PanDepth: Joint Panoptic Segmentation and Depth Completion

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

Keywords: Panoptic Segmentation, Instance Segmentation, Semantic Segmentation, Depth Completion, CNN, Multi-Task Learning.

Abstract: Understanding 3D environments semantically is pivotal in autonomous driving applications where multiple computer vision tasks are involved. Multi-task models provide different types of outputs for a given scene, yielding a more holistic representation while keeping the computational cost low. We propose a multi-task model for panoptic segmentation and depth completion using RGB images and sparse depth maps. Our model successfully predicts fully dense depth maps and performs semantic segmentation, instance segmentation, and panoptic segmentation for every input frame. Extensive experiments were done on the Virtual KITTI 2 dataset and we demonstrate that our model solves multiple tasks, without a significant increase in computational cost, while keeping high accuracy performance. Code is available at https://github.com/juanb09111/PanDepth.git.

1 INTRODUCTION

Producing a holistic representation of a given scene has become essential in computer vision. The traditional tasks and challenges, such as semantic segmentation, instance segmentation, pose estimation, edge estimation, or depth completion only provide a limited representation that alone are not enough to successfully complete more complex tasks, for instance, autonomous driving, where, in addition to estimating the distance of the objects and stuff on and around the road, it is also essential to understand the semantic context of the scene, that is, identifying the type of objects around, e.g. cars, pedestrians, road lanes, traffic signs, at the same time as the depth to such objects is estimated. This raises the need for multi-task models that are capable of solving several tasks in parallel while keeping the computational cost low.

This work is inspired by the idea of devising a model that combines panoptic segmentation and depth completion which is of high relevance in applications such as autonomous driving where understanding 3D environments semantically is pivotal for the performance of autonomous machines. We explore the hypothesis that panoptic segmentation and depth completion can use cues from one another, more explicitly, that there are depth features that contain relevant semantic cues as well there are semantic segmentation features that contain relevant depth cues.

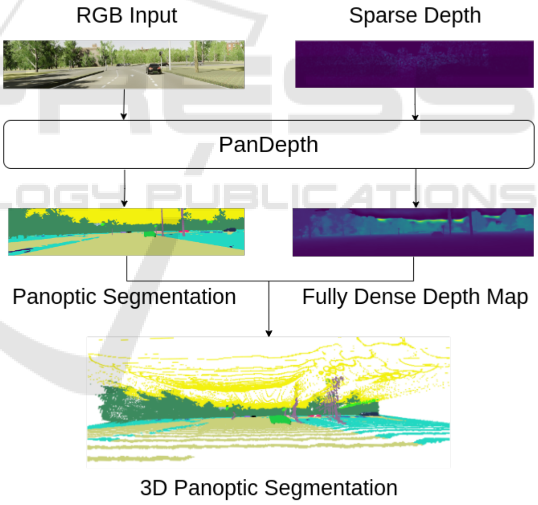


Figure 1: The proposed model (PanDepth) takes RGB images and sparse depth and returns the corresponding panoptic segmentation and fully dense depth map with which we create a 3D panoptic segmentation representation of the input frame.

Multi-task networks, not only reduce the demand for computational resources, as compared to running multiple single-task networks but also, there is empirical evidence that multi-task networks can perform better in each individual task by jointly learning features from all tasks involved (Ruder, 2017; Sener and Koltun, 2018; Lagos and Rahtu, 2022). For instance,

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

Lagos, J. and Rahtu, E.
PanDepth: Joint Panoptic Segmentation and Depth Completion.
DOI: 10.5220/0011685200003417
In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 5: VISAPP*, pages 635-643
ISBN: 978-989-758-634-7; ISSN: 2184-4321

depth features can be helpful for performing semantic segmentation and vice-versa. Even in applications where solving a single task is the primary goal, introducing features from other tasks may leverage accuracy and performance. Such paradigm is known as auxiliary tasks (Liebel and Körner, 2018a; Li and Dong, 2021), whereby solving other tasks it is possible to obtain relevant features which lead to a better performance in the main task.

We focus on solving three tasks in a joint manner, namely, semantic segmentation, instance segmentation, and depth completion using convolutional neural networks (CNNs). Combining semantic segmentation and instance segmentation into one single representation is known as panoptic segmentation (Kirillov et al., 2018). It provides a representation of an image where not only every pixel is assigned a label from a list of predefined labels, as in the case of semantic segmentation, but also, objects are detected as instances of a specific class, thus providing valuable information, such as the number of cars, people, or objects of a certain kind that are found in the image, as well as the semantic context of the non-countable stuff in the scene. Countable and non-countable objects are usually referred to as ”things” and ”stuff” in the context of computer vision (Adelson, 2001).

