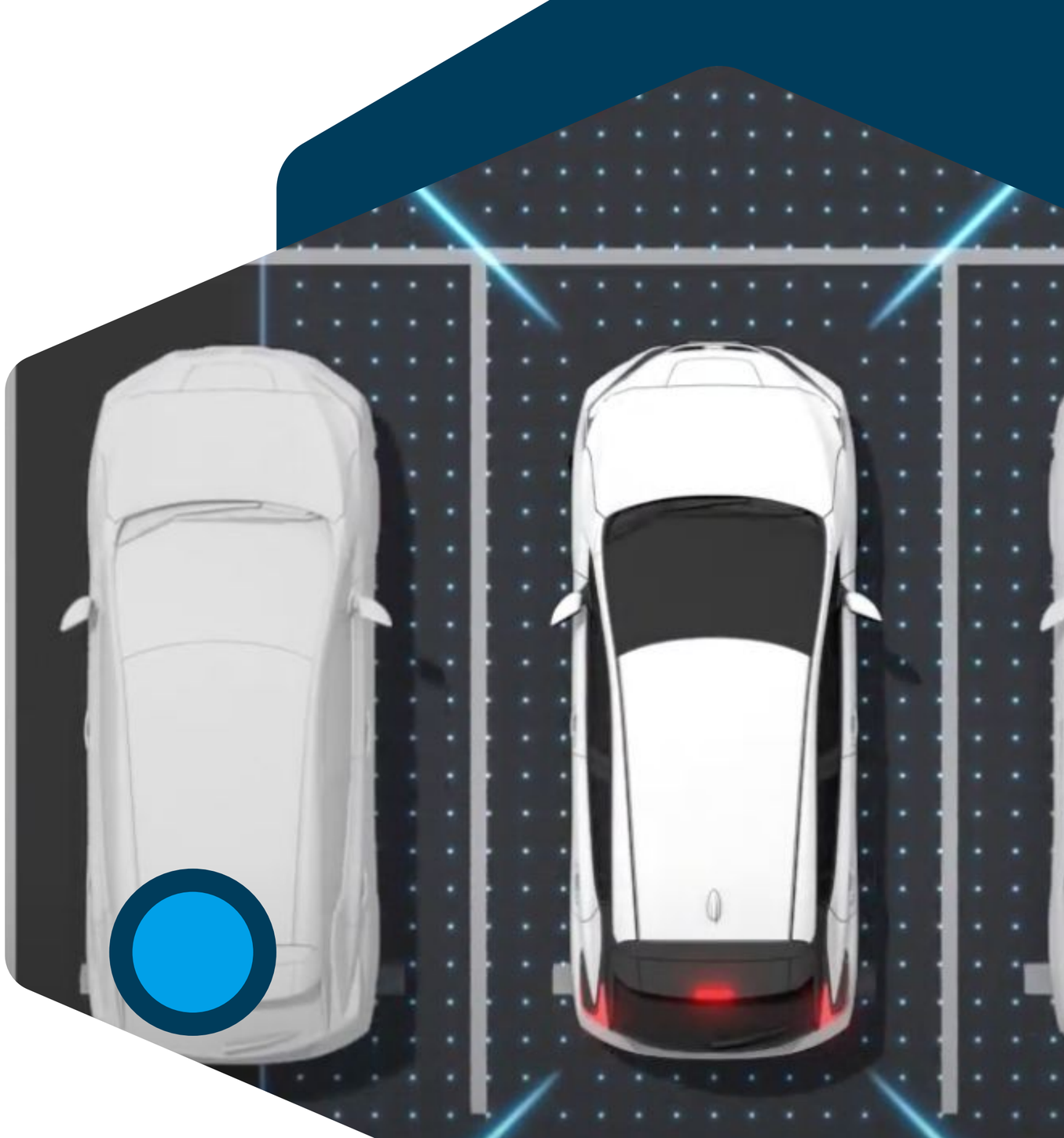
# PARKING SPOT DETECTION

## USING YOLOv8 and VGG16

→ PRESENTATION



# Introduction

- Problem Statement

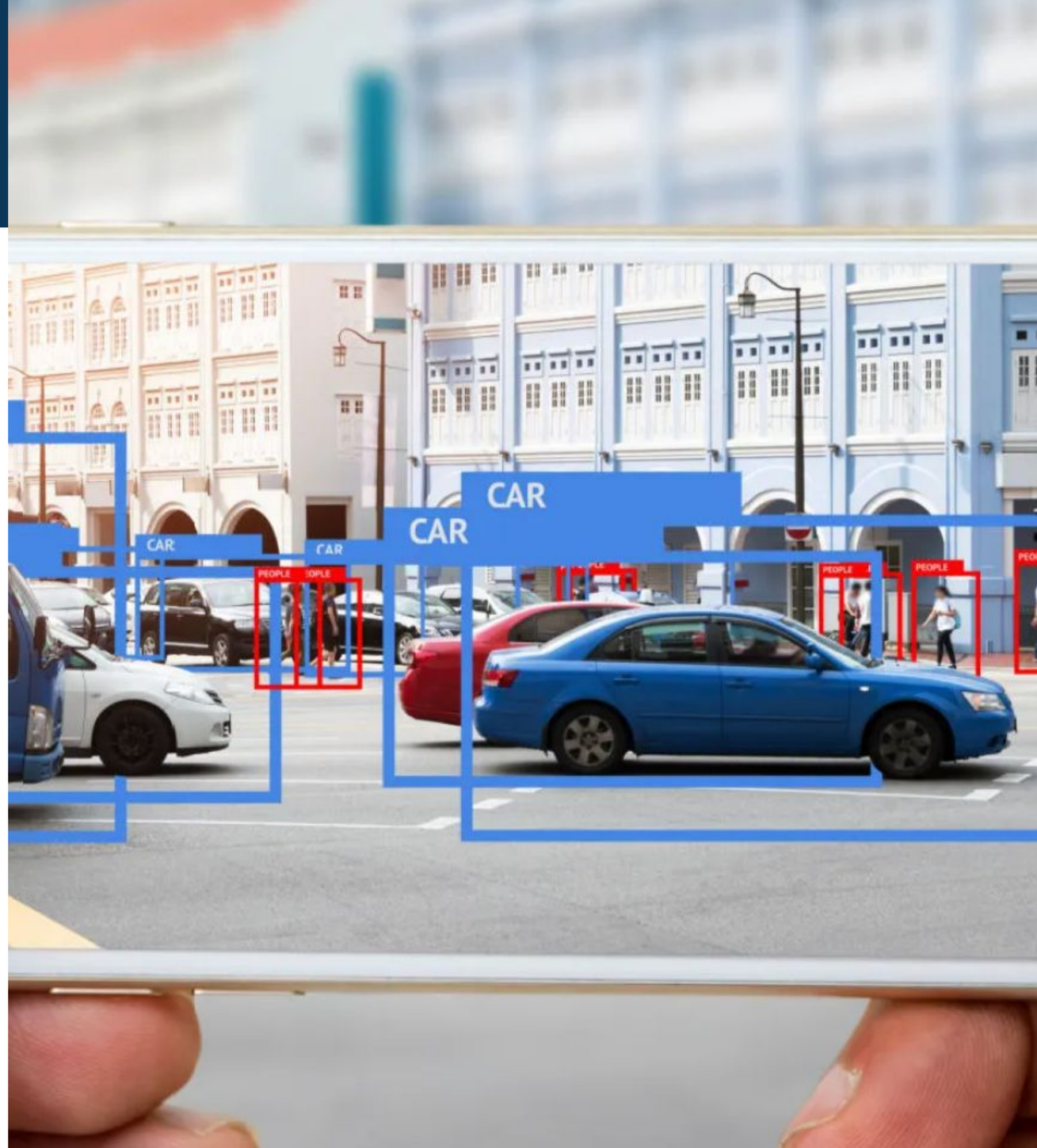
  *Finding available parking spots efficiently is a widespread urban problem.*

- Why This Problem

  *Manual search wastes time, increases emissions, causes congestion*

- Research Done

  *Literature on parking systems, computer vision-based parking detection, traditional IoT sensor-based systems.*



# Overall End-to-End Solution

1

## Scope of Solution

- Focused on detecting open parking spots from images using computer vision.

