Advances in Artificial Intelligence and Machine Learning; Research 4 (2) 2201-2212 Received 6-02-2024; Accepted 19-05-2024; Published 26-05-2024

SOccDPT: 3D Semantic Occupancy From Dense Prediction Transformers Trained Under Memory Constraints

Aditya Nalgunda Ganesh adityang5@gmail.com
Department of Computer Science
PES University
Bengaluru, Karnataka, India

Corresponding Author: Aditya Nalgunda Ganesh

Abstract

We present SOccDPT, a memory-efficient approach for 3D semantic occupancy prediction from monocular image input using dense prediction transformers. To address the limitations of existing methods trained on structured traffic datasets, we train our model on unstructured datasets including the Indian Driving Dataset and Bengaluru Driving Dataset. Our semi-supervised training pipeline allows SOccDPT to learn from datasets with limited labels by reducing the requirement for manual labeling by substituting it with pseudo-ground truth labels to produce our Bengaluru Semantic Occupancy Dataset. This broader training enhances our model's ability to handle unstructured traffic scenarios effectively. To overcome memory limitations during training, we introduce patch-wise training where we select a subset of parameters to train each epoch, reducing memory usage during auto-grad graph construction. In the context of unstructured traffic and memory-constrained training and inference, SOccDPT outperforms existing disparity estimation approaches as shown by the RMSE score of 9.1473, achieves a semantic segmentation IoU score of 46.02% and operates at a competitive frequency of 69.47 Hz. We make our code and semantic occupancy dataset public[1].

Keywords: 3D Vision, Semantic occupancy, Depth perception, Occupancy network

1. INTRODUCTION

Autonomous navigation requires 3D semantic understanding of the environment at a high frequency with a limited compute budget. The field of autonomous driving has shown significant interest in vision-based 3D scene perception due to its exceptional efficiency and abundant semantic information. When it comes to choosing an architecture, works such as [1–4] inspired from ViT [5] have the domain agnostic learning capabilities of the transformer. The transformer's versatility comes at the cost of having no good inductive priors for any domain, requiring large volumes of training data and a large volume of GPU memory to train. To apply such models on a new domain, we must be

[1] `https://adityang.github.io/SOccDPT`

Citation: Aditya Nalgunda Ganesh. SOccDPT: 3D Semantic Occupancy From Dense Prediction Transformers Trained Under Memory Constraints. Advances in Artificial Intelligence and Machine Learning 2024;4(2):126

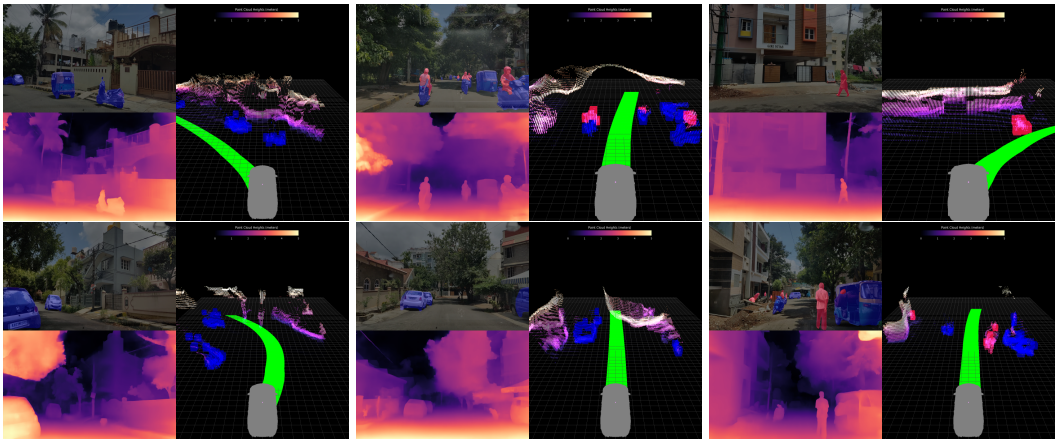


Figure 1: Above are a few frames from our Bengaluru Semantic Occupancy Dataset which is an extension of the Bengaluru Driving Dataset [6]. Each panel consists of the RGB image with 2D semantic labels on the top left, the disparity map on the bottom left and the 3D semantic occupancy on the right. The vehicle and pedestrian classes are colored in blue and red respectively. Objects without classes have been plotted as a height map for the sake of visualization. The vehicle and its future trajectory have been plotted out in grey and green respectively to aid the reader to have a better scene understanding.

efficient in making use of transfer learning and pseudo-labeling to solve the ground truth data scale problem.

