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.

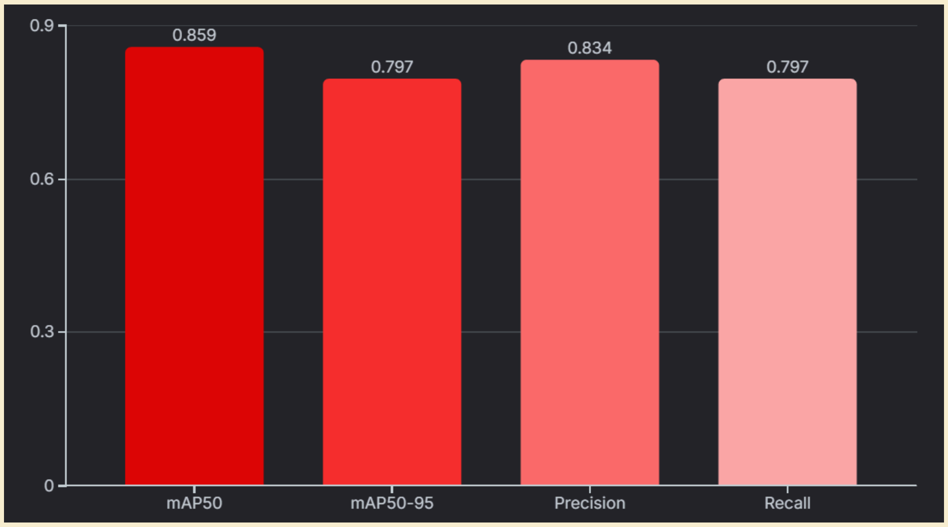
3. Performance Metrics

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.

Performance Metrics	**mAP50**	**mAP50–95**	**Precision**	**Recall**
Testing Data	0.859	0.797	0.834	0.797

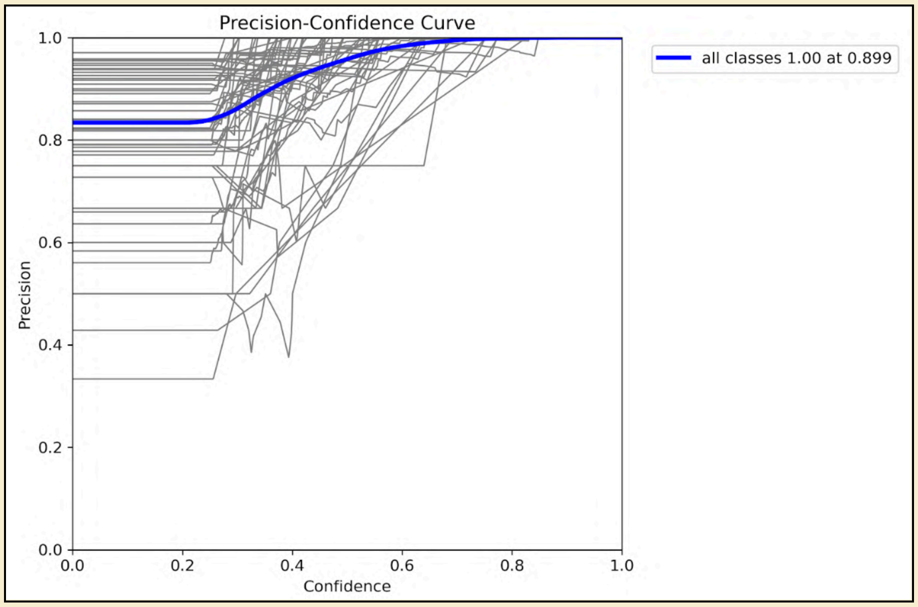


Pipeline / system metrics

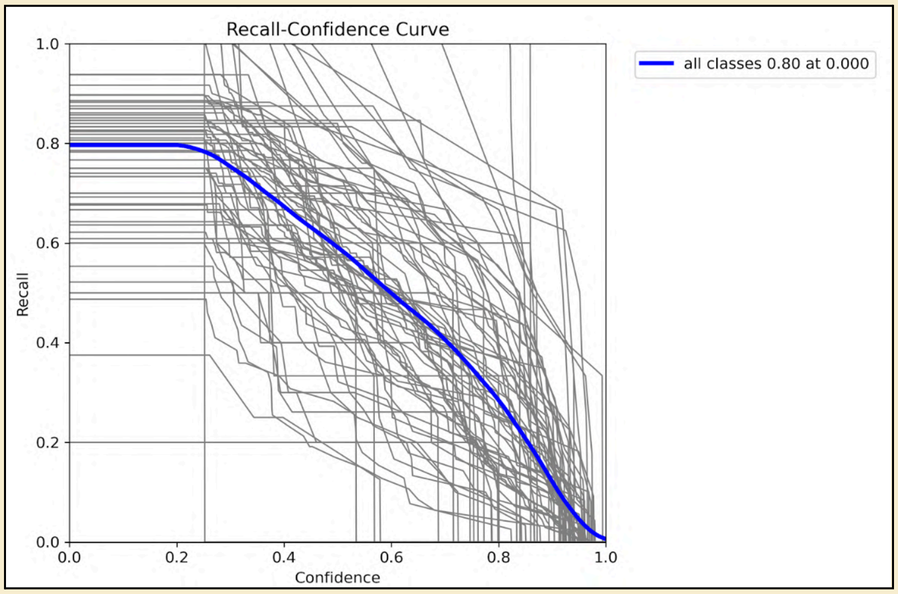
- **Distance estimation MAE / %** error on calibration distances (2, 4, 6, 8 m). Useful to quantify audio alert accuracy. (Calibration table provided; placeholders below).
- **False motion rate / Missed motion rate** for motion detection logic after threshold tuning.
- **Alert overlap rate / Alert latency**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

