WAIT TIME MONITORING FOR CAMPUS EATERIES

USING COMPUTER VISION

PRESENTED BY : TEAM 011

Ayush Trivedi
Dheeraj Pamnani
Dominic Darrah
Riya Agarwal
Sravani Bolla

1. INTRODUCTION - PROBLEM IDENTIFICATION

- Unpredictable campus eatery wait times causing frustration..

- At ASU, students often experience long, unpredictable wait times at popular campus eateries like Starbucks, Chick-fil-A, and Pitchforks.
- These delays cause frustration, missed class time, and inefficient use of short breaks.
- There is currently **no digital system** providing real-time visibility into queue lengths.

Why We Chose This Problem:

- It directly impacts a large part of the ASU community (students, staff, visitors).
- Solving it improves daily productivity, well-being, and campus operations.
- The problem is **feasible** to address using Computer Vision (CV) while maintaining privacy.
- There is a clear gap — no existing wait-time tracking solution is implemented at ASU.

Research Conducted:

- Observed dining queues, interviewed students, reviewed ASU's tech (no live tracking), and benchmarked external real-time crowd monitoring systems using computer vision.

1. INTRODUCTION - PROBLEM IDENTIFICATION

- Unpredictable campus eatery wait times causing frustration.

How It Is Being Addressed Currently:

- Students check lines manually by walking to locations.
- They sometimes rely on word-of-mouth from friends or text updates.
- Vendors do not provide real-time queue information or predictive tools.

Issues with the Current State:

- **Inefficiency:** Students waste time traveling to crowded locations and then leaving.

- **Frustration:** Lack of information causes anxiety, stress, and poor time management.

- **Lost Revenue:** Vendors miss sales opportunities from students abandoning long lines.

- **Operational Blind Spot:** Dining management cannot optimize staffing or manage peak traffic without real-time data.

- **Predictive Capability:** ASU cannot predict or balance traffic across vendors, leading to localized overcrowding during class breaks.

2. PROJECT DEFINITION

- Real-time queue tracking using cameras and computer vision.

Scope of the Solution:

- Monitor real-time queue lengths at high-traffic ASU eateries (e.g. Starbucks, Chick-fil-A, Pitchforks).
- Use overhead or wall-mounted cameras to capture queue footage.
- Analyze footage using a Computer Vision (YOLO v8) model to identify and count people to estimate wait times.
- Display live wait times via the ASU Mobile App and digital signage around campus.

Value of the Solution:

- **Utility:** Allows students to plan meals efficiently, reducing wasted trips and time.

- **Impact:** Enhances student experience, improves vendor operations, and fosters a tech-forward campus image.

- **Cost-Benefit:** Upfront investment in cameras and server infrastructure is offset by increased dining revenue, reduced congestion, and better campus resource management.

Key Stakeholders and Beneficiaries

- Students and Visitors
- Dining Services and Vendors
- ASU Administration
- App Developers

2. PROJECT DEFINITION

- Real-time queue tracking using cameras and computer vision.

How Our Solution Is Different:

- **Privacy-Focused:** Our system avoids identity recognition, using only anonymous body detection.
- **University Context:** Adapted to dynamic, fast-paced campus environments with highly variable peak times.
- **Cost-Efficient:** Uses lightweight, open-source CV models like YOLO v8 instead of expensive proprietary sensor network

Success Metrics:

- **Student Adoption:** Majority of students reporting satisfaction with queue visibility tools.

- **Operational Impact:** Significant reduction in average wait times during peak hours.

- **Vendor Sales:** Increase in transaction numbers during historically busy periods.

Case Studies:

Disney Parks
Use CV and sensors to monitor ride wait times and direct crowd flow.

Airports
Real-time TSA security checkpoint wait times using CV and Bluetooth tracking.

Theme Parks & Events
Crowd management through camera analytics and mobile app updates.

References for case studies are provided in slide 19 (Reference slide)

3. DATA ACQUISITION/ PREPARATION

- Collect and prepare cafeteria queue images for model training.

Data Sources:

- Images of cafeteria queues (Starbucks, Chick-fil-A, Pitchforks).
- Focused on varied crowd sizes, lighting conditions, and times of day.

