TALOS: Democratizing Safe Micro-Mobility via Energy-Efficient Edge AI

Kevin Davidson (Student) and Nathan Greenfield (Faculty Advisor)
Computer Science & Information Systems Department
Santa Monica College
Santa Monica, CA, USA
kevin@kevindavidson.work, greenfield_nathan@smc.edu

I. INTRODUCTION

Although municipalities strive for zero-emission transportation networks, rider safety concerns [1] hinder mass bicycle adoption. Cornerstone infrastructure projects are often delayed, cancelled, or insufficient. In line with UN SDG 11.2, we require rapid technological action to make carbon-neutral mobility a viable option for the general public.

Current bike safety technology has a significant flaw. Radar sensors lack semantic distinction, failing to differentiate between a sprinting cyclist and a speeding sedan. Even market-leading consumer systems, such as the Garmin Varia, which rely on the Doppler Effect, suffer from match-speed blindness, often completely missing aggressive tailgaters. Cyclists cannot respond best to immediate threats without actionable intelligence. Camera alternatives are either cost-prohibitive or are too slow for real-time safety. By the time the system warns a rider, they may not have enough time to react. To solve these problems, we developed TALOS, a bicycle blind-spot monitor running on a Raspberry Pi 5 with a Hailo-8L NPU.

II. DATASET ENGINEERING

To minimize false negatives, we had to overcome bias in standard datasets. COCO, which powers YOLOv8, is not well-suited for TALOS because it is biased towards the "person" class (~64,000 images), while neglecting "bicycles" (~3,000 images) [2]. COCO also contains 75 redundant classes, such as "giraffe" and "toothbrush". Even specialized automotive datasets were also not ideal because they underrepresent motorcycles, trucks, and buses. To fix this, we built a custom dataset using oversampling to force the model to learn the "motorcycle" class. Each photo with a motorcycle was duplicated four times, which raised class recall by 41%. Furthermore, trucks and buses are functionally and visually similar enough to warrant class-merging as "heavy_vehicle", reducing false negatives for buses by 35%.

III. INFERENCE OPTIMIZATION

Eco-friendly technology requires energy efficiency. Advanced Driver Assistance Systems typically rely on power-guzzling GPUs that are incompatible with small, modular, and battery-powered systems. However, running YOLOv8n on the CPU was inefficient and dangerous for real-world use because it saturated the 5V/3A power envelope yet only produced 3 frames per second. By offloading inference to the Hailo-8L NPU, we solved both the latency and power problems, increasing speed to 80 FPS and efficiency from 0.2 to 10 frames per watt. Software optimization played a major role in this increase. Standard YOLOv8 uses 32-bit floating-point weights [3], also known as FP32. We used the Hailo Dataflow Compiler to quantize the model from FP32 to Int8, or 8-bit integers, thereby reducing the continuous values of the neural network to a discrete grid. To reduce accuracy losses typically resulting from quantization, we used a calibration dataset with size n=64. Our final output is a Hailo Executable Format file, which reduced the model size by 75% in comparison to the default PyTorch (.pt) file format.

IV. MECHANICAL DESIGN

Camera stability is crucial for reliable data input. While the battery and device could fit snugly within standard storage, the external camera had to be specifically mounted rear-facing underneath the saddle. We engineered a four-part camera mounting system, with two plates limiting rotation and two other plates latched to a rigidized saddle mount with a sliding mechanism for the camera plates. Once the camera slides into the mount, we secure it with threaded fasteners to ensure stability and remove any risk of detachment.

V. ALGORITHM AND TRACKING

With a stable video feed in place, we used Time-To-Collision logic to detect rapidly approaching vehicles. We used optical expansion [4], or Tau Theory, which asserts that the rate of a vehicle/object's visual growth is inversely proportional to its time to impact. To keep track of each vehicle's approach, we use Euclidean centroid tracking. By tracking the distance between object centers between frames, the system can maintain unique IDs for each object. Then, the algorithm observes the derivative of the bounding box area. When the derivative exceeds a predetermined value, the system triggers an alert.

VI. CONCLUSION

Current ADAS are locked away behind the paywall of expensive electric cars. TALOS erases that economic barrier by providing a high-performance, open-source, cost-effective, and modular safety system that is compatible with any bicycle. TALOS makes safety and zero-emission mobility accessible to everyone, not just those who can afford premium vehicles.

REFERENCES

[1] S. L. Handy, Y. Xing, and T. J. Buehler, "Factors Associated with Bicycle Ownership and Use: A Study of Six Small U.S. Cities," *Transportation*, vol. 37, no. 6, pp. 967–985, Apr. 2010, doi: https://doi.org/10.1007/s11116-010-9269-x.

[2] T.-Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context," *arXiv.org*, May 01, 2014. https://arxiv.org/abs/1405.0312

[3] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, Dec. 2023, doi: https://doi.org/10.3390/make5040083.

[4] D. N. Lee, "A Theory of Visual Control of Braking Based on Information about Time-to-Collision," *Perception*, vol. 5, no. 4, pp. 437–459, Dec. 1976, doi: https://doi.org/10.1068/p050437.