While semantic segmentation produces a single output, pixel-wise classification, instance segmentation produces three different outputs: bounding boxes for the objects detected, a label for each bounding box, and a segmentation mask for each object detected. The outputs of both tasks, semantic segmentation, and instance segmentation, are usually fused using heuristic methods with no learnable parameters (Mohan and Valada, 2020; Xiong et al., 2019).

On the other hand, depth completion aims to produce a dense depth map from sparse depth points which cover only a few pixels from a given image. Sparse depth maps can be obtained with active depth sensors, such as lidars. When 3D points obtained with lidars are projected onto an image, only about 5% of the image is covered (Uhrig et al., 2017). The goal is then to produce a dense depth map, with depth values for all the pixels in the image, given a sparse depth map as input.

In this paper, we propose an end-to-end model for panoptic segmentation and depth completion using joint training in order to provide a more holistic representation of the input images. In contrast with other works where predictions are made based on RGB images only (Gao et al., 2022; Schon et al., 2021; Yuan et al., 2021), our model processes heterogeneous data jointly, that is, RGB images and sparse depth maps as shown in Figure 1. For most machine perception applications, active depth sensors are part of the setup, for which we consider it more relevant to integrate both RGB images as well as sparse depth maps. We also quantify the effects of joint training as compared to training every task individually, thus providing more data on the growing evidence of the advantages of multi-task networks. We conduct extensive experiments on Virtual KITTI 2 (Cabon et al., 2020), which is a relevant dataset in the context of autonomous driving that contains ground truth annotations for instance segmentation, semantic segmentation and ground truth depth maps available for the entire dataset. Although panoptic segmentation ground truth is not directly provided by Virtual KITTI 2 dataset, we use semantic and instance segmentation ground truth to generate panoptic segmentation annotations.

2 RELATED WORKS

2.1 Panoptic Segmentation

Early works in computer vision developed CNN architectures for performing semantic segmentation and instance segmentation independently with reasonable success (Long et al., 2014; Ronneberger et al., 2015; He et al., 2017). Later on, Kirillov et al. (2018) proposed a task that would combine both tasks into one, which they named panoptic segmentation. Kirillov et al. (2018) also defined a metric for assessing the performance of panoptic segmentation predictions referred to as panoptic quality (PQ), thus, providing a complete definition of the problem of panoptic segmentation with a target metric for performance comparison. Such a robust definition of the task called the attention of the community, leading to the first architectures for end-to-end panoptic segmentation using CNNs (Li et al., 2018; Hou et al., 2019; Cheng et al., 2019; Xiong et al., 2019; Liu et al., 2019; de Geus et al., 2019; Kirillov et al., 2019; Petrovai and Nedevschi, 2019).

The most common challenges that appeared with panoptic segmentation are how to optimize a shared feature extractor as well as how to combine semantic segmentation and instance segmentation predictions while keeping the computational cost low. Mohan and Valada (2020) proposed a model for panoptic segmentation which consists of two heads, namely, semantic segmentation and instance segmentation, a fusion module for combining the outputs of both heads, and a feature extractor based on a family of scalable CNNs known as EfficientNet (Tan and Le, 2019), where the resolution, depth, and width are balanced depending on the computational resources available. For multi-scale features, Mohan and Valada (2020) wrap the feature extractor into a two-way feature pyramid network (FPN). Similarly, Chen et al. (2020a) used the same concept of scalable networks to Residual Networks (ResNets) for performing panoptic segmentation.

Other approaches (Wang et al., 2020; Carion et al., 2020; Zhu et al., 2020; Cheng et al., 2021b; Li et al., 2021; Cheng et al., 2021a) have adopted transformers architecture (Vaswani et al., 2017), initially designed for text processing and sequence transduction, and integrated attention mechanisms for panoptic segmentation. In contrast with more traditional methods, with instance segmentation and semantic segmentation defined as sub-tasks, transformer-based models use queries to represent ”things” and ”stuff” classes and perform panoptic segmentation.

2.2 Depth Completion

The task of depth completion aims to transform a sparse depth map, usually obtained with active depth sensors e.g. Lidar, into a dense depth map. Lidar devices can only provide a limited amount of depth points when projected onto the corresponding image, raising the need for methods that can lead to a fully-dense representation of the depth of an entire image. Several works have used RGB images as guidance for depth completion (Qiu et al., 2018; Eldesokey et al., 2018; Gansbeke et al., 2019; Tang et al., 2019; Yang et al., 2019; Park et al., 2020; Hu et al., 2021). Jaritz et al. (2018) proposed an encoder-decoder network architecture for depth completion, based on a late fusion of RGB images and sparse depth maps. However, processing RGB and Lidar data is not trivial, since, in contrast to RGB images, sparse depth data lacks a natural grid structure unless projected onto a 2D space which also facilitates the usage of traditional 2D convolutional layers.

