SemSegDepth: A Combined Model for Semantic Segmentation and Depth Completion

Juan Pablo Lagos and Esa Rahtu[a]
Tampere University, Tampere, Finland

Keywords: Semantic Segmentation, Depth Completion, CNN, Multi-task Networks.

Abstract: Holistic scene understanding is pivotal for the performance of autonomous machines. In this paper we propose a new end-to-end model for performing semantic segmentation and depth completion jointly. The vast majority of recent approaches have developed semantic segmentation and depth completion as independent tasks. Our approach relies on RGB and sparse depth as inputs to our model and produces a dense depth map and the corresponding semantic segmentation image. It consists of a feature extractor, a depth completion branch, a semantic segmentation branch and a joint branch which further processes semantic and depth information altogether. The experiments done on Virtual KITTI 2 dataset, demonstrate and provide further evidence, that combining both tasks, semantic segmentation and depth completion, in a multi-task network can effectively improve the performance of each task. Code is available at https://github.com/juanb09111/semantic_depth.

1 INTRODUCTION

Computer vision and holistic scene understanding have become pivotal topics as we intend to provide machines with autonomous capabilities. When we, as humans, see things we unconsciously assign multiple attributes to what we see and we also perform multiple tasks simultaneously. For instance, we can effectively assess the distance of the objects we see, the quantity, the size, the texture, etc. all at once. We are also capable of understanding the world around us in its semantic complexity. On the other hand, machines can outperform humans in several tasks individually. That is the case for tasks such as object detection (Ren et al., 2016), (Zhai et al., 2017), (Redmon and Farhadi, 2018), semantic segmentation (Ronneberger et al., 2015), (Chen et al., 2018a), (Lin et al., 2016) and/or depth estimation (Chen et al., 2020), (Godard et al., 2019), (Guizilini et al., 2020), where machines have been able to successfully carry out those tasks individually. However, when it comes to performing multiple tasks, machines are still lagging behind, in comparison to humans.

In an attempt to provide a more holistic approach to the problem of scene understanding, multi-task networks have become a highly active field of research in computer vision. In addition to provide a more complete representation of a scene, there is growing evidence that multi-task networks can improve the performance of each individual task (Liebel and Körner, 2018). Panoptic segmentation, for example, combines instance segmentation, object detection and semantic segmentation (Mohan and Valada, 2020), (Cheng et al., 2020), (Wang et al., 2020), (Weber et al., 2020). To our best knowledge, only few methods have combined semantic segmentation and depth completion (Sanchez-Escobedo et al., 2018), (Zou et al., 2020). As compared to other applications, e.g. panoptic segmentation, combining semantic segmentation and depth completion poses additional challenges such as processing heterogeneous data jointly, since semantic segmentation relies on RGB images while depth completion relies on sparse depth data.

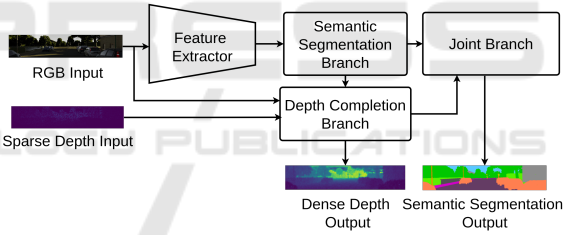


Figure 1: Overview of our proposed SemSegDepth architecture. Our model produces a dense depth map and semantics prediction given an RGB image and sparse depth as input.

[a] https://orcid.org/0000-0001-8767-0864

Lagos, J. and Rahtu, E.
SemSegDepth: A Combined Model for Semantic Segmentation and Depth Completion.
DOI: 10.5220/0010838500003124
In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022) - Volume 5: VISAPP*, pages 155-165
ISBN: 978-989-758-555-5; ISSN: 2184-4321

Semantic segmentation refers to the task of assigning a semantic label to every single pixel in an image, e.g. determining whether a pixel in an image belongs to a "car", "person", "bike" or "background". On the other hand, depth estimation, more specifically depth completion, predicts the distance of every pixel in an image, where, in most cases, a sparse depth input is provided. In applications such as autonomous driving, combining semantic segmentation and depth completion can improve the performance of the system as a whole significantly, as the machines would not only understand their surroundings semantically, but also, they would have knowledge about the proximity of the things on a given scenario.

In this paper we proposed a new end-to-end multi-task network for performing semantic segmentation and depth completion jointly. We combine two bench-marking models, namely, we use a modified version of the depth completion network proposed by Chen et al. (2020) as well as a modified version of EfficientPS (Mohan and Valada, 2020). An overview of our model SemSegDepth is shown in Figure 1. It consists of a feature extractor, a semantic segmentation branch, a depth completion branch and a joint branch. The feature extractor is a resnet50 network (He et al., 2015) wrapped in a Feature Pyramid Network (FPN) (Lin et al., 2017). Our semantic segmentation branch is based on the semantic segmentation branch of the EfficientPS architecture. The depth completion branch extends (Chen et al., 2020) by adding semantic logits as input, and finally, the joint branch further processes semantic and depth information altogether. We trained and evaluated our model on Virtual KITTI 2 (Cabon et al., 2020) and demonstrated that our SemSegDepth model improves the performance for both tasks, semantic segmentation and depth completion.

