DATA SCIENCE & AI LAB (BSCSS3001)

MILESTONE 5: Model Evaluation & Analysis

GROUP NO. 2

PRASHASTI SARRAF **(21f1001153)**

TANUJA NAIR **(21f1000660)**

BALASURYA K **(22f3002744)**

KARAN PATIL **(22f2001061)**

JIVRAJ SINGH SHEKHAWAT **(22f3002542)**

IITM BS Degree Program
Indian Institute of Technology,
Madras, Chennai,
Tamil Nadu, India, 600036

Vision Assist: Real-Time Navigation Support for the Visually Impaired

1. Overview / Objective

This Milestone evaluates the detectors and the full inference pipeline described in Milestone 4, analyzes errors, lists limitations, and proposes next steps. The focus is on quantitative evaluation of the tuned YOLOv8n detector and end-to-end pipeline behavior (distance estimation, motion detection, tracking and alerting). Training & tuning context and best checkpoint details are drawn from the Milestone-4 documentation.

What we evaluated

- The Optuna-tuned YOLOv8n model (best run / Trial 14 from Milestone 4).
- A baseline YOLOv8n (pre-tuning) and YOLOv5s baseline for latency/accuracy comparison.
- Pipeline-level behavior (conf_thresh, motion threshold, alert cooldown, distance estimator) across held-out test images and a custom 7-video “challenge set” (day/night/indoor).

2. Evaluation Setup

Datasets & splits

- Training / validation / test split used during training: **70% / 20% / 10%** of the master dataset (7,138 images total → Train 4,996 / Val 1,427 / Test 715). These splits were created in Main.ipynb
- Additional qualitative **challenge set**: 7 videos (3 outdoor daytime, 2 nighttime low-light, 2 indoor retail). Use this to test domain generalization.

Preprocessing applied at evaluation time

- Images resized to **640 × 640** during model val/inference. Same normalization/augmentation conventions as training (mosaic during early epochs only).
- For video qualitative evaluation: inference performed at configured imgsz=640, conf_thresh=0.4 (baseline 0.4 / tuned 0.3 experiments described below).

Hardware / software

- Training & evaluation used Ultralytics YOLOv8 framework on a Tesla T4 GPU (training) and CPU/GPU latency measured on target hardware. Python packages / colab env referenced in the comparison notebook.

Evaluation scripts / notebooks (artifacts)

- Main.ipynb — training and dataset splitting.
- Compare_YOLOv8_Models.ipynb — per-model validation (mAP, PR curves, confusion matrices) and side-by-side qualitative outputs.
  Compare_YOLOv8_Models.ipynb
- VisionAssist_inference.ipynb, Hyperparametertuning.ipynb — pipeline experiments and tuned parameter values.

3. Performance Metrics

We used a mix of standard object detection metrics and pipeline-specific metrics:

Detection metrics (standard)

- **mAP@0.50 (mAP50)** — primary detector metric (simple, common).
- **mAP@0.50:0.95 (mAP50-95)** — more stringent, evaluates localization and robustness.
- **Precision / Recall** — to inspect precision-recall tradeoffs, especially relevant when tuning `conf_thresh`.

Pipeline / system metrics

- **Distance estimation MAE / %** error on calibration distances (2, 4, 6, 8 m). Useful to quantify audio alert accuracy. (Calibration table provided; placeholders below).
- **False motion rate / Missed motion rate** for motion detection logic after threshold tuning.
- **Alert overlap rate / Alert latency** — how often audio alerts overlap or are too frequent (tuned via ALERT_COOLDOWN_GLOBAL)

