Enhancing Monocular 3D Object Detection with Pseudo-Depth from Foundational Monocular Depth Estimation Models

Sanchit Tanwar
Georgia Institue of Technology
stanwar8@gatech.edu

Puneet Gupta
Georgia Institue of Technology
pgupta395@gatech.edu

Abstract

Robust 3D object detection is crucial for safe autonomous driving, but most methods rely on expensive LiDAR sensors. Monocular 3D detection is challenging due to the lack of direct depth cues. In this work, we leverage recent advancements in monocular depth estimation to enhance 3D detection from monocular images. We propose a two-stage approach: 1) Generating pseudo-depth from RGB images using a state-of-the-art depth estimation model and encoding it in an HHA format suitable for convolutional processing. 2) Fusing RGB and HHA features in a multi-stream CenterNet detection model to predict 3D bounding boxes. We also explore using a 2D detector to generate region proposals for the 3D model. Our method achieves significant improvements on the KITTI dataset, especially for the car category, by effectively leveraging depth cues. The proposed approach enables improved 3D detection by exploiting depth information implicit in monocular images, reducing the need for expensive depth sensors. With further refinement, it can potentially enhance 3D detection robustness and scalability for autonomous driving.

1. Introduction

Object detection is a fundamental task in computer vision, with numerous applications in autonomous driving, robotics, and surveillance. While significant progress has been made in 2D object detection, 3D object detection remains a challenging problem due to the added complexity of understanding scene geometry. In particular, monocular 3D object detection poses a unique set of challenges, as depth information is lost in monocular images, making it difficult to comprehend the geometry of the scene, and making monocular 3d object detection an ill posed problem.

Recent advancements in monocular depth estimation have shown promising results in predicting both relative and metric depth from monocular RGB images [12], [1]. These models rely on learning visual priors of the scene, such as the typical length of a car, from large-scale datasets, these large scale datasets ensures zero shot transfer of these models on various settings including indoor and outdoor data. With the availability of such models and data, it becomes possible to leverage these priors for 3D object detection tasks.

In this work, we propose a novel approach to monocular 3D object detection that leverages recent advancements in monocular depth estimation. Our approach comprises two key phases. First, we experiment with RGB-D based 3D object detection models, substituting the original depth with pseudo-depth generated by a state-of-the-art monocular depth estimation model. Departing from the conventional point-cloud processing approach commonly employed by most modern LiDAR-based 3D object detection methods, we instead opt for a convolutional architecture identical to that used for processing RGB data. We feed this new backbone with depth information encoded in the HHA format, which has been shown to be effective in capturing geometric cues [6].

In the second phase, we extend our experiments by incorporating a 2D object detection model to generate proposals for 3D object detection. This approach is motivated by two key observations. First, 2D object detection models have consistently demonstrated higher detection rates and superior object classification capabilities compared to their 3D counterparts. Second, in real-world scenarios, 2D and 3D object detection often work in tandem, even when a dedicated 3D object detection model is available. For instance, 2D object detection remains essential for identifying classes such as red lights and traffic signs. By leveraging the objects detected by the 2D model as region proposals for the 3D object detection model, we can effectively harness the strengths of both approaches. Moreover, we propose a novel multi-head architecture in which the 3D detection outputs are generated from both the original CenterNet-like architecture and an additional head that utilizes these 2D proposals. The predictions from these two heads are then merged using non-maximum suppression (NMS), which eliminates redundant predictions and retains only the unique predic-

tions generated by each head. The approach is discussed in more details in 3 and experiments are discussed in more details in 4.

The practical implication of this experiment is far-reaching, as it opens up the possibility of leveraging the vast array of monocular datasets available for training 3D object detection models, even when depth information is not readily available. This is particularly significant because acquiring depth data can be costly and time-consuming, often requiring specialized hardware such as LiDAR sensors. By demonstrating the feasibility of using pseudo-depth generated from monocular images, our approach effectively reduces the reliance on expensive depth-sensing equipment and facilitates the widespread adoption of 3D object detection in resource-constrained settings.

Moreover, the ability to pretrain models on large-scale monocular datasets with available depth information enables us to later run these models directly with depth data obtained from more affordable sensors, such as stereo cameras, time-of-flight (ToF) sensors, or solid-state LiDAR. This approach ensures that we make the most of already collected monocular datasets, reducing the need for additional data collection efforts.