In the context of 3D semantic occupancy from monocular vision, ground truth data would refer to semantically labeled 3D point clouds with corresponding RGB images acquired from a calibrated camera sensor as shown in FIGURE 1 of our Bengaluru Semantic Occupancy Dataset. While there exist datasets [7, 8] which have labeled 3D semantic occupancy data in the context of structured traffic, the unstructured traffic scenarios remain largely underrepresented. It may not be feasible to gather large volumes of training data considering the fact that LiDAR sensors are expensive and labeling 3D semantic classes can be tedious. Hence, we make use of a set of teacher models and boosting techniques inspired from [6, 9–11] to produce labels for depth and semantic on driving video footage which we use to supervise the training of our model. We train our system on unstructured driving datasets such as the Indian Driving Dataset [12] and Bengaluru Driving Dataset [6] to ensure that our system generalizes well. Training such models requires large volume of GPU memory. We overcome this hurdle with our PatchWise training approach which keeps the GPU memory in check and this allowed us to explore higher batch sizes without altering the back-propagation algorithm.

With the goal of designing a model, that is efficient during both training and inference, we propose SOccDPT and our PatchWise training system. To ensure SOccDPT performs well in unstructured traffic scenarios, we introduce our pseudo-labeling process to generate our unstructured traffic dataset. We use a common backbone for image feature extraction and dual heads to extract disparity and semantic information of the scene. Camera intrinsics are used along with disparity to project the semantic information into 3D space.

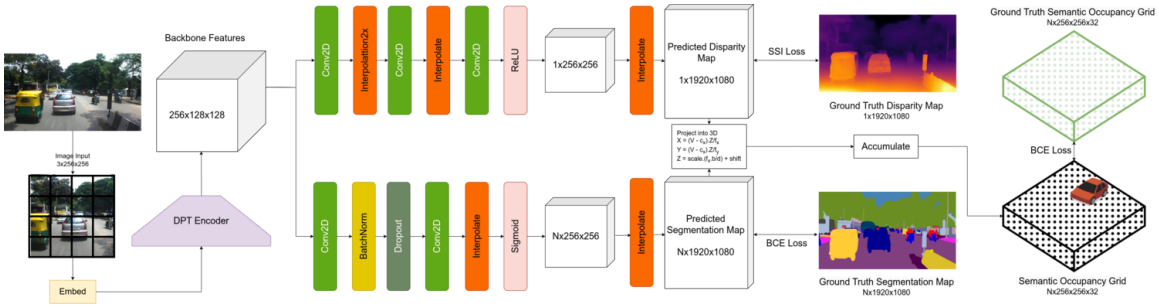


Figure 2: SOccDPT uses the ViT family for backbone feature extraction which allows us to carefully balance accuracy and compute requirements. SOccDPT takes an RGB image input of shape $3 \times 256 \times 256$ produces image features of shape 256x128x128. We then pass the extracted features to a disparity head and a segmentation head. We apply the Scale and Shift Invariant loss [13] and the Binary Cross Entropy loss for the disparity and segmentation outputs respectively. With the known camera intrinsic, we project the semantics into 3D space with the help of the disparity map and accumulate the semantics into a 3D occupancy grid of size $256 \times 256 \times 32$, thus producing a 3D semantic map from one backbone

2. RELATED WORK

Semi-Supervised Learning and Self-Supervised Learning. In the context of disparity estimation, semi-supervised learning has become very important due to the challenges involved in obtaining accurate depth information in diverse real-world environments. Several self-supervised algorithms for perceiving depth have been suggested [14–16]. These algorithms offer the advantage of utilizing only a single camera, making them suitable for easy deployment in real-world scenarios. However, they still face numerous unresolved issues. One such problem is the generation of disparity maps that lack local and temporal consistency. Watson et al. [16] addressed the temporal inconsistency by incorporating multiple consecutive frames as input. Another line of research in semi-supervised learning looks into using the existing model to generate confident annotations on unlabelled data. Since the degree of disparity is inversely related to depth, as demonstrated in FIGURE 1, slight variations in disparity for distant objects lead to significant variations in depth. Consequently, the resulting point clouds exhibit non-uniform resolution, with closer objects represented by more points compared to those farther away. There are broadly two approaches to the disparity estimation problem: monocular and stereoscopic.