Nonetheless, when mapping 3D data to 2D, valuable information regarding the geometrical relationship among the points in the 3D space is lost. Chen et al. (2020b) introduced a fuse block that exploits 3D cues by using parametric continuous convolution layers (Wang et al., 2018) while using 2D convolutions for processing RGB and later fusing the corresponding features in 2D space. Such 2D-3D fuse method is an essential building block in the proposed model, in which, with slight modifications to the model proposed by Chen et al. (2020b), we successfully map sparse depth maps to dense depth maps.

2.3 Multi Task Learning

CNNs can benefit from performing multiple tasks, as opposed to single-task networks. Branched CNNs consist of shared layers as well as task-specific layers, also known as branches. When such CNNs are trained, the weights of the shared layers are adjusted via back-propagation from each one of the branches, each one of which has one or multiple loss functions defined. In turn, the shared layers learn relevant features for all tasks, and such features are then fed to every branch. That allows for a very distinctive flow of information between the different branches. There is increasing evidence that single tasks, benefit when models are trained jointly improving the performance of each one of the tasks tackled by the network (Liebel and Körner, 2018b; Liu et al., 2018; Liebel and Körner, 2019; Zou et al., 2020a; Guo et al., 2020).

While some multi-task networks have addressed tasks relatively similar e.g. instance segmentation and semantic segmentation, other works have combined semantic segmentation and depth completion as end-to-end models (Hazirbas et al., 2016; Zou et al., 2020b; He et al., 2021). Lagos and Rahtu (2022) proposed a combined model for semantic segmentation and depth completion using RGB images and sparse depth maps, where it is demonstrated quantitatively and visually how each task outperforms equivalent single-task models for semantic segmentation and depth completion trained independently. Our model performs depth completion, instance segmentation, and semantic segmentation. We fuse instance and semantic segmentation to obtain a panoptic segmentation representation. In contrast with other methods, our model processes heterogeneous data, more specifically RGB images, and sparse depth maps using a stack of 2D-3D fuse blocks as proposed by Chen et al. (2020b)

3 ARCHITECTURE

3.1 Overview

The proposed model performs panoptic segmentation and depth completion in an end-to-end manner. It consists of a two-way feature pyramid network (FPN) as a shared feature extractor, three task-specific branches, one for each task (semantic segmentation, instance segmentation, and depth completion), one joint branch that refines the semantic logits using the resulting depth maps as guidance, and one final block for combining semantic and instance logits based on the fusion block proposed by Mohan

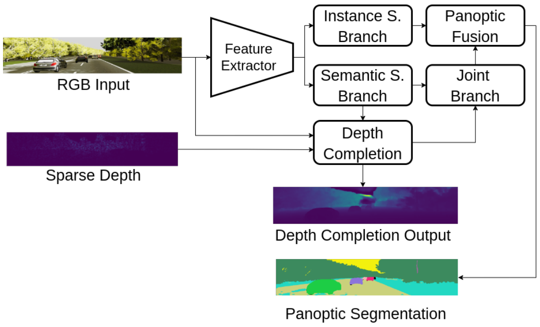


Figure 2: Overview of the proposed PanDepth architecture. Given an RGB image and sparse depth map as input, our model outputs the corresponding dense depth map and panoptic segmentation.

and Valada (2020). The inputs to our model are RGB images and sparse depth maps and the output is the corresponding panoptic segmentation representations and fully dense depth maps. The panoptic segmentation output and the resulting depth maps can be further combined to produce a 3D panoptic segmentation representation as shown in Figure 1.

3.2 Backbone

The backbone consists of a two-way FPN with an EfficientNet-B5 (Tan and Le, 2019) at the core, as shown in Figure 3. On one hand, the FPN upsamples lower resolution features and adds them together. On the other hand, the FPN downsamples higher-resolution features and adds them together. This allows for multi-scale feature extraction. The backbone returns feature maps at four different scales, downscaled by a factor of ×4, ×8, ×16, and ×32 with respect to the spatial resolution at the input.

3.3 Semantic Segmentation Branch

The semantic segmentation branch is a light-weighted structure that consists of three main building blocks based on the model proposed by Mohan and Valada (2020). Firstly, a Large Scale Feature Extractor (LSFE) extracts localized fine features. Secondly, a small-scale feature extractor based on Dense Predictions Cells (DPC), and finally, a Mismatch Correction Module (MC) is used in order to properly aggregate features at different scales. The input to this branch consists of the four feature maps returned by the backbone, they are in four different scales, ×4, ×8, ×16, and ×32. The tensors returned by the LSFE and DPC modules are aggregated as shown in Figure 3. Finally, this branch returns preliminary semantic segmentation logits of size $nc \times H \times W$, where nc is the total number of classes and $H \times W$ refers to the spatial resolution of the input $height \times width$ respectively. At a later stage, the preliminary semantic segmentation logits are refined with depth maps as guidance in the joint branch.