The KITTI dataset [5] is a widely-used benchmark for various computer vision tasks in autonomous driving, including 3D object detection. It consists of real-world driving scenarios captured in Karlsruhe, Germany, using a vehicle equipped with multiple sensors. The dataset provides a diverse set of annotated data, including stereo image pairs, 3D point clouds, and 3D object annotations for objects such as cars, pedestrians, and cyclists. KITTI has become a standard for evaluating 3D object detection algorithms in the context of autonomous driving. We benchmark our approach on KITTI dataset.

2. Related Work

This section explores relevant research in 3D object detection using LiDAR data, 2D object detection, monocular 3D object detection, and monocular depth estimation, highlighting some of the most popular and successful models in the industry.

3D Object Detection with LiDAR : LiDAR sensors provide high-fidelity depth information, enabling accurate 3D object detection. Noteworthy approaches that have achieved state-of-the-art performance on 3D object detection benchmarks include PointNet++ [13] and SECOND [23]. PointNet++ utilizes a hierarchical architecture to effectively process point clouds, setting a benchmark for accuracy in 3D object detection. Conversely, SECOND employs a sparse convolution approach, enabling real-time processing of point clouds and making it a practical choice for real-world applications. While LiDAR-based methods are highly accurate, their reliance on expensive LiDAR sensors hinders widespread adoption in cost-sensitive applications.

2D Object Detection : 2D object detection forms the foundation for many computer vision tasks. Deep learning-based methods dominate this field, with some of the most prominent and successful architectures being YOLO (You Only Look Once) [15], SSD (Single Shot MultiBox Detector) [10], and Faster R-CNN [16]. YOLO, known for its speed and single-stage detection approach, strikes a balance between accuracy and efficiency, making it suitable for real-time applications. Similarly, SSD efficiently handles objects of varying scales, while Faster R-CNN, employing a two-stage detection process, offers a compromise between speed and accuracy. The robust capabilities of 2D object detection lay the groundwork for advancements in monocular 3D object detection.

Monocular 3D Object Detection : Monocular 3D object detection presents a significant challenge due to the absence of explicit depth information. Existing methods can be broadly categorized into two approaches: image-based methods and pseudo-LiDAR based methods. Image-based methods, exemplified by FCOS3D [19] and MonoDIS [17], directly estimate 3D bounding boxes from monocular images, addressing challenges such as sparse objects and class imbalance. Conversely, pseudo-LiDAR based methods, like MVD [18] and Pseudo-LiDARNet [20], generate pseudo-LiDAR point clouds from monocular depth estimation and apply LiDAR-based 3D detection techniques. Our work falls under the pseudo-LiDAR category, investigating the effectiveness of leveraging recent advancements in monocular depth estimation for generating informative pseudo-depth representations for 3D object detection. We build on top of [11] which uses a centernet [25] head for anchor free 3d object detection.

Monocular Depth Estimation : Recent advancements in monocular depth estimation offer promising solutions for predicting depth from single images. Key works in this area that are pushing the boundaries of performance include UniDepth [12] and ZoeDepth [1]. UniDepth leverages a large-scale dataset with diverse indoor and outdoor scenes to achieve robust metric depth prediction, making it a valuable tool for various applications beyond 3D object detection. Similarly, ZoeDepth utilizes self-supervised learning and large-scale vision transformers to achieve high-quality depth estimation across various illumination and weather conditions. The progress in monocular depth estimation, as exemplified by UniDepth and ZoeDepth, motivates our approach to explore its potential for generating informative pseudo-depth for 3D object detection tasks.