Data Collection:

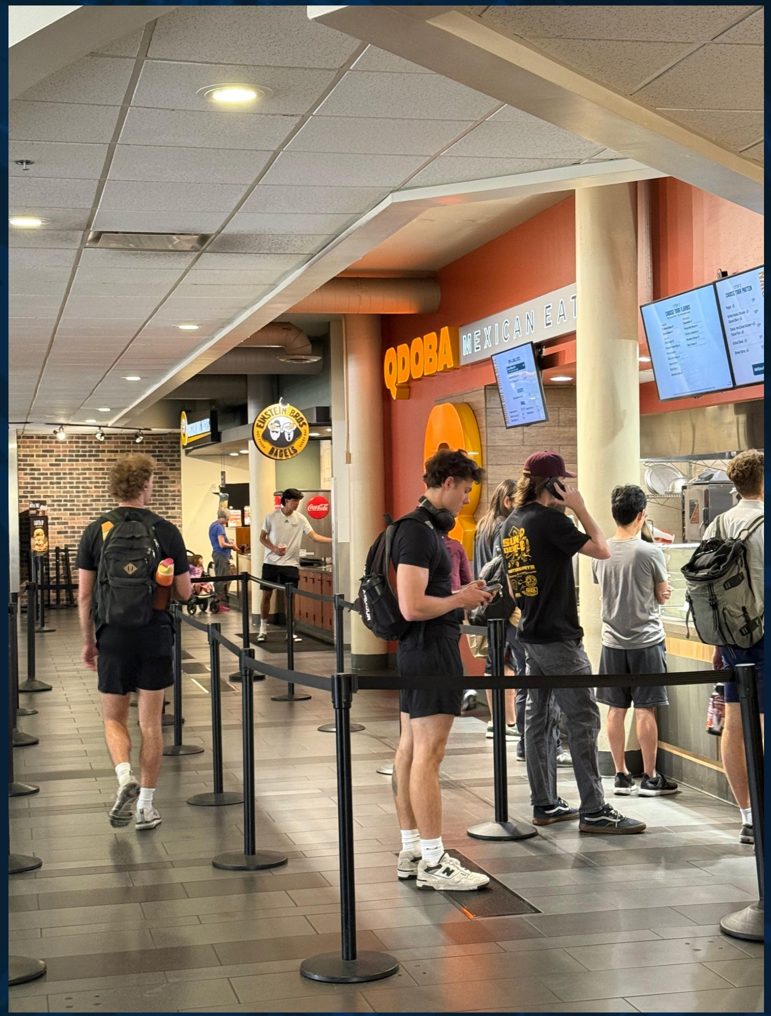
- Manually captured images on campus (with privacy protection).
- Selected frames where queues were clearly visible.

Cleaning and Preprocessing:

Removed blurry or irrelevant frames.

Resize all images to 640x640 pixels for model input.

Annotated bounding boxes around people as ground-truth labels.



3. DATA ACQUISITION/ PREPARATION

- Collect and prepare cafeteria queue images for model training.

Challenges During Data Preparation:

Lighting Variations

- Natural lighting at different eateries changed drastically (sunlight vs indoor lights).

Crowded Scenes

- Overlapping people made bounding box labeling harder.

Privacy Concerns

- Ensured all images focused on body outlines, not faces, to protect individual identity..

4. FEATURE/LABEL ENGINEERING

- Create features and labels by annotating people in queue images.

Preprocessing Steps:

Image Resizing

All images resized to 640x640 pixels for YOLOv8 input.

Frame Cleaning

Dropped blurry, empty, or irrelevant frames to maintain high data quality.

Normalization

Scaled pixel values to [0,1] range for faster and more stable model training.

Features and Labels for the Model:

Features:

- Full-frame color images capturing cafeteria queues.

Labels:

- Manually annotated bounding boxes around every detected person based on shops (single class: "person").
- Bounding boxes used to train the object detection model for people-counting, not identity recognition.

5. MODEL EVALUATION

- Tested model accuracy and gathered user feedback.