2 RELATED WORKS

2.1 Semantic Segmentation

Semantic segmentation takes image classification task to a pixel level. Fully convolutional networks have previously been used to perform dense predictions for pixel-wise segmentation (Long et al., 2014). During the last decade encoder-decoder architectures such as UNet (Ronneberger et al., 2015) became popular and achieved the state of art using what today can be considered rather simple architectures based deep convolutional networks (DCNN) that were capable of restoring the original spacial resolution with a series of upsampling layers in an end-to-end manner. However, traditional upsampling layers such as bi-linear upsampling or deconvolutional layers are computationally expensive.

Atrous convolution is a more efficient alternative and architectures such as DeepLab (Chen et al., 2016) pioneered using atrous convolution in the context of pixel-wise semantic segmentation using DCNNs. DeepLab also introduced the concept of "atrous spatial pyramid pooling" (ASPP) to enhance the network's capability of representing objects of different sizes. Later in (Chen et al., 2018a) ASPP would be optimized by using depth-wise separable convolution which would result in a faster yet stronger network. Alternatively, Kreso et al. (2016) proposed a novel and different approach to deal with the problem of scale variation in images, by using reconstructed depth from stereo images and a pixel-wise scale selection multiplexer which provides a scale-invariant image representation successfully used by a classification sub-network that finally outputs the semantic segmentation map. Other architectures (Tan and Le, 2019), (Chollet, 2016), (Mohan and Valada, 2020) would also benefit from depth-wise separable convolutional layers which are many times faster than traditional convolutional layers.

Another approach adopted in convolutional neural networks (CNN) is called gated convolutions which are based on linearizing belief nets (LBNs) (Dauphin and Grangier, 2015) that are capable of modeling a deep neural network (DNN) as linear units that can be turned on and off in a non-deterministic fashion reducing the vanishing gradient problem. LBNs were later used for language modeling by Dauphin et al. (2016) and further applied in the context of semantic segmentation by Takikawa et al. (2019) whose work tackles the problem with two branches, one of which processes the shape while the other branch processes semantic information in a classical way, and the two branches are connected with gating mechanisms.

2.2 Depth Completion

Neural networks have been largely used to produce dense depth maps out of sparse data provided by depth sensors such as Lidar and the vast majority of those networks also use RGB images for guidance (Imran et al., 2019), (Yang et al., 2019) (Xu et al., 2019), (Huang et al., 2020), (Tang et al., 2019).

The lack of ground truth dense depth maps poses a challenge for supervised learning approaches. Existing datasets like Virtual KITTI 2 (Cabon et al., 2020) provide synthetic data including dense depth maps ground truth. However, that is not the case in realistic scenarios where only sparse ground truth is available.

Other approaches like (Ma et al., 2018) are capable of learning a mapping from sparse depth and images to dense depth with no need of dense depth maps as ground truth. It is also possible to learn depth features by using surface normals as in (Qiu et al., 2018) and (Zhang and Funkhouser, 2018).

One of the challenges of working with 3D data is its non-grid nature and therefore traditional CNNs simply do not work unless the 3D points are mapped on to a 2D space. Motivated by this problem Wang et al. (2018) introduced what they called Parametric Continuous Convolution to learn features over non-grid data.

Making use of the recent continuous convolution proposed in (Wang et al., 2018), Chen et al. (2020) introduced a neural network block which extracts 2D and 3D features jointly. Such block consists of two branches running in parallel. One of the branches processes RGB features while the other branch uses continuous convolution over 3D points and finally the outputs of both branches are fused together. By stacking the same block N times they managed to effectively produce a dense depth outperforming the state of the art in 2020. Our proposed model builds upon the architecture proposed by Chen et al. (2020) for performing depth completion as described in section 3.4.

2.3 Multi Task Leaning

A CNN can effectively be trained to produce multiple outputs corresponding to different tasks. In the context of image processing, tasks such as object detection and semantic segmentation have been tackled successfully (Yao et al., 2012), (Mohan and Valada, 2020), (Kim et al., 2020). Depth estimation and semantic segmentation have also been combined in one CNN as in (Eigen and Fergus, 2014), (Hazirbas et al., 2016), (Kendall et al., 2017), (Zou et al., 2020)

Multi task networks have shown to achieve better results as whole in terms of their capability to provide a more holistic representation, but also the performance of each one of the tasks improves as a result of having a multi task CNN. Inspired by this approach, Liebel and Körner (2018) introduced the concept of "auxiliary tasks", they are side tasks that are less relevant for a given application but that potentially improved the performance of the core tasks. Moreover, Zamir et al. (2018) suggested that certain visual tasks contain underlying common and supplementary features, meaning that high level representations of an input for a specific task, may contain relevant information for solving a different task, as long as the tasks are related to one another.

More recently, He et al. (2021) proposed a multi task network for semantic segmentation and depth completion which exploits the geometric relationship between the two tasks by introducing the concept of semantic objectness, used as a constrain that describes the correlation between the semantic and the actual depth.