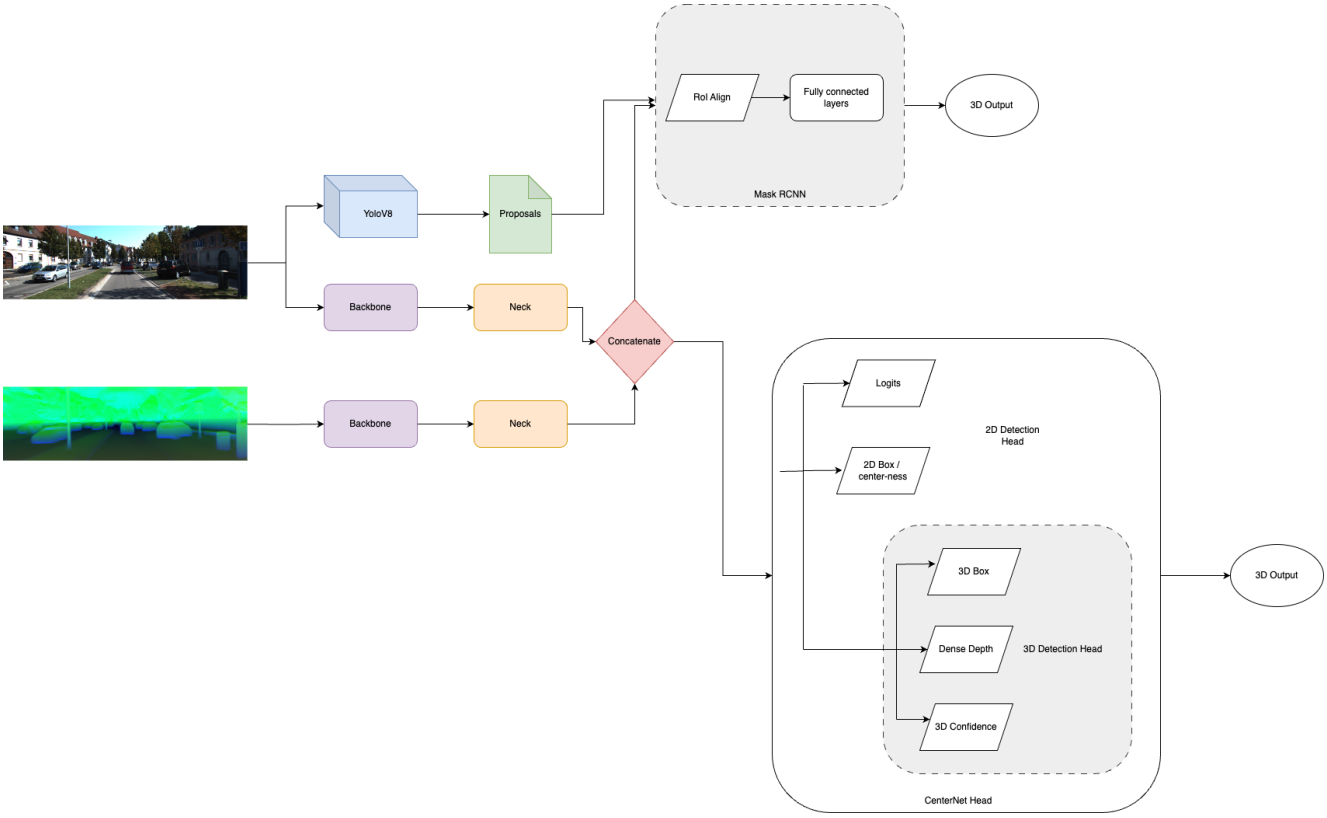

Figure 1. Overall structure of the pipeline

3. Methodology

3.1. Problem Definition

3D object detection is a problem in which we aim to predict the 2D bounding box, category, and 3D bounding box of an object in an image. The 3D bounding box is modeled using the center location $[x, y, z]$, size $[h, w, l]$, and heading angle γ of the object. Although the heading of an object is usually modeled using three degrees of freedom, in the case of objects on the road, we assume that the roll and pitch of the object are zero. Unlike 2D bounding boxes, which are modeled in the image space, 3D bounding boxes are in the 3D world space, meaning that the $[x, y, z]$ coordinates and the $[h, w, l]$ dimensions have metric units. This makes monocular 3D object detection a challenging problem because depth information is lost when projecting the 3D world onto the image space, rendering monocular 3D object detection an ill-posed problem.

3.2. Model

We chose the model proposed by [11] as our baseline. It is based on the CenterNet architecture and uses DLA-34 [24] as the backbone, FPN [9] as the neck, and CenterNet for the head. The pipeline is summarized in 1

3.2.1 CenterNet

CenterNet is a multi-head architecture that uses multiple heads to predict 2D bounding boxes, 3D bounding boxes, classes, and other attributes. Following the neck, we apply a CenterNet-based head to obtain the predictions of 2D and 3D bounding boxes. The first head predicts a heatmap with C channels for C classes, and outputs a 2D Gaussian at the center of the bounding boxes in the image space c. A second head predicts the offset o_i to the predicted center, and the final 2D bounding box location is retrieved as $c + o_i$. A third head is used to predict the width and height of the 2D bounding box. For the 3D bounding box, we predict an offset o_w for the 3D center from the predicted center c in the world space. Thus, the final 3D center will be at $c + o_w$. This offset is regressed as the projected 3D center from the 3D bounding box, and we can retrieve the final x and y coordinates using the depth of this center and the camera intrinsics, as shown in Equation 1. Three additional heads are used for decoding the 3D bounding box. The first head predicts the depth, the second head predicts the 3D size $[h, w, l]$ and heading γ, and the third head predicts the depth z. The heading is predicted by first dividing the yaw into n bins and using a classification head to predict these bins. Then, an offset is predicted to finally obtain the heading. The cam-

era projection matrix equation, where $[x_{3d}, y_{3d}, z]$ is the 3D location, $[u, v]$ are the projected points, and K is the camera intrinsic matrix, is given by:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} x_{3d} \\ y_{3d} \\ z \end{bmatrix} \tag{1}$$

The camera intrinsic matrix K can be retrieved by camera calibration. For the kitti dataset it is already available.