PRE - DEPLOYMENT

- **Model Accuracy Testing:**
  - Use a hold-out test set of cafeteria queue images.
- **Manual Verification:**
  - Compare predicted people counts against manually counted ground truth.
  - Validate on different lighting conditions (indoor, outdoor) and crowd densities.
- **User Pilot Testing:**
  - Soft launch with a few eateries and collect student feedback on estimated wait times

POST - DEPLOYMENT

- **Continuous Accuracy Monitoring:**
  - Randomly sample camera feeds weekly for manual spot checks.
  - Calculate model Precision, Recall periodically to detect model drift.
- **User Feedback Surveys:**
  - Regular surveys inside the ASU app asking about perceived wait-time accuracy.
- **Operational Metrics:**
  - Monitor changes in average wait times and vendor transaction counts after system implementation.
- **Retraining Plan:**
  - Schedule retraining every semester using newly collected cafeteria queue data.

6. DEPLOYMENT

- Installed cameras, integrated system, and launched app updates.

Estimated People, Data, Systems, and Computational Resources/Costs:

People

- 1–2 CV Engineers (model training, validation).
- 1 Backend Developer (API, integration with ASU app).
- 1 IT Technician (camera installation/maintenance).

Data

- Initial training set: ~3,000 annotated images of cafeteria queues.
- Ongoing data collection every semester (~1,000 new images/semester).

Systems

- Overhead cameras (one per key location): ~$200–$400 each.
- Central server (cloud-hosted GPU instance or ASU campus server).
- Storage for images/videos (~1 TB/year).

Computational Resources

- Training Phase: Cloud GPU (e.g., AWS EC2 p3 instance) for model development.
- Deployment Phase: Lightweight edge computing devices or backend inference server.

Estimated Costs:

- Initial setup: ~$8,000–$12,000 (cameras + server + manpower).
- Annual maintenance: ~$3,000 (camera upkeep, server costs, retraining labor).

6. DEPLOYMENT

- Installed cameras, integrated system, and launched app updates.

Estimated Workflow Updates/Costs (People, Process, Technology):

People

- Dining staff minor training: how to report camera issues (no operational burden).
- IT support for periodic system checks.

Process Changes

- Update ASU Mobile App to show real-time queue wait times.
- Display queue lengths on digital signage screens inside Student Union buildings.

Technology Changes

- Camera system installation and network connectivity setup.
- Integration of CV model inference output with backend servers and ASU APIs.

Overall Cost-Benefit:

- Initial investment recouped through improved vendor sales, better space utilization, and increased student satisfaction

7. MONITORING/MAINTENANCE

- Continuous system checks and model retraining.

Monitoring and Updating:	• Weekly manual spot checks of camera feeds. • Monthly performance reviews (Precision, Recall metrics). • Semester-based model retraining with new cafeteria data. • Continuous user feedback collection via ASU App surveys.
Unintended Incentives/Consequences (Mitigation)	• **Sudden Crowding:** Students rush vendors showing "short wait" → Smooth updates, randomized refresh times. • **Vendor Manipulation:** Vendors may alter service speed → Cross-validate queue estimates with transaction logs.
Privacy and Security (Mitigation)	• **Risk:** Potential misuse of visual data. • **Mitigation:** No raw video storage; real-time detection only; encrypted data transmission; no facial recognition.
Lessons Learned	• Scope narrowed from "wait time prediction" to "real-time queue monitoring" for feasibility. • Discovered lighting and crowding issues, requiring advanced data augmentation and model tuning.
Scalability	• Expand to all ASU campuses and venues. • Future add-ons: Wait time prediction, event crowd monitoring (libraries, gyms, etc.).

COMPUTER VISION MODEL

CV Model Description

- **Model Used:** YOLO v8 (Ultralytics, Open-Source).
- **Model Design:**
    a. One-stage object detector.
    b. Predicts bounding boxes and confidence scores directly from input images.
    c. Optimized for fast inference (real-time performance) and small objects (crowded scenes).

Which Part of the Solution Uses CV

- CV is used to **analyze camera footage** from cafeteria locations to **detect and count the number of people** in a queue in real-time.
- Output: Live queue length estimation → fed into backend → shown in ASU app and digital signage.

Why CV Is Required

- Manual counting (by staff or students) is unreliable, slow, and not scalable.
- Sensor-based alternatives (entry counters) can't distinguish people **waiting** vs **walking through**.
- Only Computer Vision can provide **accurate, anonymous, real-time queue analysis**.
- Without CV, real-time detection, automation, and dynamic updates would be impossible.