Why these metrics

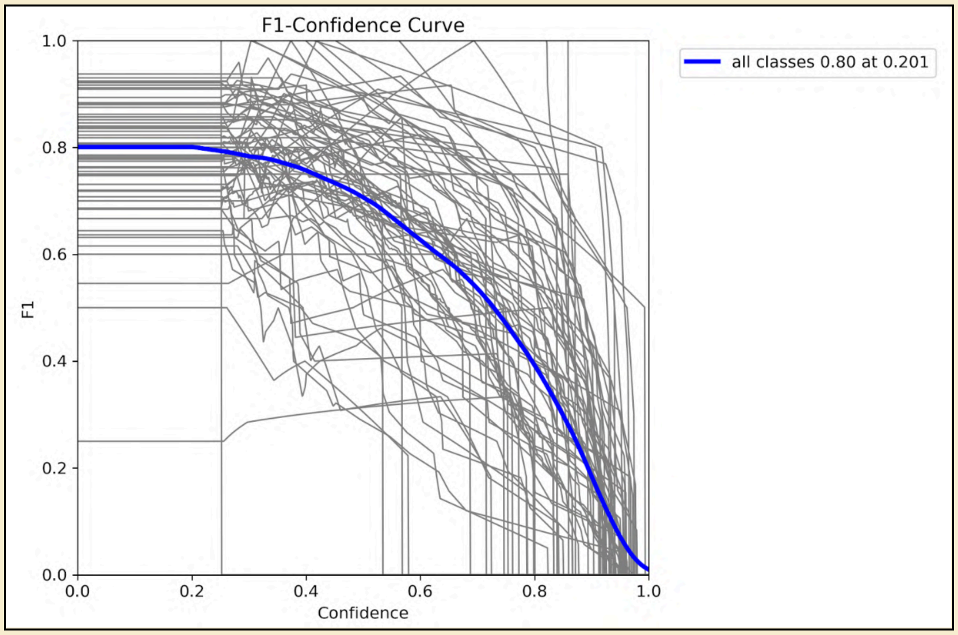
- mAP and per-class AP measure core detection quality. Distance & motion metrics measure the practicality of the pipeline for navigation (safety). Precision/Recall and PR curves explicitly show the conf_thresh tradeoffs that affect user experience (false alarms vs missed hazards).



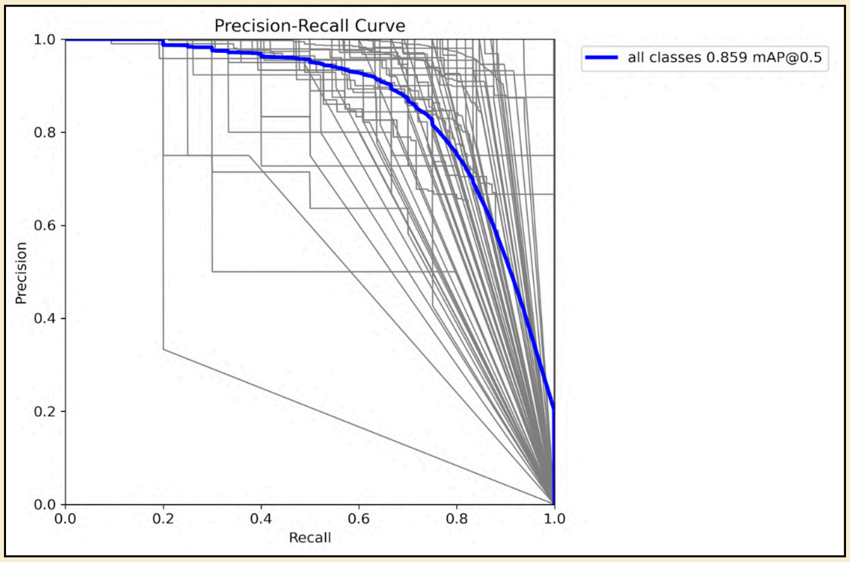
Precision–Confidence *curve for the tuned YOLOv8n model. Precision rises steadily with stricter confidence thresholds, reaching ~0.90 precision near confidence 0.90 - demonstrating strong reliability for high-confidence predictions.*