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Once we get z from the depth head, we can retrieve the $[x_{3d}, y_{3d}]$ as,

$$\begin{bmatrix} x_{3d} \\ y_{3d} \\ z \end{bmatrix} = K^{-1} \begin{bmatrix} (c + o_w)_x \cdot z \\ (c + o_w)_y \cdot z \\ z \end{bmatrix} \tag{2}$$

3.2.2 Monocular Depth estimation

To obtain depth information from monocular images, we employ the state-of-the-art model, UniDepth [12], for monocular depth estimation. In addition to predicting metric depth, UniDepth also directly predicts camera intrinsics and point clouds, which can be highly beneficial for cameras with known intrinsics. Predicted depth and pointclouds using Unidepth can be seen in 3.We first infer metric depth for both training and validation images and then convert the metric depth to HHA encoding [6] using the Depth2HHA-python toolbox [3]. Some sample HHA images are visualized in 2.

Processing depth directly with convolutional networks is suboptimal due to depth discontinuities and scale invariances. Depth maps often contain discontinuities at object boundaries, and since convolutions are invariant to translation, the depth values at different locations may vary even for similar or identical objects. This poses a challenge for processing depth images with convolutions because of the inductive biases of the model. To address these issues, most modern depth processing algorithms first convert depth to intermediate representations such as point clouds, voxels, meshes, HHA, or surface normals. For the sake of simplicity and to utilize architectures similar to those used for RGB images, we choose HHA as our intermediate representation. Although transformer-based architectures do not have image-specific inductive biases, they perform well only with large datasets. In this work, we use the KITTI dataset, which is relatively small, and thus, we opt for a convolutional network with HHA encoding.

3.2.3 Multi-stream Fusion

As HHA is a 3-channel representation, we use the same backbone and neck architecture to process HHA as we do for RGB. An alternative approach is to stack HHA with RGB, but some literature suggests that early fusion is suboptimal due to the different distributions of depth and RGB. Therefore, we create a copy of the backbone and neck to obtain two different feature maps, which are then concatenated along the channel dimension, implementing late fusion. These fused feature maps are finally passed to the head discussed in Section 3.2.1. Although other fusion techniques could be better, such as sharing features in middle layers to exploit the correlation between the two streams, we could not explore them due to time constraints.

3.2.4 Region Proposal Network

Another design choice we make is to use 2D proposals from a 2D object detection model. To generate these proposals, we train a YOLOv8 model [8] and use its inference output as proposals. We apply an ROI Align layer [7] on the concatenated feature maps discussed in Section 3.2.3. The extracted features are passed to a new head, which outputs three elements: object size, heading, and 3D center. The size is directly predicted, while the heading is predicted similarly to the CenterNet head. The 3D center is estimated by predicting an offset from the 2D center of the proposal. We use a fully connected layer on the flattened output from the ROI Align layer. This head does not provide uncertainty or classification scores, and the class predicted by the 2D detection model is considered as the final class. We merge the predictions from the two heads and apply NMS algorithm in these merged predictions, this acts as an ensemble and thus can improve the overall results.

3.2.5 Losses

There are seven loss terms in CenterNet and three in the RPN. In CenterNet, focal loss is used for heatmap prediction, which is responsible for classification. L1 loss is used for regressing the 2D center and size. For depth, a heteroscedastic regression loss is used, in which we regress both depth and standard deviation σ. This loss, given in Equation 3, was proposed in [4]. L1 loss is used for 3D center refinement and 3D size estimation. For heading, we use cross-entropy loss for the bin part and L1 loss for the regression. The number of bins is set to 12 throughout the experiments. For the region proposal network, we use the same loss as CenterNet for heading and L1 loss for 3D size and location estimation. We assign a weight of 1 to all the losses. For the loss of region proposal networks we consider only the proposals which match with a target with an IOU overlap of 0.5+ and we drop all the proposals which don't

