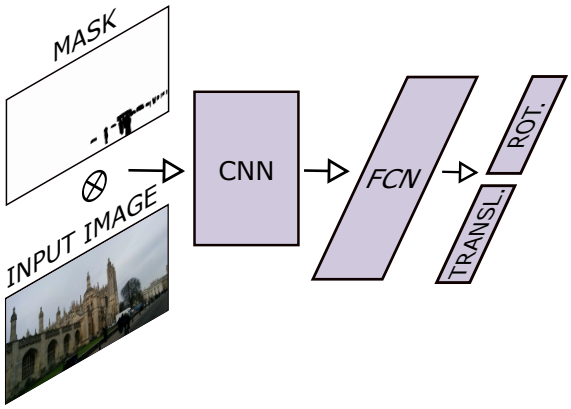


Figure 1: Diagram of the pose regression neural network. It shows the input image that is combined with a binary mask of detected objects (see Section 4).

we combine a Convolutional Neural Network (CNN), acting as a feature extractor, with a Fully Connected Network (FCN) that produces the final pose vector. Hence, following a supervised learning modality, we optimize the weights of the network in order to fit a set of training images labeled with ground truth poses. Referring to the pose estimated by the network for a training image $\mathcal{I}$ with $\hat{\mathbf{p}}_{\mathcal{I}} = [\hat{\mathbf{x}}, \hat{\mathbf{q}}]$, we minimize a loss function that expresses the distance from the ground truth pose $\mathbf{p}_{\mathcal{I}} = [\mathbf{x}, \mathbf{q}]$. Translation and rotation form two separate components, denoted by $\mathcal{L}_x$ and $\mathcal{L}_q$ respectively, which add up to the total loss function $\mathcal{L}$:

$$\mathcal{L}_x(\mathcal{I}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2 \tag{2}$$

$$\mathcal{L}_q(\mathcal{I}) = \left\|\mathbf{q} - \frac{\hat{\mathbf{q}}}{\|\hat{\mathbf{q}}\|}\right\|_2 \tag{3}$$

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

where the notation $\|\cdot\|_2$ refers to the Euclidean norm.

Due to the different units of measurement in which translation and rotation are represented, we implement the *homoscedastic loss* [13] to adapt the weights $\hat{s}_x$ and $\hat{s}_q$ that balance the magnitude of the two components of the loss. Therefore, the final objective function being minimized is the following:

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

During our experiments, $\hat{s}_x$ and $\hat{s}_q$ are initialized to 0.6 and 0.2 respectively.

Regarding the choice of the feature extractor network, it is possible to select different CNN's architectures from the state-of-the-art and combine them in the same way with a FCN in a cascade design. We follow the main trend of placing a single layer with 2048 neurons after the convolutions [14], each applying the ReLU [17] activation function to their input nodes. Before the non-linearity, we add a *Batch Normalization* layer [12] to aid the generalization capability of the network. Lastly, we use Dropout [25] as a further source of regularization and to support the learning of sparse feature representation. Therefore, the fully connected layer is linked to the final regressor of the 7-dimensional pose vector, which is not followed by any non-linearity.

4. Pose Estimation in Dynamic Scenes

In the event of a robot operation inside an urban environment, the localization can be affected by the occlusions caused by dynamic objects. In some cases, where the occluding objects are severely obstructing the field of view, we cannot expect any algorithm based on visual input to obtain any useful information on the current position. Instead, we can argue that if points of interest are included in some patches of the images, then a robust algorithm must provide the estimated pose based solely on these important clues without being distracted by the features of the dynamic objects. Hence, our objective is to empirically show the effectiveness of convolutional neural networks in focusing on the part of the image containing the relevant information. In order to confirm such a proposition, our strategy is to detect the parts of the image belonging to moving objects and to train a pose regressor using the pre-processed masked dataset. Eventually, we compare the results with those obtained by training a network on the normal dataset, in order to test if the model outcome is sensitive to the missing features contained in the masked objects. Thus, we perform an ablation study where the single varying element is the input dataset for training. For this purpose, during the experimental phase, we keep all the other factors (e.g., the hyperparameters of the networks) unchanged.