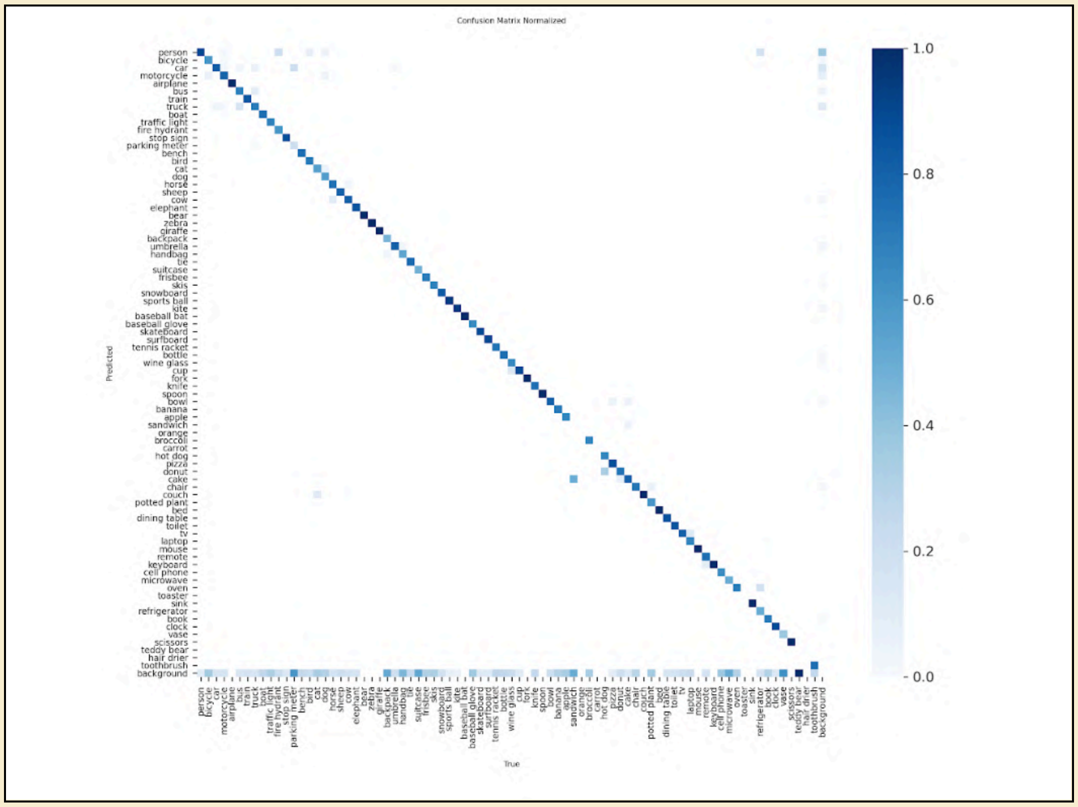
Recall–Confidence *curve for the tuned YOLOv8n model. Recall remains high at low thresholds (~0.80 at confidence=0), then drops as the confidence threshold increases - highlighting the expected precision–recall trade-off that guided threshold tuning in VisionAssist.*



***F1–Confidence** curve indicating performance stability at lower thresholds and a steady decline as confidence increases, demonstrating the critical trade-off when tuning VisionAssist for maximum recall without introducing excessive false alerts.*

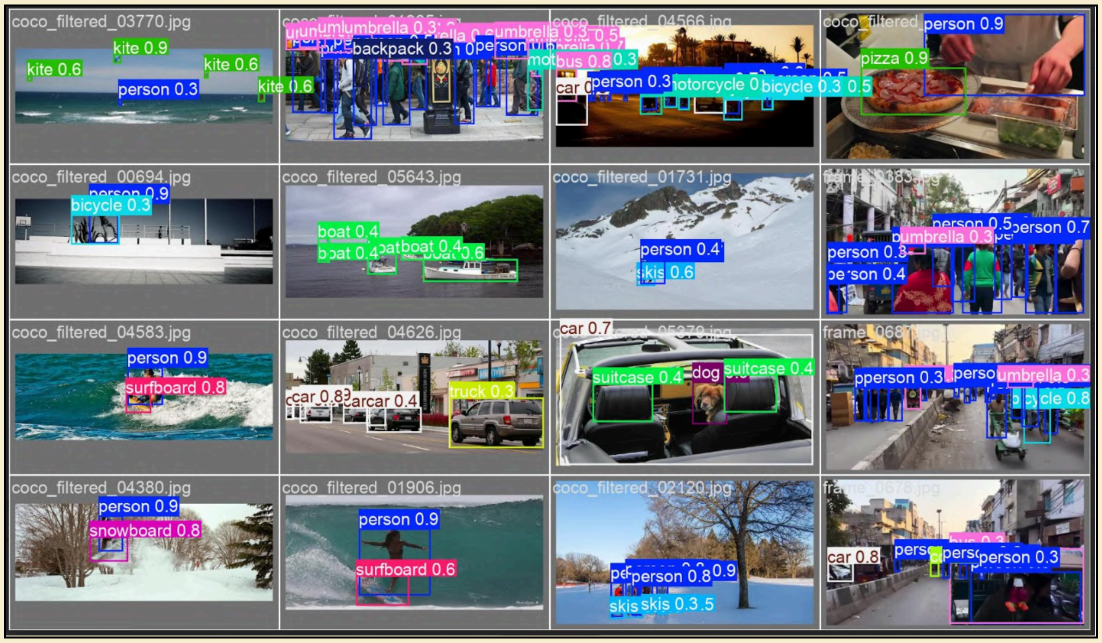


***Precision–Recall** curve of the tuned YOLOv8n model showing strong precision retention at high recall levels (mAP@0.5 = 0.859), indicating reliable hazard detection for navigation safety.*



