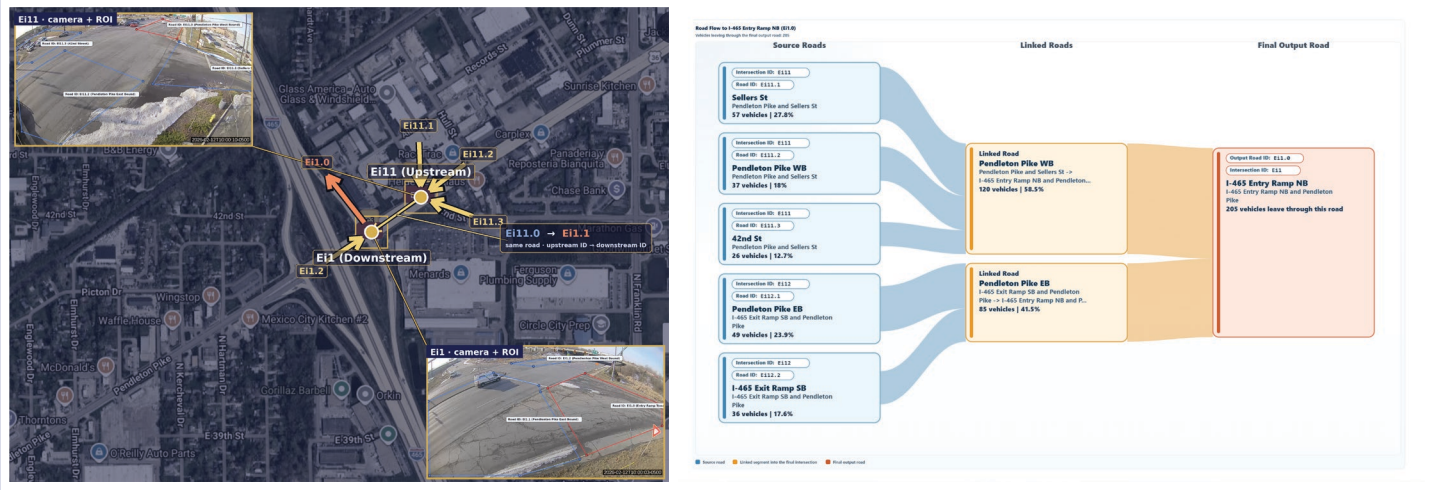
JOINT TRANSPORTATION
RESEARCH PROGRAM
INDIANA DEPARTMENT OF TRANSPORTATION
AND PURDUE UNIVERSITY

Vehicle Moving Pattern Analysis of Road Merge and Split Weaving Areas Involving Highway and Local Streets



Stanley Yung-Ping Chien, Yaobin Chen, Sajan Kumar, and Shail Desai

SPR-5019 • Report Number: FHWA/IN/JTRP-2026/27 • DOI: 10.5703/1288284318711

RECOMMENDED CITATION

Chien, S. Y.-P., Chen, Y., Kumar, S., & Desai, S. (2026). *Vehicle moving pattern analysis of road merge and split weaving areas involving highway and local streets* (Joint Transportation Research Program Publication No. FHWA/IN/JTRP-2026/27). West Lafayette, IN: Purdue University. https://doi.org/10.5703/1288284318711

AUTHORS

Stanley Yung-Ping Chien
Professor of Electrical and Computer Engineering
Elmore Family School of Electrical and Computer Engineering
Purdue University
(765) 495-7689
yschien@purdue.edu
Corresponding Author

Yaobin Chen
Professor of Electrical and Computer Engineering
Director of Transportation & Autonomous Systems Institute
Elmore Family School of Electrical and Computer Engineering
Purdue University

Sajan Kuma
Elmore Family School of Electrical and Computer Engineering
Purdue University

Shail Desai
Elmore Family School of Electrical and Computer Engineering
Purdue University

ACKNOWLEDGMENTS

This work was supported by the Joint Transportation Research Program (JTRP) SPR-5019, administered by INDOT and Purdue University. The authors would like to thank all members of the INDOT Study Advisory Committee (SAC), including Business Owner Nathan Shellhamer, Project Advisor Shuo Li, Dan McCoy, Jack Gallagher, Randy Myers, Mark Muenz, Mark Joseph, and Michelle Witkowski, for their guidance, advice, and technical support throughout the project period. Special thanks go to Dr. Darcy Bullock, Director of JTRP, and his staff at Purdue University for their administrative support and service.

JOINT TRANSPORTATION RESEARCH PROGRAM

The Joint Transportation Research Program serves as a vehicle for INDOT collaboration with higher education institutions and industry in Indiana to facilitate innovation that results in continuous improvement in the planning, design, construction, operation, management and economic efficiency of the Indiana transportation infrastructure. Learn more at engineering.purdue.edu/JTRP.

Published reports of the Joint Transportation Research Program are available at docs.lib.purdue.edu/jtrp/.

NOTICE

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views and policies of the Indiana Department of Transportation or the Federal Highway Administration. The report does not constitute a standard, specification, or regulation.

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No. FHWA/IN/JTRP-2026/27	**2. Government Accession No.**	**3. Recipient's Catalog No.**
4. Title and Subtitle Vehicle moving pattern analysis of road merge and split weaving areas involving highway and local streets		**5. Report Date** October 13, 2026 **6. Performing Organization Code**
7. Author(s) Stanley Chien (https://orcid.org/0000-0002-1492-9959) Yaobin Chen (https://orcid.org/0000-0002-9875-4083) Sajan Kuma Shail Desai		**8. Performing Organization Report No.** FHWA/IN/JTRP-2026/27
9. Performing Organization Name and Address Joint Transportation Research Program Hall for Discovery and Learning Research (DLR), Suite 204 207 S. Martin Jischke Drive West Lafayette, IN 47907		**10. Work Unit No.** **11. Contract or Grant No.** SPR-5019
12. Sponsoring Agency Name and Address Indiana Department of Transportation (SPR) State Office Building 100 North Senate Avenue Indianapolis, IN 46204		**13. Type of Report and Period Covered** Final Report **14. Sponsoring Agency Code**

15. Supplementary Notes

Conducted in cooperation with the U.S. Department of Transportation, Federal Highway Administration.

16. Abstract

The origin–destination (OD) proportions of vehicles related to highway weaving between entry ramp and next exit ramps are difficult to measure with conventional point detectors. Manual counting from videos is labor intensive and does not scale. This project developed an automated, camera-based system that measures OD proportions directly from existing interchange surveillance video, allowing the Indiana Department of Transportation (INDOT) to quantify the proportion of vehicles from local streets to highway ramps, and vice versa.

The system reconstructs each vehicle's full path through intersections in two stages. First, it detects and tracks every vehicle within each camera's field of view using a deep-learning object detector coupled with a multiobject tracker and frame stabilization. Second, it reidentifies the same vehicle traveling from upstream to downstream intersections in their corresponding recorded camera videos by comparing learned appearance embeddings, constrained by realistic vehicle travel-time windows and travel direction, so that only plausible matches are considered. Because automated cross-camera matching can produce false matches between similar-looking vehicles, the system incorporates a web-based, human-in-the-loop verification step in which an analyst confirms or rejects candidate matches before results are finalized.

From the verified matches, the system computes OD proportions and presents them in a user-friendly report with Sankey diagrams that visualize the traffic flow for all vehicles. A companion ground-truth annotation tool lets analysts generate manual counts and matches that are used to measure system accuracy. The results show that the approach reconstructs weaving OD proportions with accuracy suitable for planning and safety analysis while substantially reducing manual effort.

17. Key Words highway weaving analysis, origin–destination proportion, vehicle detection and tracking, multiobject tracking, cross-camera vehicle re-identification, computer vision, traffic surveillance video, human-in-the-loop machine learning, Sankey diagram	**18. Distribution Statement** No restrictions. This document is available through the National Technical Information Service, Springfield, VA 22161.