3.4 Instance Segmentation Branch

The instance segmentation branch is a lighter version of Mask R-CNN (He et al., 2017). Following the modifications suggested by Mohan and Valada (2020), all the convolutions were replaced by depth-wise separable convolutions (Chollet, 2016), batch normalization layers were replaced by synchronized Inplace Activated Batch Normalization layers (iABN) (Bulò et al., 2017) and the ReLU activations were replaced by Leaky ReLU.

Similar to Mask R-CNN, the instance segmentation branch consists of two stages. In the first stage, a region proposal network (RPN) returns a set of rectangular regions with a corresponding objectness score. Thereafter, a RoIAlign module extracts small feature maps of size 7×7 from the regions returned by the RPN. Subsequently, those features are used as input to two sub-branches that run in parallel, one of which regresses bounding boxes and classifies the objects of each corresponding box, and another sub-branch that regresses the corresponding masks returning an output tensor of size $NI \times 28 \times 28$, where NI corresponds to the number of instances detected.

3.5 Depth Completion Branch

This branch processes frame by frame and takes three different input types. Firstly, a sparse depth map originated from a $3D$ to $2D$ projection of a point cloud. Secondly, the corresponding RGB frame, and thirdly a preliminary semantic segmentation map as shown in Figure 3. Our depth completion branch is based on the architecture proposed by Chen et al. (2020c), upon which we made modifications in order to use preliminary semantic segmentation maps as proposed by Lagos and Rahtu (2022). At the input level, the sparse depth map is passed through two 2D convolutional layers of kernel size 3×3, while the RGB image and the semantic segmentation map are concatenated and passed through two $2D$ convolutional layers of kernel size 3×3. Subsequently, the two corresponding outputs are concatenated and, along with the original sparse depth map, they serve as input to a stack of N $2D - 3D$ Fuse Blocks. Finally, the resulting tensor from the Fuse Blocks passes through two $2D$ convolutional layers of kernel size 3×3 for refinement, yielding the final fully dense depth map.