4.1. Dataset pre-processing

Figure 2: King's College sample pictures from the *Cambridge Landmarks* dataset. These images present a high variation in the point of views as well as the numerous pedestrian and vehicles that can be detected.

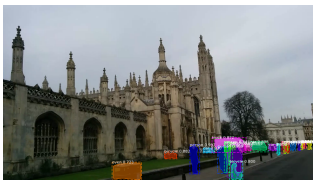
For this work, we use the King's College scenario that is part of the *Cambridge Landmarks* [14] since this representation of an urban scenario is ideal for our study case due to the presence of pedestrians and vehicles (see Figure 2). It contains 1465 images captured with a smartphone in high resolution. Ground truth is provided by a Structure from Motion (SfM) algorithm, which builds a 3D model and associates a pose to each image.

The phase of moving object segmentation is performed offline, pre-processing all the images in the dataset in advance. We used a pre-trained Mask R-CNN [9] (implemented by [2]) to segment automatically the objects (see Figure 3a). The output of Mask R-CNN is the list of object classes that are detected in the image and a mask labeling the pixels that belong to each object. From the original 80 MS COCO [16] categories, we picked those that best fit the concept of a moving object not relevant to the localization objective. Among those, we include things that could be carried by a person, e.g., a backpack, and all the animals independently from their presence in the King's college dataset, as resumed in Table 1.

Category	Person	Bicycle	Car	Truck	Handbag	Backpack	Motorcycle	Suitcase
Count	6059	2976	2716	256	248	121	93	41
Category	**Umbrella**	**Tie**	**Boat**	**Bird**	**Bus**	**Airplane**	**Dog**	**Horse**
Count	9	7	5	2	2	1	1	1

Table 1: Number of objects detected per each category (only the categories with at least one detected object are included).

Hence, the outcome of this step is a binary mask with the same size of the processed images representing whether or not the corresponding pixel has to be ignored (see Figure 3b).



(a) Object Segmentation

(b) Binary Mask

Figure 3: pre-processing steps applied to a sample image of the King's College dataset.

Following, we compute the per-channel mean and standard deviation of the pixel value on the entire dataset. Before training, we standardize the images by subtracting the mean and diving by the standard deviation so that we obtain zero-centered and unitary variance input distribution. Henceforth, we apply the binary mask setting to zero the input underlying the black part of the mask (see Figure 3b) and leaving unchanged the rest. This procedure retraces the *cutout* regularization technique [6] in the application of a zero mask after the normalization of the input. Ultimately, we training using random crops of size 224×224 of the original images. Instead, during the tests only the central crop is used.

5. Experiments and Results

5.1. Experimental Setting

As mentioned in Section 3.1, for the pose regression we train a CNN, acting like a feature extractor, coupled with a FCN that produces the final pose vector. Hence, we provide the result both based on the training with ResNetV2 [11] with 152 layers and with GoogLeNet [27]. The CNNs' implementations are found within the TensorFlow-Slim open-source library [1] and we initialized the weights with those pre-trained on the ImageNet dataset. Instead, the fully connected layers are initialized randomly using the method proposed by He *et al.* [10]. Dropout rate is set to 0.12, so that 12% of the neurons are turned off during training on average, whereas Batch Normalization momentum is set to 0.99. Adam [15] optimizer is used to minimize the loss with a learning rate $\alpha = 1e^{-4}$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$, on batches of size 64 shuffled at each new epoch. A single NVIDIA Tesla V100 has been used for each training.