3 ARCHITECTURE

3.1 Overview

We propose a CNN which takes as inputs a single RGB image and a sparse depth image, and returns the corresponding semantic segmentation image and a dense depth map in an end-to-end manner. The complete diagram of our model is shown in Figure 2. Our model consists of one feature extractor backbone, two task-specific branches, one of which is designed for performing semantic segmentation and another one which performs depth completion, and one joint branch which combines semantic and depth information. The two task-specific branches are intercommunicated at specific points. The semantic segmentation branch is based on a neural network known as "EfficientPS" for panoptic segmentation (Mohan and Valada, 2020), from where we neglect the instance segmentation branch, while the depth completion branch is based on a fusion network introduced by (Chen et al., 2020) which extracts joint 2D and 3D features. Thus, our proposed architecture combines two bench-marking models to perform semantic segmentation and depth completion jointly.

3.2 Backbone

The backbone, as shown in Figure 2, is a resnet50 feature extractor (He et al., 2015) wrapped in a FPN (Lin et al., 2017) for extracting intermediate features from the backbone in order to have feature maps at multiple scales. More specifically, the FPN returns four feature maps that are down-sampled, with respect to the input, by a factor of $\times 4$, $\times 8$, $\times 16$ and $\times 32$. These features are then fed to the semantic segmentation branch.

3.3 Semantic Segmentation Branch

The semantic segmentation head follows the architecture of the semantic segmentation branch proposed by Mohan and Valada (2020). The inputs of this branch are the four different outputs of the feature extractor, that is, four feature maps, each one with a different

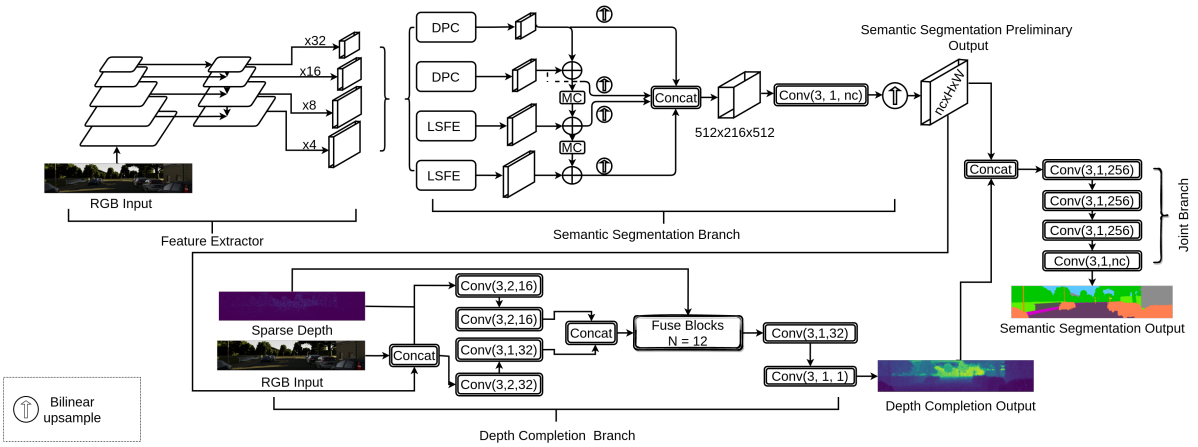


Figure 2: Diagram of the SemSegDepth architecture. The convolutional layers shown in this diagram follow the notation Conv(k,s,c) where k refers to a $k \times k$ convolutional kernel, s is the stride, and c is the number of output feature channels.

spacial resolution. This semantic segmentation head aims at capturing large-scale features as well as small-scale features and then on a later stage, such feature maps at different scales are aggregated. This branch returns semantic logits as output and the resolution of the output is $nc \times H \times W$ where nc corresponds to the number of classes.

In order to extract large-scale features, we use a Large Scale Feature Extractor (LSFE) module which consists of a stack of three layers of 3×3 convolutions and produces a feature map with 128 filters. For extracting small-scale features, we used what is known as Dense Prediction Cells (DPC) (Chen et al., 2018b) which is a modified version of ASPP (Chen et al., 2016).

Finally, in order to reduce the mismatch between small-scale features and large-scale features, we used a Mismatch Correction Module (MC). It consists of a stack of three layers of 3×3 convolutions and one bilinear upsampling layer at the very end.

3.4 Depth Completion Branch

Our depth completion branch is a modified version of the depth completion network proposed by Chen et al. (2020). To begin with, our depth completion branch receives as input, not only sparse depth and RGB as in (Chen et al., 2020), but it has one more input which corresponds to the preliminary output of the semantic segmentation branch. Thus, we are embedding semantic information into our depth completion branch.

The sparse depth, RGB and the semantic segmentation preliminary output are concatenated and passed through a stack of two 3×3 convolutional layers. The sparse depth image alone is also passed through a stack of two 3×3 convolutional layers. Then, the two outputs are concatenated and the resulting tensor, as well as the sparse depth ground truth, are the inputs of a stack of N $2D-3D$ Fuse Blocks (Chen et al., 2020). Finally, the resulting output of the Fuse Blocks passes through two convolutional layers for further refinement. The output of this branch is a fully dense depth map, that is, an image where every pixel is assigned a value of depth.

3.5 Joint Branch

Finally, in order to use depth information as guidance for the semantic segmentation branch, we concatenate the output of the depth completion branch and the preliminary output of the semantic segmentation branch. The result is passed through a joint branch that processes semantic and depth information altogether. The purpose of the joint branch is to further process the semantic segmentation preliminary output guided by depth information. It consists of a stack of four 3×3 convolutional layers. The output of the joint branch is the corresponding $nc \times H \times W$ semantic logits based on which we calculate the loss as described in section 3.6.