19. Security Classif. (of this report) Unclassified	**20. Security Classif. (of this page)** Unclassified	**21. No. of Pages** 38, including appendices	**22. Price**

EXECUTIVE SUMMARY

The Origin-Destination (OD) Proportion Analysis System provides a web-based and machine-learning-supported workflow for estimating how vehicles move through connected intersections under Entry/Merge and Exit/Split traffic conditions. The system addresses the Indiana Department of Transportation's (INDOT) needs for a practical method to move beyond isolated intersection counts and estimate vehicle movement proportions across roadway branches using traffic video data.

The research problem is challenging because adjacent traffic cameras often lack overlapping fields of view. As a result, the system must combine user-defined roadway structure, region-of-interest (ROI) annotations, machine-learning-based vehicle detection and tracking, cross-camera vehicle matching, and human verification to produce reliable OD movement counts and OD proportions.

This technical report presents the system as an integrated research and deployment workflow. The web-based workflow guides users through scenario creation, intersection labeling, video upload, ROI annotation, processing review, verification, and report generation. The Machine Learning Framework describes the analytical method used to detect, track, classify, match, and aggregate vehicle movements. The user-entered scenario data model defines the structured JSON contract that connects the web application, backend services, shared file server, and machine-learning program.

The backend and deployment architecture connects the React user interface, Python FastAPI backend services, NGINX routing, NAS-based scenario storage, and GPU-enabled machine-learning processing environment. Together, these components provide a reproducible, file-based workflow that supports scenario traceability, automated processing, human-in-the-loop validation, and report-ready analytical outputs.

The system is implemented as a web-based, unique tool that provides INDOT with a practical way to solve the OD-Proportion problem.

CONTENTS

1. INTRODUCTION . . . 8
2. RELATED WORK . . . 8
   2.1 Highway Weaving and Arterial OD Vision Systems . . . 9
   2.2 Cross-Camera Vehicle Re-Identification . . . 9
   2.3 Timer Filters for ReID . . . 9
3. HUMAN IN THE LOOP MACHINE LEARNING FRAMEWORK . . . 9
   3.1 Merge and Split OD-Proportion Scenarios . . . 9
   3.2 Hierarchical Road Network Representation and ROI Definition . . . 9
   3.3 Machine Learning Processing Framework . . . 11
4. USER PROVIDES THE OD-PROPORTION SCENARIOS . . . 11
5. MACHINE LEARNING BASED AND HUMAN IN THE LOOP OD-PROPORTION ANALYSIS . . . 11
   5.1 Vehicle Detection and Fine-Grained Classification . . . 11
   5.2 Ground Truth Annotation . . . 12
   5.3 Within-Intersection Vehicle Tracking . . . 12
   5.4 Cross-Camera Vehicle Matching . . . 12
      5.4.1 Encoder Configuration and Feature Representation . . . 12
      5.4.2 Travel-Time Filtering for Candidate Pairs . . . 12
      5.4.3 Assignment-Based Matching and Match Selection . . . 13
   5.5 Human-in-the-Loop Matched Vehicle Pair Verification . . . 13
   5.6 OD-Proportion Estimation . . . 13
   5.7 OD Proportion Analysis Result and Report . . . 14
6. BACKEND ARCHITECTURE OF THE OD PROPORTION ANALYSIS SYSTEM . . . 14
7. EXPERIMENT RESULTS . . . 14
   7.1 Deployment Sites . . . 14
   7.2 Detection and Within-Intersection Tracking Performance . . . 15
      7.2.1 Vehicle Detection and Classification . . . 15
      7.2.2 Within-Intersection Tracking . . . 15
   7.3 Cross-Camera Vehicle Matching Performance . . . 15
      7.3.1 Three-Encoder Ensemble . . . 15
      7.3.2 Per-Site Performance . . . 15
      7.3.3 Effect of Resolution and Time of Day . . . 15
      7.3.4 Contribution of the Travel-Time Filter . . . 15
   7.4 Human-in-the-Loop Verification . . . 16
   7.5 Final Overall Results and Evaluation . . . 16
      7.5.1 Within-Intersection Turning-Movement Count Accuracy . . . 16
      7.5.2 End-to-End Precision . . . 16
   7.6 Summary . . . 17
8. CONCLUSION . . . 17
REFERENCES . . . 17
APPENDICES . . . 19
   Appendix A. Example OD Analysis Report . . . 19
   Appendix B. The OD-Proportion Analysis Backend System . . . 19

LIST OF TABLES

Table 7.1 Deployment Sites. 14

Table 7.2 Within-Intersection Tracking Accuracy Analysis Over Separate Deployment Sites (720p Only). 15

Table 7.3 Encoder Comparison (Pooled Pendleton Evaluation, 786 Pairs). 15

Table 7.4 Cross-Site Matching Performance, Three-Way Ensemble. 15

Table 7.5 Time of the Day Matching Performance for the Pendleton Pike Scenario. 16

Table 7.6 Matching F1 Score by Filtering Configuration. 16

Table 7.7 Effect of Human-in-the-Loop Verification on End-to-End Precision. 16

Table 7.8 OD Count Accuracy (Predicted vs. Ground-Truth Vehicle Counts). 16

Table 7.9 Origin–Destination Flow Shares at Pendleton Pike (EI11). 16

Table 7.10 Origin–Destination Flow Shares at Pendleton Pike (EI1). 16

Table 7.11 End-to-End Precision (720p). 17

LIST OF FIGURES

Figure 1.1 Origin-Destination Problem. 8

Figure 1.2 Actual Roadway Network for OD Proportion Analysis. 8

Figure 3.1 Split Example for OD Proportion Analysis. 9

Figure 3.2 Tree Diagram (Labels with E = Merge, X = Split). 10

Figure 3.3 ROI-Based Road Annotation for Vehicle Tracking. 10

Figure 3.4 Machine Learning Framework From CCTV Videos to OD-Proportion Outputs. 11

Figure 5.1 Cost Matrix and Assignment-Based Vehicle Matching 13

1. INTRODUCTION

Origin–Destination (OD) proportions in a multi-intersection network describe how vehicles entering the area from a given road (origin) are distributed among the available exit roads (destinations) through a series of interconnected intersections, and vice versa. Knowledge of OD proportions is fundamental to traffic engineering practice and supports applications such as signal timing optimization, capacity analysis, incident management, and long-term infrastructure planning to improve or expand transportation systems. Transportation agencies need reliable methods to understand how vehicles move through connected intersections, especially when traffic enters or exits an interested road through multiple surrounding roads. This information supports:

- Traffic operations and congestion analysis
- Signal timing evaluation
- Capacity and safety planning
- Long-term roadway improvement decisions

Conventional approaches to estimating OD proportions suffer from well-known limitations. Household travel surveys (Stopher & Greaves, 2007) are costly and conducted infrequently, limiting their temporal relevance. License plate reader systems (Chiou et al., 2012) require extensive hardware deployment across multiple locations and raise privacy concerns. Inductive loop detectors and roadside radar systems (Federal Highway Administration, 2016) capture traffic volumes but provide little insight into the routing decisions vehicles make at intersections. Vision-based approaches are promising, but some multicamera vehicle-tracking systems use overlapping views of the same roadway area to link vehicles across cameras (Luna et al., 2022; Zhang et al., 2024). Our setting is different: each intersection is observed by one camera, and cameras at adjacent intersections do not share a field of view. Other approaches use GPS-tagged taxi trajectories (Yuan et al., 2010) or mobile phone data (Iqbal et al., 2014), which may not be available for routine OD analysis.

This study addresses the OD proportion problem in a network of connected intersections centered on a designated road (called the focal road). The objective is to determine how many vehicles exiting (or entering) the network via the focal road originally entered (or exited) through each of the other connecting roads (Figure 1.1). For generality, we assume that the available data consists of only one video stream from a single closed-circuit television camera (CCTV) installed at each intersection, with no overlapping videos between the video obtained in adjacent road intersections. This setup reflects realistic deployment conditions, as municipalities often already maintain such camera infrastructure for traffic monitoring and incident detection, or can deploy temporary cameras at selected intersections when needed.