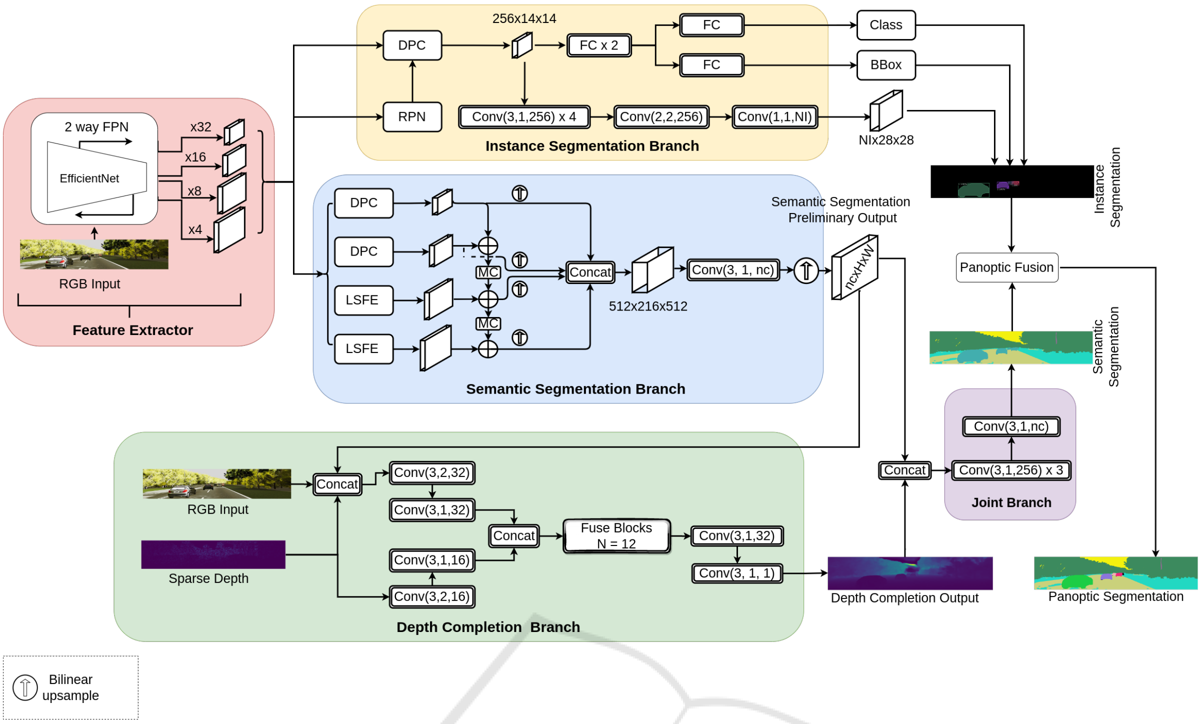


Figure 3: PanDepth architecture. Our model consists of a feature extractor, three task-specific branches (*i.e.* instance segmentation, semantic segmentation, and depth completion), a joint branch, and a panoptic fusion module. The convolutional layers in this diagram follow the notation Conv(k,s,c) $\times n$ representing a stack of n convolutional layers where k refers to a kernel of size $k \times k$, s is the stride, c is the number of output feature channels, and FC represents a fully connected layer.

3.6 Joint Branch

We use the depth completion output as guidance for refining the semantic segmentation preliminary output. This branch, albeit simple, successfully leverages the performance of the semantic segmentation task. It consists of four stacked $2D$ convolutions of kernel size 3×3. The input to this branch is the concatenation of the output of the depth completion branch, that is, a fully dense depth map, and the preliminary semantic segmentation output. Finally, this branch returns a tensor of size $nc \times H \times W$, where nc is the total number of classes in the dataset, H and W correspond to the original height and width of the model's input respectively.

3.7 Loss Functions

Semantic Segmentation. We used the weighted per-pixel log-loss for semantic segmentation. It is defined as follows:

$$L_{semantic} = -\sum_{i} w_i p_i \log \hat{p}_i, \tag{1}$$

where i is the pixel index, $w_i = \frac{4}{WH}$ if pixel i is within the 25% worst predictions, $w_i = 0$ otherwise. W and H correspond to the width and height of the input image respectively, p_i and $\hat{p}_i$ are the ground truth and the predicted probability for pixel i of belonging to class label $c \in p$ respectively. The predicted probability $\hat{p}_i$ is computed using the Softmax function defined as:

$$Softmax(x_n) = \frac{exp(x_n)}{\sum_m exp(x_m)}. \tag{2}$$

Instance Segmentation. We adopted the loss functions for instance segmentation as defined in Mask R-CNN (He et al., 2017). There are loss functions defined for the two stages of this branch. In the first stage (the RPN), we calculate two losses, namely, objectness score loss (L_{os}) and object proposal loss (L_{op}. For the second stage, we calculate three losses, classification loss L_{cls}, bounding-box regression loss L_{box}, and mask loss L_{mask}. The total loss for the instance segmentation branch is given by:

$$L_{instance} = L_{os} + L_{op} + L_{cls} + L_{box} + L_{mask} \tag{3}$$

Depth Completion. We used Mean Squared Error (MSE) as loss for the depth completion branch. The MSE was calculated and averaged over the pixels for which the corresponding ground truth depth values were available in the sparse depth map. The loss function is defined by

$$L_{depth} = \frac{1}{N}\sum_{i}(\hat{y}_i - y_i)^2, \tag{4}$$

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

Joint Loss. In addition to the loss function related to each task, we compute a loss involving each one of the tasks performed by our model, in particular, semantic segmentation, instance segmentation, and depth completion. This loss is simply the sum of every specific loss as in e.q. 5

$$L_{joint} = L_{semantic} + L_{instance} + L_{depth}. \tag{5}$$

4 EXPERIMENTS

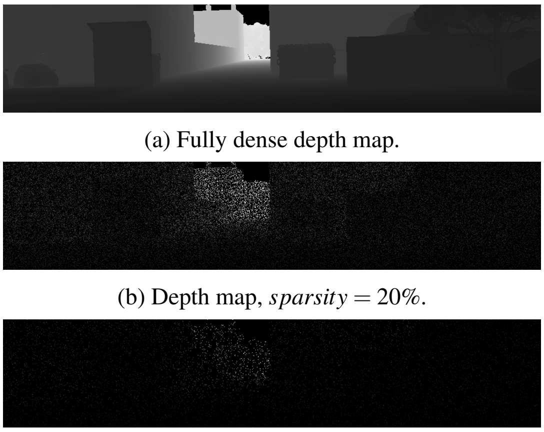
4.1 Implementation Details

We trained our model for 50 epochs on one machine with four 32GB graphics processing units (GPUs) running in parallel. The loss functions were optimized using Adam algorithm with a learning rate set to 0.0002.

