Eren Işık

(+90) 505 548 35 96 | erenisik@std.iyte.edu.tr | linkedin/in/eren-işık-271914349 | github/Ursidae0

Education

İzmir Institute of Technology — İzmir, Türkiye
B.Sc. Computer Engineering — GPA: 3.25/4 — *2023 – 2027*

Experience

Cengaver Rover — İzmir, Türkiye
Software Team Member — TEKNOFEST University Rover Competition — *September 2025 – Present*

- Rewrote drive firmware from MicroPython to C (Raspberry Pi Pico SDK), replacing a polling loop with a 1 kHz real-time control loop; added CRC-8 packet integrity, hardware watchdog, and Dead Man's Switch — all absent in the original.
- Designed a custom binary communication protocol over USB CDC with packet framing and error detection; implemented the matching host-side driver as a ROS 2 node on the Jetson.
- Built a multi-layer safety system — software disarm on communication timeout, hardware watchdog reset, and GPIO-level motor cutoff. (ensuring fail-safe behavior on link loss or firmware fault)

Projects

Sparse GPU Kernels (Paper Reproduction) | GitHub | *CUDA, C++, Docker* — December 2025 – January 2026

- Deployed the Sputnik CUDA library (Gale et al., SC'20) inside an NVIDIA NGC container, adapting container configurations for compatibility with updated Linux kernel and NVIDIA driver versions.
- Benchmarked SpMM and SDDMM kernels on an RTX 4070 Super against cuSPARSE baselines; achieved 7.8x speedup over dense cuBLAS at 99% sparsity with peak SpMM throughput reaching 4.3 TFLOPs.
- Reproduced the paper's 70.5% sparsity break-even point (paper reports ∼71% on V100), confirming kernel efficiency scales across GPU generations.

Neural Network Inference on NVIDIA Jetson Nano | GitHub | *Python, Docker, CUDA* — December 2025

- Deployed and compared three TensorRT-optimized models (PeopleNet, SSD-Inception-v2, SSD-MobileNet-v2) on NVIDIA Jetson Nano (Tegra X1); PeopleNet achieved fastest inference at ∼650 ms/image on the embedded GPU.
- Profiled SM efficiency across all models using nvprof; fused convolution kernels reached up to 99.9% SM utilization while NMS post-processing was identified as the primary bottleneck at ∼12.6% SM efficiency.

Computer Vision Project Portfolio | GitHub | *Python, OpenCV, scikit-learn, scikit-fuzzy* — December 2025 – January 2026

- **People Counting** — extracted morphological features (closing/opening) from foreground masks and trained an MLP to estimate crowd count; best method achieved MAPE of 10.64% with 86.27% of estimates falling within 15% of ground truth.
- **Retinal Blood Vessel Extraction (Paper Reproduction)** — reproduced MATLAB-based hybrid FCM + supervised classification pipeline in Python for retinal vasculature extraction from fundus images; achieved AUC of 0.86 and 92.6% pixel accuracy on the DRIVE test set; attributed the residual performance gap to the paper's proprietary Root Guided Decision Tree, unavailable in standard ML libraries.

Technical Skills

Programming: Python, C/C++, CUDA
Libraries: scikit-learn, OpenCV, NumPy, Matplotlib, scikit-fuzzy
Developer Tools: Git, Docker, CMake, Bash, nvprof, Raspberry Pi Pico SDK (ARM Cortex-M0+)
Languages: Turkish (Native), English (B2)