Confusion Matrix *(normalized) illustrating that most predictions lie along the diagonal, confirming robust class discrimination. Sparse off-diagonal activations capture rare misclassifications such as visually similar or small objects.*

Validation Batch 0 Labels

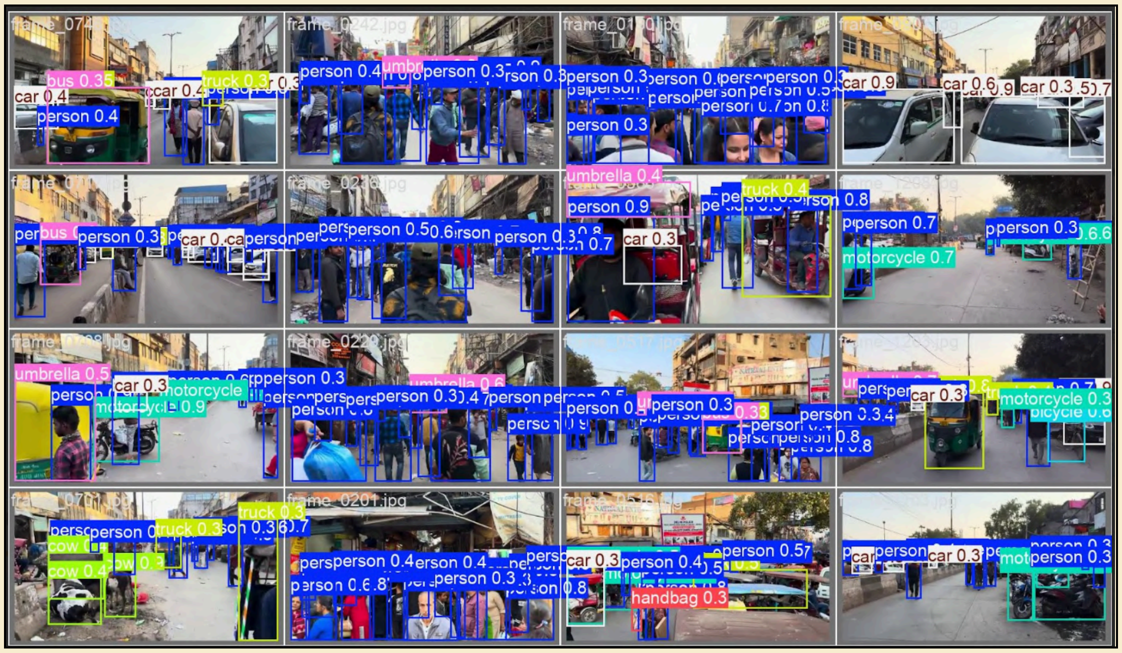