Monocular and Stereo Disparity Estimation. Diverse neural network architectures, including variational auto-encoders, convolutional neural networks, generative adversarial networks and recurrent neural networks, have demonstrated their efficacy in tackling the task of depth estimation. Within this framework, two methods are commonly employed: monocular, where depth is estimated from a single input image, and stereoscopic depth estimation, where depth is estimated from a pair of images provided as input to the system. Monocular approaches such as [13–16] take advantage of depth cues such as occlusion boundaries, parallel lines and so on to understand the 3D scene.

Bird's Eye View (BEV) Architectures. Since disparity is inversely proportional to depth, errors in depth estimation grow quadratically with disparity errors. To address these errors, some approaches

look to operate in the Bird's Eye View space which is directly proportional to depth. Obtaining a top-down view of a scene offers a comprehensive understanding of the surrounding environment, effectively capturing both static and dynamic elements. BEV architectures, exemplified by [17–19], generate this top-down map, which can be utilized for path planning purposes. This top down map is essentially a segmentation map that would highlight the road, non-drivable space, parking areas, vehicles, pedestrians and so on. The concept of predicting BEV from multiple camera perspectives has demonstrated performance comparable to LiDAR-centric methods [20, 21]. However, a limitation of this approach is the absence of 3D information about the scene, such as unclassified objects, potholes, and overhanging obstacles.

3D Occupancy Networks. Difficult to classify 3D obstacles become harder and harder to catch as the research community and industry chase the long tail of nines. Recent approaches look to build generic 3D object detection free of ontology by way of 3D Occupancy Networks. Achieving an effective representation of a 3D scene is a fundamental objective in perceiving 3D environments. One direct approach involves discretizing the 3D space into voxels within an occupancy grid [21, 22]. The voxel-based representation is advantageous for capturing intricate 3D structures, making it suitable for tasks like LiDAR segmentation [21, 23] and 3D scene completion [24, 25]. A recent method, TPVFormer [26], addresses memory optimization by representing the 3D space as projections on three orthogonal planes. Despite the significant progress made by these approaches, they do not specifically tackle the issue of existing dataset biases towards structured traffic.

3. PROPOSED WORK

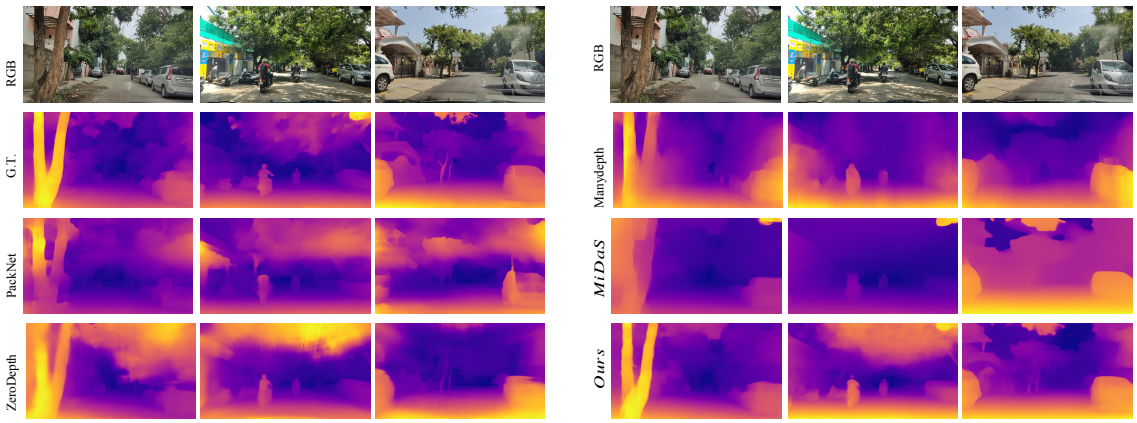


Figure 3: Qualitative results comparing frames in BDD to Midas [27] versions, monodepth [14], manydepth [16], ZeroDepth [28]. As we can see, all the existing approaches do not address the diversity that is seen in unstructured traffic