3.6 Loss Functions

Semantic Segmentation. For semantic segmentation we computed the cross-entropy loss for every pixel. The loss for a pixel i is defined as:

$$L_{semantic} = -\sum_{i} p_i \log \hat{p}_i, \quad (1)$$

where i is the pixel index, p_i is the ground truth and $\hat{p}_i$ is the log Softmax value of the predicted probability for pixel i. The log Softmax function is defined as:

$$LogSoftmax(x_i) = \log\left(\frac{exp(x_i)}{\sum_j exp(x_j)}\right) \quad (2)$$

Depth Completion. For depth completion we used the squared error average across all the pixels in the image for which the ground truth labels were available. The loss function for depth completion is then defined as:

$$L_{depth} = \frac{1}{N}\sum_i (\hat{y}_i - y_i)^2, \quad (3)$$

where N is the number of pixels, $\hat{y}_i$ and y_i are the predicted value and the ground truth for pixel i, respectively.

Joint Loss. In addition using a loss per task, we implemented a joint loss function in order to leverage the correlation between both tasks, namely semantic segmentation and depth completion. The joint loss is simply the sum of each individual loss as follows:

$$L_{joint} = L_{semantic} + L_{depth}. \quad (4)$$

4 EXPERIMENT SETUP

4.1 Implementation Details

We implemented the network on PyTorch and used PyTorch *DistributedDataParallel* for data parallelism during training. We used one machine with four 16GB graphics processing units (GPU) for training. We optimized the loss function using the stochastic gradient descent (SGD) with an initial learning rate set to 16×10^{-4}, momentum set to 0.9 and weight decay set to 5×10^{-5}.

4.2 Dataset

The experiments were done on the Virtual KITTI 2 dataset (Cabon et al., 2020). Virtual KITTI 2 is a synthetic video dataset which provides ground truth annotations for multiple tasks, namely, instance segmentation, semantic segmentation, multiple object tracking (MOT), optical flow, depth estimation, object detection and camera pose. It also provides stereo images for every scene. Besides, every sequence is recreated with subtle changes in the viewing angles, more specifically, $\pm 15°$ *and* $\pm 30°$ horizontal rotations and changes in the weather conditions such as foggy, cloudy, rainy, morning and sunset. We used 500 samples for training, 125 for evaluation, and 200 samples for testing.

Virtual KITTI 2 provides fully dense depth maps as depth ground truth, meaning that the depth values are provided for every single pixel in the input image. However, in order to reproduce real conditions where the ground truth depth is acquired with sensors such as LIDAR, which can only provide sparse values within a given range, we filtered out all the points exceeding a distance of 50 meters and then we randomly sampled the ground truth. Hence, our synthetic depth ground truth consists of sparse depth images containing 8000 depth values per image, where each point is within a range of 50 meters. We cropped all the images to a resolution of 200×1000 ($H \times W$).

4.3 Evaluation

In order to be able to evaluate the performance of our model, we used the mean Intersection-over-Union (mIoU) for evaluating the performance of semantic segmentation and root mean squared error (RMSE) for depth completion. The mIoU is defined as:

$$mIoU = \frac{1}{nc}\sum_l \frac{TP_l}{TP_l + FN_l + FP_l}, \quad (5)$$

where nc is the number of classes, TP_l, FN_l and FP_l are the number of true positive, false negative and false positive pixels respectively, labeled as class l. The RMSE metric is defined as:

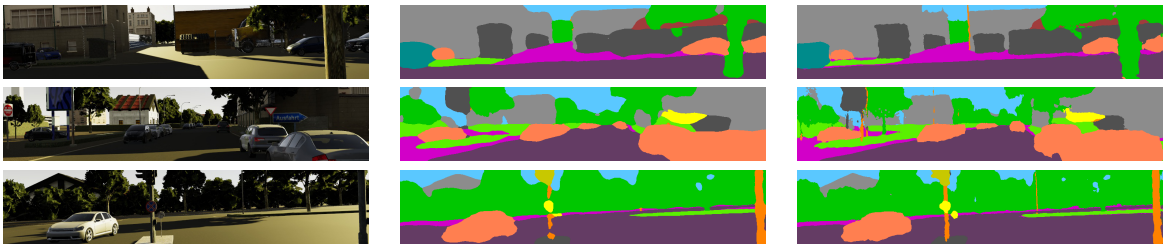
$$RMSE = \sqrt{(1/N)\sum_i (\hat{y}_i - y_i)^2}, \quad (6)$$

where N is the number of pixels, $\hat{y}_i$ and y_i are the predicted value and the ground truth for pixel i, respectively.

4.4 Baseline

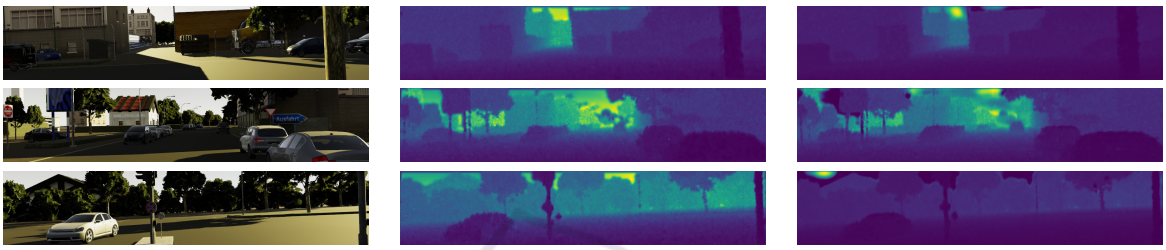
In our work, the main purpose is to quantify the performance improvement of each individual task by having them as joint tasks in one single model. Therefore, our baseline consists of two models, one for each task. In this section, we describe each baseline model.