4.2 Dataset

We trained and tested our models on Virtual KITTI 2 (Cabon et al., 2020). It is a synthetic dataset that provides ground truth annotations for semantic segmentation, instance segmentation, depth estimation, and optical flow for the entire dataset. It consists of five scenes named *"Scene01"*, *"Scene02"*, *"Scene06"*, *"Scene18"*, and *"Scene20"* which account for a total of 2126 unique frames of stereo images that are augmented to recreate 10 different environment conditions: clone, fog, morning, overcast, rain, sunset, and four angle variations corresponding to ±15° and ±30° around the vertical axis. All in all, Virtual KITTI 2 contains 21260 RGB stereo frames.

In our experiments, we discarded the angle variation splits, ±15° and ±30°, to reduce redundancy in the dataset and kept the other six splits for training, evaluation, and testing. We trained on scenes *"Scene01"*, *"Scene06"*, and *"Scene20"*, evaluated on *"Scene18"* and tested on *"Scene02"*. We resized the input frames to 200px height and 1000px width.



(a) Fully dense depth map.

(b) Depth map, $sparsity = 20\%$.

(c) Depth map, $sparsity = 5\%$.

Figure 4: Depth maps visualization at different sparsity levels.

Pre-Processing. Since Virtual KITTI 2 is a synthetic dataset, it provides fully dense depth maps for every single frame, however, in order to recreate real conditions as close as possible, we sampled the ground truth maps and set the sparsity to 20%, meaning that only 20% of the pixels from any given image would have a depth value available. On the other hand, non-ground-truth maps were sampled to have a sparsity of 5%. Under real-world conditions, 3D scenes are mapped with laser scanner devices, and when the 3D points are projected onto a 2D plane, they account for approximately 5% coverage of the entire image. The ground truth, however, is usually obtained by merging consecutive maps together, thus increasing the sparsity to around 20% (Uhrig et al., 2017). Figure 4 depicts the visual contrast between different sparsity levels.

Panoptic Segmentation Annotations for Virtual KITTI 2. Although Virtual Kitti 2 does not provide panoptic segmentation annotations directly, it is possible to use semantic segmentation and instance segmentation annotations to generate ground truth panoptic segmentation annotations. All the scripts are provided in the code repository. Thus, we hope to increase the interest of the community in this dataset as well as other possible datasets for which this approach might be found suitable and useful.

4.3 Evaluation Metrics

We calculated the standard COCO metrics (Lin et al., 2014) for every task. More specifically, we computed the Intersection over Union (IoU) for semantic segmentation, Mean Average Precision (mAP) for object detection, as well as PQ, recognition quality (RQ),

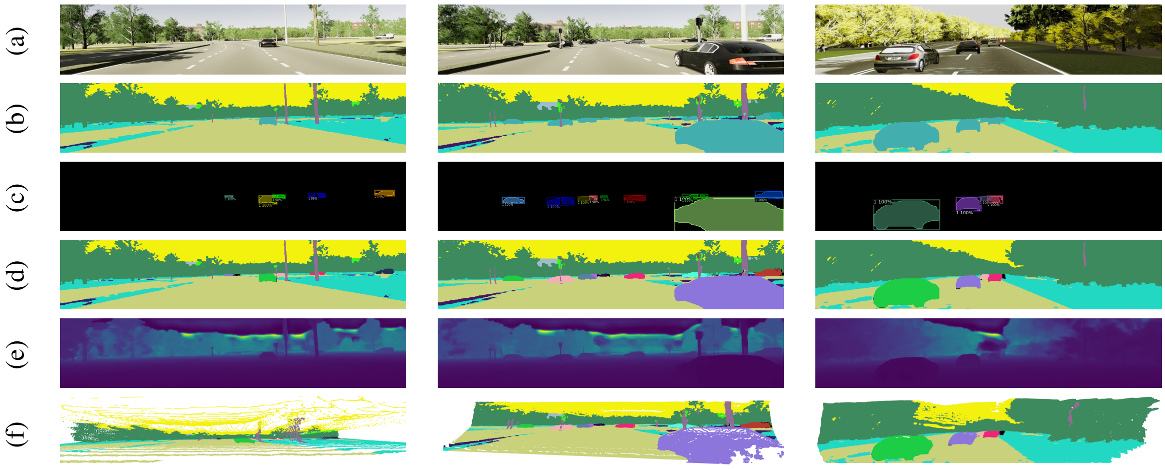


Figure 5: Panoptic segmentation and depth completion results on Virtual KITTI 2. Rows from top down show: *(a)* RGB input images, *(b)* semantic segmentation, *(c)* instance segmentation, *(d)* panoptic segmentation, *(e)* depth completion output, and *(f)* 3D panoptic segmentation.

Table 1: Results of our model compared to baselines.

