EN.601.661: Computer Vision Project Proposal

Title: Multi-Task Perception Model for Autonomous Vehicles
Simultaneous Object Detection, Semantic Segmentation, and Depth Estimation

Spring 2025
March 5th, 2025

Team Members

- **Minh Lê** (*JHED ID: mle20*)
- **Yi Lu** (*JHED ID: ylu174*)
- **Ritwik Rohan** (*JHED ID: rrohan2*)
- **Yuhan Yong** (*JHED ID: yyong2*)

Problem Area

Autonomous driving technology has become more and more popular nowadays as one of the most important applications of computer vision. In reality, it needs to process multiple tasks at the same time to operate safely and efficiently. These tasks include object detection, understanding the road (semantic segmentation), and depth estimation, which might cause serious accidents if they do not occur quickly at the real-time. However, running each of these tasks separately can result in low processing speeds, often reaching only 5-10 frames per second (FPS), which is insufficient for real-time decision-making and can lead to hazardous delays.
To solve this, we propose a multi-task model that uses a shared encoder to process all tasks together. Our inspiration comes from [7]**Tesla's Hydranet**, where a single neural network processes multiple perception tasks simultaneously, enabling more efficient and coordinated decision-making. This approach will help the model work faster, reduce delays, and remain stable, making autonomous systems safer and more reliable, thus reducing inference time.

Context

In this hot area, numerous researchers have made significant contribution to the current autonumous driving technology, in terms of multi tasks of semantic segmentation, depth estimation. The key up-to-date contributions of the researchers include: adaptation of a single model for multiple tasks, realization of real-time measurement, and become robust with datasets. Specifically, researchers utilized Backbone network to complete the two tasks separately.
(1) **Semantic Segmentation**: It is a task of per-pixel label classification. Researchers proposed Light-Weight [1]**RefineNet** on top of [2]**MobileNet-v2** as the baseline architecture. (2) **Depth Estimation**: This task aims to find the distance of each pixel from the observer. Researchers use [1]**knowledge distillation approach** that is based on the idea of having a large pre-trained teacher (expert) network (or an ensemble of networks), and using its logits, or predictions directly, as a guiding signal for a small network along with original labels. [1]Nekrasov et al. (2019) demonstrated this approach on NYUDv2-40 and KITTI, achieving **13ms** inference time with **6.5 GFLOPs** on 640 × 480 images, making it suitable for real-time use.

Datasets

To tackle the proposed problem, we will use data from the KITTI Vision Benchmark Suite of the Karlsruhe Institute of Technology and Toyota Technical Institute as a starting point to add object detection to the model, considering that the foundation code uses KITTI for segmentation and depth estimation as their dataset. KITTI is widely used for autonomous driving research and includes datasets for **RGB images, depth maps, and annotated object** labels for object detection, semantic segmentation, and depth estimation. The KITTI Object Detection dataset, part of the larger KITTI dataset) contains **14,999 images with 9 different classes** (i.e., car, pedestrian, cyclist, etc.). In addition, KITTI's small-scale dataset allows for faster run time when testing the decoder.

Computational Resources

To train our multi-task autonomous driving model, which integrates semantic segmentation, depth estimation, and object detection, the minimum computational requirement is 16GB VRAM.

Component	**Estimated GPU Memory Usage**
MobileNetV2 Encoder	∼1GB
Semantic Segmentation Decoder (RefineNet)	3–5GB
Depth Estimation Decoder (RefineNet)	3–5GB
Object Detection Decoder (YOLO/SSD/PointPillars)	4–6GB

Proposed Solution

Overview

We propose a multi-task perception model that simultaneously performs semantic segmentation, depth estimation, and object detection using a shared encoder-decoder architecture. A [2]**MobileNetV2** encoder will extract features from monocular RGB images, reducing redundancy and improving computational efficiency. Task-specific decoders include [1]**RefineNet for segmentation and depth estimation**, and three object detection models (**[4]YOLO, [5]SSD, and [6]PointPillars**) for comparative analysis. This approach ensures real-time performance while optimizing accuracy for autonomous driving.