SemSegNet_b. This is our semantic segmentation baseline. As proposed by Mohan and Valada (2020), the semantic segmentation task is carried out by an architecture which is composed by a feature extractor and a semantic segmentation branch as shown in Figure 2. The depth completion branch and the joint branch are removed from this model.



(a) RGB (b) SemSegNet_b (c) SemSegDepth (ours)

Figure 3: Semantic Segmentation results on Virtual KITTI 2. Column *a* shows the RGB image input, column *b* shows the semantic segmentation results using the baseline model SemSegNet_b and column *c* shows the the semantic segmentation results using our model SemSegDepth.



(a) RGB (b) DepthNet_b (c) SemSegDepth (ours)

Figure 4: Depth completion results on Virtual KITTI 2. Column *a* shows the RGB image input, column *b* shows the depth completion results using the baseline model DepthNet_b and column *c* shows the the depth completion results using our model SemSegDepth.

DepthNet_b. This baseline network is the model proposed by Chen et al. (2020) which corresponds to the depth completion branch of the model shown in Figure 2 without concatenating the "Semantic Segmentation Preliminary Output" at the input. Hence, the only inputs are the RGB Input and the Sparse Depth.

We compared our model to this baseline networks and the results are presented in section 4.5. We also present the results obtained with different configurations of our model as ablation studies in section 4.6.

4.5 Results

The quantitative results for semantic segmentation and depth completion are shown in Table 1. The best performance is highlighted in bold letters.

Table 1: Results of our model compared to baseline networks.

Method	mIoU	RMSE(mm)
SemSegNet_b	0.520	-
DepthNet_b	-	580.2
SemSegDepth (ours)	**0.5932**	**458.2**

As shown in Table 1, our model outperforms each one of the baseline networks in every specific task. By combining both tasks, depth completion and semantic segmentation, our model achieved a significant improvement in the mIoU metric as compared to the semantic segmentation baseline model SemSegNet_b. On the other hand, there was also a major improvement in the depth completion task. Qualitatively, the results are shown in Figure 3 for the semantic segmentation task and in Figure 4 for the depth completion task.

4.6 Ablation Studies

In addition to the baseline models, namely SemSegNet_b and DepthNet_b, we also designed four other networks which correspond to slight modifications of our model. These networks can be understood as intermediate steps from the baseline networks to our final model. In this section we describe each one of these models.

SemNet_depth_gt. This is an extension of the semantic segmentation baseline network SemSegNet_b. This model is based on the model we proposed, shown in Figure 2, and modified by removing the depth completion branch. The input to the joint branch is the concatenation of the Semantic Segmentation Preliminary Output and the depth completion sparse depth ground truth. The purpose behind this model is to

evaluate whether or not providing depth information to the semantic segmentation baseline network can improve the performance for the task of semantic segmentation. The architecture of this model is shown in Figure 5.

SemNet_depth_dense_gt. In contrast to SemNet_depth_gt, in this model, the input to the joint branch is the concatenation of the Semantic Segmentation Preliminary Output and a dense depth map ground truth. Figure 6 shows the architecture of this model.

DepthNet_semantic_gt. As shown in Figure 7, this is a modification of the depth completion baseline network DepthNet_b. Similar to SemNet_depth_gt, we wanted to study whether or not providing reliable semantic information could improve the performance of the depth completion task alone. Therefore, we added one more input to the DepthNet_b network, it corresponds to the semantic segmentation ground truth image which is concatenated with the RGB input at the first concatenation layer.

SemSeg_Depth_a. This model, as shown in Figure 8, is based on SemNet_depth_gt, where, instead of using the depth completion sparse depth ground truth as input to the joint branch, we predict a dense depth map using DepthNet_b and pass it then as input to the joint branch.

SemSeg_Depth_b. Similar to SemSegDepth shown in Figure 2. This model combines semantic segmentation and depth completion in a multi-task network. However, different to SemSegDepth, the Semantic Segmentation Preliminary Output is not an input to the depth completion branch. Instead, we use another instance of the semantic segmentation branch to extract semantic features. All in all, this model consists of a feature extractor, two semantic segmentation branches (one of which works as an input to the depth completion branch), a depth completion branch and a joint branch. The architecture of this model is shown in Figure 9

SemSeg_Depth_c. This model follows the exact same architecture as SemSeg_Depth_b. The difference lies in the loss calculation. While in SemSeg_Depth_b we only calculate the semantic segmentation and the depth completion loss as in eq. 1 and eq. 3, respectively, in SemSeg_Depth_c we also calculate the joint loss as in eq. 4.

The quantitative results of all the models used in the ablation studies are shown in Table 2.

Table 2: Ablation Experiments.