Method	mIoU	mAP	RMSE(mm)	PQ	RQ	SQ
Semantic_only	0.380	-	-	-	-	-
Instance_only	-	0.691	-	-	-	-
Depth_only	-	-	623	-	-	-
SemSegDepth	0.387	-	677	-	-	-
PanDepth(ours)	0.413	0.597	653	0.384	0.450	0.467

and segmentation quality (SQ) for panoptic segmentation. In addition to the COCO metrics, we computed the root means squared error (RMSE) to evaluate the performance of the depth completion task.

4.4 Results

We compared the proposed PanDepth model against equivalent models where only one of the task-specific branches of PanDepth is enabled. Such models are listed in Table 1 as *"Semantic_only"*, *"Instance_only"*, and *"Depth_only"*. Table 1 also shows the performance of the proposed model PanDepth compared to SemSegDepth (Lagos and Rahtu, 2022), a joint-learning model for semantic segmentation and depth completion. SemSegDepth is a multi-task learning model that follows an architecture similar to that of our model PanDepth. On one hand, it consists of task-specific branches with a shared backbone. On the other hand, the input comprises heterogeneous data, namely, RGB frames and sparse depth maps. However, our model solves more tasks, thus providing a more holistic representation of the input scenes, while keeping high accuracy in all evaluation metrics as shown in Table 1. The qualitative results of the proposed model can be inspected visually in Figure 5, where the output of every individual task is depicted as well as a 3D panoptic segmentation reconstructed using the corresponding depth completion output and panoptic segmentation output.

Our model outperforms SemSegDepth in both the accuracy of the semantic segmentation task, as measured by the mIoU metric, and the depth completion task, as measured by the RMSE metric. The proposed PanDepth model also outperforms the semantic-segmentation-only model (*"Semantic_only"*) providing more evidence of the advantages of joint-learning. Although the single-task models *"Instance_only"* and *"Depth_only"*, for instance segmentation and depth completion respectively, show an increase in accuracy compared to PanDepth, as reported by the mAP and the RMSE, the proposed PanDepth model provides a more complete scene understanding of 3D environments which is a favorable trade-off in autonomous driving applications where holistic scene representations are highly valuable.

It is also important to note that the size of our model does not increase significantly despite solving multiple tasks. That is due to sharing structures such as the feature extractor and relatively small model branches as shown in Table 2.

Table 2: Model size.

Structure	Params
Backbone (EfficientNet-B5)	25.2M
2-way FPN	1.5M
Semantic Branch	1.2M
Instance Branch	53.1M
Depth Branch	1.9M
Joint Branch	1.2M
PanDepth Total Params	84M

5 CONCLUSIONS

This paper presents an end-to-end model for panoptic segmentation and depth completion using heterogeneous data as input, namely RGB images, and sparse depth maps. Our model yields a better scene understanding by providing a semantic representation of 3D environments. We propose a joint-learning method to perform multiple tasks, specifically semantic segmentation, instance segmentation, depth completion, and panoptic segmentation. Through a rigorous set of experiments, we demonstrate, quantitatively and qualitatively, the advantages of joint learning and multi-task models. Our model solves multiple computer vision tasks, keeping high-accuracy results compared to other strong baselines, without a significant increase in computational cost.

REFERENCES

Adelson, E. H. (2001). On seeing stuff: the perception of materials by humans and machines. In *IS&T/SPIE Electronic Imaging*.

Bulò, S. R., Porzi, L., and Kontschieder, P. (2017). In-place activated batchnorm for memory-optimized training of dnns. *CoRR*, abs/1712.02616.

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

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. *CoRR*, abs/2005.12872.

Chen, L., Wang, H., and Qiao, S. (2020a). Scaling wide residual networks for panoptic segmentation. *CoRR*, abs/2011.11675.

Chen, Y., Yang, B., Liang, M., and Urtasun, R. (2020b). Learning joint 2d-3d representations for depth completion. *CoRR*, abs/2012.12402.

Chen, Y., Yang, B., Liang, M., and Urtasun, R. (2020c). 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. (2019). Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. *CoRR*, abs/1911.10194.

Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., and Girdhar, R. (2021a). Masked-attention mask transformer for universal image segmentation. *CoRR*, abs/2112.01527.

Cheng, B., Schwing, A. G., and Kirillov, A. (2021b). Per-pixel classification is not all you need for semantic segmentation. *CoRR*, abs/2107.06278.

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

de Geus, D., Meletis, P., and Dubbelman, G. (2019). Fast panoptic segmentation network. *CoRR*, abs/1910.03892.

Eldesokey, A., Felsberg, M., and Khan, F. S. (2018). Confidence propagation through cnns for guided sparse depth regression. *CoRR*, abs/1811.01791.

Gansbeke, W. V., Neven, D., Brabandere, B. D., and Gool, L. V. (2019). Sparse and noisy lidar completion with RGB guidance and uncertainty. *CoRR*, abs/1902.05356.

