MT3D-Seg: Unified Multi-Task Learning Framework for 3D Object Detection and Drivable Area Segmentation in Smart Mobility

Firas Jendoubi[1], Redouane Khemmar[1], Romain Rossi[1] and Madjid Haddad[2]

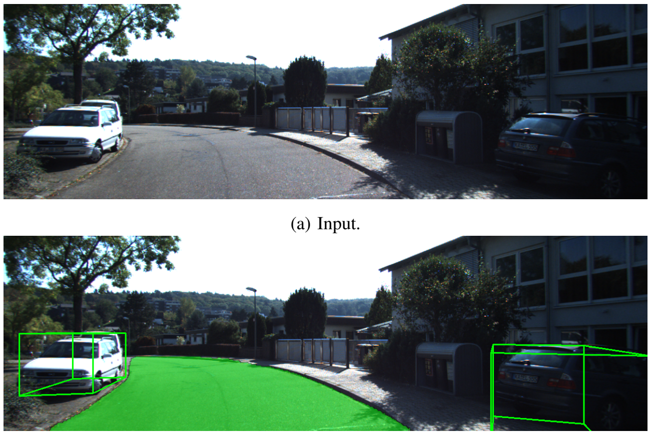
***Abstract*— Robust perception is essential for ensuring the safe operation of autonomous systems in smart mobility. Multi-task learning (M-TL) enhances perception by simultaneously addressing multiple tasks, improving efficiency, and optimizing performance. This paper introduces MT3D-Seg, a novel MT-L framework that integrates 3D object detection and drivable area segmentation, with a primary focus on 3D object detection. Trained on the KITTI dataset, MT3D-Seg leverages shared feature representations to handle complex road scenarios effectively. Experimental results demonstrate high accuracy across both tasks while reducing computational overhead, making it a promising solution for real-time environmental perception in autonomous driving.**

I. INTRODUCTION

Advancements in autonomous driving necessitate efficient real-time perception systems for tasks like object detection, drivable area segmentation, and 3D spatial understanding. Traditionally, these tasks were handled by separate models, increasing computational costs. While 2D detection models like YOLO [1] and Faster R-CNN [2] are fast and accurate, 3D detection models such as PointNet [3] and VoxelNet [4] enhance spatial understanding but remain computationally demanding. Multi-Task Learning (M-TL) improves efficiency by sharing feature extraction across tasks. Models like YOLOP [5] and MultiNet [6] integrate 2D detection and drivable area segmentation, optimizing performance for real-time applications. However, integrating 3D object detection into M-TL frameworks remains a challenge due to its complexity.

This work addresses this gap by introducing an M-TL model that combines 3D object detection and drivable area segmentation in a unified architecture. Using a shared encoder-decoder structure, our approach enhances spatial perception while optimizing computational efficiency, making it suitable for real-time autonomous systems. Figure 1 illustrates our model's input and output transformations.

The paper is structured as follows: Section II reviews related work, Section III details our methodology, Section IV describes training procedures, Section V presents experimental results, and Section VI concludes with findings and future directions.



(a) Input.

(b) Output.

Fig. 1: Illustration of model processing: the input image (a) is transformed by our model to generate 3D object detections and drivable area segmentation (b).

II. RELATED WORK

In this section, we explore deep learning solutions for the two core tasks object detection and drivable area segmentation, and provide an overview of related multi-task learning approaches.

A. Object Detection

2D object detection methods include one-stage models like YOLO and SSD [7], which prioritize speed, and two-stage models like Faster R-CNN [2], which improve accuracy at the cost of inference time. 3D detection can be image-based, as in Mono3D [8], or rely on point clouds, like PointNet [3] and PV-RCNN [9]. Mono3D uses geometric reasoning to infer 3D bounding boxes from monocular images, offering a cost-effective alternative to LiDAR-based approaches.

B. Drivable Area Segmentation

Semantic segmentation models like DeepLab, SegNet [10], and U-Net [11] enable precise pixel-wise predictions crucial for autonomous driving. These architectures leverage atrous convolutions, encoder-decoder designs, and skip connections to enhance segmentation accuracy and efficiency.