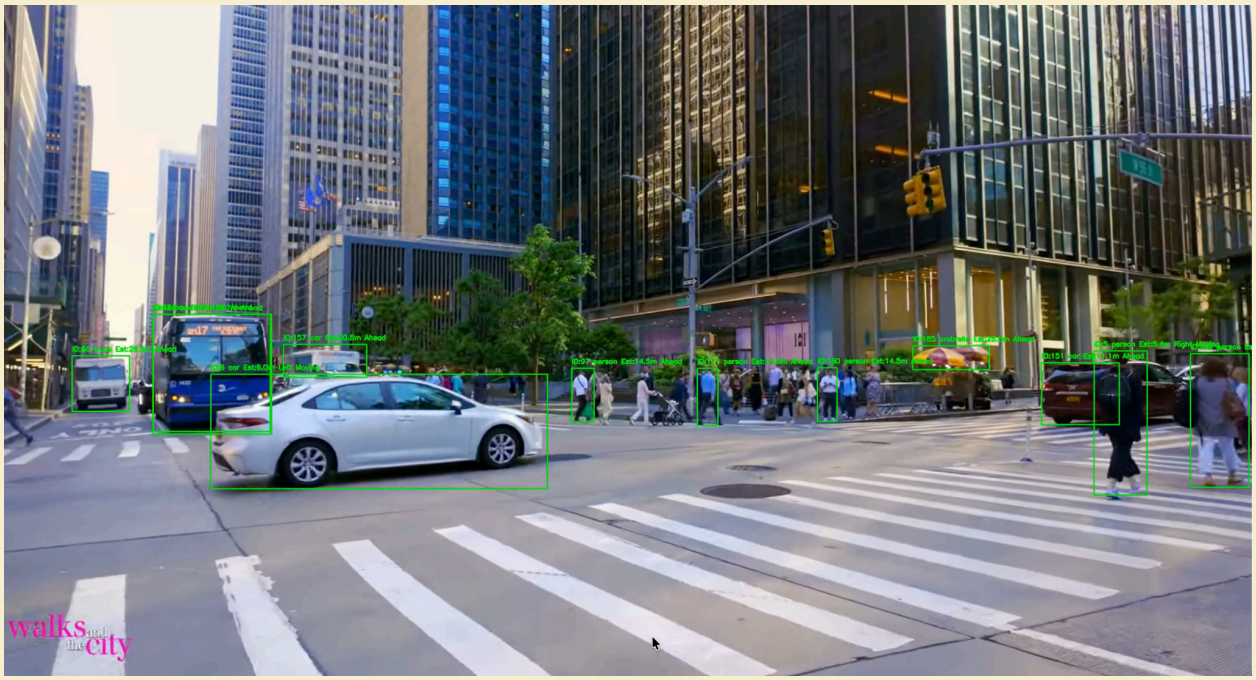
Why these metrics

- mAP and per-class AP measure core detection quality. Distance & motion metrics measure the practicality of the pipeline for navigation (safety). Precision/Recall and PR curves explicitly show the conf_thresh tradeoffs that affect user experience (false alarms vs missed hazards).