2

## Key Stakeholders

- Parking management companies, city planners, drivers.

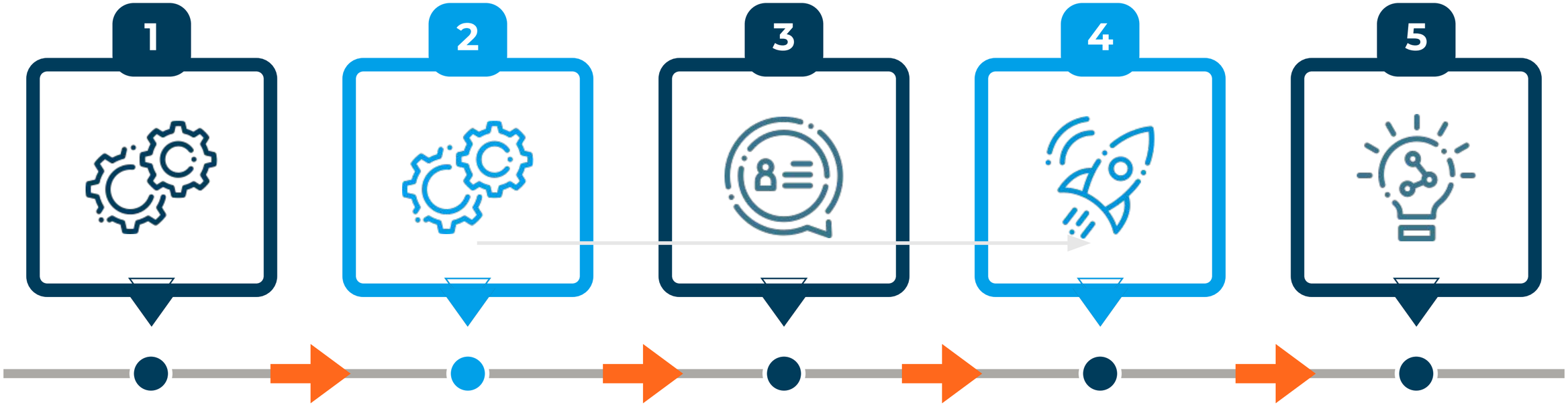
3

## Current Status-Quo

- Manual monitoring
- IoT-based sensors (expensive, maintenance issues)

# EXECUTION

## ROADMAP



# VALUE, COSTS, AND SUCCESS METRICS

**Valu**
**e**

Fa ter parking, less traffic, improved experience

**Success**
**Metrics**

Accu ~85%, Speed (processing under 5s/image)

**Cost-Benefit Tradeoff**

Cheaper than IoT sensors, scalable with existing cameras

**Cost-Benefit Tradeoff**

Google Colab GPU, OpenCV, Ultralytics YOLOv8, TensorFlow.