C. Multi-Task Learning

M-TL frameworks enhance efficiency by sharing representations across tasks. Mask R-CNN [12] extends Faster R-CNN with instance segmentation, while MultiNet [6] and YOLOP [5] combine detection, segmentation, and lane detection within a unified model. However, most existing

*This work was not supported by any organization

[1]University Rouen Normandy, ESIGELEC, IRSEEM ,76000 Rouen, France firas.jendoubi@groupe-esigelec.org, redouane.khemmar@esigelec.fr, romain.rossi@esigelec.fr

[2]SEGULA Technologies, 92000 Nanterre, France madjid.haddad@segula.fr

M-TL frameworks focus on 2D tasks, highlighting the need for models that integrate 3D detection to improve spatial perception in autonomous systems.

III. Methodology

We propose MT3D-Seg, an efficient feed-forward network for multi-task learning that jointly performs 3D object detection and drivable area segmentation. As illustrated in Figure 2, MT3D-Seg consists of a shared encoder for feature extraction and two task-specific decoders. This design enhances efficiency by leveraging shared representations while optimizing task-specific outputs.

A. Encoder

MT3D-Seg adopts CSPDarknet [13] as the encoder backbone, a lightweight yet powerful architecture used in YOLOv4 [14] and YOLOP [5]. CSPDarknet enhances gradient flow, reduces computational cost, and facilitates efficient feature propagation. Additionally, the encoder integrates Spatial Pyramid Pooling (SPP) [15] and Feature Pyramid Network (FPN) [16] modules. SPP captures multi-scale features, while FPN merges semantic information across layers, improving object detection and segmentation performance.

B. 3D Detection Decoder

The 3D detection head in MT3D-Seg extends the anchor-based YOLO framework, optimized for multi-task learning. Inspired by lightweight detection models [17], it employs an FPN [16] and a Path Aggregation Network (PAN) [18] to enhance multi-scale feature extraction. The detection head predicts: 2D bounding boxes to locate objects in the image plane. 3D bounding box parameters, including object center (x, y, z), dimensions (width, height, length), orientation (yaw angle), and depth estimation (distance from the camera). These predictions enable robust 3D scene understanding, facilitating obstacle detection for autonomous navigation.

C. Segmentation Decoder

The segmentation decoder in MT3D-Seg follows a lightweight design inspired by YOLOP. The bottom layer of the FPN feeds into the segmentation branch, producing feature maps of size $(W/8, H/8, 256)$. We apply three upsampling steps using nearest interpolation to restore the output to $(W, H, 2)$, where each pixel represents the probability of being part of the drivable area or background. Unlike traditional Fully Convolutional Networks (FCNs) [19], MT3D-Seg integrates multi-scale features efficiently, improving segmentation in complex road scenes. The simplified decoder structure ensures high-precision predictions while maintaining real-time inference speeds, making it well-suited for autonomous driving applications.

IV. M-TL Development & Training Details

A. M-TL Loss Function

The loss function of MT3D-Seg is a critical component that drives the optimization process for both 3D object detection and drivable area segmentation. We define the global loss L_{global} as a weighted sum of the individual losses associated with these two tasks, as in equation 1:

$$L_{\text{global}} = \beta_1 L_{\text{3D}} + \beta_2 L_{\text{segmentation}} \tag{1}$$

where β_1 and β_2 are coefficients that adjust the relative importance of each task in the overall optimization. The 3D detection loss L_{3D} encompasses several components: the loss for the center of the bounding box L_{center}, the dimensions of the bounding box L_{size}, the distance to the ground truth L_{distance}, the orientation of the bounding box $L_{\text{orientation}}$, and the classification loss L_{class}. This can be expressed as in equation 2:

$$L_{\text{3D}} = \alpha_1 L_{\text{center}} + \alpha_2 L_{\text{size}} + \alpha_3 L_{\text{distance}} + \alpha_4 L_{\text{orientation}} + \alpha_5 L_{\text{class}} \tag{2}$$