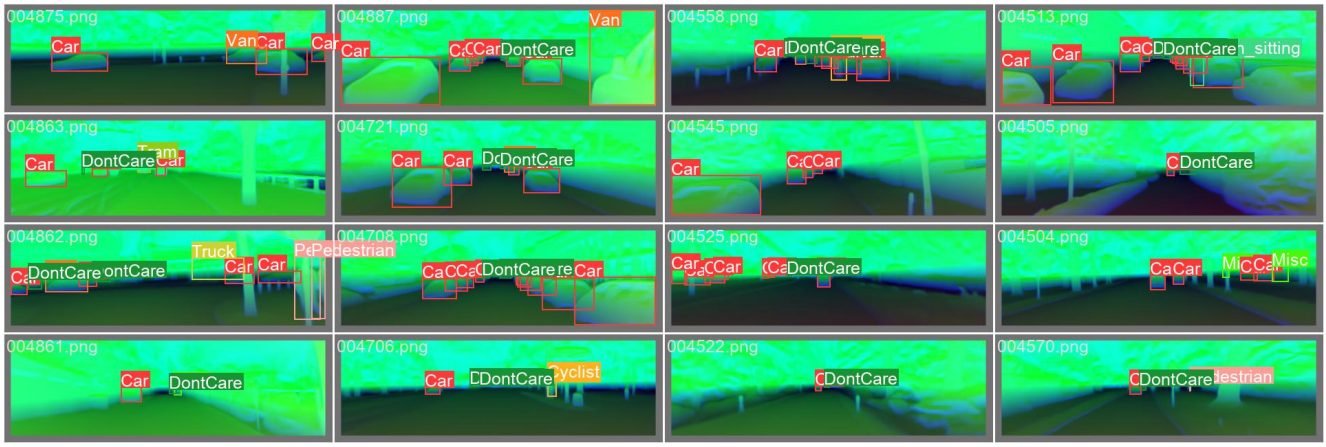

Figure 2. Sample depth maps generated by the UniDepth [12] and converted to the HHA representation, which encodes horizontal disparity, height above ground, and angle with gravity.

Figure 3. Visualization of predicted depth and the pointclouds using UniDepth [12] model. Left is the monocular image, the center is the predicted depth visualized with magma encoding and right is the visualization of pointclouds.

have any target i.e False positives and also all the targets with no proposals i.e False negatives.

$$L = \frac{\sqrt{2}}{\sigma}||d - d*||_1 + \log\sigma \tag{3}$$

4. Experiments and Results

4.1. Training

We evaluated our model on the KITTI 3D object detection dataset, which provides 7,481 images split into training and validation sets. Although test images are also provided, we evaluate and compare our models only on the validation set, as the ground truths for the test split are not available, and a test server is provided instead. Following the work we build upon [11], we use 3,712 images in the training split and 3,769 images in the validation split. We train and evaluate models on three categories of the KITTI dataset: pedestrians, cars, and cyclists. We report 3D detection Average Precision (AP) and Birds-Eye View (BEV) detection at an Intersection over Union (IoU) threshold of 0.7. Similar to other works, we also split our results based on KITTI's default easy, moderate, and hard samples.

Our model was trained for 140 epochs with a learning rate of 0.00125 and a learning rate scheduler that decays the learning rate by 0.1 after 90 and 120 epochs. We also employed weight decay with a weight decay factor of 0.00001 and a warm-up period of 5 epochs. From the beginning of our experiments, we used the Ranger optimizer [21] with AdamW as the base optimizer, which has been shown to work well on multiple benchmarks, although we did not compare it with Adam or SGD in our setting. For most of the training, we used a single A40 GPU and trained models with batch sizes of 32 and 24. We applied several data augmentation techniques, including random flipping, random cropping, translation, and random scaling.

For the region proposal experiments, we used YOLOv8m as our 2D object detection model. This model was trained for 100 epochs with most hyperparameters set to their default values. The use of a pre-trained 2D object detection model allowed us to generate high-quality region proposals, which were then used as input to our 3D object detection pipeline.

4.2. Results

In this section we present the results of the design choices we made, we also share the experiments we conducted which failed and the hypothesis behind these fail-

Method	3D IOU@0.7			BEV@0.7		
	Easy	Mod.	Hard	Easy	Mod.	Hard
CenterNet [25]	0.60	0.66	0.77	3.46	3.31	3.21
MonoGRNet [14]	11.90	7.56	5.76	19.72	12.81	10.15
MonoDIS [17]	11.06	7.60	6.37	18.45	12.58	10.66
M3D-RPN [2]	14.53	11.07	8.65	20.85	15.62	11.88
MonoPair [4]	16.28	12.30	10.42	24.12	18.17	15.76
Ours (RGB)	16.218	14.357	13.55	23.30	20.43	19.08
Ours (HHA)	**28.81**	22.52	19.12	**37.24**	27.01	25.04
Ours (RGB + HHA)	27.32	**23.03**	**19.33**	34.99	**29.31**	**25.42**

Table 1. 3D object detection performance comparison for the Car category on the KITTI validation set. The proposed RGB+HHA model outperforms the RGB-only baseline and several other state-of-the-art monocular 3D object detection methods. The results demonstrate the effectiveness of incorporating pseudo-depth information for improving 3D localization accuracy.