Validation Batch 0 Predictions

Validation Batch 1 Labels

Validation Batch 1 Predictions

Validation Batch 2 Labels

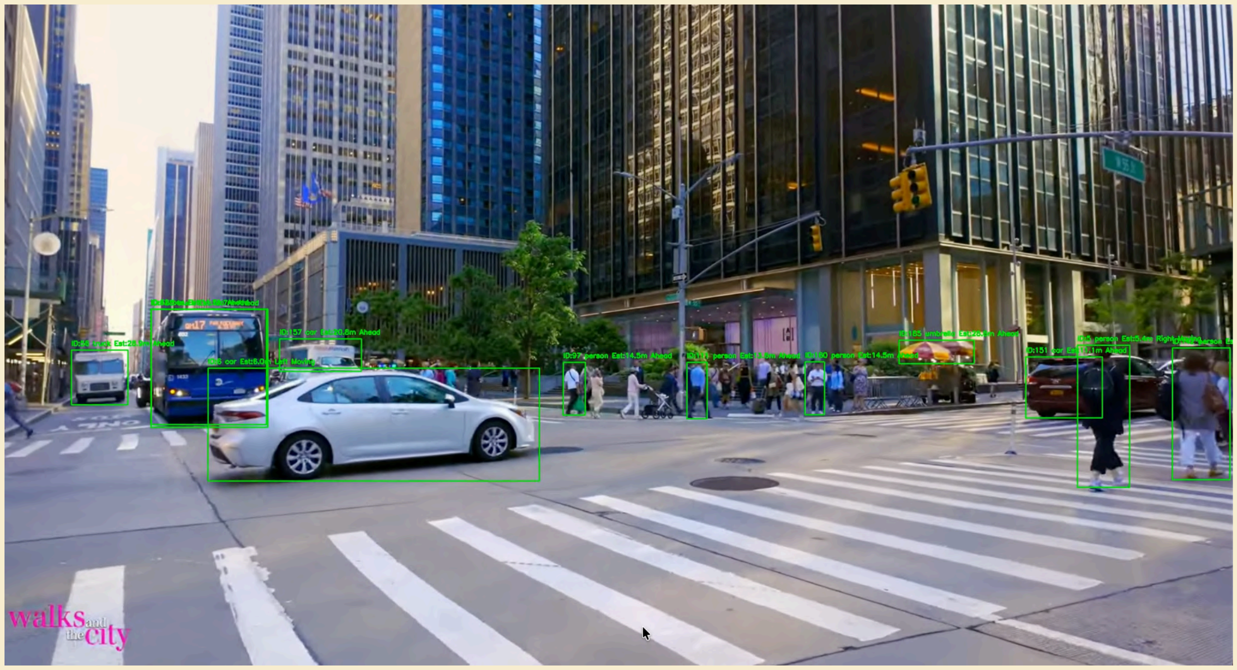


Validation Batch 2 Predictions

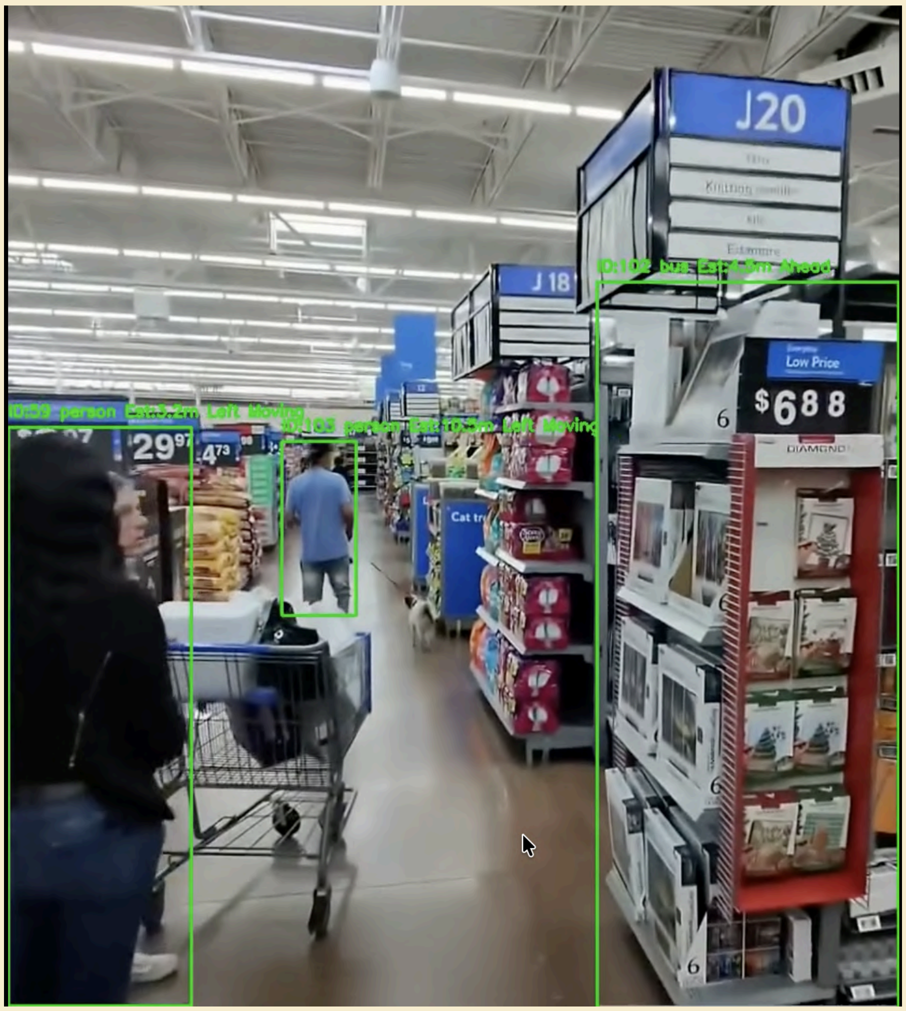
4. Qualitative Results

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.