Each component is assigned a coefficient α_i to ensure a balanced contribution to the total loss, facilitating effective learning across tasks. The segmentation loss $L_{\text{segmentation}}$ employs Cross Entropy Loss with Logits, represented as in equation 3:

$$L_{\text{segmentation}} = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \right] \tag{3}$$

Where N is the number of pixels, y_i is the ground truth label, and p_i is the predicted probability for each pixel. By combining these loss functions, the global loss effectively guides the training process, enhancing the accuracy of both object detection and segmentation, and promoting a more integrated understanding of complex scenes, making it particularly suitable for applications in smart mobility.

B. KITTI Dataset

We trained MT3D-Seg on the KITTI dataset [20], a widely used benchmark for autonomous driving. KITTI provides 3D object detection and drivable area annotations, enabling precise localization and navigation. However, it lacks lane segmentation annotations, unlike datasets such as BDD100K [21]. This limitation prevents us from incorporating lane detection, emphasizing the need for more comprehensive datasets for integrated autonomous driving solutions.

C. Implicit Function Theorem for Backpropagation

We incorporate the Implicit Function Theorem (IFT) into backpropagation, following the MinBackProp approach [22], to enhance stability and efficiency. Unlike gradient surgery [23] or weight balancing [24], which require manual tuning, IFT resolves gradient conflicts implicitly, optimizing without extra hyperparameters. IFT reformulates gradient computation as an implicit function, preventing numerical instability in multi-task learning. In traditional backpropagation, direct gradient computation can be unstable, especially in complex settings. By applying IFT, we compute stable gradients efficiently, as shown in Equation 4:

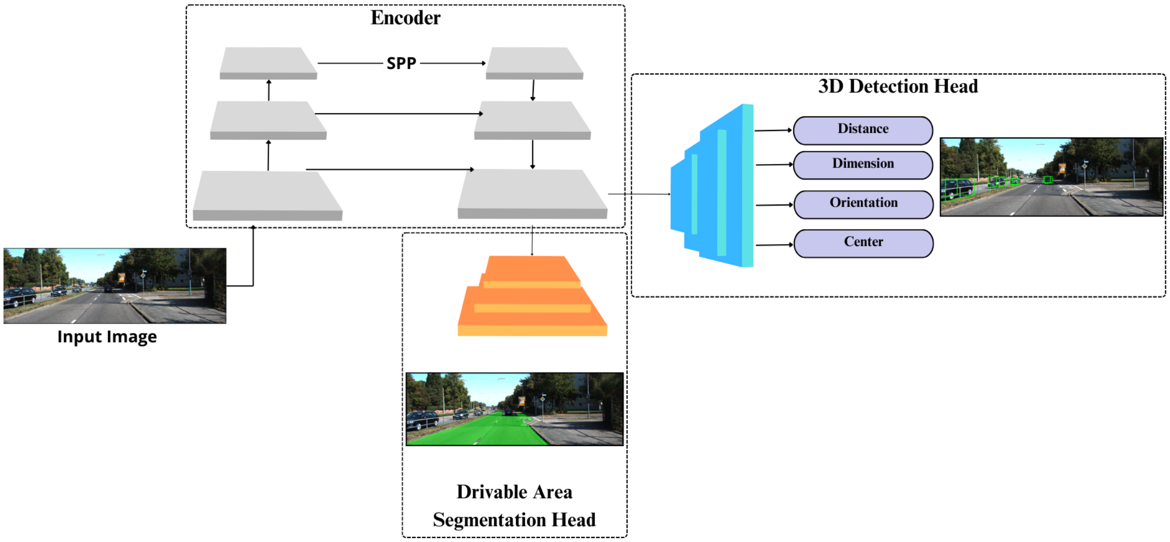


Fig. 2: Architecture of MT3D-Seg with a shared encoder and two decoders for 3D object detection and drivable area segmentation.

$$\frac{dL}{d\theta} = -\left(\frac{\partial f}{\partial y}\right)^{-1} \frac{\partial f}{\partial x}\frac{dL}{dx}, \tag{4}$$

Where $f(x, y) = 0$ satisfies IFT conditions. This implicit differentiation enables robust gradient computation with reduced overhead.