Figure 1.2 shows a merge example of the actual roadway network used for OD-proportion analysis. The figure illustrates that the problem is not limited to counting vehicles at a single intersection; rather, the objective is to estimate how vehicles move between connected intersections and how traffic is distributed across different roadway branches.

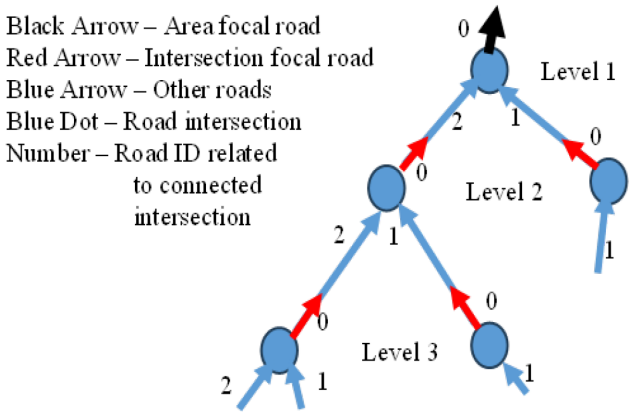


Figure 1.1 Origin-Destination Problem.



Figure 1.2 Actual Roadway Network for OD Proportion Analysis.

The main contribution of this project is the development of a scalable framework that decomposes the OD-proportion problem into five key subproblems: (1) user interface for problem description, (2) vehicle detection and tracking within individual intersections, (3) vehicle matching across adjacent intersections without overlapping camera views, (4) human-in-the-loop to eliminate false results, and (5) integration of these results to generate OD-proportions. Together, human-in-the-loop and machine learning techniques provide an end-to-end solution for OD estimation under realistic data and deployment constraints.

2. RELATED WORK

Three lines of prior work were briefly reviewed related to the proposed framework: highway weaving and arterial OD vision systems, cross-camera vehicle re-identification (ReID), and timer filters for ReID.

2.1 Highway Weaving and Arterial OD Vision Systems

W. Lin et al. (2024) introduced a vision-based pipeline for highway weaving sections that uses cross-camera vehicle matching to recover lane-change OD. Qiu, Lin, Chien, et al. (2024) extended this idea to a highway segment between adjacent entry and exit ramps with a single time-window filter. Related work from the same group uses existing highway camera videos for lane-wise traffic monitoring, with automatic adaptation to changes in camera angle and zoom (Qiu, Lin, Christopher, et al., 2024).

2.2 Cross-Camera Vehicle Re-Identification

Vehicle ReID has been driven primarily by the VeRi-776 benchmark (Liu et al., 2016), the CityFlow multicamera tracking corpus (Tang et al., 2019), and the annual AI City Challenge series (Wang et al., 2024), which have standardized many of the architectures referenced in this report. FastReID (L. He et al., 2020) systematizes the ResNet-IBN-A backbone family; TransReID (S. He et al., 2021) adds patch-shuffle and side-information embeddings for camera-aware retrieval; k-reciprocal reranking (Zhong et al. 2017) is a widely used postprocessing step. More recent work explores large pretrained vision-language models, such as CLIP (Radford et al., 2021), as zero-shot retrieval backbones. Our ReID approach is motivated by the observation that no single ReID encoder provides a dominant feature vector for vehicle matching.

2.3 Timer Filters for ReID

The closest prior work on travel-time-based filtering for cross-camera matching is a class of methods called camera-link models. These methods learn the typical travel time between each camera pair from the data and reject candidate matches whose travel times do not fit. Hsu et al. (2020) learn the travel-time rule from labeled trajectories; Huang et al. (2023) use a transformer; Y. Lin et al. (2024) learn it without labels. All three assign a single rule for travel time determination per camera pair, which lumps together left-turn, straight-through, and right-turn vehicles that share those two cameras. Instead, for this study, a time-varying window is dynamically learned to capture changes in travel time at signalized arterials: vehicles that catch a green light and vehicles that wait through a red phase (downstream traffic queues that affect travel time).

3. HUMAN IN THE LOOP MACHINE LEARNING FRAMEWORK

This report describes a framework and process to solve the OD-Proportion Analysis Problem. The framework decomposes the analysis into five steps:

(1) Define a systematic hierarchy to describe the intersections and roads in an area for OD-Proportion analysis and provide a web-based user interface for the user to specify all necessary information for OD-Proportion analysis.



Figure 3.1 Split Example for OD Proportion Analysis.

(2) Apply various machine-learning models to detect and track vehicles within each intersection.
(3) Apply machine learning methods to match vehicles between adjacent intersections.
(4) Human-in-the-loop to remove the false positive vehicle matches.
(5) Integrate all the model outputs to generate final OD-proportion results.

3.1 Merge and Split OD-Proportion Scenarios

As shown in Figure 1.2 and Figure 3.1, respectively, the OD Proportion Analysis problem can occur in both vehicles merging from multiple roads to an area focal road and vehicles splitting from an area focal road to multiple roads. The system supports both merge and split scenarios.

3.2 Hierarchical Road Network Representation and ROI Definition

The OD Proportion Analysis method represents the study area as a hierarchical road network, which is shown in Figure 3.2. In this structure, each intersection is represented as a node, and the connected intersections are organized by their relationship to that intersection's focal road. Merge scenarios use the Ei labeling convention, while Split scenarios use the Xi labeling convention. This level-order labeling approach helps preserve the parent–child relationship between connected intersections and provides a consistent method for representing complex roadway networks. It also reduces ambiguity during scenario setup by ensuring that each intersection is labeled according to its position within the roadway tree. Figure 3.3 shows an example intersection with the intersection focal road specified in a red rectangle (region of interest [ROI]) and other roads specified in blue rectangles.

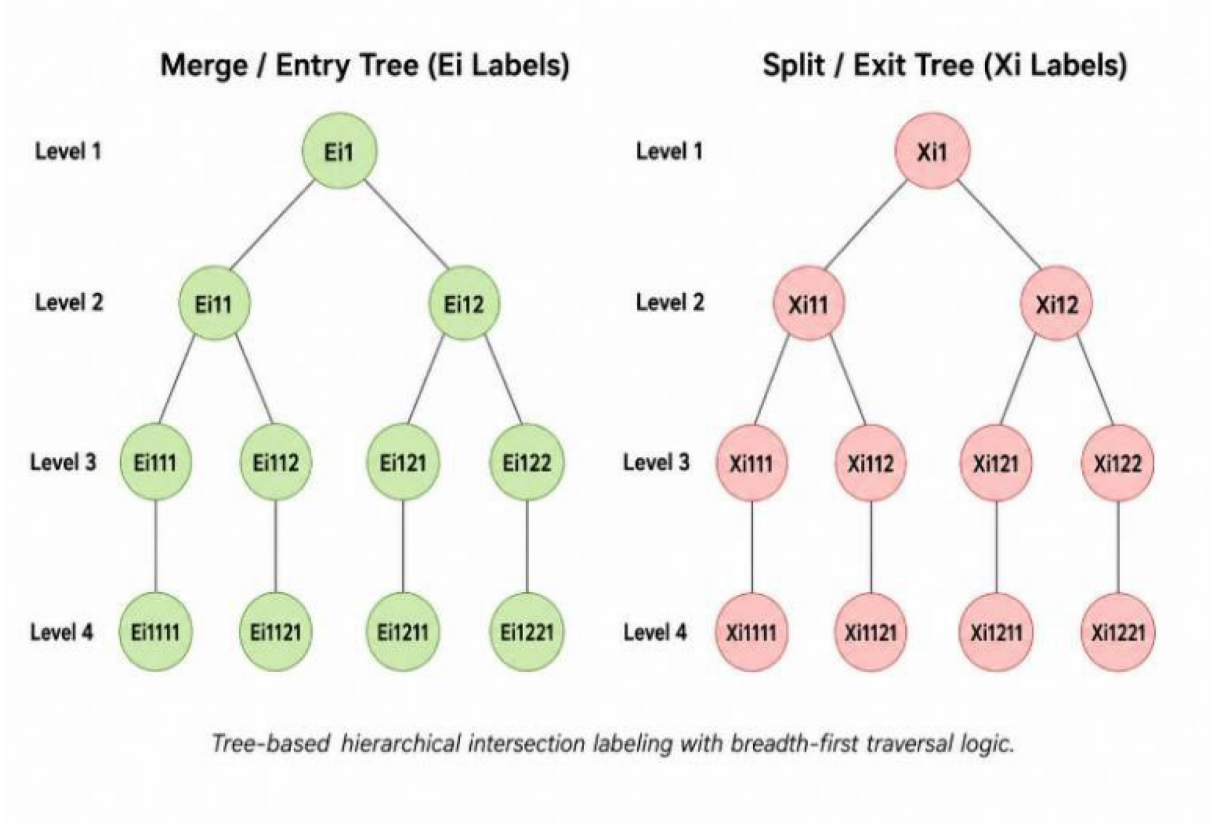


Figure 3.2 Tree Diagram (Labels with E = Merge, X = Split).