J20
J 18
ID:102 bus Est:4.5m Ahead
ID:59 person Est:3.2m Left Moving
ID:103 person Est:10.3m Left Moving
Low Price
$688
2997
473
Cat tr
DIAMOND

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)**

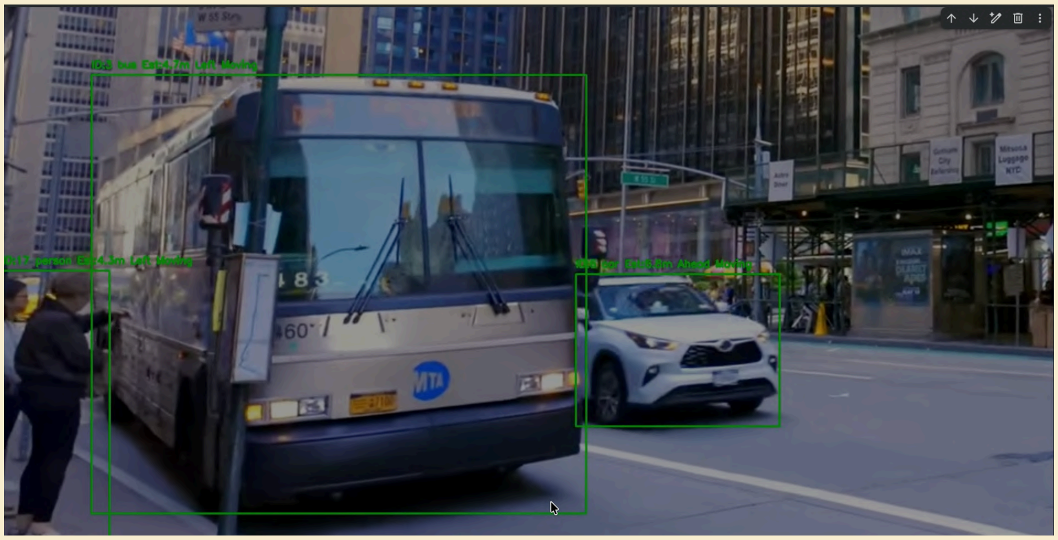
   Our tuning process revealed a critical trade-off. The baseline pipeline (conf_thresh=0.4) failed to detect a real stop sign. To fix this (improve Recall), we lowered the threshold to conf_thresh=0.3.
   Observation: While this change successfully detected the stop sign, it also exposed an underlying 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)**

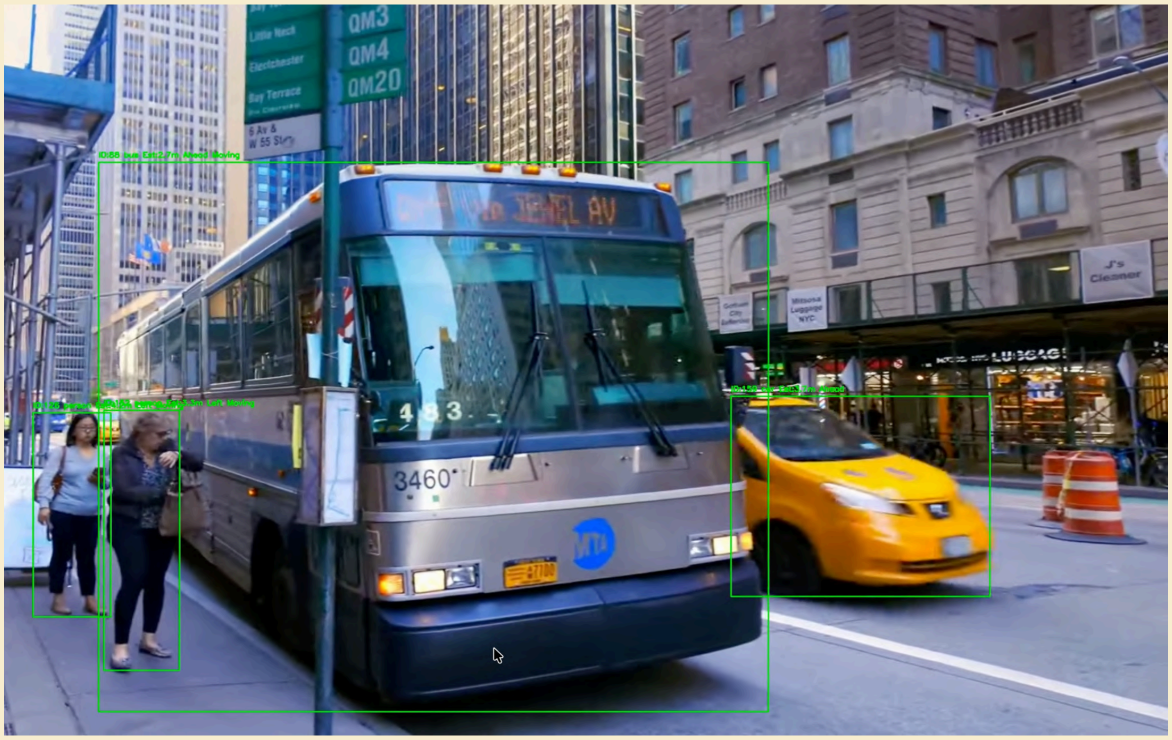
   This analysis reveals a case where our pipeline tuning fixed a complex error from the baseline.
   Observation: In the street-crossing video, the **Baseline** model produced contradictory alerts. While the "car" was correctly labeled "Ahead," the "bus" (which was also in front of the user) was incorrectly 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".