Workflow Updates

Validation and Monitoring

People

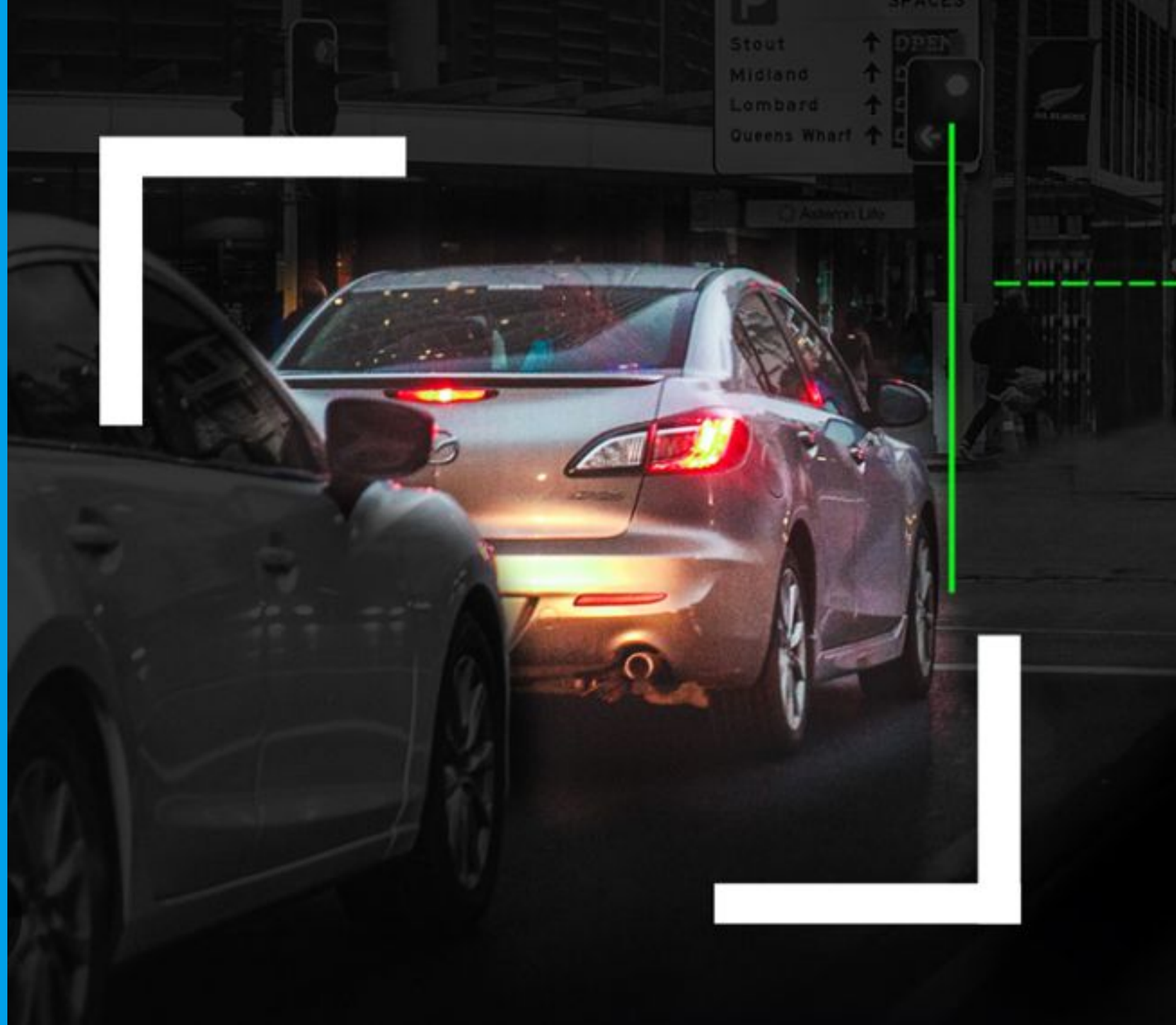
Process

Technology

Pre deployment

Pre deployment

# CV Model Section



# CV MODEL ARCHITECTURE

## Object Detection (YOLOv8)

- Purpose: Detect and crop cars from parking lot images.
- Pre-trained model used: YOLOv8n (nano version for speed).

## Classification (VGG16)

- Purpose: Classify cropped spots as Open or Closed.
- Modified VGG16: Added GAP + Dense Layers.