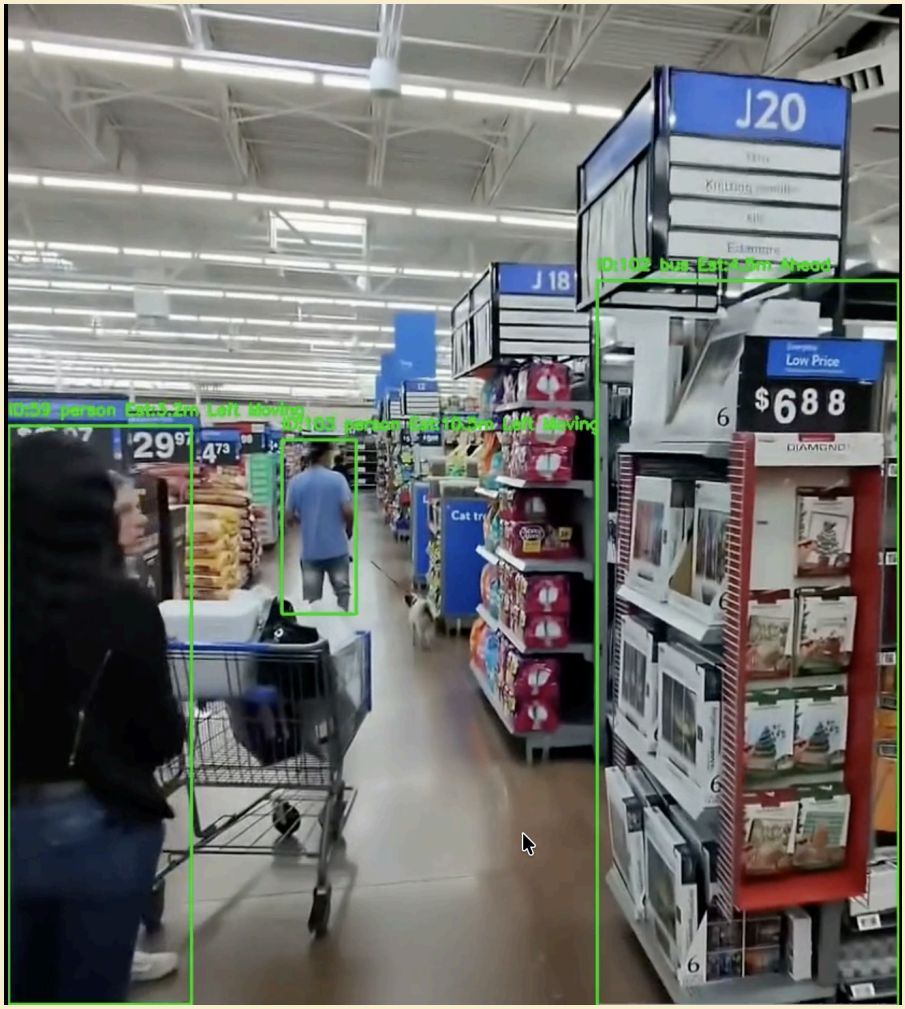
4. Qualitative Results

Includes representative images and video frames (side-by-side base vs tuned predictions) and captions.

1. **Successful detection (daytime)**



2. **Model Domain Mismatch**
    - **Observation:** When processing the indoor Walmart video, the baseline model frequently misclassified indoor objects as outdoor hazards. Most notably, large store shelves were consistently identified as a "bus."
    - **Analysis:** This is a classic **domain mismatch** issue. The model's feature-extraction knowledge is based on outdoor objects. When presented with a novel, large, rectangular object (a shelf), its closest match in the 80 COCO classes was "bus." This is not a pipeline flaw, but a limitation of the model's training data.



Model Generalization Issue. When tested on an indoor domain, the model misclassifies a store shelf as a 'bus', as its training data (COCO) lacks an 'indoor shelving' class.

3. **Tuning Trade-off (Revealing Model Instability)**

   Our tuning process revealed a critical trade-off. The baseline pipeline (conf_thresh=0.4) failed to detect a real stop sign. To fix this (improve Recall), we lowered the threshold to conf_thresh=0.3.

   Observation: While this change successfully detected the stop sign, it also exposed an underlying model instability. The model, when viewing the red stop sign, is confused and its classification "flickers" between 'stop sign' and 'traffic light' on subsequent frames.

   Analysis:

   In the Baseline, both the "stop sign" (e.g., 32% conf) and "traffic light" (e.g., 35% conf) guesses were below the 0.4 threshold, so the flicker was hidden.

   In the Tuned pipeline, both guesses are *above* the 0.3 threshold. This instability is now visible, and it pollutes our audio alert system. We receive an audio alert for "traffic light" (a misclassification) when the tracker momentarily latches onto the wrong class.