COMPUTER VISION MODEL

Model Training Details

- **Data:** ~100 annotated images of cafeteria queues (custom and open-source).
- **Preprocessing:** Image resizing and cropping for robustness.
- **Model:** YOLO v8 (pre-trained on COCO, fine-tuned on our queue dataset).

External Validation Plan (Pre and Post Deployment)

- Pilot launch at selected eateries; real-world people counts compared against manual counts.
- Ongoing weekly random manual audits post-deployment.

Outcome-Action Pairings

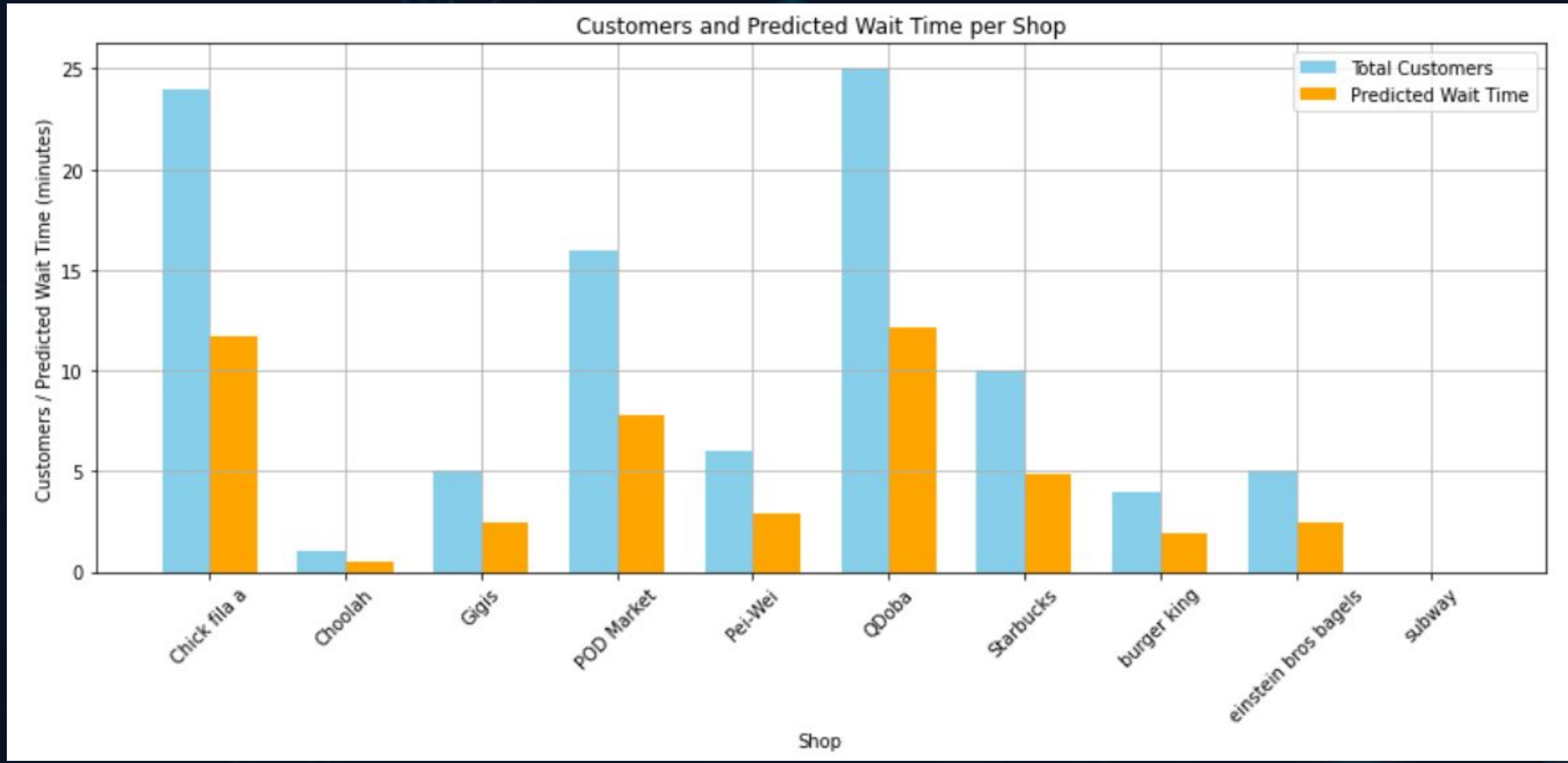
- **TP (True Positive):** Correct people detection → Accurate wait time displayed.
- **FP (False Positive):** Wrong detection → Slightly inflated wait time (acceptable).
- **FN (False Negative):** Missed detection → Slightly underestimated wait time (monitored).
- **TN (True Negative):** Correct no detection → No queue shown.