Figure 3.3 ROI-Based Road Annotation for Vehicle Tracking.

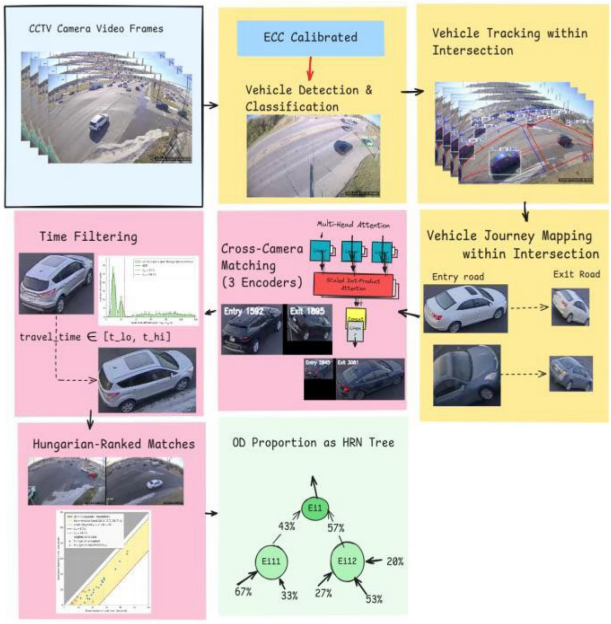


Figure 3.4 Machine Learning Framework From CCTV Videos to OD-Proportion Outputs.

3.3 Machine Learning Processing Framework

The Machine Learning Service provides an automated video-processing and vehicle-matching workflow for the OD Proportion Analysis System. In this setting, a single fixed camera observes each intersection. The views of different cameras are assumed not to overlap. Therefore, vehicles cannot be continuously tracked from one intersection to another.

The overall framework begins with uploaded CCTV video frames and user-defined roadway information. Vehicles in the recorded videos are detected and classified in each frame, tracked within each intersection, mapped to roadway regions of interest, and then matched across adjacent intersections using appearance features and travel-time constraints. The human-in-the-loop process removes false-positive vehicle matches. True vehicle matches are then aggregated to compute OD movement counts and OD proportions (Figure 3.4).

4. USER PROVIDES THE OD-PROPORTION SCENARIOS

A graphical user interface (GUI) was designed and developed to guide users through the complete OD Proportion Analysis workflow. The interface provides a structured, step-by-step process, so users can consistently create, configure, process, review, and report on each analysis scenario. The GUI supports the following major workflow functions:

- Scenario creation and setup
- Location confirmation on Google map
- Intersection labeling
- Video upload and frame capture
- Region-of-interest annotation
- Execute data processing software
- Data processing status review
- Human-in-the-loop vehicle matching verification
- Analysis result review
- Report generation

The OD-Proportion Analysis starts with the user specifying the areas and roads to be studied and recording camera videos that clearly cover all roads connected to each intersection; then the user uses the Web-based User Interface OD Proportion Analysis Tool to systematically provide the necessary information for this analysis. Specifically, the following data are collected.

- Study location on Google Maps, roadway context, and connecting-road information
- Scenario type, defined as Entry/Merge or Exit/Split
- Video length, video capture time of day, and number of videos available for analysis
- Intersection and road labels using the Ei/Xi and .0/.1/.2 conventions
- Uploaded video files, representative frame captures, and ROI annotation inputs
- Final user review information required before the scenario is submitted for processing

The step-by-step workflow reduces setup errors and improves consistency between scenarios. Since OD analysis depends on connected information such as intersection labels, road names, uploaded videos, captured frames, and ROI coordinates, the interface ensures that each required step is completed before processing begins.

The entire user-provided information through the web-based user interface is organized into a JSON-based data model and a video folder for further analysis.

5. MACHINE LEARNING BASED AND HUMAN IN THE LOOP OD-PROPORTION ANALYSIS

5.1 Vehicle Detection and Fine-Grained Classification

Vehicle detection is the first machine-learning stage in the OD Proportion Analysis workflow. The system uses a YOLO-11v object detector (Redmon et al., 2016) to identify traffic-relevant vehicle classes in each video frame. The detector focuses on vehicle categories such as cars, motorcycles, buses, and trucks (T.Y. Lin et al., 2014). After the initial detection step, a fine-grained vehicle classification ML-model, locally trained using the collected training data, is applied to support future cross-camera vehicle matching. This classification step assigns more detailed vehicle body-style categories, such as sedan, coupe, hatchback, pickup, SUV, van, light truck, bus, compact, minivan, crossover, and delivery or step van. These additional vehicle-type labels help reduce incorrect candidate matches by ensuring that vehicles are compared against visually and semantically similar vehicle classes.

This stage outputs detected vehicle bounding boxes, traffic class labels, fine-grained vehicle categories, and vehicle image crops. These outputs can be used as the input for optional ground truth generation (for AI model development and performance

analysis purposes only), and will be used for within-intersection vehicle tracking and later cross-camera vehicle matching.

5.2 Ground Truth Annotation

The Ground Truth Annotation and Validation System is a component of the OD Proportion Analysis tool. It supports manual vehicle counting, vehicle ID assignment, and tracking verification within an intersection. The manually annotated data can serve as benchmark data for evaluating machine-generated detections, tracks, and matched vehicle pairs within each intersection.

For each scenario, the Ground Truth Web Application can generate a scenario-specific report summarizing manually verified vehicle counts and tracking information. It also produces raw Excel sheets that provide the Machine Learning team with structured data for detailed review, model evaluation, and algorithm improvement. The Ground Truth Web Application comprises multiple interfaces that support scenario selection, manual annotation, vehicle-tracking review, and ground-truth output generation.

5.3 Within-Intersection Vehicle Tracking

After vehicles are detected, the system tracks them across frames within each intersection. Within-intersection tracking maintains each vehicle's identity as it moves through the camera view and determines the roadway path it follows. The system records the track ID, the approach and exit roads, and the timestamps for ROIs on the entry and exit roads. The tracking workflow uses BoT-SORT (Aharon et al., 2022) to associate detected vehicles across consecutive frames. BoT-SORT combines motion-based information and appearance similarity to maintain track identity as vehicles move. This is important in intersection videos because vehicles may be temporarily occluded by other vehicles, may move through crowded queues, or may appear at different scales as they approach or leave the camera view.

The method also incorporates StrongSORT-style (Du et al., 2023) motion compensation to reduce errors caused by small camera movements, which are common with pole-mounted traffic cameras due to wind or ground vibration. By compensating for camera movement, the tracking system can more reliably maintain vehicle identity and determine whether a vehicle has passed through the correct ROI.

A vehicle is counted as passing through an ROI only when its detected bounding box sufficiently overlaps the ROI polygon and is present for enough consecutive frames. This prevents false movement assignments caused by brief bounding-box overlaps, adjacent-lane interference, or partial visibility at the image border.

5.4 Cross-Camera Vehicle Matching

Cross-camera vehicle matching is the central machine-learning challenge in the OD Proportion Analysis System. Because adjacent intersection cameras do not have overlapping fields of view, a vehicle leaving one intersection cannot be continuously tracked until it appears at the next intersection. Therefore, the system treats this task as a vehicle re-identification problem.