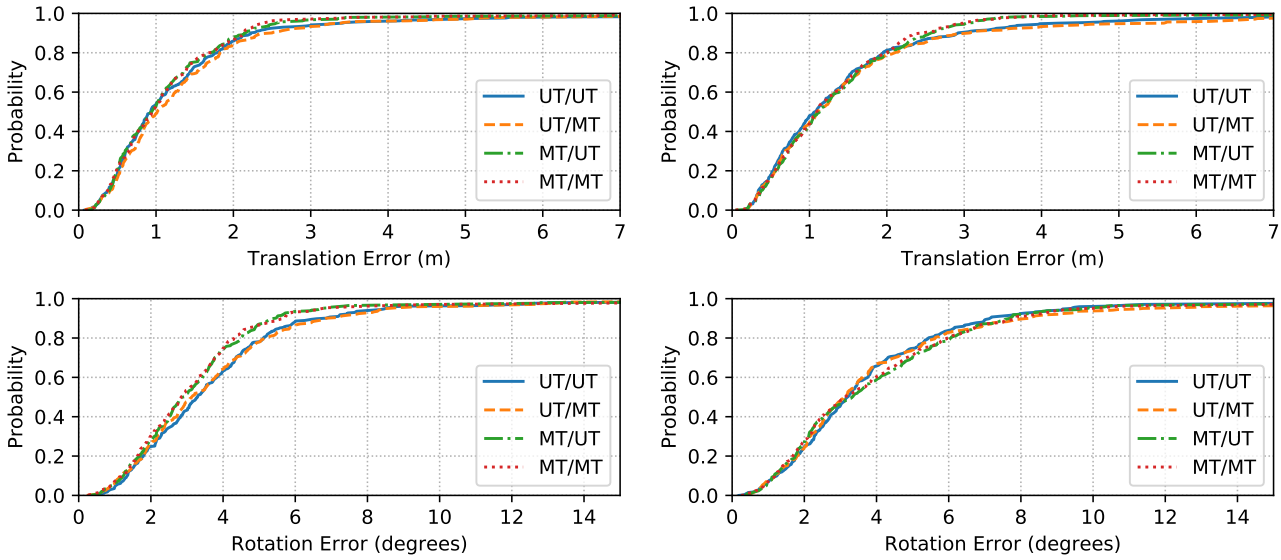
5.2. Comparison of the Results

Herein, we discuss the results obtained with the networks (i.e., ResNet and GoogLeNet) using the masking procedure either at training time or at test time or during both phases. Hence, we obtained the mean and median error statistics for four different combinations that we name: Unmasked Training + Unmasked Test (UT/UT), Unmasked Training + Masked Test (UT/MT), Masked Training + Unmasked Test (MT/UT), and Masked Training + Masked Test (MT/MT).

	UT/UT	UT/UM	MT/UT	MT/MT
GoogLeNet				
Median Error	0.93m, 3.29°	1.01m, 3.12°	0.94m, 2.86°	0.95m, 2.84°
Mean Error	1.36m, 3.85°	1.39m, 3.84°	1.21m, 3.40°	1.19m, 3.39°
ResNetV2 152				
Median Error	1.06m, 3.18°	1.10m, 3.12°	1.13m, 3.31°	1.09m, 3.12°
Mean Error	1.58m, 3.99°	1.73m, 4.20°	1.40m, 4.21°	1.33m, 4.20°

Table 2: Median and mean errors of translation and rotation for the four different combinations of Masking and Unmasking images at training and test time.

Comparing the mean and median values in Table 2, it is possible to observe that a method which outperforms the others does not clearly emerge. While the median translation error is slightly lower in the UT/UT approach, for the rotation it is the contrary. This result is reflected in the



(a) GoogLeNet Localization Performance

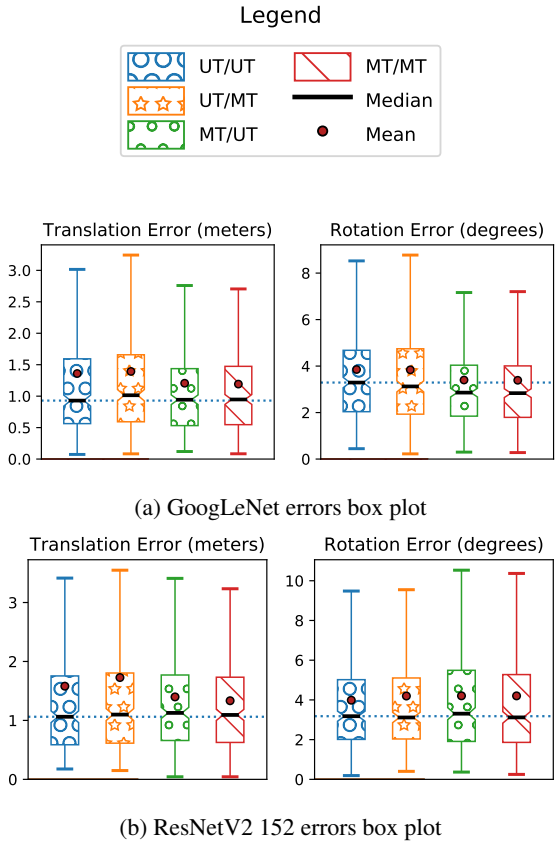
(b) ResNetV2 152 Localization Performance