3.1 SOccDPT Architecture

As described in FIGURE 2, SOccDPT uses the Dense Prediction Transformer [4, 27] backbones to efficiently extract image features. We then use independent heads to produce the disparity and

segmentation maps. Instead of penalizing the model for generating the output in an inaccurate scale, we address the issue of arbitrary scale in the disparity map by estimating the scale and shift relative to the ground truth for every frame. This estimation process involves aligning the prediction with the ground truth using a least-squares criterion. Once the segmentation and disparity maps are computed, we make use of the camera intrinsics to project the semantics into 3D space. Consider a point on the image plane at position (u, v) with disparity $D(u, v)$ and 2D semantics $S_{2D}(u, v)$. This point corresponds to the 3D point (x, y, z) as shown in FIGURE 1 from which we can assert the 3D semantics correspondence to be $S_{3D}(x, y, z) \longleftarrow S_{2D}(u, v)$

$$(x, y, z) = (\frac{b \cdot (u - o_x)}{D(u, v)}, \frac{b \cdot f_x \cdot (v - o_y)}{f_y \cdot D(u, v)}, \frac{b \cdot f_x}{D(u, v)}) \tag{1}$$

In order to train our network, we started off by building a baseline model $V1$ which consists of 2 separate backbones, one for disparity and the other for segmentation. This informs us of the performance of the dense prediction transformer on unstructured traffic datasets. We improve upon $V1$ by having a common backbone in $V2$ which lead to optimizations in speed and memory consumption. This came at the cost of the accuracy of both the segmentation and disparity. This is due to the fact that the network would be learning the features and intricacies of both the tasks from scratch simultaneously. To address this, $V3$ makes a minor modification to $V2$ which allows us to load in the disparity estimation backbone from $V1$. This allows $V3$ to have a backbone which is proficient in the disparity estimation task. When starting from this point, the backbone and segmentation head only have to learn the task of image segmentation, without making any major alterations to the existing disparity estimation. This provided an improvement in how much the model was able to learn with the same data.

3.2 PatchWise Training

Our PatchWise system offers a solution to GPU memory limitations during neural network training. Instead of updating all weights simultaneously, which can lead to ”out of memory” errors, PatchWise updates a subset of the model’s weights at a time. This approach enables the training of larger networks and the use of larger batch sizes on systems with limited GPU memory, although it increases training time. The implementation details are described in FIGURE 1.

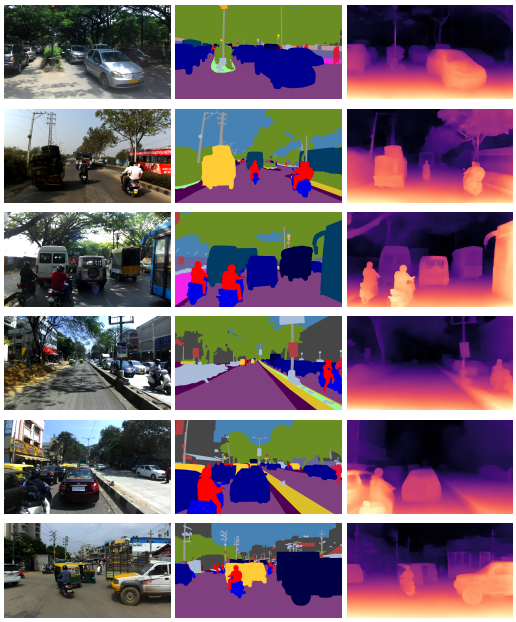
3.3 Pseudo-Ground Truth Labels for Semi-Supervision

The ability of vision-based networks to learn and accurately predict based on image input is limited by their receptive field and their overall learning capacity (number of parameters). We generate pseudo-labels using existing disparity estimation and image segmentation models by feeding in segments of the image which allows the model to focus on smaller regions. This approach trades compute time for higher accuracy. We augment the Indian Driving Dataset [12] with depth labels and the Bengaluru Driving Dataset [6] with 2D semantic labels, enhancing their utility for training.