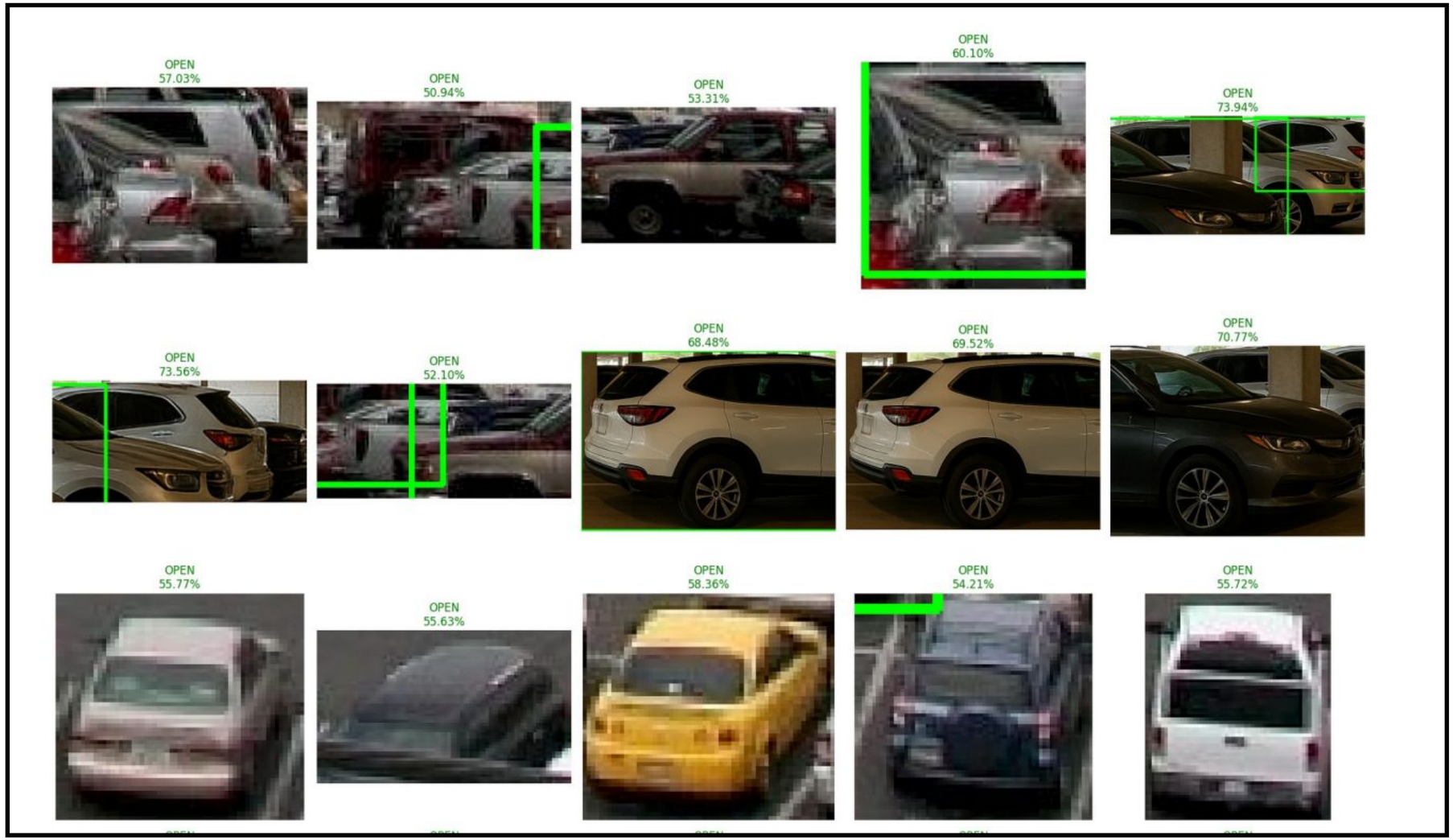
## CV Model Evaluation (Proof-of-Concept)

- Training Accuracy: 96%
- Validation Accuracy: ~60%
- Issues: Overfitting (small dataset).

# OUTCOME-ACTION PAIRINGS AND COST ANALYSIS

| Outcome | Model Prediction | Real Situation | Action Taken | Cost Impact |
|---|---|---|---|---|
| **TP** | Open | Open | Display spot to driver | Positive |
| **TN** | Closed | Closed | Hide spot from driver | Positive |
| **FP** | Open | Closed | Mislead driver to occupied spot | High (driver frustration, system distrust) |
| **FN** | Closed | Open | Hide a usable spot | Medium (opportunity loss) |