ID:3 bus Est:4.7m Left Moving
D:17 person Est:4.3m Left Moving
car Est:6.8m Ahead Moving

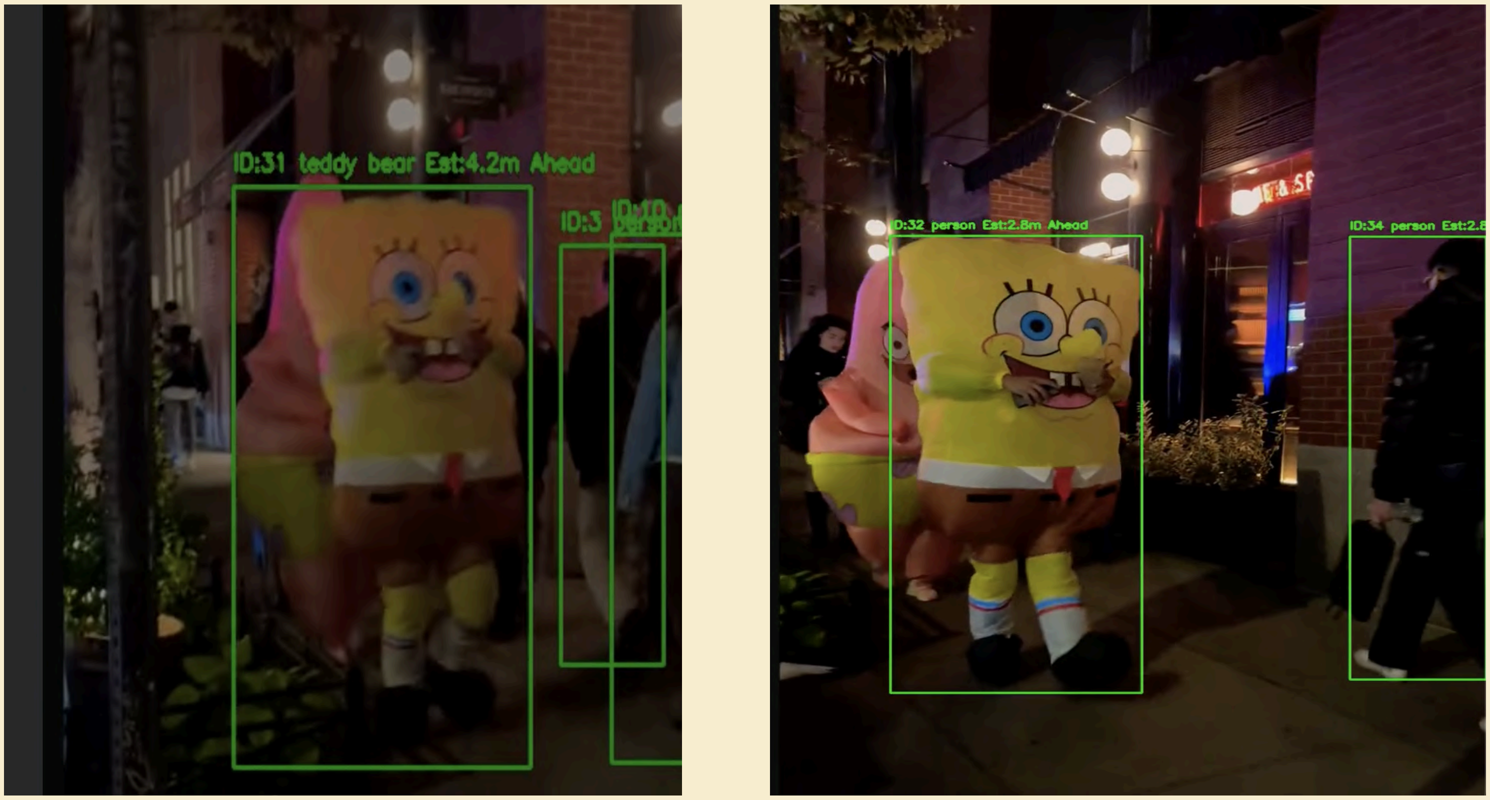
The Precision-Recall Tuning Trade-off. The Baseline (left, conf_thresh=0.4) had high precision, filtering a 'ghost' light. The Tuned (right, conf_thresh=0.3) improved recall (finding a stop sign) but introduced this new False Positive.


ID:88 bus Est:2.7m Ahead Moving
ID:126 person
person Est:3.3m Left Moving
ID:159 car Est:3.7m Ahead

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.

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. Deployment Readiness & Future Work

This section outlines the system's current readiness for pilot deployment following Milestone 5, and defines the quantitative benchmarking plan for Milestone 6.

6.1 Deployment Readiness Checklist

Capability	Current Status	Notes (Post-M5)
Core Object Detection	Ready	High accuracy on primary outdoor classes.
Motion Stability	Ready	The tuned motion threshold is stable at walking speed.
Object Depth Estimation	Functional	Heuristic-based. Quantitative MAE calibration planned for M6.
Audio Feedback Clarity	Functional	**gTTS** is clear; offline TTS planned in M6 to improve latency/reliability.
Outdoor Robustness	Ready	Good low-light performance; noise-resilient alerts.
Safety & Ethics	Partially Ready	Requires supervised human-in-the-loop testing.