Method	3D IOU@0.5			BEV@0.5		
	E	M	H	E	M	H
RGB	12.48	11.74	10.92	14.26	12.16	11.45
HHA	13.28	12.14	11.50	15.18	12.53	12.34
RGB+HHA	10.91	8.50	7.92	12.01	8.5	8.36

Table 2. 3D object detection performance for the Pedestrian category on the KITTI validation set. The HHA-only model achieves the best results, indicating the importance of depth information for detecting smaller objects like pedestrians. The RGB+HHA model shows slightly lower performance compared to the individual modalities.

Method	3D IOU @0.5			BEV@0.5		
	E	M	H	E	M	H
RGB	8.13	5.57	5.47	8.42	6.26	5.69
HHA	4.14	3.31	2.27	5.03	3.55	3.07
RGB+HHA	10.32	6.63	6.12	10.93	7.29	7.18

Table 3. 3D object detection performance for the Cyclist category on the KITTI validation set. The RGB+HHA model significantly outperforms the individual RGB and HHA models, highlighting the complementary nature of color and depth information for detecting cyclists, which are often more challenging due to their smaller size and complex appearance.

Method	2d AP@0.5		
	Easy	Moderate	Hard
RGB	71.26	47.18	46.95
HHA	38.80	27.54	23.96
RGB+HHA	75.50	49.96	49.32

Table 4. 2D object detection performance for the Cyclist category on the KITTI validation set. The RGB+HHA model achieves the highest Average Precision (AP) scores across all difficulty levels, demonstrating improved 2D localization accuracy by combining color and depth cues.

Method	2d AP@0.5		
	Easy	Moderate	Hard
RGB	96.95	88.10	79.67
HHA	89.86	85.65	77.74
RGB+HHA	90.28	88.70	80.08

Table 5. 2D object detection performance for the Car category on the KITTI validation set. The RGB-only model achieves the best results, suggesting that color information plays a more dominant role in detecting cars compared to depth information. The RGB+HHA model shows comparable performance to the RGB-only baseline.

ures. We began our experimentation with the base model provided by [11]. The model using only RGB, shown in Table 1, presents the results obtained when we attempted to replicate the original findings. Subsequently, we replaced RGB with HHA, and as evident from the quantitative results, this substitution led to significant improvements in both metrics. The task becomes easier for the model when depth information is readily available. Due to the lack of public repositories providing results for the cyclist and pedestrian classes on the validation split, we compare our results solely with the RGB model.

For the cyclist class, we observed that the results using only HHA were inferior compared to the RGB model, as shown in Table 3. To investigate this discrepancy, we examined the 2D detection mean Average Precision (mAP) for the class, presented in Table 4. It is apparent from the table that the detection rate of cyclists in HHA was substantially lower than in RGB. This observation is intuitively reasonable, as cyclists are relatively small compared to cars, making their detection in depth maps more challenging. However, when multiple modalities are employed, the results improve significantly, demonstrating that the model effectively learns to leverage the strengths of both modalities. The RGB modality contributes to better detection rates, while the depth map facilitates more accurate 3D fitting. This synergy ultimately leads to superior overall results in comparison to using RGB alone.

From the quantitative results of 1, it is evident that depth maps help in improving the results significantly, we plotted the results using the lidar data. From 4 it is evident that yellow boxes of RGB+HHA model fit much better to ground truth in comparison to the white boxes.