# OPEN AND CLOSED IDENTIFICATION




## Insights

**To minimize False Positives (FP)**

- **Set a higher prediction threshold (e.g., 0.7 instead of 0.5).**

**Tradeoff**

- **Fewer available spots detected, but more trustworthy.**

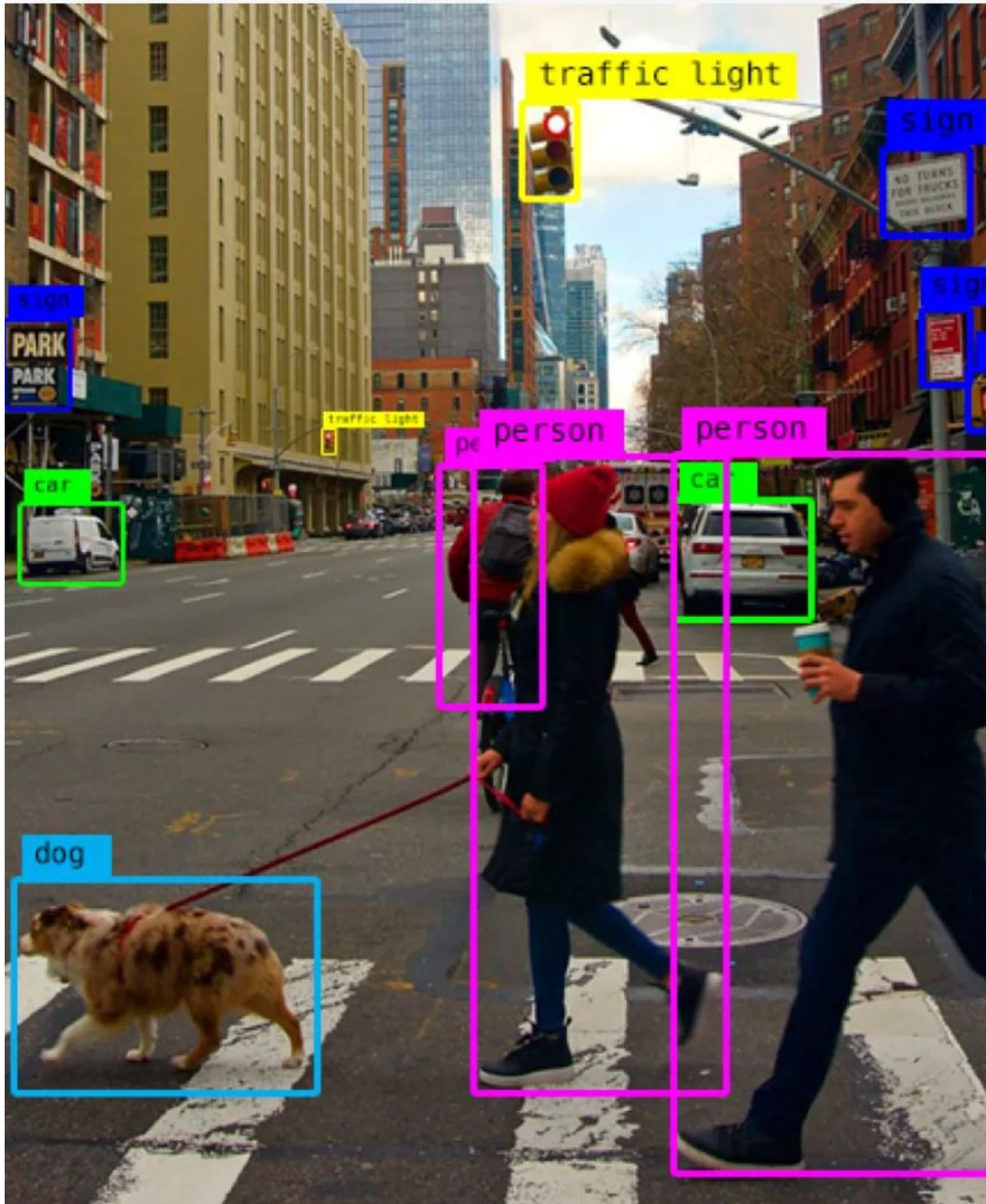
# BIASES AND LIMITATIONS

1 Unintended Consequences

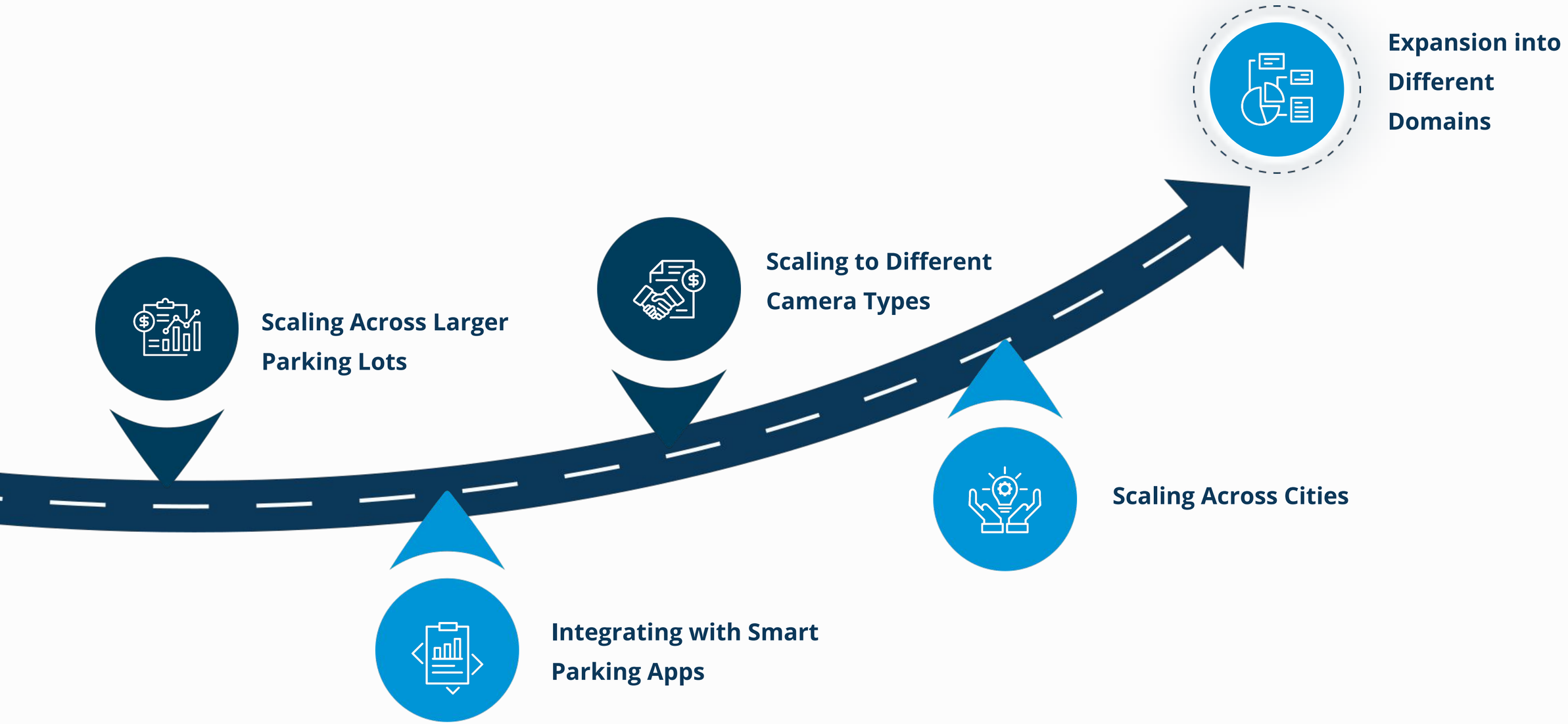
2 Privacy/Security

3 Bias

4 Augment dataset with more diverse samples.



# SCALING POTENTIAL



# TARGET SEGMENT

**Scaling Potential**

- Deploy to large parking lots
- Integrate into smart apps
- Use different camera feeds
- Expand to new cities

**Lessons Learned**

- Small datasets = Overfitting risk
- Need diverse data (lighting, weather)
- Open-source models need tuning
- Think about deployment early

# SUMMARY

| Team Member | Tasks Performed |
|---|---|
| Anwesha Banerjee | Content for PPT and Insights from data |
| Sagar Prasad | Code-Detection & Cropping |
| Ryan Karim | Proposal |
| Kevin M | Code-Classification |
| Zifan Wang | Dataset Research |

# REFERENCES


- YOLOv8: Ultralytics
- VGG16: Keras Applications
- TensorFlow documentation
- Githubs Links of dataset

# THANK YOU