Method	mIoU	RMSE(mm)
SemSegNet_b	0.520	-
DepthNet_b	-	580.2
SemNet_depth_gt	0.542	-
SemNet_depth_dense_gt	**0.638**	-
DepthNet_semantic_gt	-	833.7
SemSeg_Depth_a	0.5421	1497.0
SemSeg_Depth_b	0.5463	438.4
SemSeg_Depth_c	0.5841	**429.7**
SemSegDepth (ours)	0.5932	458.2

It is important to note that neither SemNet_depth_gt nor DepthNet_semantic_gt are significantly better in terms of their performance, despite having as input the corresponding sparse depth map ground truth and semantic segmentation ground truth respectively. SemNet_depth_dense_gt outperforms all the other models for semantic segmentation, suggesting that reliable depth data contains information useful for other tasks such as semantic segmentation. However, SemNet_depth_dense_gt relies heavily on a fully dense depth map ground truth as input, which is available in virtual environments only, whereas in real environments such ground truth is not available, hence sparse depth maps are a far more common.

SemSeg_Depth_a shows to perform better on the task of semantic segmentation but much worse performance for depth completion. On the other hand, sharing the backbone weights as in SemSeg_Depth_b and SemSeg_Depth_c, demonstrates to be a better performing approach. Furthermore, SemSeg_Depth_c outperforms SemSeg_Depth_b by calculating a joint loss as in eq. 4, even when both architectures are exactly the same.

Finally, in an attempt to share as many weights as possible, our proposed model also shares the weights of the semantic segmentation branch, yielding better performance in the semantic segmentation task. This further highlights the relevance of having two task-specific branches sharing weights in the network as in our proposed model SemSegDepth. Our model learns high-level features containing information for both tasks intrinsically, which is significantly more accurate as compared to having access to the complementary ground truth.

5 CONCLUSIONS

In this paper, we propose an end-to-end multi-task network for semantic segmentation and depth completion. It combines a modified version of two bench-marking models, more specifically, we used the model proposed by Chen et al. (2020) for depth completion and our semantic segmentation branch is based on the semantic segmentation branch of the EfficientPS model proposed by Mohan and Valada (2020).

With the proposed model, we successfully provide further evidence that multi-task networks can significantly improve the performance of each individual task by learning features jointly. Our model successfully predicts the fully dense depth map as well as the semantic segmentation image in a scene, given an RGB image and a sparse depth image as inputs to our model. In addition to that, our ablation studies demonstrate quantitatively, that our multi-task network outperforms, by a large margin, equivalent single-task networks.

REFERENCES

Cabon, Y., Murray, N., and Humenberger, M. (2020). Virtual kitti 2.

Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2016). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*, abs/1606.00915.

Chen, L., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018a). Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611.

Chen, L.-C., Collins, M. D., Zhu, Y., Papandreou, G., Zoph, B., Schroff, F., Adam, H., and Shlens, J. (2018b). Searching for efficient multi-scale architectures for dense image prediction.

Chen, Y., Yang, B., Liang, M., and Urtasun, R. (2020). Learning joint 2d-3d representations for depth completion.

Cheng, B., Collins, M. D., Zhu, Y., Liu, T., Huang, T. S., Adam, H., and Chen, L.-C. (2020). Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation.

Chollet, F. (2016). Xception: Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357.

Dauphin, Y. and Grangier, D. (2015). Predicting distributions with linearizing belief networks. *CoRR*, abs/1511.05622.

Dauphin, Y. N., Fan, A., Auli, M., and Grangier, D. (2016). Language modeling with gated convolutional networks. *CoRR*, abs/1612.08083.

Eigen, D. and Fergus, R. (2014). Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *CoRR*, abs/1411.4734.

Godard, C., Aodha, O. M., Firman, M., and Brostow, G. (2019). Digging into self-supervised monocular depth estimation.

Guizilini, V., Ambrus, R., Pillai, S., Raventos, A., and Gaidon, A. (2020). 3d packing for self-supervised monocular depth estimation.

Hazirbas, C., Ma, L., Domokos, C., and Cremers, D. (2016). Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *Asian Conference on Computer Vision (ACCV)*.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

He, L., Lu, J., Wang, G., Song, S., and Zhou, J. (2021). Sosd-net: Joint semantic object segmentation and depth estimation from monocular images. *CoRR*, abs/2101.07422.

Huang, Z., Fan, J., Cheng, S., Yi, S., Wang, X., and Li, H. (2020). Hms-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion.

Imran, S., Long, Y., Liu, X., and Morris, D. (2019). Depth coefficients for depth completion.

Kendall, A., Gal, Y., and Cipolla, R. (2017). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *CoRR*, abs/1705.07115.

Kim, D., Woo, S., Lee, J.-Y., and Kweon, I. S. (2020). Video panoptic segmentation.

Kreso, I., Causevic, D., Krapac, J., and Segvic, S. (2016). Convolutional scale invariance for semantic segmentation. In *GCPR*.

Liebel, L. and Körner, M. (2018). Auxiliary tasks in multi-task learning.

Lin, G., Milan, A., Shen, C., and Reid, I. (2016). Refinenet: Multi-path refinement networks for high-resolution semantic segmentation.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection.

Long, J., Shelhamer, E., and Darrell, T. (2014). Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038.

Ma, F., Cavalheiro, G. V., and Karaman, S. (2018). Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera.