Figure 4: Localization performances shown as a cumulative distribution plot of the error for the translation and rotation separately. Comparing the four approaches, we can notice that the trends are overall similar.

lower plot of Figure 4a, in which the MT/MT (or MT/UT) approach exhibits higher probability of obtaining lower rotation error with GoogLeNet, whereas with ResNet (upper plot in Figure 4b) it shows a mildly better performance in the translation for the last percentile of frames.

Inspecting the boxplots in Figure 5, we notice that the medians of MT/MT fall in the 95% confidence intervals of the UT/UT respective medians (shown through the notches and a blue dotted line), apart from the rotation error of GoogLeNet. Therefore, we can conclude that the medians do not differ with 95% confidence. From one point of view, this evidence could imply that the UT/UT approach already incorporates the capability of masking irrelevant information contained in the input images. On the other side, it validates the prior assumption that features contained in dynamic object are not influencing the pose estimation and can be hidden without harming the accuracy of the results.

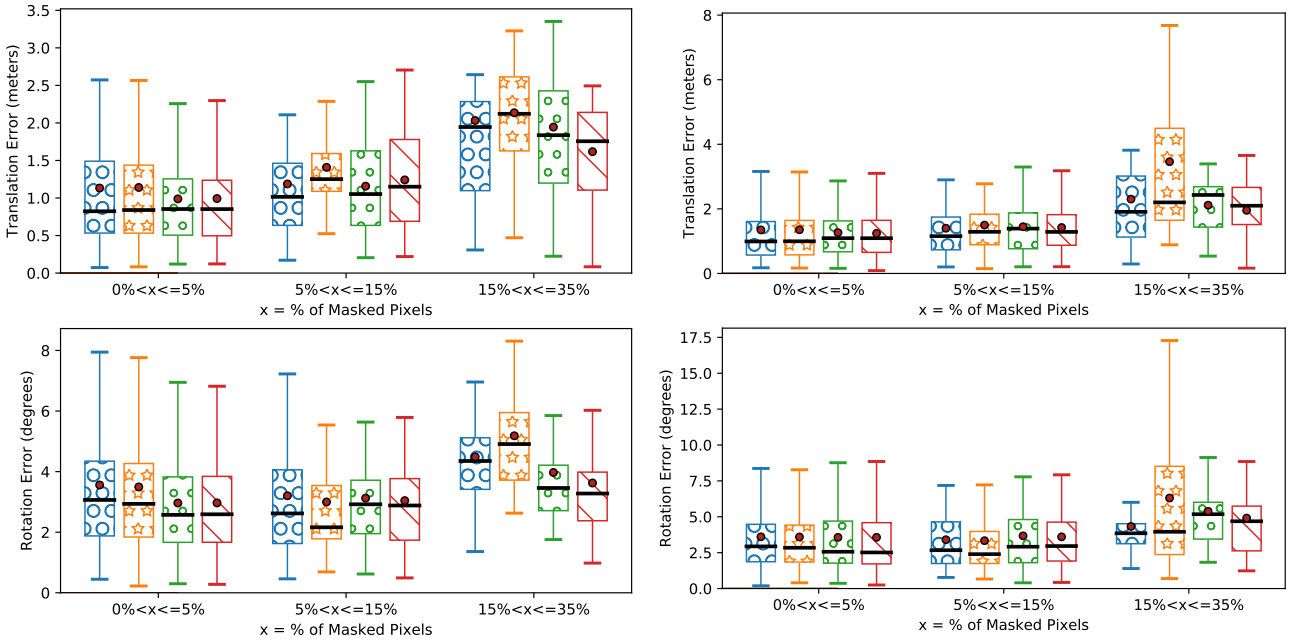
Ultimately, we study possible relationships between the portion of image that can be masked and the error in the localization. With this regard, we bin the test images by the percentage of pixels belonging to detected dynamic objects over the total number of pixels, i.e., 224×224. Since the majority of images has lower than 5% masked pixels and very few over 35%, in Figure 6 we show the results for the bins: 0% to 5% masked pixels, which contains 207 images; 5% to 15%, containing 56 images; 15% to 35% containing 28 images. Thus, the boxplot (Figure 6) reveals an apparent connection between the increase in the localization error, both translation and rotation part, and the portion of the im-