Algorithm 1 details the IFT-based backpropagation process, ensuring stable parameter updates:

Algorithm 1 IFT-based Backpropagation

```
Require: Parameters $\theta$, loss $L(\theta)$, function $f(x, y) = 0$
for each iteration do
    Compute $L(\theta)$, $\frac{\partial L}{\partial x}$, $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial y}$
    Apply IFT: $\frac{dL}{d\theta} \leftarrow -\left(\frac{\partial f}{\partial y}\right)^{-1} \frac{\partial f}{\partial x}\frac{\partial L}{\partial x}$
    Update $\theta$ with $\frac{dL}{d\theta}$
end for
return Optimized $\theta$
```

By leveraging IFT, MinBackProp achieved 100% stability and a 10× speedup. Similarly, our model benefits from improved convergence, reduced computation time, and enhanced robustness, making it highly suitable for real-time autonomous perception tasks.

V. Experimental Results and Analysis

This section presents the evaluation of our M-TL model for 3D object detection and drivable area segmentation, highlighting its robustness and efficiency for autonomous driving. The model was trained with a learning rate of 10^{-4} using an NVIDIA A100 GPU with 48 GB of memory.

A. Traffic 3D Object Detection

For the results section of our 3D object detection, we provide both qualitative and quantitative evaluations to demonstrate the effectiveness of our approach. As shown in Figure 3, the qualitative results indicate that our model accurately detects 3D objects in complex traffic scenes, producing clear and reliable bounding boxes.

Our quantitative evaluation, summarized in Table I, assesses 3D object detection performance using Recall and Mean Average Precision (mAP) at IoU 0.7. These metrics enable comparison with single-task models and provide insight into our model's ability to detect 3D objects in complex scenes. Class detection is evaluated, as accurate 3D bounding box regression relies on precise region proposals and object classification. We also evaluate key components of our 3D pipeline:

- **Dimension Prediction**: We assess dimension estimation using the Dimension Score (DS) as defined in [27]. The DS is computed as equation 5:

$$DS = \min\left(\frac{V_{pd}}{V_{gt}}, \frac{V_{gt}}{V_{pd}}\right), \tag{5}$$

Where V_{pd} and V_{gt} represent the predicted and ground truth object volumes, respectively.

- **Principal Box Estimation**: The accuracy of the predicted 3D bounding box center is measured using the Center Score (CS) from [27], which accounts for projected center coordinates and bounding box dimensions. Given x and y as the projected center coordinates in pixels, and w and h as the width and height of the 2D bounding box, the CS is defined as in equation 6:

$$CS = \frac{2 + \cos\left(\frac{x_{gt} - x_{pd}}{w_{pd}}\right) + \cos\left(\frac{y_{gt} - y_{pd}}{h_{pd}}\right)}{4}. \tag{6}$$

- **Orientation Evaluation**: Following the KITTI benchmark, we assess orientation accuracy using the Orientation Score (OS), defined as equation 7:

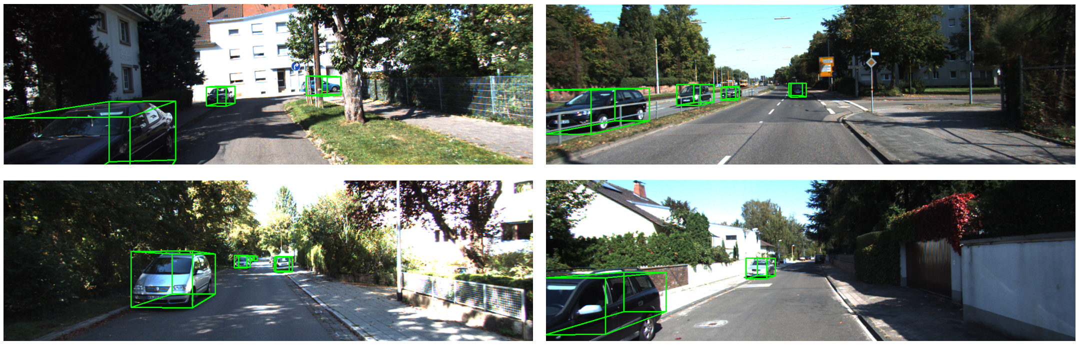


Fig. 3: Visualization of the traffic 3D Object detection results of our model.