Mohan, R. and Valada, A. (2020). Efficientps: Efficient panoptic segmentation. *CoRR*, abs/2004.02307.

Qiu, J., Cui, Z., Zhang, Y., Zhang, X., Liu, S., Zeng, B., and Pollefeys, M. (2018). Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. *CoRR*, abs/1812.00488.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *CoRR*, abs/1804.02767.

Ren, S., He, K., Girshick, R., and Sun, J. (2016). Faster r-cnn: Towards real-time object detection with region proposal networks.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597.

Sanchez-Escobedo, D., Lin, X., Casas, J. R., and Pardas, M. (2018). Hybridnet for depth estimation and semantic segmentation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1563–1567.

Takikawa, T., Acuna, D., Jampani, V., and Fidler, S. (2019). Gated-scnn: Gated shape cnns for semantic segmentation. *CoRR*, abs/1907.05740.

Tan, M. and Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. *CoRR*, abs/1905.11946.

Tang, J., Tian, F.-P., Feng, W., Li, J., and Tan, P. (2019). Learning guided convolutional network for depth completion.

Wang, H., Luo, R., Maire, M., and Shakhnarovich, G. (2020). Pixel consensus voting for panoptic segmentation.

Wang, S., Suo, S., Ma, W.-C., Pokrovsky, A., and Urtasun, R. (2018). Deep parametric continuous convolutional neural networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Weber, M., Luiten, J., and Leibe, B. (2020). Single-shot panoptic segmentation.

Xu, Y., Zhu, X., Shi, J., Zhang, G., Bao, H., and Li, H. (2019). Depth completion from sparse lidar data with depth-normal constraints.

Yang, Y., Wong, A., and Soatto, S. (2019). Dense depth posterior (DDP) from single image and sparse range. *CoRR*, abs/1901.10034.

Yao, J., Fidler, S., and Urtasun, R. (2012). Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 702–709.

Zamir, A. R., Sax, A., Shen, W. B., Guibas, L. J., Malik, J., and Savarese, S. (2018). Taskonomy: Disentangling task transfer learning. *CoRR*, abs/1804.08328.

Zhai, Y., Fu, J., Lu, Y., and Li, H. (2017). Feature selective networks for object detection.

Zhang, Y. and Funkhouser, T. A. (2018). Deep depth completion of a single RGB-D image. *CoRR*, abs/1803.09326.

Zou, N., Xiang, Z., Chen, Y., Chen, S., and Qiao, C. (2020). Simultaneous semantic segmentation and depth completion with constraint of boundary. *Sensors*, 20(3).

APPENDIX

All the models introduced in section 4.6 are presented here as appendices.

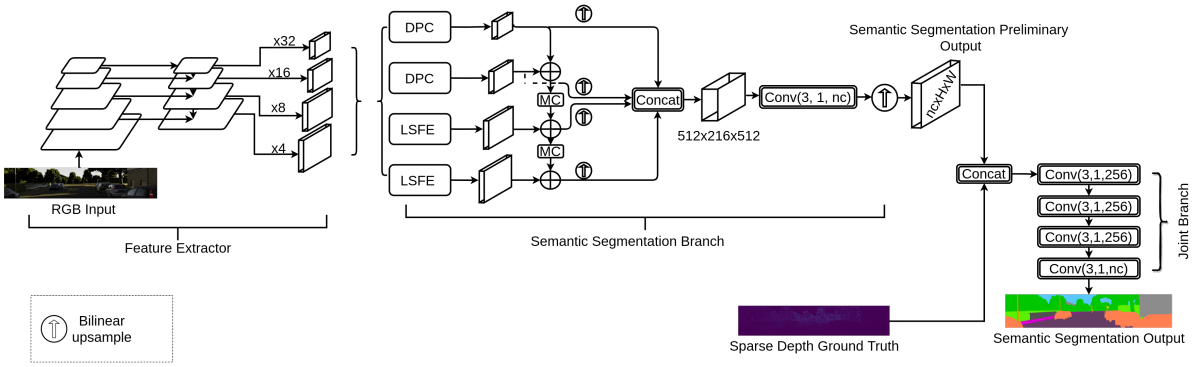


Figure 5: SemNet_depth_gt architecture.

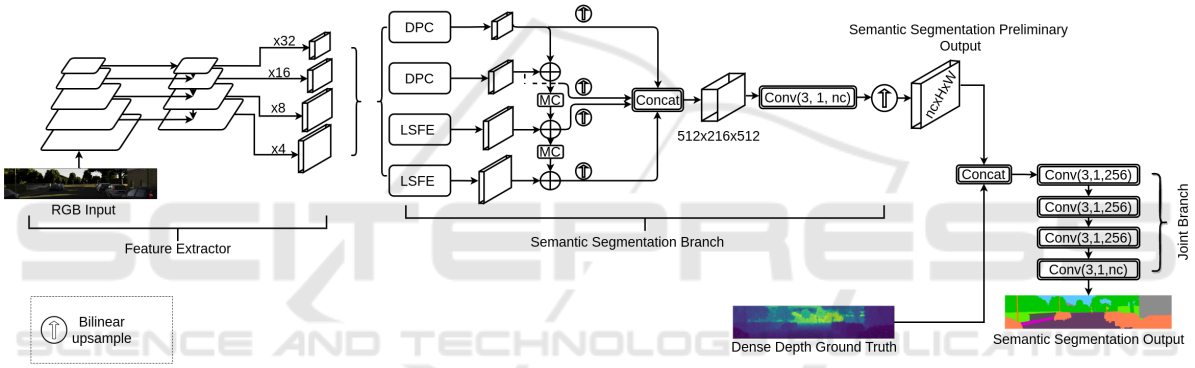


Figure 6: SemNet_depth_dense_gt architecture.