(a) GoogLeNet errors box plot

(b) ResNetV2 152 errors box plot

Figure 5: Box plots of the translation and rotation errors.



(a) GoogLeNet errors box plot

(b) ResNetV2 152 errors box plot

Figure 6: Translation and rotation errors on the test images grouped by the percentage of pixels that could be masked with the proposed method (for the legend see Figure 5). It shows a slight relationship between the portion of image that is obscured and the increase in the mean/median error.

age that is covered by dynamic objects especially when this is a significant part, e.g., more than 15%. Furthermore, in the test set there are 3 more images over the 35% threshold. These are not included in the plot since their mean error is markedly higher than the other bins' means and would not make possible a clear visualization. Anyway, this evidence further confirms a relationship between localization error and size of dynamic objects.

5.3. Saliency Maps Visualization

In this Section, we investigate the contribution that each pixel is supposed to give to the final pose estimate. For this purpose, we make use of the saliency maps produced by *SmoothGrad* [24] technique combined with the *Integrated Gradients* method [26]. *Integrated Gradients* (*IG*) accumulates the contribution given by the pixels in the images that lie in the straight interpolation line between the original image, e.g., the one for which we would like to visualize the saliency, and a baseline image, e.g., a black picture which is supposed to have a neutral pose estimation (high error). Hence, it integrates the gradient of the network output with respect to each input image by computing the Riemman approximation of the integral, with a discrete number of steps m. Naming x the original image and x' the baseline, and calling F the function represented by the neural network, we calculate the saliency for the pixel i as:

$$IG_i(x) = (x_i - x'_i) \times \sum_{k=1}^{m} \frac{\partial F(x'_i + \frac{k}{m} \times (x_i - x'_i))}{\partial x_i} \times \frac{1}{m} \tag{6}$$

In our experiments, we use $m = 40$ integration steps.

Furthermore, *SmoothGrad* (*SG*) sharpens the saliency maps by taking into account the possible fluctuations of the backpropagated gradients. In fact, the authors showed that the gradient is sensitive to slight variations of the input. Henceforth, they proposed to smooth the maps by averaging together the backpropagated gradient of multiple input instances created by applying a Gaussian filter. For this reason, *SmoothGrad* is compatible with any saliency algorithm since by itself does not compute the maps. As it follows, the computation takes a saliency function S applied to an image x, and iterates n times sampling additive noise from a Gaussian normal distribution $\mathcal{N}(0, \sigma^2)$ with zero mean and standard deviation σ:

$$SG(x) = \frac{1}{n} \times \sum_{k=1}^{n} S(x + \mathcal{N}(0, \sigma^2)) \tag{7}$$

In our experiments we parametrized *SG* with $n = 35$ and $\sigma = 0.1 \cdot (x_{max} - x_{min})$, i.e., the 10% of the pixel intensity range.

In figure 7, we draw the obtained saliencies of the UT/UT and MT/MT approaches using GoogLeNet for a couple of test frames in which two representative scenes are illustrated: first, a person on a bike is in the foreground 7a, second, vehicles parked in front of a building 7b. It is possible to observe that in the first case the UT/UT approach masks effectively most of the cyclist almost as well as the MT/MT approach. On the contrary, the second frame shows that the gradient "leaks" inside the shape of the white camper making it clearly visible. This effect could possibly mean that the vehicle features carry useful information since it is always parked at the same spot in all the frame of the dataset.