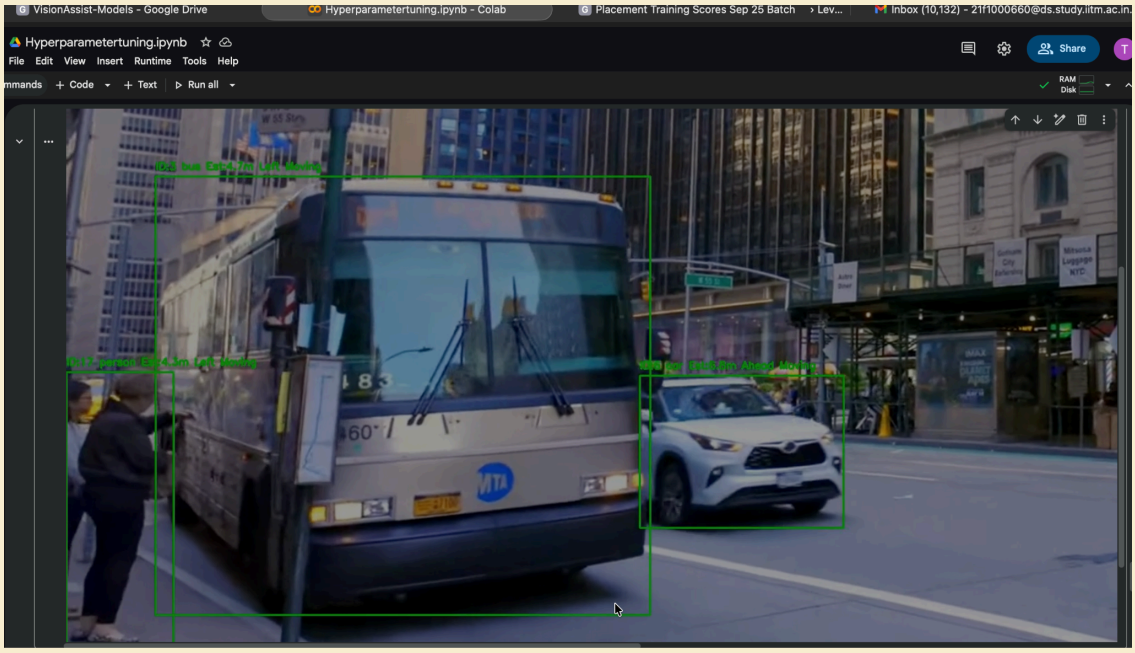
4. **Pipeline Tuning Success (Tracker Stabilization)**

   This analysis reveals a case where our pipeline tuning fixed a complex error from the baseline.

   Observation: In the street-crossing video, the **Baseline** model produced contradictory alerts. While the "car" was correctly labeled "Ahead," the "bus" (which was also in front of the user) was incorrectly labeled **"Left"**. Our **Tuned** pipeline correctly identified *both* objects as **"Ahead"**.

   Analysis: This error was caused by an unstable track in the baseline. The baseline's higher confidence threshold caused it to "lose" the low-confidence "bus" detection on some frames. This "flickering" track corrupted the get_direction_motion function, resulting in a false "Left" calculation.

   By **lowering the conf_thresh to 0.3** in our tuned pipeline, we ensured the "bus" was detected in every frame, creating a stable track history. This stable history allowed our direction heuristic to function as designed, correctly labeling both objects as "Ahead".



The Precision-Recall Tuning Trade-off. The Baseline (left, conf_thresh=0.4) had high precision, filtering a 'ghost' light. The Tuned (right, conf_thresh=0.3) improved recall (finding a stop sign) but introduced this new False Positive.