WORKING DEMO

- Proof of Concept Demo
  - Walk-through of Computer Vision Model
  - Accurately identify customers in queue to order
    - Predicting wait time for customer
      - **Wait time** refers to the duration between a customer joining the eatery line and placing their order.
      - Initially, the average wait time was assumed to be 30 seconds with a standard deviation of 10 seconds. After analyzing annotated images based on customer count, the observed average wait time per customer was 29.19 seconds with a standard deviation of 5.88 seconds.

```
Average wait time per customer: 29.19 seconds
Standard deviation: 5.88 seconds
```

	shop	customer_count	predicted_wait_time(seconds)	predicted_wait_time(minutes)
0	Chick fila a	24	700.50	11.68
1	Choolah	1	29.19	0.49
2	Gigis	5	145.94	2.43
3	POD Market	16	467.00	7.78
4	Pei-Wei	6	175.12	2.92
5	QDoba	25	729.69	12.16
6	Starbucks	10	291.88	4.86
7	burger king	4	116.75	1.95
8	einstein bros bagels	5	145.94	2.43
9	subway	0	0.00	0.00



CONCLUSION

1. SOLUTION OVERVIEW

- Built a real-time Computer Vision system to detect and count people in cafeteria queues at ASU.
- Deployed YOLOv8 model on camera feeds to estimate live wait times, integrated with ASU app and digital signage.
- Focused on anonymous, privacy-preserving detection without storing facial or identity data.