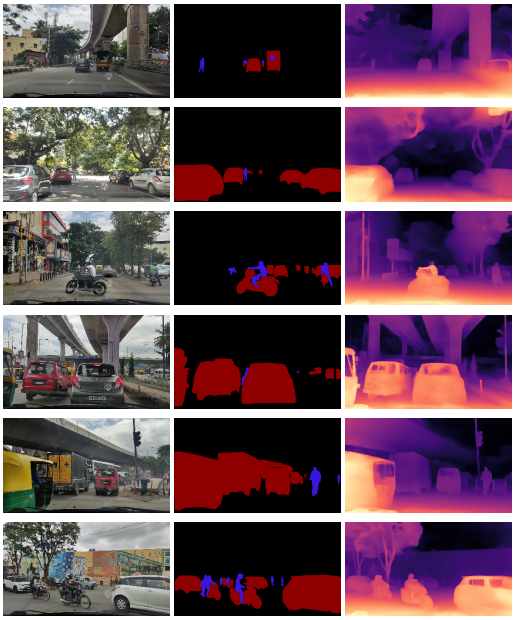
Algorithm 1: PatchWise

PatchWise (net, train_percentage, train_step);
Input : PyTorch Module net, training percentage train_percentage, training function train_step
Output: Trained neural network

```
N ← length(net.parameters);
M ← round(N × train_percentage);
num_iterations ← ⌈N/M⌉;
updated_weights ← {};
saved_weights ← {};
for index, param in net.parameters do
    saved_weights[index] ← param;
end
for net_patch_index in range(0, num_iterations) do
    start_index ← net_patch_index × M;
    end_index ← min(start_index + M, N);
    train_indices ← range(start_index, end_index)  for index, param in net.parameters do
        param ← saved_weights[index];
        param.requires_grad ← bool(
         index ∈ train_indices
        );
    end
    train_step(net);
    save_indices ← range(start_index, end_index);
    for index, param in net.parameters do
        if index ∈ save_indices then
            updated_weights[index] ← param;
        end
    end
end
for index, param in net.parameters do
    param ← updated_weights[index];
end
return net;
```



(a) Depth Boosting for the Indian Driving Dataset



(b) Semantic Segmentation auto-labeling for the Bengaluru Driving Dataset

Figure 4: **Auto Labeling**. We use Depth Boosting to generate depth labels for the Indian Driving Dataset. We use Depth Boosting to generate depth labels for the Indian Driving Dataset. We have the RGB frames on the left, segmentation map in the middle and our depth labels on the right in the sub-figures. We would like to highlight the detail in the automatically generated disparity maps and segmentation maps.

Depth Boosting. Monocular depth estimation systems use a lot of the depth cues used by humans including occlusion boundaries, parallel lines, edges, vanishing points and the shape and size of objects. Altering the resolution of the image affects the clarity of these depth cues. Taking inspiration from the depth boosting techniques [6, 10, 11], we merge the disparity maps from the various resolutions, we generate a high-resolution disparity map with global consistency. We use this method to generate disparity labels for the Indian Driving Dataset as shown in FIGURE 4a. The depth images on the left are colored by inverse depth (or disparity), such that pixels representing objects closer to the camera are brighter and those representing objects further away are darker.

Semantic Segmentation auto-labeling. To produce high resolution 2D semantic labels, we take inspiration from PointRend [9]. We take an image as input and produce a coarse intermediate segmentation map using a pre-trained segmentation model. This coarse map is gradually up-sampled using bi-linear interpolation and only the regions of the resized map with high uncertainty are refined. The uncertain regions typically include the boundaries of objects. The uncertain region is refined by a lightweight multi-layered perceptron. Its input is a feature vector that is extracted through interpolation from the feature maps, which intern has been computed by the base model. As shown in FIGURE 4b we have auto-labelled vehicles in red and humans in blue.

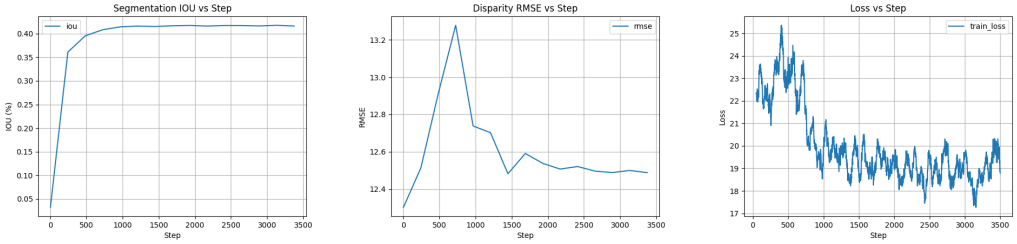


Figure 5: While training $SOccDPT_{V3}$ we start with the pre-trained depth backbone. As a result, the initial disparity metrics (RMSE, a1, a2, a3) are good while the initial IoU score is under 5%. Within the first few epochs, the IoU score starts growing steadily, and we observe a small spike in the disparity metrics as the depth head adjusts to the changes made to accommodate the segmentation head