6. Conclusions

This paper addressed the problem of the camera global localization in the case of dynamic object presence when using a CNN for pose regression. For this end, a preprocessing step of the camera images with an object segmentation network has been proposed. In particular, the network has been used to mask the pixels corresponding to pedestrians and vehicles in a dataset representing an urban scenario. Following, a neural network for pose regression has been trained using this dataset and the original one. Consequently, the translation and rotation error were calculated testing the approaches both on normal and masked images. Ultimately, the results of the four combinations were compared together verifying the effectiveness of masking dynamic objects on the final predicted pose. Complementary, this test would expose the robustness' degree of CNNs with respect to dynamic objects' features. Experimental results showed that the performances of the two training approaches are similar, with a slight reduction of the error when hiding occluding objects from the views. Therefore, whilst the pose estimation would benefit overall from removing the pedestrians and other possibly moving objects by blackening them out, CNNs appear to inherently be able to extract salient features through learning. Feature work will investigate the reconstruction of the patches hidden by moving objects using an in-painting technique and how it could relate to the pose estimation. Additionally, extensive experiments will be carried out on a dataset appropriately designed for such study.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 4

(a) Example of saliency with a person in the foreground.

(b) Example of saliency with the presence of vehicles in the scene.

Figure 7: Saliency maps created by combining *SmoothGrad* with *Integrated Gradients*.

[2] W. Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN, 2017. 4

[3] A. Bachrach, S. Prentice, R. He, and N. Roy. Range–robust autonomous navigation in gps-denied environments. *Journal of Field Robotics*, 28(5):644–666, 2011. 1

[4] B. Bescos, J. M. Fácil, J. Civera, and J. Neira. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4):4076–4083, 2018. 2

[5] M. Cummins and P. Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008. 1

[6] T. DeVries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 2, 4

[7] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew. Deep learning for visual understanding: A review. *Neurocomputing*, 187:27–48, 2016. 1

[8] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2, 4

[10] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1026–1034, 2015. 4

[11] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 4

[12] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 3

[13] A. Kendall and R. Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5974–5983, 2017. 1, 3

[14] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 1, 3, 4

[15] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 4

[17] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010. 3

[18] F. Nex and F. Remondino. Uav for 3d mapping applications: a review. *Applied geomatics*, 6(1):1–15, 2014. 1

[19] N. Piasco, D. Sidibé, C. Demonceaux, and V. Gouet-Brunet. A survey on visual-based localization: On the benefit of heterogeneous data. *Pattern Recognition*, 74:90–109, 2018. 2

[20] N. Radwan, A. Valada, and W. Burgard. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robotics and Automation Letters*, 3(4):4407–4414, 2018. 1

[21] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 2

[22] L. Riazuelo, L. Montano, and J. Montiel. Semantic visual slam in populated environments. In *2017 European Conference on Mobile Robots (ECMR)*, pages 1–7. IEEE, 2017. 2

[23] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 2

[24] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017. 2, 6

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 3

[26] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org, 2017. 2, 6

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 4

[28] V. Tsakanikas and T. Dagiuklas. Video surveillance systems-current status and future trends. *Computers & Electrical Engineering*, 70:736–753, 2018. 1

[29] A. Valada, N. Radwan, and W. Burgard. Deep auxiliary learning for visual localization and odometry. In *IEEE International Conference on Robotics and Automation*, pages 6939–6946. IEEE, 2018. 1

[30] Z. Wang, Q. Zhang, J. Li, S. Zhang, and J. Liu. A computationally efficient semantic slam solution for dynamic scenes. *Remote Sensing*, 11(11):1363, 2019. 2

[31] S. Wangsiripitak and D. W. Murray. Avoiding moving outliers in visual slam by tracking moving objects. In *2009 IEEE international conference on robotics and automation*, pages 375–380. IEEE, 2009. 2