Report: Practical to Advanced Deep Learning in Computer Vision

Alexander Ziller
alex.ziller@tum.de

Leonhard Feiner
leo.feiner@tum.de

Abstract

During the practical we have been working on improving pose regression networks accuracy by multi-task learning with semantic information. In this final report we are shortly reporting our approaches and results. We could show that multi-task learning outperforms state-of-the-art pose regression methods without semantic information. Yet it does not outperform the more complex architecture of Radwan et al.

1. Introduction

1.1. Problem formulation

Radwan et al [4] showed that semantic information can improve pose estimation i.e. estimating location and rotation of a given image from a scene. Our task was to see whether a simpler architecture could also profit from this.

1.2. Baseline approaches

1.2.1 Mapnet

As baseline for our pose estimation results we are using Mapnet[1]. Mapnet uses a pretrained ResNet-34 as a feature extraction network and afterwards several linear layers to regress a six dimensional pose vector (see grey part in Figure 1). Note that we are not comparing to MapNet++, which includes GPS and VO data.

1.2.2 Segmentation network

To test whether multi-task learning of poses and semantics also boosts the latter we implemented a model which uses a ResNet-34 as feature extraction network and afterwards several upsampling layers with skip connections from the feature extraction layers. This was inspired by the famous U-Net paper[5]. Note that this architecture was also used in the multi-task model.

1.3. Dataset

We are using the DeepLoc dataset[4]. It includes a training and validation set each consisting of stereo RGB images, depth data and semantic information for every second datapoint.

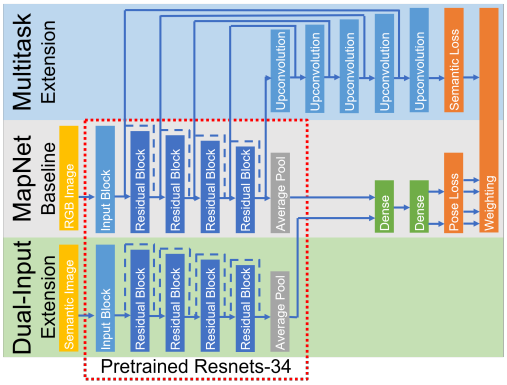


Figure 1: Visualization of architectures (grey: MapNet, green+grey: Dual-Input model, blue: segmentation network, grey+blue: Multi-task model)

2. Our approaches

2.1. Dual input model

The simplest approach to test if semantic information could boost pose estimation is to feed it as additional input to the network (see green part of Figure 1). Because the input data has ten input channels we could not use a pretrained feature extraction network, which has three input channels for RGB data, naively. We implemented and tested several approaches to solve this problem. Most of them were based on the idea to convert labels to RGB data as it is done in visualization and feed it then to the network. The other idea was to bring it to one channel and create a fourth input layer for the feature extraction network by using the average weights of the other three pretrained layers. Empirically it showed that the conversion to RGB data without shared weights in the feature extraction part gives best results and therefore is considered in the following as Dual input model.

2.2. Multi-task model

This model is a fusion of MapNet from 1.2.1 and the segmentation network described in 1.2.2 (see Figure 1). It takes a RGB image as input and outputs semantic segmentation and a pose estimation. The weighting of Losses is essential to performance and is described in 2.3

2.3. Loss formulation

The loss is comprised of different individual loss terms, measuring performance on absolute and relative translation and rotation output.

$$\mathcal{L}^{combined} = \mathcal{L}^{weighted}_{trans,abs} + \mathcal{L}^{weighted}_{rot,abs} + \mathcal{L}^{weighted}_{trans,rel} + \mathcal{L}^{weighted}_{rot,rel} \tag{1}$$

Each of these individual losses is weighted. For MapNet and Dual input model this is according to the geometric weighting of the MapNet paper [1]. In the following equations we use X as $X \in \{(trans, abs); (trans, rel); (rot, abs); (rot, rel)\}$.

$$\mathcal{L}^{weighted}_{X} = e^{-\alpha_X}\mathcal{L}_X + \alpha_X \tag{2}$$

For Multi-task the losses are combined by a learned weighting as described in [3].

$$\mathcal{L}^{weighted}_{X} = \frac{1}{2\sigma^2_X}\mathcal{L}_X + \log(\sigma_X) \tag{3}$$

$$\mathcal{L}^{weighted}_{semantics} = \frac{1}{\sigma^2_X}\mathcal{L}_X + \log(\sigma_X) \tag{4}$$

In these Loss terms α and $\log(\sigma)$ are learnable weights.