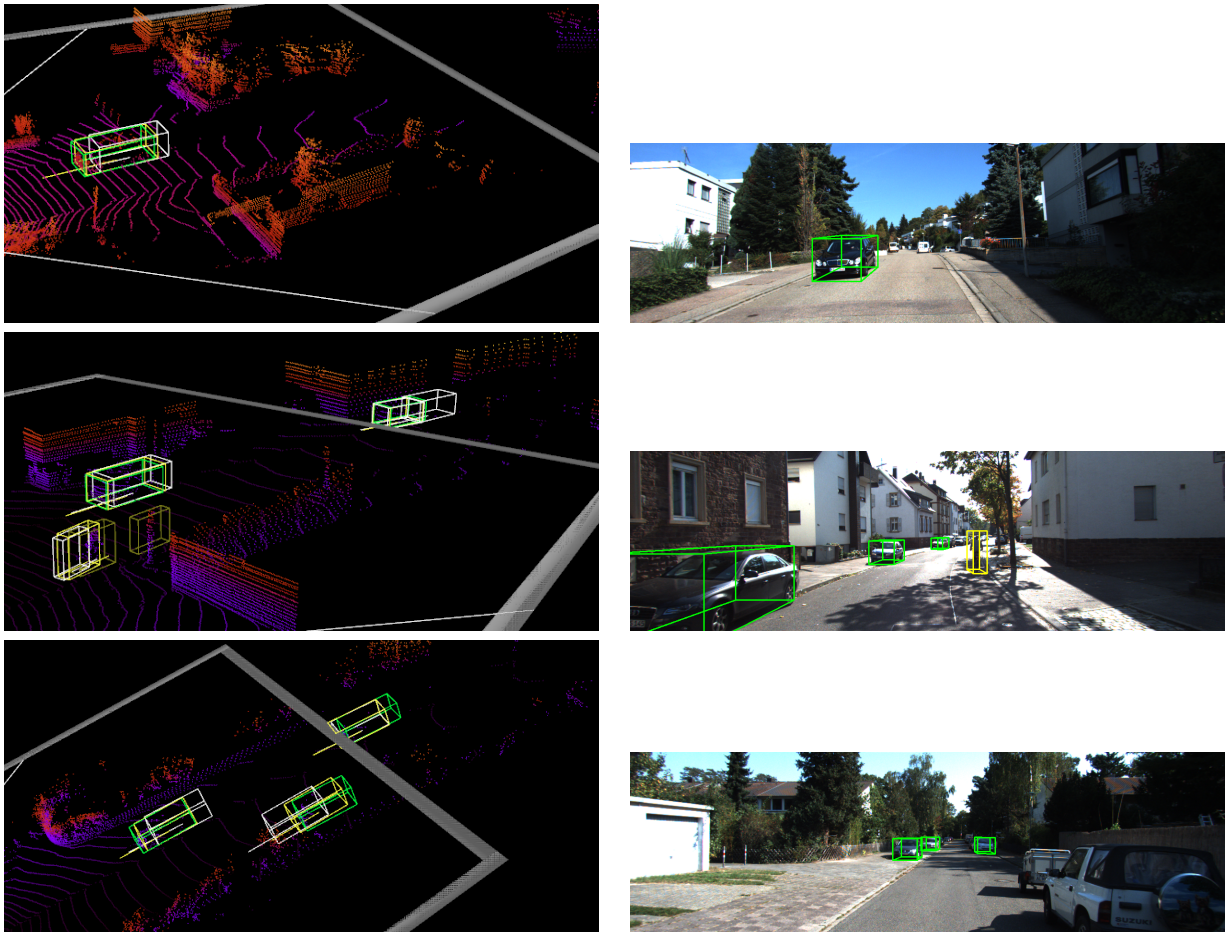

Figure 4. Visualizations of 3D object detection results using point cloud data. The ground truth 3D bounding boxes are shown in green, while the predicted boxes from the RGB-only model are in white and those from the RGB+HHA model are in yellow. The RGB+HHA model demonstrates improved 3D localization accuracy compared to the RGB-only baseline. The darker yellow boxes in second image depict the ground truth for cyclists. (Right) Corresponding detection results from the RGB+HHA model projected onto the RGB image.

4.3. Failed Experiments

We conducted two experiments that did not yield the expected results. The first experiment involved replacing the backbone and neck of the model with a more modern pipeline developed by [22]. Although this task was relatively straightforward, the model's performance was subpar. Due to time constraints and the prioritization of other tasks, we decided to discontinue these experiments. The suboptimal results could be attributed to two major differences in the architecture. Firstly, the backbone of Damo-yolo has a higher downsampling ratio compared to the original DLA-based backbone. While the DLA backbone creates a final feature map with a 1/4 downsampling ratio, the Damo-yolo backbone downsamples the feature map to 1/8. Secondly, the neck of Damo-yolo produces feature maps at multiple resolutions, similar to the neck of DLA, but the CenterNet head is only applied to the final feature map with the highest resolution. As an alternative approach, we experimented with progressively upsampling the feature map at a constant resolution and concatenating these feature maps. However, this approach proved to be suboptimal because the downsampling rate of the lowest resolution feature map was very high, and basic interpolation-based upsampling is not effective in this scenario. Unfortunately, due to time limitations, we were unable to explore additional strategies to address this issue.

