Project Proposal: Cross-Architecture Motion Blur Robustness via Feature-Level Knowledge Distillation

Kamren James
M.S. Robotics
UCR
kjame001@ucr.edu

Thi Thi Khine
M.S. Robotics
UCR
tkhin002@ucr.edu

Sharmeen Kazi
M.S. Computational Data Science
UCR
skazi013@ucr.edu

Abstract

Mobile robots often suffer from detection degradation due to motion blur and rolling shutter artifacts, particularly when using lightweight edge processors. This project investigates whether **Feature-Level Knowledge Distillation (KD)** can consistently recover lost accuracy across different generations of object detectors. We propose a pipeline where a “Teacher” model (trained on sharp images) supervises “Student” models (exposed to synthetic motion blur) via dynamic temperature scaling and attention transfer. We will benchmark this approach across **YOLOv8n**, **YOLOv11n**, and **YOLOv26n** to determine if the distillation benefits are architecture-agnostic, ultimately deploying the most robust model to a Viam Rover 2 (Raspberry Pi 5).

1 Problem Statement & Motivation

Autonomous robots operate in dynamic environments where rapid movement induces severe motion blur due to vibration. Standard detectors, trained on clean internet datasets (COCO), fail catastrophically under these conditions. While heavier models can handle blur, they are too slow for our **50ms latency target**. We aim to transfer the "blur-invariance" of a heavy teacher to a nano-scale student without adding inference latency.

2 Methodology

We have implemented a custom **Distillation Trainer** in PyTorch that extends the standard Ultralytics training loop. Our approach consists of three key components:

1. **Synthetic Blur Augmentation:** A custom pipeline (‘BlurAugment‘) applies randomized directional motion blur and Gaussian noise to training batches on-the-fly, creating a domain gap between the Teacher (who sees sharp images) and the Student (who sees blurred images).
2. **Feature-Level Loss:** Unlike standard Logit Distillation, we calculate the MSE loss between the intermediate feature maps of the Student and Teacher. This forces the Student to learn geometric features (edges/shapes) that align with the Teacher’s "clean" perception, effectively ignoring the blur artifacts.
3. **Cross-Architecture Validation:** We will train and compare three distinct student architectures—YOLOv8-Nano, YOLOv11-Nano, and YOLOv26-Nano—to evaluate if newer architectures respond better to distillation.

3 Dataset and Evaluation

Training Data: 2017 COCO train/val annotation dataset, augmented with synthetic blur.
Test Data: A "Golden Test Set" of 600 *real-world* blurry images captured from the moving robot.
Metrics: mAP@50 (Accuracy) and Inference Latency (ms) on Raspberry Pi 5 / Jetson Orin Nano.