TinyML-Based Adaptive Speed Control for Car Robot: A Comparative Approach

Student: Alexandru Petriceanu[1] | **Responsible Professor**: Qing Wang[1]
[1]EEMCS, Delft University of Technology, Netherlands

TU Delft

1. Background

- **Adaptive speed control** heavily relies on how a navigator perceives their surroundings. Previously relied on ground truth systems like:
  - Stereo vision [1]
  - LiDAR [2]
  - Radar [3]
- **Problems**: Computationally and energy inefficient**.** Infeasible for large-scale deployment.
- **Solution:** ML-powered depth estimation model.
  - **Use a U-Net** or **pyramid architecture model** to predict depths from images, using monocular vision [4, 5].
  - **Computationally cheaper,** better fit for everyday use.
- **State-of-the-art models** need GPUs and extensive resources to function [6, 7].
- Therefore, use **TinyML**, a subset of small models that run on cheap microcontrollers.
- For testing, use the Raspberry Pi Pico, which has only **264 KB of RAM** and **2 MB of flash memory**. Also uses the RP2040 chip, which has **a dual core Arm Cortex-M0+** processor.

2. Research Question

What is the post-compression efficiency of TinyML depth perception models when run on the Raspberry Pi Pico?

- **Subquestions:**
  - What **other literature** is there on TinyML depth estimation?
  - Is running the **monocular depth estimation** task on the Raspberry Pi Pico **feasible**?
  - What effects do **compression** techniques such as **quantization** and **pruning** have on depth perception models?
  - What are representative **metrics** for **efficiency**?

Figure 1: Raspberry Pi Pico

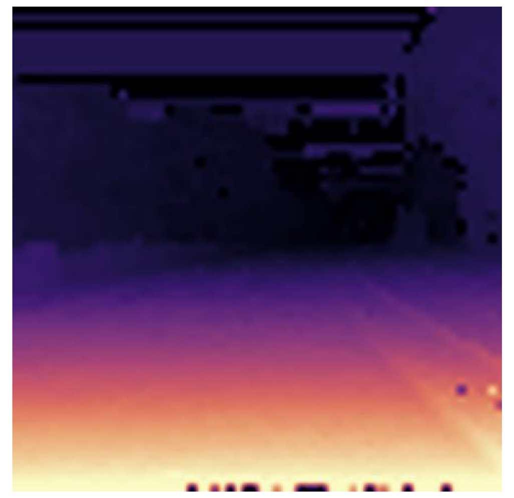
3. Methodology

Figure 2.1: Before Preprocessing

- **Three models** were selected and compared:**L-EfficientUNet** [4] , **L-Enet** [8], **μPyD-Net** [5]
- One more original model was added to evaluate the efficiency of LSTMs: **Temporal-μPyD-Net**
- Models were trained and tested on the **Eigen split** of the **KITTI dataset** [9, 10]. This dataset contains over 60 recordings from stereo vision cameras and sparse LiDAR depth maps.

***Figure 2.2**: Post-Preprocessing*

***Figure 2.3**: Semi Global Matching (SGM)*



- The supervision label was the **SGM (Semi Global Matching) disparity map** obtained from each stereo image pair [11].
- Grayscale images from the left camera were cropped and resized to 32x32 pixels to be used as input.
- Models were trained using berHu (reverse Huber) loss and an Adam optimizer for 100 epochs.
- **Evaluation** was performed using threshold accuracy, inference time, and memory used.
- The **top two best-performing** models were run for inference on the Raspberry Pi Pico.

4. Results

Model	δ < 1.25	δ < 1.25²	δ < 1.25³
L-EfficientUNet	54.40%	70.08%	80.77%
L-ENet	55.88%	73.24%	83.35%
μPyD-Net	74.32%	83.95%	88.44%
Temporal-μPyD-Net	74.38%	83.68%	88.40%