TABLE I: Comparison of traffic 3D detection models. The values show recall, mAP at IoU 0.7, DS, CS, OS, and processing speed (FPS).

Network (Type)	Recall (%)	mAP70 (%)	DS	CS	OS	Speed (FPS)
Mono3D (No M-TL)	8.6	15.0	-	-	-	0.5
PointNet (No M-TL)	11.1	72.5	-	-	-	3.1
VoxelNet (No M-TL)	41.0	83.2	-	-	-	10.3
Complex-YOLO (No M-TL) [25]	85.1	63.6	-	-	-	50.0
BirdNet+ (No M-TL) [26]	86.7	51.4	-	-	-	10.0
Lightweight 3D (No M-TL) [17]	-	-	0.88	0.96	0.92	-
Joint Monocular 3D (No M-TL) [27]	-	-	0.962	0.918	0.974	-
MT3D-Seg (M-TL)	84.2	77.5	0.98	0.971	0.972	48.1

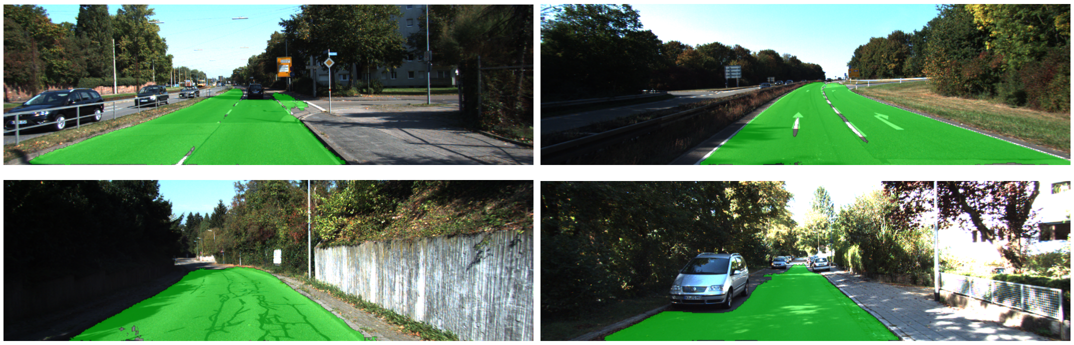


Fig. 4: Visualization of the drivable area segmentation results of our model.

$$OS = \frac{1 + \cos(\alpha_{gt} - \alpha_{pd})}{2}. \tag{7}$$

Given that no existing multi-task learning model integrates 3D object detection as a primary task, we compare our results to state-of-the-art single-task models focused solely on 3D object detection to establish a performance benchmark. By incorporating these diverse evaluation criteria, we provide a comprehensive analysis of our model's performance, highlighting its strengths in both accuracy and computational efficiency. While our results are competitive with models like VoxelNet, BirdNet+ [26], and PointNet, our approach stands out for its efficiency. Unlike these models, which require high-dimensional processing and intensive LiDAR data treatment, our model offers a more balanced trade-off between performance and speed, making it well-suited for real-time applications.

B. Drivable Area Segmentation

The qualitative results, illustrated in Figure 4, emphasize our model's effectiveness in segmenting drivable areas across diverse scenes, showcasing consistent clarity and accuracy in delineating navigable regions. These visual examples affirm the model's adaptability to complex, real-world scenarios and highlight its precision in segmentation, which is critical for autonomous driving applications.