Model	$RMSE^{\downarrow}$	$a1^{\uparrow}$	$a2^{\uparrow}$	$a3^{\uparrow}$	$FPS^{\uparrow}$ (Hz)	$Parameters^{\downarrow}$
$MiDaS\ Swin2_T$[4]	23.325	0.5944	0.7816	0.8585	**82.2656**	**14.84M**
$MiDaS\ DPT_H$[27]	21.861	0.5527	0.7712	0.8558	13.4394	123.1M
$MiDaS\ DPT_L$[27]	13.36	0.6888	0.8514	0.9192	6.3142	42.3M
$Manydepth$[16]	31.3599	0.4508	0.6668	0.7850	20.3578	35.34M
$ZeroDepth$[28]	36.3419	0.3882	0.6552	0.7875	2.363	232.59M
$PackNet$[29]	50.6722	0.2257	0.4936	0.6870	8.7504	128.29M
$SOccDPT_{V1}$	13.3782	0.6854	0.8442	0.9172	39.1141	84.3M
$SOccDPT_{V2}$	26.2383	0.4879	0.7181	0.8309	69.6503	42.3M
$SOccDPT_{V3}$	**12.4075**	**0.6935**	**0.8588**	**0.9265**	69.4733	42.3M

Table 1: We compare SOccDPT's disparity metrics on the Bengaluru Driving Dataset, FPS and number of parameters with existing approaches. $SOccDPT_{V3}$ outperforms the models in terms of accuracy while maintaining a high FPS and small model size

4. EXPERIMENTS

4.1 Experimental Setup

We train SOccDPT on a laptop with an Intel i7-12700H (20 threads) and NVIDIA GeForce RTX 3070 Laptop GPU with 8 GB VRAM. With the goal of focusing performance in unstructured traffic, our network has been trained on the Indian Driving Dataset [12] and the Bengaluru Driving Dataset [6]. In TABLE 2, we present the set of hyper-parameters which produce optimal results. We evaluate on the metrics Intersection over Union (IoU), Root Mean Squared Error (RMSE), threshold errors ($a1$, $a2$, $a3$). Here, a_i is the fraction of predictions where the threshold gt/pred or pred/gt is less than 1.25^i.

Method	Dataset	Hyperparameters			$RMSE^{\downarrow}$	$a1^{\uparrow}$	$a2^{\uparrow}$	$a3^{\uparrow}$	$IoU^{\uparrow}$ (%)
		BS	EP	LR		$\delta < 1.25^1$	$\delta < 1.25^2$	$\delta < 1.25^3$	
V1	IDD	12*	0.5	0.00001	11.2353	0.7717	0.8991	0.9211	42.48
	BDD	12*	0.5	0.00001	13.3782	0.6854	0.8442	0.9172	41.73
V2	IDD	6	0.95	0.00001	27.6473	0.5302	0.7084	0.8134	26.29
	BDD	6	0.95	0.00001	26.2383	0.4879	0.7181	0.8309	34.75
V3	IDD	6	0.95	0.0001	**9.1473**	**0.7807**	**0.9009**	**0.9416**	43.50
	BDD	6	0.95	0.0001	12.4075	0.6935	0.8588	0.9265	**46.02**

Table 2: **Ablation Study** SOccDPT's hyper-parameters and the metrics achieved. RMSE, a1, a2, a3 are disparity metrics and IoU is the segmentation metric. The hyper-parameters are batch size (BS), Encoder Percentage (EP) and learning rate (LR). Models with a * have had their two heads and backbones trained separately

4.2 Datasets

We make use of the IDD, BDD and BSOD datasets. The Indian Driving Dataset (IDD) [12] has a total of about 7974 frames with 6993 and 981 frames for training and testing respectively. The Bengaluru Driving Dataset (BDD) [6] has a total of about 3629 frames. We split it to have 10% for testing and the remainder for training. We present our Bengaluru Semantic Occupancy Dataset by extending BDD with 3D semantic occupancy labels by picking a voxel size of 50 cm and applying a voting filter to drop the voxels with fewer than 10 points.