For each candidate vehicle pair, the system evaluates whether a vehicle observed leaving an upstream intersection is likely the same vehicle entering a downstream intersection. Matching decisions are based on vehicle appearance, vehicle class consistency, travel-time feasibility, and assignment-based matching logic. The following subsections describe the general idea. Details are provided in Kumar et al. (2026).

5.4.1 Encoder Configuration and Feature Representation

Vehicle matching is based on the vehicles' feature vectors. The system uses machine-learning encoders to generate feature vectors from cropped vehicle images. These feature vectors represent visual characteristics such as vehicle shape, color, body style, lighting response, and visible design features. Because no single encoder is reliable under all roadway conditions, the method combines multiple encoders to improve matching robustness.

The method averages three complementary encoders: a curriculum-trained TransReID encoder (Bengio et al., 2009), a multicorridor-trained TransReID encoder, and a large pretrained vision transformer encoder (Dosovitskiy et al., 2021). Each encoder emphasizes different visual characteristics and has some failure patterns. For example, one encoder may be more effective under occlusion, another may be more robust across lighting and camera-angle variation, and another may capture fine-grained appearance differences.

For each tracked vehicle, multiple cropped images are sampled along the vehicle track. Crops are processed through encoders, and the resulting feature vectors are combined into a track-level descriptor. This reduces the effect of temporary occlusion, motion blur, or frame-level detection noise.

5.4.2 Travel-Time Filtering for Candidate Pairs

For each candidate vehicle pair, the travel-time difference Δt is the downstream ROI arrival time minus the upstream ROI departure time, using timestamps on a common time basis. Appearance-based candidate pairs provide the samples used to learn a travel-time window for each connecting road. Here, $f(\Delta t)$ is the Gaussian kernel density estimate of these travel-time differences, and f_peak is its maximum density. Equation 5.1 sets the density threshold τ to 25% of f_peak. Equation 5.2 defines t_lo and t_hi as the smallest and largest travel-time values where the estimated density is at least τ. The interval [t_lo, t_hi] is then used to filter candidate pairs before one-to-one assignment. Pairs outside this interval are excluded.

$$\tau = 0.25 f_{\text{peak}} \qquad \text{Equation 5.1}$$

$$t_{\text{lo}} = \min\left\{\Delta t : f\left(\Delta t\right) \geq \tau\right\}, \quad t_{hi} = \max\left\{\Delta t : f\left(\Delta t\right) \geq \tau\right\} \qquad \text{Equation 5.2}$$

This step matters because many vehicles may look similar, especially in common body styles and colors. Travel-time

filtering reduces false matches by limiting candidate pairs to the learned travel-time interval for the connecting road.

In signalized arterial environments, travel times may vary because some vehicles pass through a green light while others wait at a red light or in a downstream queue. The learned interval can span both free-flow and queued travel-time modes, including the gap between them. The road-specific window is updated as traffic conditions change rather than using a fixed travel-time threshold.

5.4.3 Assignment-Based Matching and Match Selection

After applying vehicle-class and travel-time filters, the remaining candidate pairs are scored using fused visual similarity. The similarity scores from the multiple encoders are combined into a single matching cost (Equation 5.3). A lower cost indicates that two vehicle observations are more likely to represent the same vehicle.

$$\text{cost} = 1 - \text{fused similarity} \qquad \text{Equation 5.3}$$

The candidate-pair costs are organized into a cost matrix, where rows correspond to upstream vehicle tracks and columns to downstream vehicle tracks (Figure 5.1). The matching problem is then solved using one-to-one assignment logic, ensuring that each upstream vehicle matches at most one downstream vehicle and each downstream vehicle matches at most one upstream vehicle (Kuhn, 1955). This prevents multiple vehicles from being incorrectly assigned to the same downstream observation.

The accepted matches are used as the cross-intersection vehicle correspondences required for OD movement estimation.

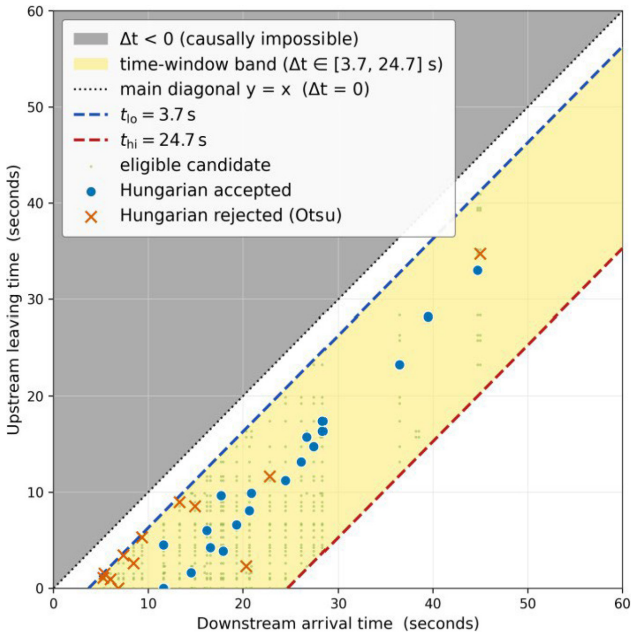


Figure 5.1 Cost Matrix and Assignment-Based Vehicle Matching

5.5 Human-in-the-Loop Matched Vehicle Pair Verification

The ML models' generated vehicle matches contain a significant number of false positives. Therefore, these results can only serve as a baseline. However, since we are only interested in true-positive matches and want to eliminate false positives, human help can be applied. Since all vehicles to be checked are already paired, this reduces the human matching workload by over 90% compared to nonpaired matching, making efficient human-in-the-loop match verification practically feasible. The web-based user interface retrieves machine-matched vehicle pairs and corresponding vehicle images, then displays vehicle-pair images side-by-side for user verification. This enables the user to efficiently indicate all false-positive vehicle-pair matches.

5.6 OD-Proportion Estimation

After vehicle detection, within-intersection tracking, ROI-based movement mapping, cross-camera matching, and human-in-the-loop verification are completed, the accepted vehicle matches are aggregated to estimate OD movement counts and OD proportions. The OD-proportion output describes how vehicles entering or exiting through the focal road are distributed across the connected roadway branches. The OD-proportion formulation uses upstream and downstream intersection vehicles to estimate the contribution of matched vehicles to movement proportions across connected roadway branches. This formulation connects the earlier machine-learning outputs, including tracked vehicle counts and matched vehicle pairs, to the final OD movement estimates.

The focal road of a downstream and an upstream intersection as U_0 and D_0, respectively, and other roads are numbered as i (i = 1, . . . , n). Using an example of vehicles on multiple roads merging to one area-focal-road through two intersections, U and D, the road connecting the upstream to downstream intersections is numbered as U_k and D_0, respectively. Let UX_i or DX_j be the number of vehicles tracked from U_i or D_j to U_0 or D_0. Equation 5.4 shows the number of vehicles observed in the focal road of an intersection coming from other roads in the same intersection:

$$UX_0 = \sum_{i=1}^{n} UX_i,\ DX_0 = \sum_{j=1}^{m} DX_j \qquad \text{Equation 5.4}$$

Let $UTX_0 = DTX_k$ be the matched vehicles from U_0 to D_k. Equation 5.5 shows that DTX_k is a fraction c of UX_0 and a function of UX_i:

$$DTX_k = UTX_0 = cUX_0 = c\sum_{i=1}^{n} UX_i = \sum_{i=1}^{n} cUX_i \qquad \text{Equation 5.5}$$

Let ai be a coefficient factor no greater than 1 for cUX_i vehicles tracked to D_k, and let $b = DTX_k/DX_k$:

$$DTX_k = bDX_k = bUX_0 = \sum_{i=1}^{n} a_i cUX_i \qquad \text{Equation 5.6}$$

Thus,

$$DX_0 = \sum_{j=1}^{m} DX_j,$$
$$= \sum_{p=1}^{k-1} DX_p + \sum_{p=k+1}^{m} DX_p + DX_k \qquad \text{Equation 5.7}$$
$$= \sum_{p=1}^{k-1} DX_p + \sum_{p=k+1}^{m} DX_p + \sum_{i=1}^{n} \frac{ca_i UX_i}{b}$$

The formulation in Equation 5.7 defines the relationship between upstream and downstream vehicle movements and matched vehicles at connected intersections. The variables represent roadway-level vehicle counts, matched vehicle movements, and coefficient factors used to estimate vehicle distribution across connected roadway branches.

