Smart Eyes on the Yellow Line

Smart Eyes on the Yellow Line

Automating compliance, enhancing safety.

FACILITATOR: Mr. M Surya Rao

GUIDED BY: Mr. Kumar Sourav & Mr. Aritra De

TEAM MEMBERS: Sania Zia, Amritpal Singh, Aaditya Prabhakar, Ankit Kumar, Akash Kumar Singh, Manish Patel.

May 6

PROBLEM STATEMENT AND SOLUTION STRATEGY

THE PROBLEM:

Currently, JSP township junctions have STOP signages to ensure vehicles pause and turn safely. However, there's no automated system to verify compliance. Cameras are installed, but monitoring is entirely manual.

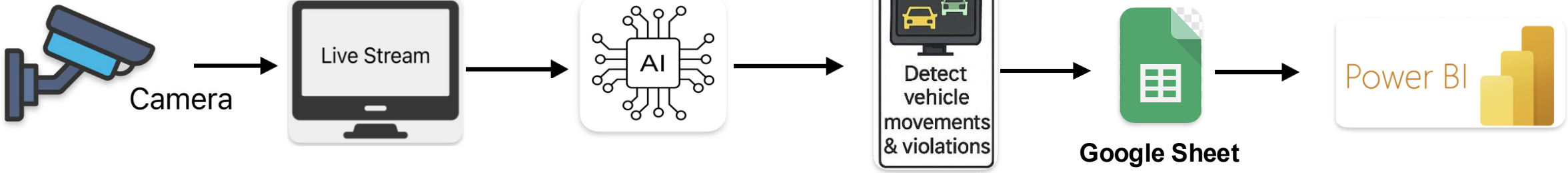
OBJECTIVE:

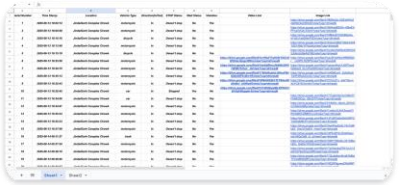
Using AI vision, we aim to automate violation detection through existing cameras and send real-time alerts via WhatsApp. This ensures 24/7 compliance and enhances safety by preventing incidents.

SOLUTION STRATEGY:

We use a YOLO-based deep learning model to detect and track vehicles within a defined ROI. It checks if each vehicle stops for 3 seconds and labels the result as a violation or non-violation. All details, along with images and 10-second clips, are logged to a Google Sheet.