4.3 Ablation Study

In TABLE 2, we present the set of hyper-parameters for SOccDPT's $V1$, $V2$, and $V3$. We observe that $V1$ produces good disparity and segmentation metrics while also being the largest network in terms of the number of parameters and the slowest to run as shown in TABLE 1. $SOccDPT_{V1}$ has two independent backbones which explain the larger number of parameters and increased inference time. While $SOccDPT_{V2}$ shows an improvement in speed and reduction in the number of parameters, it takes a performance hit in terms of disparity and segmentation accuracy, as this network is being trained from scratch. $SOccDPT_{V2}$ introduces the common backbone which reduces the compute requirements, but since this entire network is being trained from scratch, it has no priors regarding either semantics or disparity estimation. We introduce this prior into $SOccDPT_{V3}$ by changing the architecture of $V2$ to allow us to load in pre-trained weights from the disparity backbone. As seen in FIGURE 5, $SOccDPT_{V3}$ starts off with good RMSE scores for disparity estimation and poor IoU for segmentation, which is as expected. Through the course of training, the IoU steadily climbs. Initially, we see a spike in RMSE which comes back down over several epochs. $SOccDPT_{V3}$ has similar timing and memory characteristics when compared to $SOccDPT_{V2}$ as it is only a minor modification that allows us to load in the disparity backbone. But this small change allows $SOccDPT_{V3}$ to vastly outperform $SOccDPT_{V2}$ without requiring additional training data.

4.4 Comparison with Existing Methods

$SOccDPT_{V3}$'s performance exceeds existing disparity estimation approaches on unstructured traffic scenarios presented from the Bengaluru Driving Dataset. As shown in TABLE 1, $SOccDPT_{V3}$ shows the best accuracy in disparity estimation while also maintaining a high FPS and keeping compute requirements low. As shown in FIGURE 3, our model provides very detailed disparity maps compared to existing approaches while performing in real-time and keeping memory requirements low.

5. CONCLUSIONS

Existing disparity and segmentation approaches have come far, but do not specifically address the challenge in the autonomous vehicle context in unstructured traffic scenarios. We use depth boosting and semantic auto-labeling to build a self-supervised training pipeline, which can take videos as input and train a 3D semantic occupancy network. $SOccDPT$ uses a multi-headed Dense Transformer based architecture to take advantage of this self-supervised pipeline, to learn 3D semantic occupancy in the context of autonomous navigation in unstructured traffic. Our PatchWise training system allowed us to explore training with larger batch sizes which would not have been possible with memory-constrained hardware. These models show potential in their ability to learn 3D semantic occupancy from monocular vision and operate at real time.

References

[1] Graham B, El-Nouby A, Touvron H, Stock P, Joulin A, et al. Levit: A Vision Transformer in Convnet's Clothing for Faster Inference. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV). 2021;12259-12269.

[2] Jaegle A, Borgeaud S, Alayrac JB, Doersch C, Ionescu C, et al. Perceiver IO: A General Architecture for Structured Inputs & Outputs. 2021. Arxiv Preprint: https://arxiv.org/pdf/2107.14795

[3] Li J, Xia X, Li W, Li H, Wang X, Xiao X, et al. Next-Vit: Next Generation Vision Transformer for Efficient Deployment in Realistic Industrial Scenarios. 2022. ArXiv Preprint: https://arxiv.org/pdf/2207.05501

[4] Liu Z, Hu H, Lin Y, Yao Z, Xie Z, Wei Y, et al. Swin Transformer V2: Scaling up Capacity and Resolution. In: International Conference on Computer Vision and Pattern Recognition (CVPR); 2022:11999-2009.

[5] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, et al. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. 2020. Arxiv preprint: https://arxiv.org/pdf/2010.11929

[6] Ganesh AN, Pobbathi Badrinath D, Kumar HM, PS, Narayan S. Octran: 3D Occupancy Convolutional Transformer Network in Unstructured Traffic Scenarios. Transformers for Vision workshop. Google Spotlight Presentation at the Transformers for Vision Workshop. CVPR. 2023.

[7] Fong WK, Mohan R, Hurtado JV, Zhou L, Caesar H, et al. Panoptic Nuscenes: A Large-Scale Benchmark for Lidar Panoptic Segmentation and Tracking. 2021. ArXiv preprint: https://arxiv.org/pdf/2109.03805

[8] Geiger A, Lenz P, Stiller C, Urtasun R. Vision Meets Robotics: The Kitti Dataset. Int J Robot Res. 2013;32:1231-1237.