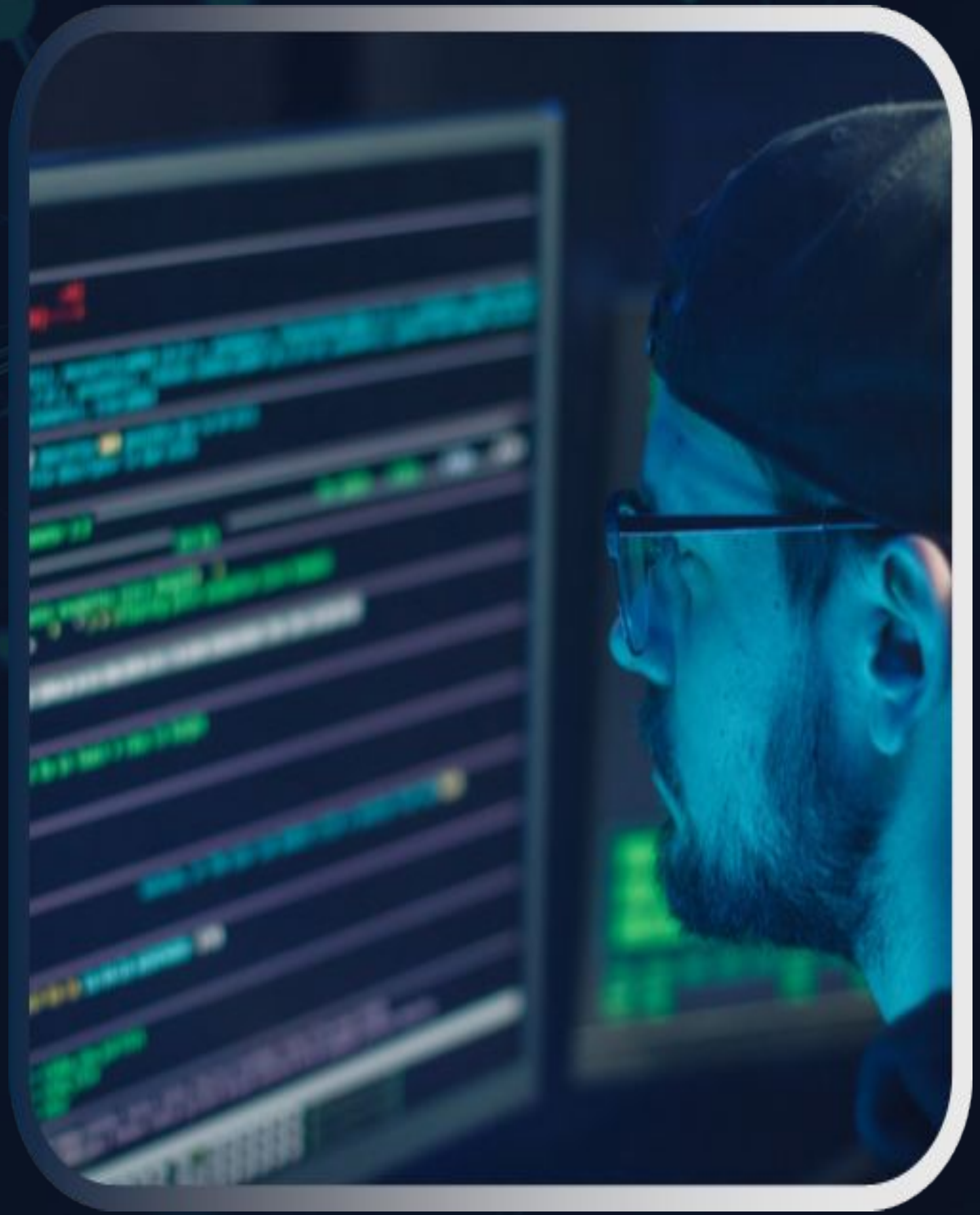
2. KEY FINDINGS

- Real-time queue updates improve student satisfaction and dining vendor efficiency.
- Computer Vision approach is scalable and operationally practical for campus environments.
- Shops like Chick-fil-A and Qdoba have the highest customer counts and predicted wait times compared to others.
- Smaller shops like Choolah, Burger King, and Einstein Bros Bagels have fewer customers and much shorter predicted wait times.

CONCLUSION continued

3. LIMITATIONS

- Reduced accuracy in dense, poorly lit environments; occasional misses due to occlusion or non-ideal poses.
- Static cameras miss dynamic queue movements without advanced tracking.
- Struggles in dense crowds; plan to use instance segmentation models like Mask R-CNN.
- Current pre-trained YOLOv8 models miss occasional customers standing in queues, due to non-ideal poses or occlusion.



4. FUTURE WORK

- Integrate predictive analytics to estimate future wait and service times based on historical patterns and transaction rates.
- Expand the system to libraries, gym facilities, event spaces, and other areas.
- Upgrade detection models by using instance segmentation models like Mask R-CNN for better performance in crowded scenes.
- Develop a continuous learning system with live feedback and auto-retraining for improved accuracy.
- Fine-tune models specifically for university eatery queues and extend to handle dynamic moving queues.

Tabulated Task Summary

Team Member	CV Model / Research	Presentation Content Creation	Final Presentation Delivery
Ayush Trivedi	YOLO v8 model implementation, tuning	Slides 15-17	Conducted live demo and conclusions
Dheeraj Pamnani	Model validation - outcome scenarios; wait time calculation	Slides 12-14	Walked through monitoring plan and Computer Vision model architecture
Dominic Darrah	Data collection, cleaning, annotation strategy, privacy considerations	Slides 6-8	Presented data acquisition & preprocessing steps
Riya Agarwal	Problem scope identification, value and impact analysis	Slides 2-5	Opened the presentation and discussed impact & value add
Sravani Bolla	Model validation, pre/post deployment metrics; proposed resource plan	Slides 9-11	Explained validation process & deployment pipeline

Each member played an active role in the early-stage discussions, from identifying the core problem to ideating the project solution.

References

- https://www.ultralytics.com/blog/revolutionizing-queue-management-with-ultralytics-yolov8-and-openvino
- https://github.com/ultralytics/ultralytics
- Blackwells Capital Outlines Plans for AI at Disney Theme Parks (Crowd Management, Predictive Maintenance, Real-Time Ticket Price Changes)
- CrowdVision - Automated passenger tracking using video analytics
- YOLO11 Real-time Queue Management | Ultralytics
- AI LLM to assist in syntax

THANK YOU!