For merge-type scenarios, the system estimates the contributions of vehicles from upstream or local roads to traffic entering the focal roadway. For split-type scenarios, the same framework can be applied in reverse to estimate how vehicles leaving the focal roadway are distributed across downstream or local roads.

The final OD outputs include movement counts, movement proportions, matched vehicle-pair information, vehicle snapshots, and report-ready data files. These outputs are later used by the Vehicle Verification UI and the web-based analytical report.

5.7 OD Proportion Analysis Result and Report

The OD proportion analysis results are presented in a Web user interface and a PDF report. The result contains the following information:

1. Description of the user-entered OD-Proportion problem, including the image map with the road annotations.
2. The vehicle detection counts on each road of each intersection. If the ground truth was generated earlier, the percentage error is displayed.
3. The number of vehicles tracked from each road to the intersection focal road in each intersection. If the ground truth was generated earlier, the percentage error is displayed.
4. The number of vehicles matched from the two ends of the same road at the adjacent intersection. If the ground truth was generated earlier, the percentage error is displayed.
5. The number of vehicles matched from each road to the area focal road. If the ground truth was generated earlier, the percentage error is displayed.

An example of the OD-proportion analysis result PDF output is in Appendix A.

6. BACKEND ARCHITECTURE OF THE OD PROPORTION ANALYSIS SYSTEM

The backend architecture of the Origin–Destination Proportion Analysis system was developed to support a complete, file-based, machine-learning-assisted traffic analysis workflow. The backend serves as the system integration layer connecting the web-based user interface, shared scenario storage, machine-learning processing pipeline, and analytical reporting components. Its design emphasizes structured data exchange, scenario-level traceability, persistent file management, and reproducible processing across the full analysis lifecycle.

The backend system is implemented as a Python-based API service that coordinates data exchange between the user browser, NGINX web server, application server, NAS-based file storage, and machine learning processing server. Through REST API requests, the frontend sends scenario metadata, uploaded videos, map thumbnails, intersection images, and region-of-interest annotations to the backend. The backend validates these inputs and stores them within a scenario-specific workspace on the shared NAS file server. Appendix B details the Backend system.

7. EXPERIMENT RESULTS

The OD-Proportion Analysis methodology was evaluated on field video collected from five deployment areas in Indiana, each observed by nonoverlapping, pole-mounted traffic cameras. Results are reported stage by stage—vehicle detection and within-intersection tracking (Section 7.2), cross-camera matching (Section 7.3), and human-in-the-loop verification (Section 7.4). Section 7.5 combines the stage results into an overall OD-proportion analysis report. Because each stage feeds the next, the cumulative effect of every stage determines OD precision and accuracy.

7.1 Deployment Sites

The evaluation locations span a range of roadway environments, camera resolutions, and traffic conditions, as shown in Table 7.1.

All sites share the same constraint that each intersection is observed by a single camera, and adjacent-intersection cameras have no overlapping field of view, so a vehicle cannot be tracked continuously from one intersection to the next. Most areas comprise two adjacent intersections; one comprises three.

TABLE 7.1
Deployment Sites.

Site	Location	Resolution	Intersections Evaluated	Scenario	Human Matching
Pendleton Pike	Indianapolis	720p	2	Split/Merge	Yes
I-74 at Rampart Rd	Shelbyville	720p	2	Split	Yes
US-31 at Michigan St	Indianapolis	540p	2	Merge	Yes
SR-331 at Jackson St	Indianapolis	540p	2	Merge	Yes
SR-9 at Martindale	Indianapolis	480p	3	Split/Merge	Yes

Camera resolution varies across sites (480p, 540p, 720p). Camera resolution is a primary factor in matching quality: higher-resolution video yields more vehicle details and appearance detail that matching depends on, whereas 480p crops retain substantially less usable detail (Section 7.3).

7.2 Detection and Within-Intersection Tracking Performance

7.2.1 Vehicle Detection and Classification

Vehicle detection is performed by a two-stage subsystem. A YOLOv11 detector first localizes traffic-relevant vehicle classes—car, motorcycle, bus, and truck—in each frame. A second, locally trained fine-grained classifier then assigns each detected vehicle a body-style category (sedan, coupe, hatchback, pickup, SUV, van, light truck, bus, compact, minivan, crossover, and delivery/step van). Across both stages, the subsystem performs best under 720p camera videos. The detector achieves precision of approximately 0.95 and recall in the 95%–97% range for the observed traffic-relevant vehicles. The fine-grained classifier reaches top-1 accuracy 0.899 and top-3 accuracy 0.942 (F1 0.753) on a validation set of more than 1,000 images. Remaining errors concentrate on partially occluded vehicles, vehicles far from the camera, and visually ambiguous body styles.

7.2.2 Within-Intersection Tracking

Tracking uses BoT-SORT with StrongSORT-style motion compensation to maintain vehicle identity through occlusion, queueing, and small movements of pole-mounted cameras caused by wind and vibration. Tracking quality is characterized by how many true origin-to-destination journeys survive each downstream gate (Table 7.2). Performance is reported using average accuracy, which evaluates the algorithm on vehicle tracking within the intersection between entry and exit roads. Low-resolution cameras clearly produce unacceptably low average accuracy. Later subsections discuss only scenarios using 720p cameras.

7.3 Cross-Camera Vehicle Matching Performance

Cross-camera matching decides whether a vehicle leaving an upstream intersection is the same vehicle arriving at a downstream intersection, with no overlapping camera view between them. Performance is reported using retrieval mean average precision (mAP) and Rank-1 accuracy—which measure how well the correct downstream vehicle is ranked among the gated candidates—together with the bipartite matching F1 score, the accuracy of the final one-to-one vehicle assignments. The two differ by construction: mAP is computed over the features, vehicle classes, and travel-time-gated candidate pool (so it is high once gating removes implausible candidates), while F1 reflects the accuracy of the matches actually used for OD counting.

TABLE 7.2
Within-Intersection Tracking Accuracy Analysis Over Separate Deployment Sites (720p Only).

Scenario	Camera Resolution	Average Accuracy
Pendleton Pike	720p	0.98
I-74 Rampart	720p	0.92
Jackson St.	540p	0.45
Martindale St.	480p	0.18

7.3.1 Three-Encoder Ensemble

No single appearance encoder is reliable across all roadway conditions, so the system averages the feature vectors of three complementary encoders: a curriculum-trained TransReID encoder, a multicorridor-trained TransReID encoder, and a large pretrained Vision Transformer encoder. Averaged over the pooled evaluation, the three-way mean-fusion ensemble matches or exceeds every individual encoder and has the lowest cross-site variance. As shown in Table 7.3, no single encoder wins every site, so fusing them yields the most stable deployment behavior. A fourth (FastReID) encoder was tested and never improved results and was therefore excluded.

7.3.2 Per-Site Performance

Table 7.4 reports the performance of the deployed ensemble model at the sites using 720p cameras. Pendleton (720p) was used to train and fine-tune encoders, whereas Rampart Intersection (720p) was used to test the transfer accuracy of our ensemble method.

7.3.3 Effect of Resolution and Time of Day

Table 7.5 shows the controlled time-of-day comparison, where general matching precision averaged 0.8 across the morning, midday, and evening. Performance is lower in the early morning and evening, when light is lower and traffic is heavier. The performance is better in good lighting with no glare, mostly at noon.

7.3.4 Contribution of the Travel-Time Filter

The travel-time window is the dominant driver of matching accuracy. Table 7.6 shows the travel-time filter effect on the matching performance: the learned, road-specific travel-time filter improved matching F1 significantly.