Model Architecture

The model consists of a lightweight **MobileNetV2** encoder followed by three decoders. The **semantic segmentation** decoder, based on RefineNet, predicts pixel-wise class labels to distinguish road elements. The **depth estimation** decoder, also using RefineNet, predicts relative distances for spatial awareness. For **object detection**, we compare three architectures: **YOLO** for single-shot detection with high speed, **SSD** for multi-scale anchor-based detection, and **PointPillars**, which processes LiDAR data for enhanced 3D object detection. This comparative study allows us to evaluate detection accuracy versus computational cost in a multi-task setting. The model takes a single monocular RGB image, preprocessed via normalization and resizing. The output includes: (1) a **segmentation map** with per-pixel classifications, (2) a **depth map** estimating spatial distance, and (3) **bounding boxes** with class labels and confidence scores for detected objects.

Training Strategy, Loss Functions and Evaluation Metrics

Instead of training separate models, we employ a shared encoder with distinct loss functions to optimize each task. For **semantic segmentation**, we use a weighted **Cross-Entropy Loss** with class balancing. For **depth estimation**, we adopt the **Inverse Huber Loss (BerHu)**, which is robust to outliers. For **object detection**, loss functions vary per model: **YOLO** combines localization, classification, and confidence losses, **SSD** uses Smooth L1 loss for bounding boxes and Cross-Entropy for classification, while **PointPillars** employs focal loss to handle class imbalance. We plan to use [3]**weighted loss functions**. For evaluation metrics, we plan to evaluate segmentation using **mean Intersection over Union (mIoU)**, depth estimation with **Root Mean Squared Error (RMSE)** and **Absolute Relative Error (Abs Rel)**, and object detection using **mean Average Precision (mAP)**. Additionally, **Frames per Second (FPS)** will measure real-time feasibility.

Challenges and Considerations

Key challenges include **dataset imbalance**, as different datasets provide asymmetric annotations. We address this using [1]**knowledge distillation**, generating synthetic labels via a teacher network to ensure consistent training. Another issue is **hyperparameter tuning**, as all decoders will require different learning rates, anchor configurations, and optimization strategies, necessitating extensive experimentation. We plan to use techniques like LRscheduler, LRFinder to handle this. One more issue is having different **loss functions** for all decoders based on application. For this we plan to use a weighted loss function for which the weights will acts as a hyperparameter. Lastly, Training and inference require significant memory and processing power, which can limit real-time performance, especially on resource-constrained hardware. **Computational constraints** can be mitigated by model pruning to balance memory efficiency and inference speed.

References

[1] V. Nekrasov, T. Dharmasiri, A. Spek, T. Drummond, C. Shen, and I. Reid, *Real-Time Joint Semantic Segmentation and Depth Estimation Using Asymmetric Annotations*. arXiv preprint arXiv:1809.04766, 2019. Available at: `https://arxiv.org/pdf/1809.04766`

[2] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. Google Inc., arXiv preprint arXiv:1801.04381, 2018. Available at: `https://arxiv.org/pdf/1801.04381`

[3] A. Kendall, Y. Gal, and R. Cipolla, *Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics*. University of Cambridge and University of Oxford, arXiv preprint arXiv:1705.07115, 2018. Available at: `https://arxiv.org/pdf/1705.07115`

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, *YOLO: Unified, Real-Time Object Detection*. arXiv preprint arXiv:1506.02640, 2015. Available at: `https://arxiv.org/pdf/1506.02640`

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, *SSD: Single Shot MultiBox Detector*. arXiv preprint arXiv:1512.02325, 2016. Available at: `https://arxiv.org/pdf/1512.02325`

[6] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, *PointPillars: Fast Encoders for Object Detection from Point Clouds*. arXiv preprint arXiv:1812.05784, 2018. Available at: `https://arxiv.org/pdf/1812.05784`

[7] R. T. Mullapudi, I. Radosavovic, and D. Ramanan, *HydraNets: Specialized Dynamic Architectures for Multi-Task Learning*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018. Available at: `https://openaccess.thecvf.com/content_cvpr_2018/papers/Mullapudi_HydraNets_Specialized_Dynamic_CVPR_2018_paper.pdf`