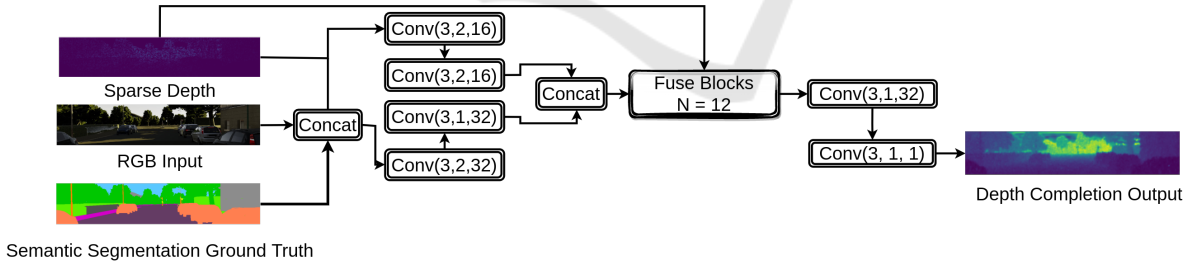


Figure 7: DepthNet_semantic_gt architecture.

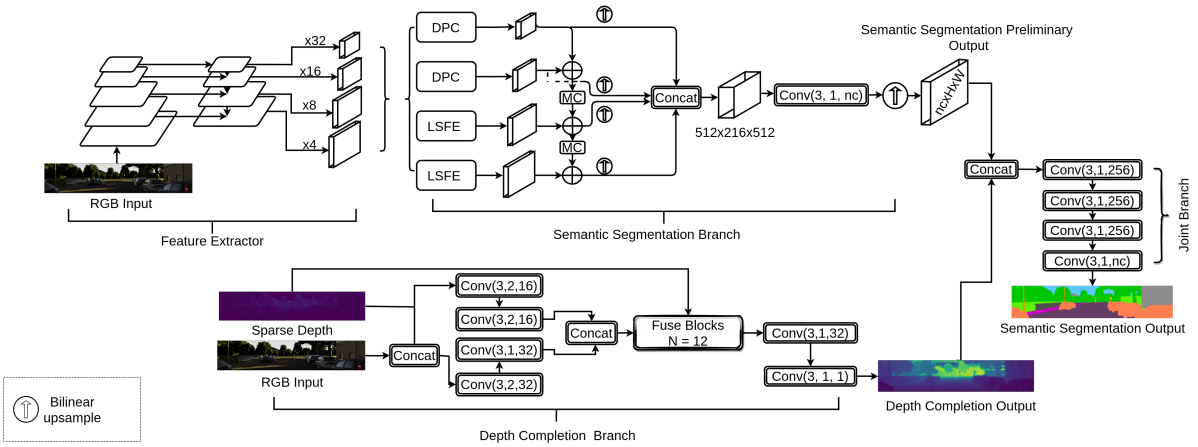


Figure 8: SemSegDepth_a architecture.

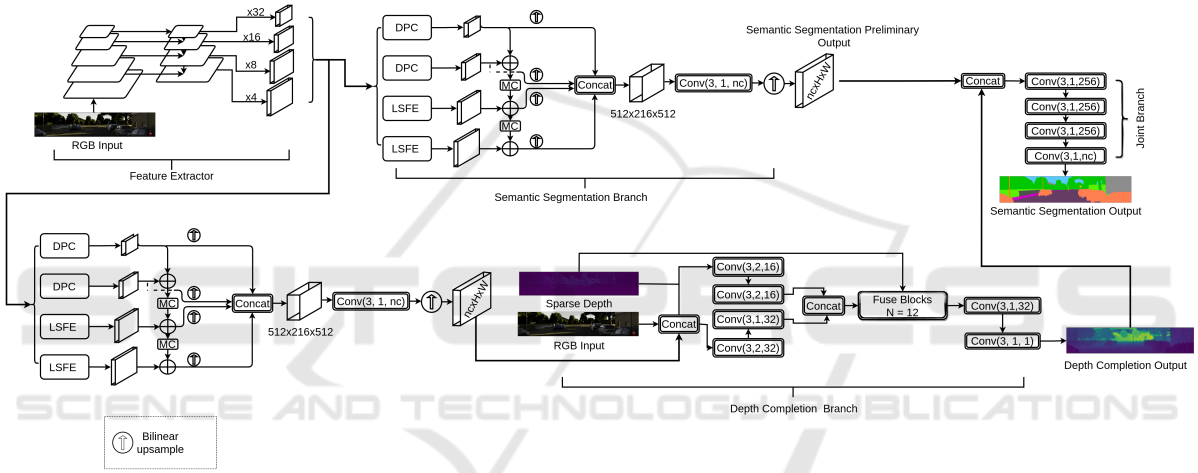


Figure 9: SemSegDepth_b architecture.