TABLE 7.3
Encoder Comparison (Pooled Pendleton Evaluation, 786 Pairs).

Encoder	mAP	Rank-1	F1
TransReID, curriculum fine-tuned	0.75	0.73	0.65
TransReID,multi-corridor fine-tuned	0.76	0.70	0.66
Vision Transformer (ViT-L/16), fine-tuned	0.74	0.74	0.59
Three-way mean fusion (deployed)	0.80	0.74	0.74

TABLE 7.4
Cross-Site Matching Performance, Three-Way Ensemble.

Site (Resolution)	mAP	Rank-1	F1
Pendleton Pike (720p)	0.88	0.74	0.80
I-74 at Rampart Rd (720p)	0.69	0.65	0.63

TABLE 7.5
Time of the Day Matching Performance for the Pendleton Pike Scenario.

Site (Resolution)	Time of Day	Traffic	Lighting	mAP	Rank-1	F1
Pendleton Pike (720p)	Morning peak	High	Low sun/glare	0.893	0.867	0.801
Pendleton Pike (720p)	Midday off-peak	Low	Bright, even	0.919	0.843	0.803
Pendleton Pike (720p)	Evening peak	High	Low sun/dusk	0.884	0.836	0.790

TABLE 7.6
Matching F1 Score by Filtering Configuration.

Configuration	F1 Score (Pendleton Pike)
Appearance only (baseline)	0.17
+ learned travel-time window	0.38
+ travel-time + class filter	0.43

TABLE 7.7
Effect of Human-in-the-Loop Verification on End-to-End Precision.

	Matching-Stage Precision	Overall Precision
Automatic (preverification)	0.70	0.59
After verification	0.96	0.85

TABLE 7.8
OD Count Accuracy (Predicted vs. Ground-Truth Vehicle Counts).

Site	Intersection	Accuracy	Detected/ Ground Truth
Pendleton Pike	EI1	98%	516/523
Pendleton Pike	EI11	86%	261/305

TABLE 7.9
Origin–Destination Flow Shares at Pendleton Pike (EI11).

Input Road	Out Road	Share of flow
EI11.1	EI11.0	2.7%
EI11.2	EI11.0	73.4%
EI11.3	EI11.0	23.9%
Total	-	100%

TABLE 7.10
Origin–Destination Flow Shares at Pendleton Pike (EI1).

Input Road	Out Road	Share of flow
EI1.1	EI1.0	65.9%
EI1.2	EI1.0	34.1%
Total	**-**	**100%**

7.4 Human-in-the-Loop Verification

Automatic matching still produces false positives, that is, similar-looking vehicles, fragmented tracks, and viewpoint differences are unavoidable without continuous tracking. Because the system needs only true-positive matches for OD counting, a human verifier reviews the candidate pairs the model proposes and rejects the false ones. Crucially, the reviewer sees already-paired candidates rather than recounting raw video, which reduces the verification effort by more than 95% relative to manual hand-counting (roughly 5–10 s per pair) and makes human verification practical at deployment scale.

Verification raises end-to-end precision from 0.59 (automatic) to 0.85 because rejecting false pairs drives matching-stage precision to effectively 1.00 while retaining true matches; see Table 7.7 for observing effects of Human verification. The interface applies three operating bands: pairs scoring 0.85 or above are auto-accepted, 0.60–0.85 are queued for review, and below 0.60 are rejected.

7.5 Final Overall Results and Evaluation

7.5.1 Within-Intersection Turning-Movement Count Accuracy

The application-level metric of interest to practitioners is how closely the system's per-road vehicle counts match ground-truth counts. At intersections with light traffic, the system reproduced ground-truth counts exactly. At intersections with busier roads, accuracy was more variable (Table 7.8), ranging from ~80%–100%.

7.5.2 End-to-End Precision

Combining the stage precisions gives the overall system precision. Before verification, the five-stage product defined in Equation 7.1 (detection, tracking, matching, and the downstream detection and tracking of the matched vehicle) yields a mean precision of 0.59; human verification of the matching pairs raises the matching factor to approximately 1.00 and the overall precision to 0.85 at 720p (Table 7.11).

$$P = P_{up,det} \times P_{up,track} \times P_{match} \times P_{down,det} \times P_{down,track}$$

Equation 7.1

The flow shares for EI11 and EI1 are shown in Table 7.9 and Table 7.10, respectively.

As shown in Table 7.11, a matched origin–destination vehicle passes through five processing stages: detection and tracking at the upstream intersection, cross-camera matching, and detection and tracking again at the downstream intersection.

TABLE 7.11
End-to-End Precision (720p).

	Detection (Upstream)	Tracking (Upstream)	Matching	Detection (Downstream)	Tracking (Downstream)	Overall
Program generated (Before verification)	0.95	0.97	0.70	0.95	0.97	0.59
After verification	0.95	0.97	1.00	0.95	0.97	0.85

Reading across the first row as an example, the per-stage precisions are 0.95 (upstream detection), 0.97 (upstream tracking), 0.70 (matching), 0.95 (downstream detection), and 0.97 (downstream tracking); their product is 0.59. The matching stage is the weakest factor, so the human-in-the-loop verification in Section 7.4 targets it: once the reviewer removes false matches, matching precision approaches 1.00, and overall end-to-end precision rises to 0.85.

7.6 Summary

On the corridor with 720p cameras, the system achieves matching mAP 0.90/F1 0.80 and, after verification, 0.85 end-to-end precision. Performance degrades predictably with lower camera resolution (540p). These results strongly recommend deploying a 720p traffic camera with human-in-the-loop verification. We expect the results to improve with 1080p cameras, although we did not have a chance to use them in the experiments.

8. CONCLUSION

This project answers a practical question for traffic engineers: when vehicles pass through a set of connected intersections, how many of them travel from each incoming road to each outgoing road? Answering it is hard in the real world because tracking many vehicles among many connected roads over a long time is tedious. The solution described in this report combined machine learning and human-in-the-loop techniques to provide a practical solution. Since the reviewer only checks already-paired vehicles instead of re-counting the whole video, this step is more than 95% faster than counting by hand.

Tested on field video from five sites in Indiana, the system with 720p cameras detects vehicles with about 95%–97% accuracy and, reaches a matching mAP of about 0.90 and an F1 of about 0.80. After the human review step, the end-to-end precision reaches 0.85. Just as important for real use, the system delivers all of this from the single, nonoverlapping cameras that agencies already have in the field, along with a web-based tool that guides a user from setting up a scenario to producing a final report.

Overall, the results show that reliable origin–destination proportions can be produced from existing traffic cameras with a modest human-in-the-loop step, giving transportation agencies a practical, low-cost way to understand how vehicles move through their road networks.

REFERENCES

Aharon, N., Orfaig, R., & Bobrovsky, B.-Z. (2022). *BoT-SORT: Robust associations multi-pedestrian tracking*. arXiv. https://doi.org/10.48550/arXiv.2206.14651

Bengio, Y., Louradour, J., Collobert, R., &Weston, J. (2009). Curriculum learning. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning* (pp. 41–48). https://doi.org/10.1145/1553374.1553380

Chiou, Y.-C., Lan, L. W., Tseng, C.-M., & Fan, C.-C. (2012). Optimal locations of license plate recognition to enhance the origin-destination matrix estimation. *Asian Transport Studies*, *2*(1) 80–92. https://doi.org/10.11175/eastsats.2.80

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). *An image is worth 16x16 words: Transformers for image recognition at scale*. International Conference on Learning Representations, Vienna, Austria. https://openreview.net/forum?id=YicbFdNTTy

Du, Y., Zhao, Z., Song, Y., Zhao, Y., Su, F., & Gong, T. (2023). StrongSORT: Make DeepSORT great again. *IEEE Transactions on Multimedia*, *25*. https://doi.org/10.1109/TMM.2023.3240881

Federal Highway Administration. (2016). *Traffic monitoring guide* (Report No. FHWA-PL-17-003). https://www.fhwa.dot.gov/policyinformation/tmguide/tmg_fhwa_pl_17_003.pdf