[9] Cheng B, Parkhi O, Kirillov A. Pointly-supervised instance segmentation. 2021. Arxiv Preprint: https://arxiv.org/pdf/2104.06404v1

[10] https://summit.sfu.ca/_flysystem/fedora/2022-11/etd22069.pdf

[11] Miangoleh SM, Dille S, Mai L, Paris S, Aksoy Y. Boosting Monocular Depth Estimation Models to High-Resolution via Content-Adaptive Multi-Resolution Merging. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE PUBLICATIONS. 2021; 9680-9689.

[12] Varma G, Subramanian A, Namboodiri AM, Chandraker M, Jawahar CV. Idd: A Dataset for Exploring Problems of Autonomous Navigation in Unconstrained Environments. 2018 Arxiv Preprint: https://arxiv.org/pdf/1811.10200

[13] Ranftl R, Lasinger K, Hafner D, Schindler K, Koltun V. Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer. IEEE Trans Pattern Anal Mach Intell. 2022;44:1623-1637.

[14] Godard C, Aodha OM, Brostow GJ. Unsupervised Monocular Depth Estimation With Left-Right Consistency.In Proceedings of the IEEE conference on computer vision and pattern recognition. 2017:270-279.

[15] Godard C, Aodha OM, Firman M, Brostow G. Digging Into Self-Supervised Monocular Depth Estimation. In Proceedings of the IEEE/CVF international conference on computer vision 2019:3828-3838.

[16] Watson J, Aodha OM, Prisacariu V, Brostow G, Firman M. The Temporal Opportunist: Self-Supervised Multi-Frame Monocular Depth. IEEE Computer Society, June 2021. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2021;1164-1174.

[17] Li Z, Wang W, Li H, Xie E, Sima C, et al. Bevformer: Learning Bird's-Eye-View Representation From Multi-Camera Images via Spatiotemporal Transformers. 2022. ArXiv preprint: https://arxiv.org/pdf/2203.17270

[18] Reiher L, Lampe B, Eckstein L. A sim2real Deep Learning Approach for the Transformation of Images From Multiple Vehicle-Mounted Cameras to a Semantically Segmented Image in Bird's Eye View. In: 23rd International Conference on Intelligent Transportation Systems (ITSC). IEEE PUBLICATIONS; 2020;1-7.

[19] Roddick T, Cipolla R. Predicting Semantic Map Representations From Images Using Pyramid Occupancy Networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition 2020; 6: 11138–11147.

[20] Liu Z, Tang H, Amini A, Yang X, Mao H, et al. Bevfusion: Multi-Task Multisensor Fusion With Unified Bird's-Eye View Representation. In: IEEE International Conference on Robotics and Automation (ICRA); 2023.

[21] Zhu X, Zhou H, Wang T, Hong F, Ma Y, et al. Cylindrical and Asymmetrical 3D Convolution Networks for Lidar Segmentation. IEEE Computer Society, June 2021. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2021;9934-9943.

[22] Zhou Y, Tuzel O. Voxelnet: End-To-End Learning for Point Cloud Based 3D Object Detection. IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE PUBLICATIONS; 2018: 4490-4499.

[23] Ye D, Zhou Z, Chen W, Xie Y, Wang Y, et al. Lidarmultinet: Towards a Unified Multi-Task Network for Lidar Perception. Clin Orthop Relat Res. 2022.

[24] Cao AQ, de Charette R. Monoscene: Monocular 3D Semantic Scene Completion. In: CVPR; 2022:3981-3991.

[25] Yan X, Gao J, Li J, Zhang R, Li Z, et al. Sparse Single Sweep Lidar Point Cloud Segmentation via Learning Contextual Shape Priors From Scene Completion. AAAI Conference on Artificial Intelligence. 2021;35:3101-3109.

[26] Huang Y, Zheng W, Zhang Y, Zhou J, Lu J. Tri-Perspective View for Vision-Based 3D Semantic Occupancy Prediction. 2023. ArXiv preprint: https://arxiv.org/pdf/2302.07817

[27] Ranftl R, Lasinger K, Hafner D, Schindler K, Koltun V. Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer. IEEE Trans Pattern Anal Mach Intell. 2022;44:1623-1637.

[28] Guizilini V, Vasiljevic I, Chen D, Ambru R, Gaidon A. Towards Zero-Shot Scale -Aware Monocular Depth Estimation. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV). 2023:9199-9209.

[29] Guizilini V, Ambrus R, Pillai S, Raventos A, Gaidon A. 3D Packing for Self-Supervised Monocular Depth Estimation. In: Proceedings of the international conference on computer vision and pattern recognition (CVPR); 2020:2482-2491.