CS482/682 Final Project Report Group 21

Multi-Task Learning of Monocular Depth and Segmentation from Video

Kaleab A. Kinfu (kkinfu1), Zhuohong He (zhe17) William J Howe (whowe1)

1 Introduction

Background For autonomous vehicles, depth and segmentation are crucial prerequisites to perform various tasks such as perception, navigation, and planning. Therefore, active 3D vision sensors are becoming very popular. However, many sensors are prohibitively expensive. Thus, reliably computing depth from passive sensors such as video cameras that are cheap and rugged is extremely desirable. In this work, we investigate if we can achieve state-of-the-art monocular depth estimation or improve accuracy by exploring multitask training with segmentation.

Related Work Godard *et al.* [1] demonstrates an effective model that jointly learns depth and pose estimation to produce a video frame given the surrounding frames. Additionally, it was shown that incorporating an appearance loss based on local structure significantly enhanced depth estimation performance compared to simple pixelwise L1 loss.

2 Methods

Dataset We aim to validate the effectiveness of the proposed method on the KITTI-360 dataset (a follow up to the KITTI dataset). Recorded in Karlsruhe, Germany, it includes camera images, laser depth scans, GPS measurements, and IMU measurements over a driving distance of 73.7km.

Method Self-supervised learning for monocular depth estimation poses the learning problem as a novel view-synthesis task which entails predicting a hidden video frame by projecting from neighboring

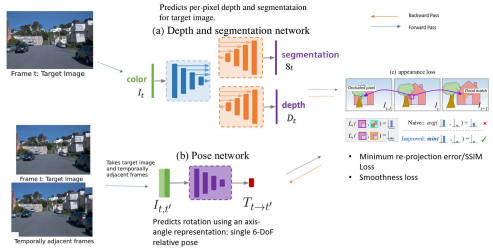

Figure 1: **Monocular Depth and Segmentation Estimator Model**. *(a) Depth and segmentation network*: We use a standard, fully convolutional, U-Net to predict depth and segmentation. *(b) Pose network:* Pose between a pair of frames is predicted with a separate pose network. *(c) Per-pixel minimum reprojection loss.* Images adapted from [1].

frames. We can extract depth as an intermediary of image synthesis. Since an extensive number of possibly incorrect depths can reconstruct the target image given the predicted pose between two views, conventional methods commonly enforce smoothness in depth maps to handle this uncertainty. Therefore, we can formulate the problem as the minimization of a photometric reprojection loss between a target RGB image and a reconstructed image from pixels of the neighbouring frames and a smoothness loss on the depth. Moreover, we aim to evaluate if learning both depth estimation and segmentation at the same time can be complementary. We are motivated by the fact that instance segmentation and depth both involve identifying clusters of similar pixels. Certain objects like pedestrians have approximately constant depths, so we can use segmentation to inform depth estimation and vice versa. We accomplish multitask learning by passing the features generated by the encoder

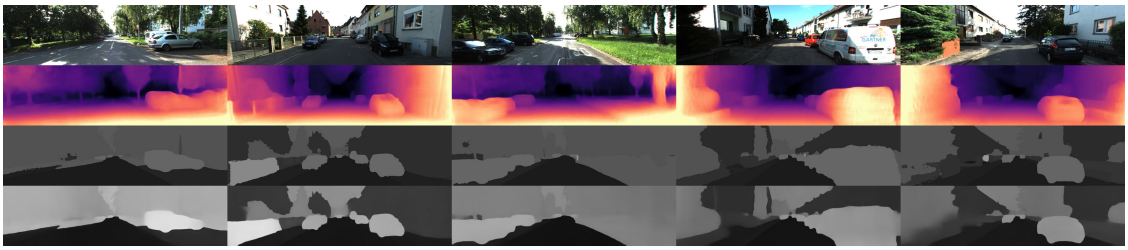

Figure 2: Performance of the Depth + Instance model on KITTI-360. *1st row*: RGB input, *2nd row*: estimated depth, *3rd row:* ground-truth instance segmentation, *4th row:* predicted instance segmentation

through both the segmentation the depth decoder, as shown in Figure 1. The segmentation output is then minimized separately on the same smoothness loss. The multitask loss is a weighted sum of the depth reprojection and segmentation losses. Due to hardware limitations, we trained each model for 20 epochs on a batch size of 8. For hyperparameter tuning, the learning rate was iteratively lowered from our initial guess to ensure the training loss curve was smooth. Similarly, for the weighted sum hyperparameter, we found that an 5.5:1 ratio of (depth loss):(segmentation loss) was ideal multitask loss.

3 Results

To quantify performance, we evaluate on the KITTI-360 dataset. As a baseline, we use the Monodepth2 pre-trained on the full KITTI dataset [1]. In this work, we trained the proposed models on 9,000 frames from KITTI-360 and evaluated on 2,000 frames. The different models we experiment on are: (i) depth model, only the depth network is trained, (ii) depth + semantic model: depth and segmentation network is trained on RGB and semantic segmentation data, (iii) depth + instance model: depth and segmentation network is trained on RGB and instance segmentation data.

Evaluation consisted of four metrics: (1) absolute relative error which averages L1 error in pixel values, (2) squared relative error which averages L2 error, (3) RMSE between pixels, and (4) the proportion of pixels with (grnd. truth):(output) ratios $< \delta$. Table 1 shows that our depth model trained on few KITTI-360 samples outperforms the baseline trained on the full KITTI. We attribute this to domain shift. Our depth+instance multitask model outperforms all the other methods. The model likely learned a strong underlying representation of both tasks in the encoder. On the contrary, our depth+semantics model performed worse likely because the two tasks are dissimilar. Semantic segmentation enforces the same output value for instances of the same class whether they are far or close to the camera. On the other hand, instances have different labels creating an positive effect on depth estimation. Figure 2 illustrates the qualitative results of the methods on randomly selected images of the benchmark.

Table 1: Performance of the model on KITTI-360 dataset. The best score for each metric is bolded.

	Lower is better			*Higher is better*		
Method	Abs Rel	Sq. Rel	RMSE	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Baseline (monodepth2) [1]	0.525	**10.122**	**13.624**	0.333	0.597	0.768
Depth	0.555	16.409	15.942	0.403	0.652	0.795
Depth + Semantics						
* Depth	0.642	17.142	16.153	0.306	0.555	0.728
* Semantics	**0.051**	**0.002**	**0.008**	**0.947**	**0.976**	**0.991**
Depth + Instance						
* Depth	**0.459**	10.783	14.258	**0.443**	**0.692**	**0.827**
* Instance	0.117	0.015	0.064	0.888	0.963	0.982

References

[1] Godard, C., Aodha, O. M., and Brostow, G. J. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision* (2019), pp. 3827–3837.