3. Results

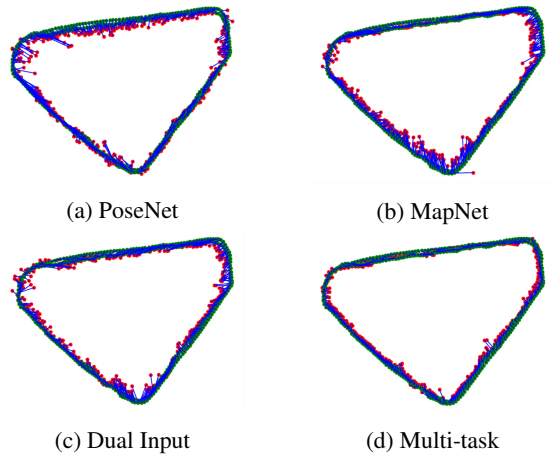
3.1. Pose estimation errors

To evaluate pose estimation we calculate mean and median of translation and rotation error on validation data. We also compare it to results given in [4] which used the same dataset. See Figure 2 for visualization of pose estimation errors.

Model	Source	median	mean	median	mean
PoseNet	Our	3.64m	4.44m	3.13°	4.27°
PoseNet	[4]	2.42m		3.66°	
MapNet	Our	3.42m	3.91m	3.23°	4.48°
Dual-Input	Our	3.26m	3.54m	2.82°	4.32°
Mult-task	Our	1.07m	1.31m	2.96°	4.32°
VlocNet	[4]	0.68m		3.43°	
VlocNet++	[4]	0.37m		1.93°	

3.2. Segmentation results

Both segmentation and mult-task model reach a pixel accuracy of 96%. The difference is too low to conclude that one of these methods is better i.e. on this dataset pose regression did not boost segmentation results significantly.



(a) PoseNet (b) MapNet

(c) Dual Input (d) Multi-task

Figure 2: Pose graphs on validation data. Green dots: Ground truth; Red dots: Predicted points; Blue lines: Connecting corresponding dots

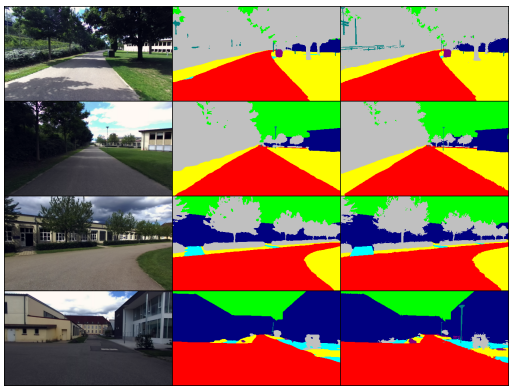


Figure 3: Semantic segmentation input, result and target for validation data

4. Challenges

Throughout the practical we faced several challenges. Some of them should be mentioned as they are critical to training and evaluation.

- Dataset does not contain test set. We used validation data as test data. Bad practice but ensures comparability to other papers.
- Only every second image has semantic ground truth.
- Random initialization influences results significantly.
- Dataset had error in groundtruth until January.

4.1. Attention visualization

To make it more visual which features the network is interested in we implemented the visualization technique from [2]. See Figure 4.

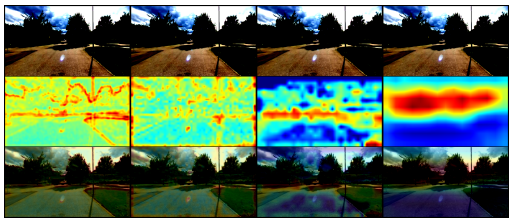


Figure 4: Backpropagated attention in layers 1 to 4 of the feature extraction network.

5. Conclusion

Overall we can conclude that multi-task learning is a simple and light weight way to improve pose estimation with neural networks. However, our model was not able to compete with VLocNet++, which is using a more complex architecture and more input signals such as GPS and visual odometry.

Code references

We used external code from following ressources:

- Base repository: `https://github.com/NVlabs/geomapnet`
- Utils to calculate IoU for semantic segmentation: `https://github.com/CSAILVision/semantic-segmentation-pytorch/blob/master/utils.py`
- Calculation of attention maps: `https://github.com/1Konny/gradcam_plus_plus-pytorch/blob/master/gradcam.py`

References

[1] Samarth Brahmbhatt et al. "Geometry-Aware Learning of Maps for Camera Localization". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[2] Aditya Chattopadhay et al. "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks". In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2018, pp. 839–847.

[3] Alex Kendall, Yarin Gal, and Roberto Cipolla. "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7482–7491.

[4] Abhinav Valada Noha Radwan and Wolfram Burgard. "VLocNet++: Deep Multitask Learning for Semantic Visual Localization and Odometry". In: 3.4 (2018), pp. 4407–4414. DOI: 10.1109/LRA.2018.2869640.

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.