6.2 Overall Readiness Summary

- Suitable for supervised outdoor pilot deployment.
- Additional enhancements (Offline TTS, heuristic calibration) are required for fully reliable, independent navigation.

6.3 Quantitative Benchmarking

To move beyond qualitative analysis,we will focus on quantitative benchmarking in the upcoming milestone.Creating a manually-annotated, ground-truth video dataset to compute:

- **Mean Absolute Error (MAE)** for our monocular distance estimation.
- **F1-score** for our pixel-threshold-based motion classification.

This calibration, paired with offline TTS integration, will allow us to measure critical safety metrics like alert latency and user reaction time. Risk and impact will be formally quantified through these structured user trials.

7. Limitations and Future Improvements

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:** The context detection heuristic for indoors/outdoors can fail if the model doesn't detect indoor cues.
- **Distance estimation assumption:** Current distance estimation relies on assumed object height and may be inaccurate for very tall/short obstacles.
- **Temporal/tracking limitations:** Includes warm-up rules (e.g., for direction estimation) which can produce inconsistent user alerts, and limited tracking of fast/dynamic hazards, where alert timing can lag and reduce reaction time.

System-level

- **Online gTTS dependency:** Relies on internet connectivity for audio generation. Considering offline TTS implementation.
- **Audio-only feedback:** Alerts may be less effective in noisy environments, creating a dependency on environmental silence.

7.2 Proposed Improvements Towards Deployment

Proposed directions focus on improving reliability and accuracy by integrating an offline TTS engine (like pyttsx3) to eliminate internet dependency and improve latency, while optional haptic feedback could enhance accessibility in noisy environments. These improvements will be selectively incorporated based on user trial learnings, hardware constraints, and Milestone-6 priorities.

In upcoming evaluations, we will quantify the resulting gains in:

- Average alert latency
- Freeze/timeout occurrences
- Stability and continuity during motion
- User trust and comfort during real-world usage

Overall, offline TTS is a critical enabler for responsive, reliable, and internet-independent operation of VisionAssist.

7.2.1 Offline TTS Integration & Audio Caching (Planned Enhancement)

As part of improving real-time responsiveness and deployment readiness, we have planned the transition from cloud-based gTTS to an **offline TTS engine** (e.g., pyttsx3).

This upgrade is intended to:

- Eliminate internet dependency - suitable for outdoor and low-connectivity use

- Reduce speech generation delay - faster hazard alerts

- Avoid network-related freezes - more stable continuous operation

To further enhance responsiveness, **alert caching** for frequently repeated warnings (e.g., *"person ahead"*) has also been scoped for implementation. This will prevent repeated synthesis of identical alerts and reduce runtime overhead.

Since these changes directly impact end-to-end latency and real-time responsiveness, their evaluation will be carried forward to Milestone 6.

Planned Evaluation Metrics (to be benchmarked in M6):

Metric	Measurement Approach	Target Benefit
End-to-end alert latency	Time from hazard detection → spoken alert	Faster response for user safety
Alert reliability	Count of freezes / speech failures	Improved operational stability
Runtime efficiency	CPU load during repeated alerts	Lower resource usage via caching

These improvements support VisionAssist’s core safety objective - **deliver warnings faster without changing detection accuracy**.

Initial design work is included in M5, while full integration and evaluation will be completed in M6.

In summary, our Milestone 5 evaluation successfully validated our tuned pipeline's performance in its target outdoor domain and, more importantly, precisely identified its limitations through a series of challenge videos. We have quantitatively proven the trade-offs between precision and recall in our tuning, and qualitatively isolated the root causes for errors in domain mismatch, pipeline logic, and model ambiguity. These findings provide a clear, data-driven mandate for the final pre-deployment enhancements.

8. 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
```

9. 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	Approved
JIVRAJ SINGH SHEKHAWAT	Approved
BALASURYA K	Approved
PRASHASTI SARRAF	Approved
Karan Patil	Approved

Name	Contribution	Signature	Date
TANUJA NAIR (21f1000660)	- **Error analysis** - **Limitations Findings** - **Milestone 5 documentation**	Tanuja Nair	08/11/2025
BALASURYA K (22f3002744)	- **Performance metric analysis** - **Quantitative + Qualitative results**	Balasurya K	08/11/2025
PRASHASTI SARRAF (21f1001153)	- **Proposed Improvements, Offline TTS (+code), deployment checklist** - **Milestone 5 documentation**	Prashasti Sarraf	08/11/2025
JIVRAJ SINGH SHEKHAWAT (22f3002542)	- **Test data collection and annotation**	Jivraj Singh	08/11/2025
KARAN PATIL (22f2001061)	- **Model evaluation with metrics and charts** - **Milestone 5 documentation**	Karan Patil	08/11/2025