Gao, N., He, F., Jia, J., Shan, Y., Zhang, H., Zhao, X., and Huang, K. (2022). Panopticdepth: A unified framework for depth-aware panoptic segmentation.

Guo, P., Lee, C., and Ulbricht, D. (2020). Learning to branch for multi-task learning. *CoRR*, abs/2006.01895.

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., Gkioxari, G., Dollár, P., and Girshick, R. B. (2017). Mask R-CNN. *CoRR*, abs/1703.06870.

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.

Hou, R., Li, J., Bhargava, A., Raventos, A., Guizilini, V., Fang, C., Lynch, J. P., and Gaidon, A. (2019). Real-time panoptic segmentation from dense detections. *CoRR*, abs/1912.01202.

Hu, M., Wang, S., Li, B., Ning, S., Fan, L., and Gong, X. (2021). Penet: Towards precise and efficient image guided depth completion. *CoRR*, abs/2103.00783.

Jaritz, M., de Charette, R., Wirbel, É., Perrotton, X., and Nashashibi, F. (2018). Sparse and dense data with cnns: Depth completion and semantic segmentation. *CoRR*, abs/1808.00769.

Kirillov, A., Girshick, R. B., He, K., and Dollár, P. (2019). Panoptic feature pyramid networks. *CoRR*, abs/1901.02446.

Kirillov, A., He, K., Girshick, R. B., Rother, C., and Dollár, P. (2018). Panoptic segmentation. *CoRR*, abs/1801.00868.

Lagos, J. P. and Rahtu, E. (2022). Semsegdepth: A combined model for semantic segmentation and depth completion. In Farinella, G. M., Radeva, P., and Bouatouch, K., editors, *Proceedings of the 17th International Joint Conference on Computer Vision, Imag-*

ing and Computer Graphics Theory and Applications, VISIGRAPP 2022, Volume 5: VISAPP, Online Streaming, February 6-8, 2022, pages 155–165. SCITEPRESS.

Li, B. and Dong, A. (2021). Multi-task learning with attention : Constructing auxiliary tasks for learning to learn. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 145–152.

Li, Y., Chen, X., Zhu, Z., Xie, L., Huang, G., Du, D., and Wang, X. (2018). Attention-guided unified network for panoptic segmentation. *CoRR*, abs/1812.03904.

Li, Z., Wang, W., Xie, E., Yu, Z., Anandkumar, A., Alvarez, J. M., Lu, T., and Luo, P. (2021). Panoptic segformer. *CoRR*, abs/2109.03814.

Liebel, L. and Körner, M. (2019). Multidepth: Single-image depth estimation via multi-task regression and classification. *CoRR*, abs/1907.11111.

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

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

Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312.

Liu, H., Peng, C., Yu, C., Wang, J., Liu, X., Yu, G., and Jiang, W. (2019). An end-to-end network for panoptic segmentation. *CoRR*, abs/1903.05027.

Liu, S., Johns, E., and Davison, A. J. (2018). End-to-end multi-task learning with attention. *CoRR*, abs/1803.10704.

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

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

Park, J., Joo, K., Hu, Z., Liu, C., and Kweon, I. S. (2020). Non-local spatial propagation network for depth completion. *CoRR*, abs/2007.10042.

Petrovai, A. and Nedevschi, S. (2019). Multi-task network for panoptic segmentation in automated driving. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 2394–2401.

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.

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

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098.

Schon, M., Buchholz, M., and Dietmayer, K. (2021). MGNet: Monocular geometric scene understanding for autonomous driving. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE.

Sener, O. and Koltun, V. (2018). Multi-task learning as multi-objective optimization. *CoRR*, abs/1810.04650.

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

Tang, J., Tian, F., Feng, W., Li, J., and Tan, P. (2019). Learning guided convolutional network for depth completion. *CoRR*, abs/1908.01238.

Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., and Geiger, A. (2017). Sparsity invariant cnns. *CoRR*, abs/1708.06500.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

Wang, H., Zhu, Y., Adam, H., Yuille, A. L., and Chen, L. (2020). Max-deeplab: End-to-end panoptic segmentation with mask transformers. *CoRR*, abs/2012.00759.

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*.

Xiong, Y., Liao, R., Zhao, H., Hu, R., Bai, M., Yumer, E., and Urtasun, R. (2019). Upsnet: A unified panoptic segmentation network. *CoRR*, abs/1901.03784.

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

Yuan, H., Li, X., Yang, Y., Cheng, G., Zhang, J., Tong, Y., Zhang, L., and Tao, D. (2021). Polyphonicformer: Unified query learning for depth-aware video panoptic segmentation. *CoRR*, abs/2112.02582.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. (2020). Deformable DETR: deformable transformers for end-to-end object detection. *CoRR*, abs/2010.04159.

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

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