Project Proposal Report

# Smart BusCounter AI: Automatic Bus Counting Using Computer Vision

**Propose by: Chantharith Ny** (Computer Vision and AI Club)

## I. Background and Motivation

Public transportation systems, especially in developing countries like Cambodia, often lack real-time data on bus occupancy. This makes it difficult for operators to:

- Optimize routes and fleet allocation
- Monitor overcrowding
- Improve passenger experience

While some modern systems use ticket scans or sensors, these solutions are expensive and not easily scalable. Using **affordable front and back cameras** with AI-based counting can provide an accurate, low-cost, and scalable alternative.

## II. Problem Statement

- Buses often lack real-time passenger counting systems.
- Manual counting is inefficient and error-prone.
- Many buses are equipped only with basic surveillance cameras.
- There is no AI system in place that can utilize these camera feeds to count passengers boarding and exiting in real time.

## III. Objectives

- Develop a computer vision system to count passengers from front and back bus cameras.

- Use object detection and tracking models to identify entry and exit events.
- Aggregate counts from both cameras to provide an accurate real-time passenger number.
- Deploy the model to work in real-time or near-real-time with limited hardware resources.

# IV. Scope of Work

| No. | Task | Description |
|---|---|---|
| 1. | Literature Review | Explore methods for object detection and people counting using computer vision. |
| 2. | Data Collection | Collect bus footage from front and back cameras. |
| 3. | Data Annotation | Label passenger movements for supervised learning. |
| 4. | Model Development | Train and fine-tune object detection/tracking models. |
| 5. | Integration | Combine both camera streams to compute passenger count. |
| 6. | App Development | Develop an interface/dashboard to show real-time count. |
| 7. | Testing | Test system accuracy in different conditions (crowded, low light, etc.). |
| 8. | Deployment | Deploy on edge devices or local servers in buses. |

# V. Methodology

| Step | Description | Tools/Tech Used |
|---|---|---|
| 1. Data Collection | Gather real footage from buses. | Bus-mounted cameras |

| | | |
|---|---|---|
| 2. Annotation | Use tools to label passenger entries/exits. | CVAT, LabelImg |
| 3. Object Detection | Detect passengers in frames. | YOLOv8, EfficientDet |
| 4. Tracking | Track individuals across frames and views. | DeepSORT, ByteTrack |
| 5. Movement Analysis | Classify movement as entry or exit. | Optical flow, custom logic |
| 6. Camera Synchronization | Align front and back camera timestamps. | Video timestamp analysis |
| 7. Count Aggregation | Calculate net passenger count. | Custom counting algorithm |
| 8. Interface | Display count live or export data. | Streamlit, React, Flask |
| 9. Deployment | Run on local devices or edge servers. | Raspberry Pi, Jetson Nano |

# VI. Expected Outcomes

- A dual-camera AI system that accurately counts passengers boarding and exiting the bus.
- A dashboard or API that displays live passenger count.
- Reduced cost compared to commercial counting systems.
- High scalability for public transport in Cambodia.

# VII. Significance of The Project

- Supports intelligent transportation systems.
- Provides real-time data for public transport operators.
- Helps optimize route management and improve passenger safety.
- Enables smart city initiatives without high hardware costs.

# VIII. Budget Estimate

| Item | Description | Estimated Cost (USD) |
|---|---|---|
| Edge Device | For onboard AI processing (e.g., Jetson Nano) | $100 – $200 |
| Camera Equipment | Front and back bus cameras | Already installed or $40–$80 |
| Cloud/Server (optional) | For model updates or data sync | $10 – $30/month |
| Annotation Tools | Open-source | Free |
| ML Libraries | PyTorch, YOLOv8 | Free |
| **Total** | – | **$150 – $250** one-time, minimal monthly cost |

# IX. Challenges and Risk Management

| Challenge | Description | Mitigation |
|---|---|---|
| Occlusion | People overlap in crowded conditions | Use tracking + multiple camera views |
| Lighting Issues | Nighttime or poor lighting affects accuracy | Use IR-compatible cameras or preprocessing |
| Camera Sync | Front and back video misaligned | Timestamp-based sync or hardware sync setup |
| Model Generalization | Different buses, angles, and movement patterns | Train on diverse datasets, fine-tune per environment |
| Privacy Concerns | Capturing faces or identities | Blur faces, comply with data protection laws |

# X. Future Work & Expansion

- Add face-blur for privacy protection.
- Expand to multi-bus systems with central tracking.
- Integrate with GPS for location-aware analysis.
- Build predictive models for route demand forecasting.
- Collaborate with city authorities for broader smart transport systems.

# XI. References & Learning Resources

### Papers:


1. "YOLOv8 for Real-Time Object Detection" – Ultralytics
2. "DeepSORT: Simple Online and Realtime Tracking with a Deep Association Metric"


### Tools:

- YOLOv8, OpenCV, DeepSORT, PyTorch
- LabelImg, CVAT

### Public Datasets:

- MOTChallenge, CityPersons, CrowdHuman

### Learning:

- PapersWithCode
- YouTube: Roboflow, Computer Vision Zone
- Fast.ai / Coursera AI courses

# XII. Conclusion

This project proposes an AI-powered system to count passengers using dual-camera setups in buses. By leveraging real-time computer vision, the system offers an efficient and scalable alternative to traditional counting methods. With wide applicability in Cambodia’s growing transport sector, it represents a practical and impactful use of AI in public service.