The second major experiment that did not succeed was the region proposal network (RPN). Although we developed a complete pipeline for this experiment and trained the model, which demonstrated good performance as noted from the loss curve 5, we encountered difficulties in debugging the detection decoder for the ROI head within the given timeframe. We plan to investigate and resolve this issue as part of our future work, as it has the potential to address the lower overall detection rate of the multi-modal architecture, as evident from Table 5. These results show that the major improvement we get in 3D detection is due to better local-

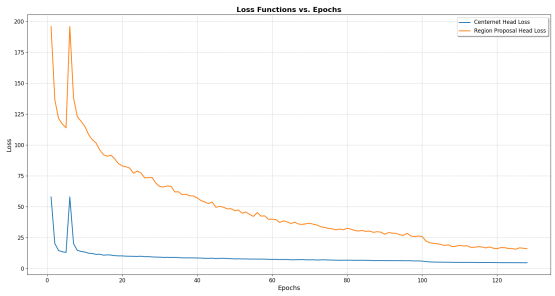

Figure 5. Validation loss curves during training for the two different detection heads in the model.

ization of the boxes, but overall the rate of false negatives is higher in RGB+HHA in comparison to only RGB model. By successfully integrating the RPN into our pipeline, we aim to improve the detection performance and further enhance the effectiveness of our multi-modal approach for 3D object detection.

5. Conclusion

In this work, we explored leveraging recent advancements in monocular depth estimation to improve 3D object detection from monocular images. By utilizing pseudo-depth generated by a state-of-the-art monocular depth estimation model and encoding it in the HHA format, we were able to significantly boost the performance of a CenterNet-based 3D object detection model, especially for the car class. A multi-stream fusion approach combining RGB and HHA depth features proved most effective. We also investigated incorporating a 2D object detection model to generate region proposals for the 3D detection pipeline, though encountered some implementation challenges. Overall, our experiments demonstrate the potential of monocular depth estimation to enable improved 3D object detection by providing useful depth cues from RGB images alone. This can reduce the need for expensive depth sensors and allow leveraging of large existing monocular datasets. Further work is needed to refine the region proposal integration and expand evaluation to additional classes and datasets.

6. Acknowledgement

We would like to acknowledge the professor and TA's of CS7643 for providing an opportunity to build an open ended project. We would also like to thank them and georgia tech for providing computation resources through Google cloud and PACE which were used for running all the experiments conducted in this work.

References

[1] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 1, 2

[2] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9287–9296, 2019. 6

[3] Xiaokang Chen. Depth2HHA-python: Converting depth maps to hha encodings. https://github.com/charlesCXK/Depth2HHA-python, 2018. 4

[4] Yongjian Chen, Lei Tai, Kai Sun, and Mingyang Li. Monopair: Monocular 3d object detection using pairwise spatial relationships. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12093–12102, 2020. 4, 6

[5] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2

[6] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13*, pages 345–360. Springer, 2014. 1, 4

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 4

[8] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLO, Jan. 2023. 4

[9] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[10] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016. 2

[11] Xinzhu Ma, Yinmin Zhang, Dan Xu, Dongzhan Zhou, Shuai Yi, Haojie Li, and Wanli Ouyang. Delving into localization errors for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4721–4730, 2021. 2, 3, 5, 6

[12] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. *arXiv preprint arXiv:2403.18913*, 2024. 1, 2, 4, 5

[13] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2

[14] Zengyi Qin, Jinglu Wang, and Yan Lu. Monogrnet: A geometric reasoning network for monocular 3d object localization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8851–8858, 2019. 6

[15] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2

[16] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 2

[17] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1991–1999, 2019. 2, 6

[18] Kaixuan Wang and Shaojie Shen. Mvdepthnet: Real-time multiview depth estimation neural network. In *2018 International Conference on 3D Vision (3DV)*, pages 248–257, 2018. 2

[19] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 913–922, 2021. 2

[20] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8445–8453, 2019. 2

[21] Less Wright and Nestor Demeure. Ranger21: a synergistic deep learning optimizer. *arXiv preprint arXiv:2106.13731*, 2021. 5

[22] Xianzhe Xu, Yiqi Jiang, Weihua Chen, Yilun Huang, Yuan Zhang, and Xiuyu Sun. Damo-yolo: A report on real-time object detection design. *arXiv preprint arXiv:2211.15444*, 2022. 7

[23] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 2

[24] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2403–2412, 2018. 3

[25] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2, 6