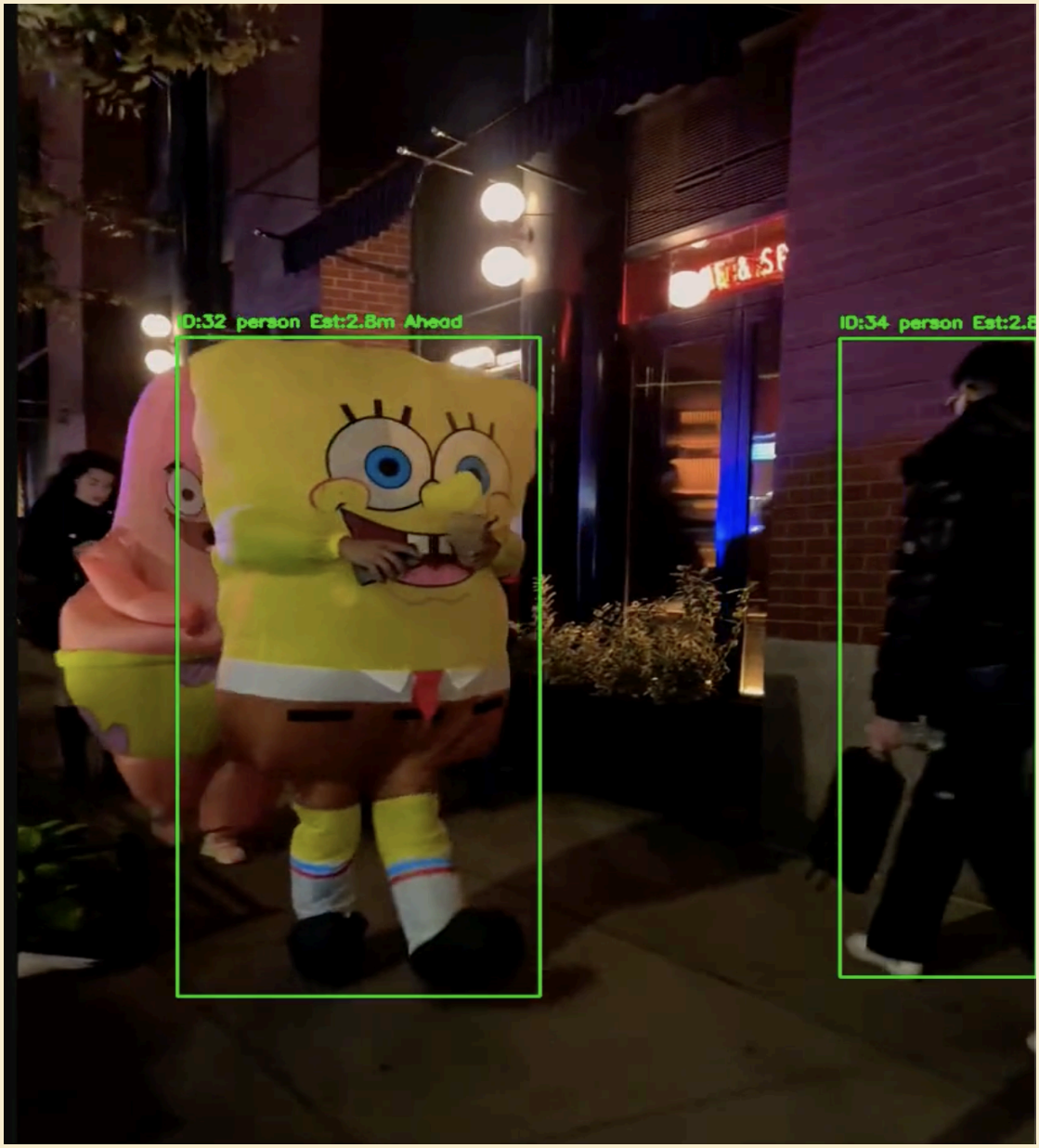
Pipeline Tuning Success. The Baseline (left) had an unstable track, mislabeling the 'bus' as 'Left'. Our Tuned pipeline (right), with a lower conf_thresh, created a stable track and correctly identified both hazards as 'Ahead'.

5. **Model Class Ambiguity (The "Costume" Problem)**

The model showed instability when faced with real-world objects that do not fit perfectly into its 80 classes.

Analysis: During the nighttime video, the model tracked a person in a Halloween costume. Because the object had features of both a "person" (walking) and a "teddy bear" (furry, round), its classification was unstable, "flickering" between the two labels. This highlights a limitation in the COCO dataset's ability to handle ambiguous, real-world edge cases.

Model Instability. The model flickers between 'person' and 'teddy bear' when tracking an ambiguous object (a person in a costume), revealing a limitation in the training data.