***Table 1:** Accuracies for 64x64 resolution*

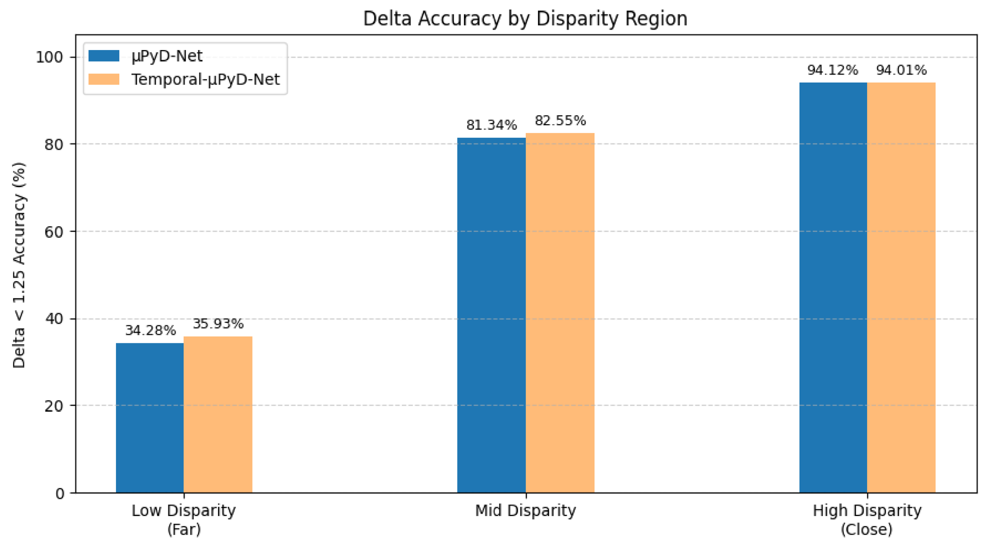


***Figure 3:** μPyD-Net vs Temporal- μPyD-Net on 32x32 resolution*

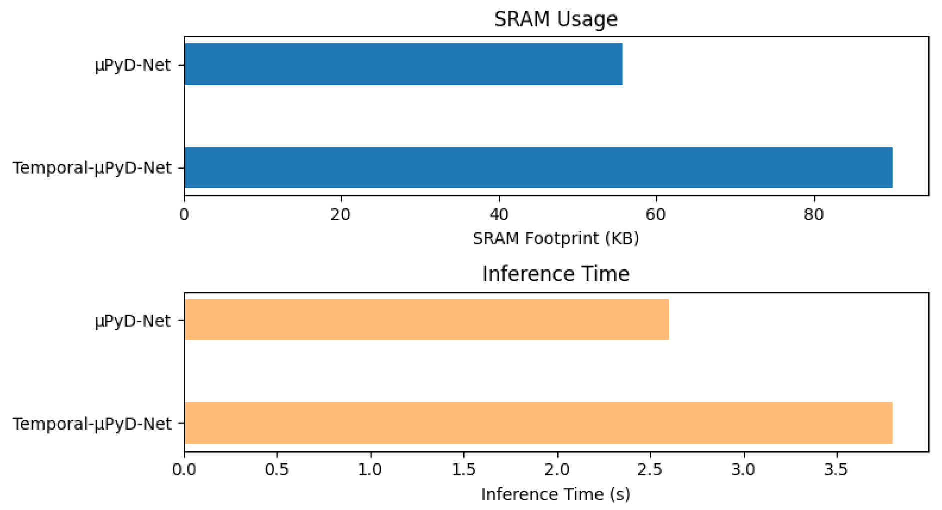


***Figure 4:** μPyD-Net vs Temporal- μPyD-Net SRAM and Inference Time (32x32)*

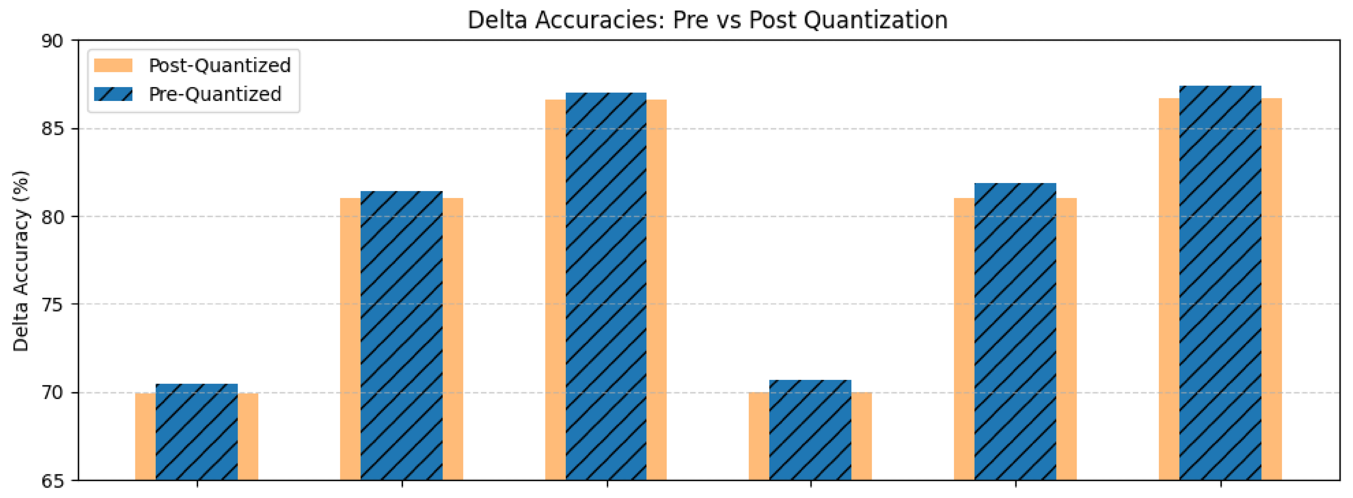


***Figure 5:** Pre- vs Post-Quantization Accuracies*

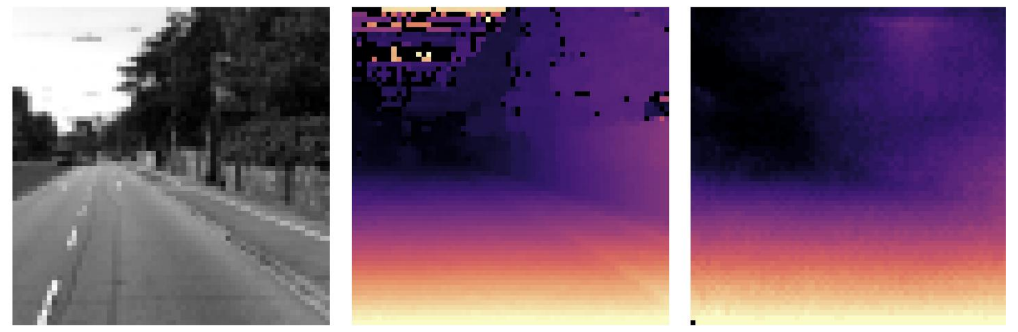


***Figure 6:** μPyD-Net Prediction*

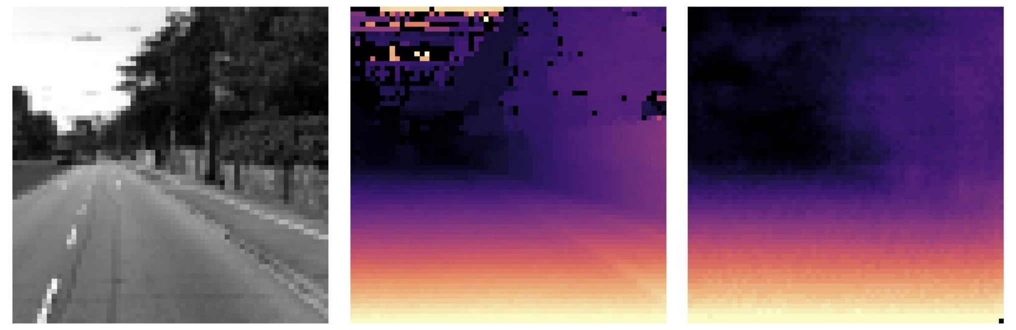


***Figure 7:** Temporal-μPyD-Net Prediction*

***Figure 8:** μPyD-Net Prediction*

***Figure 9:** Temporal-μPyD-Net Prediction*

***Figure 10:** μPyD-Net Prediction (Overlayed)*

***Figure 11:** Temporal-μPyD-Net Prediction (Overlayed)*

5. Conclusion and Future Work

- Results show that running the depth perception task **is feasible** on the Raspberry Pi Pico. For **practical applications**, however, either more work on **fine-tuning inference time** or using **a better board** is recommended.
- Moreover, by measuring accuracies **pre-quantization** versus **post-quantization**, we can tell that full INT8 quantization does not affect accuracy in any meaningful way.
- Finally, we can conclude that **disparity maps** produced by the **SGM algorithm** are good supervision labels [11] and **berHu loss** facilitates good training results by not getting stuck in local minima and being a popular choice when it comes to depth estimation [5].

Citations:
[1] Zaarane, A., Slimani, I., Al Okaishi, W., Atouf, I., & Hamdoun, A. (2020). Distance measurement system for autonomous vehicles using stereo camera. **[2]** Li, Y., & Ibanez-Guzman, J. (2020). Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. **[3]** Sun, S., Petropulu, A. P., & Poor, H. V. (2020). MIMO radar for advanced driver-assistance systems and autonomous driving: Advantages and challenges. **[4]** Liu, X., Chen, J., Zhou, Y., Zhou, Y., Wang, J., Ou, X., & Lei, H. (2023, July). L-EfficientUNet: Lightweight End-to-End Monocular Depth Estimation for Mobile Robots. **[5]** Peluso, V., Cipolletta, A., Calimera, A., Poggi, M., Tosi, F., Aleotti, F., & Mattoccia, S. (2021). Monocular depth perception on microcontrollers for edge applications. *IEEE Transactions on Circuits and Systems for Video Technology*, *32*(3), 1524-1536. **[6]** Godard, C., Mac Aodha, O., Firman, M., & Brostow, G. J. (2019). Digging into self-supervised monocular depth estimation. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 3828-3838). **[7]** Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., & Zhao, H. (2024). Depth anything v2. Advances in Neural Information Processing Systems, 37, 21875-21911. **[8]** Kim, S., Nam, J., & Ko, B. (2019). Fast Depth Estimation in a Single Image Using Lightweight Efficient Neural Network. *Sensors* **[9]** Eigen, D., Puhrsch, C., & Fergus, R. (2014). Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, *27*.
[10] Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *The international journal of robotics research*, *32*(11), 1231-1237. **[11]** Hirschmüller, Heiko. (2005). Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2. 807-814. 10.1109/CVPR.2005.56.