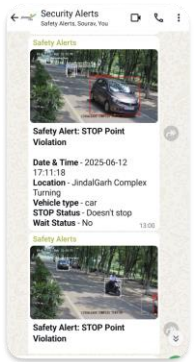
Smart Detection & Alert Workflow Schema





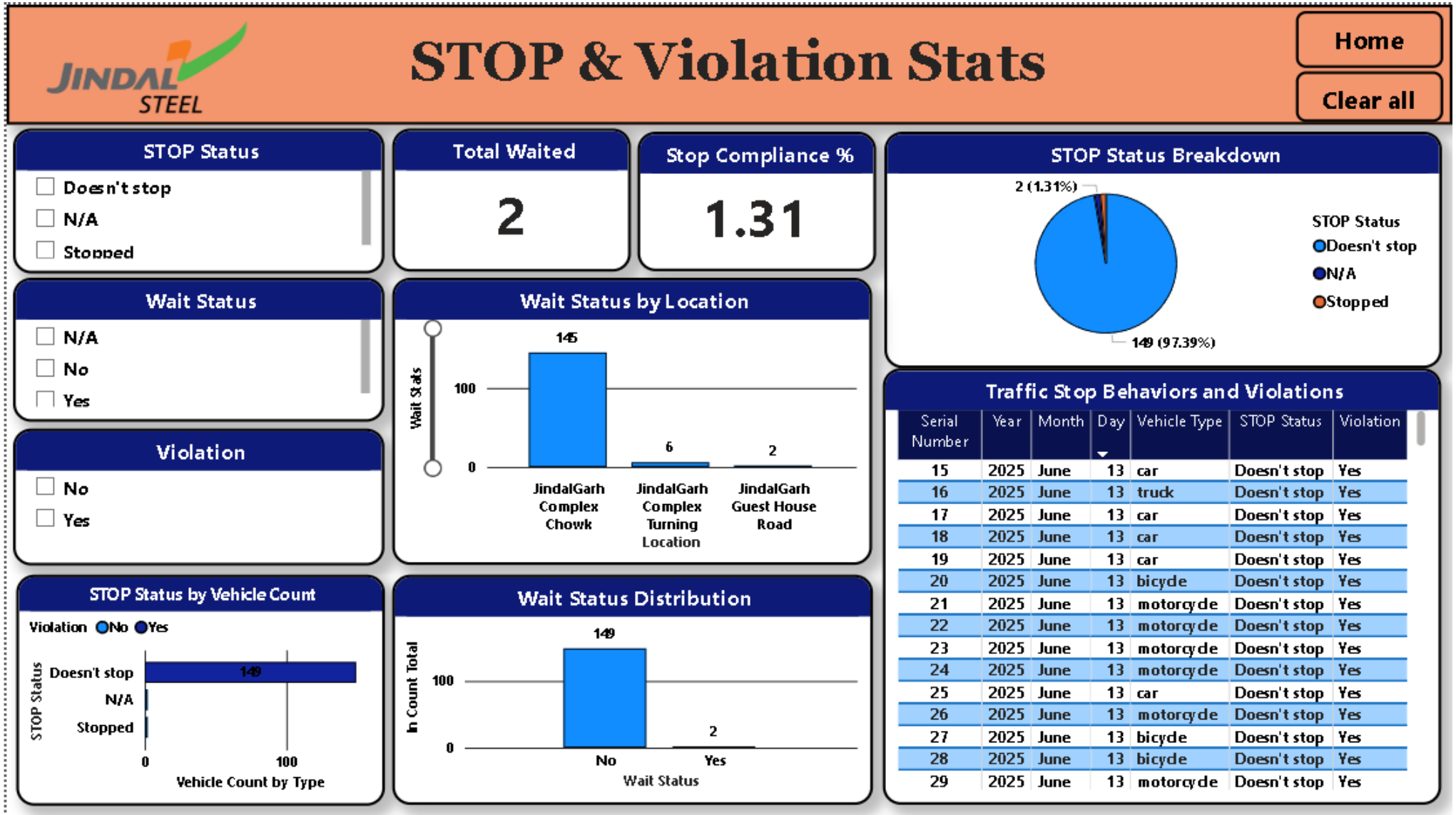
Google Script generated template message

Alert on WhatsApp



Live Detection in Action

Actionable Alerts & Dynamic Insights



IMPLEMENTATION

•Capture real-time camera feed and define ROI

•Detect and classify vehicles with white boxes.

•Track closest point near yellow line and movement direction (IN/OUT)

•For IN: Green box if stopped ≥ 3 sec, Red if < 3 sec.

•On violation (red boundary), save image + 10-sec video and log to Google Sheet.

•Send alerts via WhatsApp; Power BI visualizes traffic and violations

BENEFITS

- Ensure full compliance and zero incidents at junctions.
- Real-time detection with WhatsApp alerts.
- Achieved in-house for <10% of vendor cost
- Auto-captured images/videos as violation proof.
- Scalable across all JSP cameras and locations.

TECHNOLOGY USED

- VS Code
- Python Script
- Google Colab
- Google Apps Script
- Google Sheet
- WhatsApp
- Power BI Dashboard

Challenges, Learnings & Way Forward

Challenges

- High server requirements for real-time video processing
- Cameras placed too high – number plates unreadable
- PTZ cameras frequently rotated, disrupting detection
- No dedicated VA platform – models run on Windows terminals

Learnings

- Built real-time logic for continuous detection
- Integrated Python with Google tools, Power BI, and WhatsApp
- Automated WhatsApp alerts via desktop scripting
- Optimized system resources across modules

Way Forward

- Develop AI for fire, intrusion & no-parking detection
- Standardize WhatsApp alerts via databases for multi-usecase handling
- Shift to a unified, high-performance VA deployment platform

THANK YOU