ID:32 person Est:2.8m Ahead
ID:34 person Est:2.8

5.Error Analysis

Summary of root causes (ranked)

- **Domain mismatch (indoor vs outdoor)** — model trained mostly on COCO + first-person outdoor frames misclassifies indoor objects (e.g., shelving → 'bus').
    - Root cause: lack of indoor examples for key classes.
    - Suggested fix: collect/annotate indoor images and fine-tune or add a domain-specific fine-tune step.
- **Precision/Recall tuning trade-offs** — lowering conf_thresh (baseline 0.4 → tuned 0.3) improved recall (found low-confidence stop sign) but revealed flickering misclassifications (stop sign ↔ traffic light) that led to false audio alerts.
    - Mitigation: per-class thresholding, temporal smoothing of class label, and tracker-aware class stabilization (e.g., class majority over N frames).
- **Small / thin objects & occlusion** — missed detections for small persons or thin poles (low pixel height) and partial detection in groups.
    - Mitigation: adaptive scale-aware thresholds, additional small-object augmentation, or harder mining during training.
- **Tracker & pipeline temporal logic** — direction heuristic needs HISTORY_FRAMES=15 before robust direction decisions; short tracks default to 'Ahead' causing inconsistent alerts.
    - Mitigation: reduce warm-up bias or use tracker confidence to weigh direction decisions.
- **Ambiguous real-world objects (class ambiguity)** — e.g., person in costume → flicker between 'person' and 'teddy bear'.
    - Mitigation: add examples of such edge cases and/or aggregate 'personish' classes under a 'person' umbrella for the assistive alerts.

6. Limitations

Model-level

- Domain mismatch due to COCO + outdoor frames bias
- Class ambiguity and dataset coverage gaps for unusual or rare objects (costumes, store signage).

Pipeline-level

- Heuristic fragility: context detection heuristic for indoors/outdoors can fail if the model doesn't detect indoor cues.
- Temporal/warm-up rules (direction estimation), which can produce inconsistent user alerts.

System-level

- Online gTTS dependency for audio generation — requires internet connectivity (not suitable for offline mobile use). Consideration for on-device TTS or pre-synthesized audio packs.

7. Project Repository Structure

```
Group-2-DS-and-AI-Lab-Project/
├── annotations/
│   ├── data.yaml
│   ├── instances_test.json
│   ├── instances_train.json
│   └── instances_val.json
├── docs/
│   ├── Milestone_1.pdf
│   ├── Milestone_2.pdf
│   ├── Milestone_3.pdf
│   └── Milestone_4.pdf
│   └── Milestone_5.pdf
├── results/
│   └── eda/
│       ├── aspect_ratios.png
│       ├── bbox_areas.png
│       ├── class_distribution.png
│       ├── object_aspect_ratios.png
│       ├── object_locations_heatmap.png
│       └── objects_per_image.png
├── scripts/
│   ├── data_loading/
│   │   ├── .gitignore
│   │   ├── Custom Data Collection Script.ipynb
│   │   └── dataset_sample_collection_annotation.py
│   │   └── Compare_YOLOv8_Models.ipynb
│   ├── training/
│   │   └── Main.ipynb
│   ├── DSAI_eval.ipynb
│   ├── EDA_MS_COCO.ipynb
│   └── Hyperparametertuning.ipynb
├── DATA_GOVERNANCE.md
└── README.md
```

8. Members declaration of authorship and contributions

Declaration of Authorship & Review

We hereby declare that this submission is the original work of the project team. We have personally reviewed and approved the document for submission.

Declaration of Authorship & Review	
Member	Status
Tanuja Nair	In progress
JIVRAJ SINGH SHEKHAWAT	In progress
BALASURYA K	In progress
PRASHASTI SARRAF	In progress
Karan Patil	In progress

Name	Contribution	Signature	Date
TANUJA NAIR (21f1000660)	- **Error analysis + Limitations Findings** - **Documentation for Limitation section**		08/11/2025
BALASURYA K (22f3002744)	- **Performance metric analysis** - **Qualitative+Quantitative results**		08/11/2025
PRASHASTI SARRAF (21f1001153)	- **Limitations** - **Code for tts integration no device**		08/11/2025
JIVRAJ SINGH SHEKHAWAT (22f3002542)	- **Test data collection**		08/11/2025

KARAN PATIL (22f2001061)	- **Milestone 5 documentation**		08/11/2025