Table II presents the quantitative performance of MT3D-Seg, benchmarked against MultiNet, DLTNet, PSPNet [28], and YOLOP. Using Intersection over Union (IoU) to assess segmentation accuracy, our model achieves competitive results. While YOLOP attains a slightly higher IoU, our model offers a notable advantage in processing speed due to its two-task setup versus YOLOP's three-task structure. This efficiency makes it well-suited for real-time applications, ensuring fast and accurate scene analysis. These results highlight the model's effectiveness for autonomous driving perception.

TABLE II: Quantitative comparison of drivable area segmentation with other models, showing mean Intersection over Union (mIoU) and speed in frames per second (FPS).

Network (Type)	mIoU (%)	Speed (FPS)
MultiNet (M-TL)	71.6	8.6
DLTNet (M-TL)	71.3	9.3
PSPNet (M-TL)	89.6	11.1
YOLOP (M-TL)	91.5	41.0
MT3D-Seg (M-TL)	89.8	48.1

VI. Conclusion & Perspectives

This paper introduces a network architecture MT3D-Seg that integrates 3D object detection and drivable area segmentation within a multi-task learning framework, enhancing computational efficiency and scene understanding for autonomous driving. Trained and evaluated on the KITTI dataset, the model demonstrates robust performance across both tasks. While effective, its current scope is limited to 3D object detection and drivable area segmentation. Future work will extend its capabilities to include tasks such as lane detection, real-time tracking, and environmental classification, while further optimizations will refine its performance to meet or exceed state-of-the-art benchmarks in autonomous driving and smart mobility applications.

References

[1] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of yolo algorithm developments," *Procedia computer science*, vol. 199, pp. 1066–1073, 2022.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[3] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[4] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4490–4499.

[5] D. Wu, M.-W. Liao, W.-T. Zhang, X.-G. Wang, X. Bai, W.-Q. Cheng, and W.-Y. Liu, "Yolop: You only look once for panoptic driving perception," *Machine Intelligence Research*, vol. 19, no. 6, pp. 550–562, 2022.

[6] M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun, "Multinet: Real-time joint semantic reasoning for autonomous driving," in *2018 IEEE intelligent vehicles symposium (IV)*. IEEE, 2018, pp. 1013–1020.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. 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*. Springer, 2016, pp. 21–37.

[8] T. He and S. Soatto, "Mono3d++: Monocular 3d vehicle detection with two-scale 3d hypotheses and task priors," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 8409–8416.

[9] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 529–10 538.

[10] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[13] M. Mahasin and I. A. Dewi, "Comparison of cspdarknet53, cspresnext-50, and efficientnet-b0 backbones on yolo v4 as object detector," *International journal of engineering, science and information technology*, vol. 2, no. 3, pp. 64–72, 2022.

[14] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[16] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[17] A. Mauri, R. Khemmar, B. Decoux, M. Haddad, and R. Boutteau, "Lightweight convolutional neural network for real-time 3d object detection in road and railway environments," *Journal of Real-Time Image Processing*, vol. 19, no. 3, pp. 499–516, 2022.

[18] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8759–8768.

[19] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[20] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[21] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2636–2645.

[22] D. Sungatullina and T. Pajdla, "Minbackprop–backpropagating through minimal solvers," *arXiv preprint arXiv:2404.17993*, 2024.

[23] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient surgery for multi-task learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5824–5836, 2020.

[24] S. Liu, Y. Liang, and A. Gitter, "Loss-balanced task weighting to reduce negative transfer in multi-task learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 9977–9978.

[25] M. Simony, S. Milzy, K. Amendey, and H.-M. Gross, "Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds," in *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018, pp. 0–0.

[26] A. Barrera, C. Guindel, J. Beltrán, and F. García, "Birdnet+: End-to-end 3d object detection in lidar bird's eye view," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–6.

[27] H.-N. Hu, Q.-Z. Cai, D. Wang, J. Lin, M. Sun, P. Krahenbuhl, T. Darrell, and F. Yu, "Joint monocular 3d vehicle detection and tracking," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5390–5399.

[28] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.