He, L., Liao, X., Liu, W., Liu, X., Cheng, P., & Mei, T. (2020). *FastReID: A PyTorch toolbox for general instance re-identification*. arXiv. https://doi.org/10.48550/arXiv.2006.02631

He, S., Luo, H., Wang, P., Wang, F., Li, H., & Jiang, W. (2021). TransReID: Transformer-based object re-identification. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 15013–15022). https://doi.org/10.1109/ICCV48922.2021.01474

Hsu, H.-M., Wang, Y., & Hwang, J.-N. (2020). Traffic-aware multi-camera tracking of vehicles based on ReID and camera link model. In *MM '20: Proceedings of the 28th ACM International Conference on Multimedia* (pp. 964–972). https://doi.org/10.1145/3394171.3413863

Huang, H.-W., Yang, C.-Y., & Hwang, J.-N. (2023). *Multi-target multi-camera vehicle tracking using transformer-based camera link model and spatial-temporal information*. arXiv. https://doi.org/10.48550/arXiv.2301.07805

Iqbal, M. S., Choudhury, C. F., Wang, P., & González, M. C. (2014). Development of origin-destination matrices using mobile phone call data. *Transportation Research Part C: Emerging Technologies*, *40*, 63–74. https://doi.org/10.1016/j.trc.2014.01.002

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, *2*(1–2), 83–97. https://doi.org/10.1002/nav.3800020109

Kumar, S., Chien, S., Chen, Y., Desai, S., & Shellhamer, N. (2026). Origin–destination proportion analysis using machine learning on non-overlapping camera videos. In *2026 IEEE 9th*

International Conference on Multimedia Information Processing and Retrieval (MIPR) (pp. 347–353). https://doi.org/10.1109/MIPR70517.2026.00062

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer vision – ECCV 2014* (pp. 740–755). https://doi.org/10.1007/978-3-319-10602-1_48

Lin, W., Qiu, M., Chien, S., Christopher, L. A., & Chen, Y. (2024). Highway origin-destination vehicle weaving analysis using field camera videos. In *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)* (pp. 1925–1931). https://doi.org/10.1109/ITSC58415.2024.10920155

Lin, Y., Lockyer, S., & Zhang, N. (2024). *City-scale multi-camera vehicle tracking system with improved self-supervised camera link model*. arXiv. https://doi.org/10.48550/arXiv.2405.11345

Liu, X.-C., Liu, W., Ma, H., & Fu, H.-Y. (2016). Large-scale vehicle re-identification in urban surveillance videos [Proceedings]. 2016 IEEE International Conference on Multimedia and Expo (ICME), Seattle, WA, United States. https://doi.org/10.1109/ICME.2016.7553002

Luna, E., SanMiguel, J. C., Martínez, J. M., & Escudero-Viñolo, M. (2022). Online clustering-based multi-camera vehicle tracking in scenarios with overlapping FOVs. *Multimedia Tools and Applications*, *81*, 7063–7083. https://doi.org/10.1007/s11042-022-11923-2

Qiu, M., Lin, W., Chien, S., Christopher, L. A., Chen, Y., & Hu, S. (2024). Enhancing vehicle re-identification and matching for weaving analysis. In *2024 IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 1–7). https://doi.org/10.1109/AVSS61716.2024.10672578

Qiu, M., Lin, W., Christopher, L. A., Chien, S., Chen, Y., & Hu, S. (2024). Real-time lane-wise traffic monitoring in optimal ROIs. In *2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR)* (pp. 8–14). https://doi.org/10.1109/MIPR62202.2024.00009

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, PMLR 139* (pp. 8748–8763). https://proceedings.mlr.press/v139/radford21a.html

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 779–788). https://doi.org/10.1109/CVPR.2016.91

Stopher, P. R., & Greaves, S. P. (2007). Household travel surveys: Where are we going? *Transportation Research Part A: Policy and Practice*, *41*(5), 367–381. https://doi.org/10.1016/j.tra.2006.09.005

Tang, Z., Naphade, M., Liu, M.-Y., Yang, X., Birchfield, S., Wang, S., Kumar, R., Anastasiu, D., & Hwang, J.-N. (2019). CityFlow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 8797–8806).

Wang, S., Anastasiu, D. C., Tang, Z., Chang, M.-C., Yao, Y., Zheng, L., Rahman, M. S., Arya, M. S., Sharma, A., Chakraborty, P., Prajapati, S., Kong, Q., Kobori, N., Gochoo, M., Otgonbold, M.-E., Alnajjar, F., Batnasan, G., Chen, P.-Y., Hsieh, J.-W., … Chellappa, R. (2024). The 8th AI City Challenge. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp.7261–7272).https://doi.org/10.1109/CVPRW63382.2024.00722

Yuan, J., Zheng, Y., Zhang, C., Xie, W., Xie, X., Sun, G., & Huang, Y. (2010). T-drive: Driving directions based on taxi trajectories. In *GIS '10: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems* (pp. 99–108). https://doi.org/10.1145/1869790.1869807

Zhang, H., Fang, R., Li, S., Miao, Q., Fan, X., Hu, J., & Chan, S. (2024). Multi-camera multi-vehicle tracking guided by highway overlapping FoVs. *Mathematics*, *12*(10), Article 1467. https://doi.org/10.3390/math12101467

Zhong, Z., Zheng, L., Cao, D., & Li, S. (2017). Re-ranking person re-identification with k-reciprocal encoding. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3652–3661). https://doi.org/10.1109/CVPR.2017.389

APPENDICES

Appendix A. Example OD Analysis Report

Appendix B. The OD-Proportion Analysis Backend System

Appendix A. Example OD Analysis Report

OD Proportion Analysis Report

---

1. Scenario Details

This report summarizes vehicle movement patterns for the selected scenario. Results include the traffic flow structure, vehicle counts, origin-destination proportions, data visuals, and supporting visual thumbnails used in the analysis.

Scenario Name: I-465 NB and Pendleton Pike

Scenario Type: Local Roads to Highway (Merge)

Scenario ID: SC0113

Analysis ID: SC0113_Entry_Output

Location Coordinates: 39.832204, -86.035787

Analysis Generated: July 16, 2026

Scenario Analysis Version: v001

User Provided Scenario Information:

User Notes: There are two intersections for this Pendleton Pike Merge Scenario. Intersection 1 Video Recording Date: 02/11/2026 (1:00 PM to 1:10 PM) Intersection 2 Video Recording Date: 02/11/2026 (1:00 PM to 1:10 PM)

2. Intersection Details

Intersection Information and Input/Output Road Details

This section provides detailed information about each configured intersection within the selected scenario, including the associated input roads (roads entering the intersection) and output roads (roads exiting the intersection).

Intersection 1: Ei1 - I-465 Entry Ramp NB and Pendleton Pike

- **Output Road:** Ei1.0 - I-465 Entry Ramp NB
- **Input Road 1:** Ei1.1 - Pendleton Pike WB
- **Input Road 2:** Ei1.2 - I-465 Exit Ramp NB
- **Input Road 3:** Ei1.3 - Pendleton Pike EB

Intersection 2: Ei11 - Pendleton Pike and Sellers St

- **Output Road:** Ei11.0 - Pendleton Pike WB
- **Input Road 1:** Ei11.1 - Sellers St
- **Input Road 2:** Ei11.2 - Pendleton Pike WB
- **Input Road 3:** Ei11.3 - 42nd St

Table: Intersection Connectivity

From Intersection	From Road	To Intersection	To Road
Pendleton Pike and Sellers St (Ei11)	Pendleton Pike WB (Ei11.0)	I-465 Entry Ramp NB and Pendleton Pike (Ei1)	Pendleton Pike WB (Ei1.1)

3. Birds Eye View of the scenario

The figure below presents the user-captured Bird's-Eye View of the intersections within the selected scenario.



4. Intersection Thumbnails (Region of Interest)

The images below display the captured and annotated thumbnails extracted from the intersection video recordings.

Intersection 1: Ei1 - I-465 Entry Ramp NB and Pendleton Pike

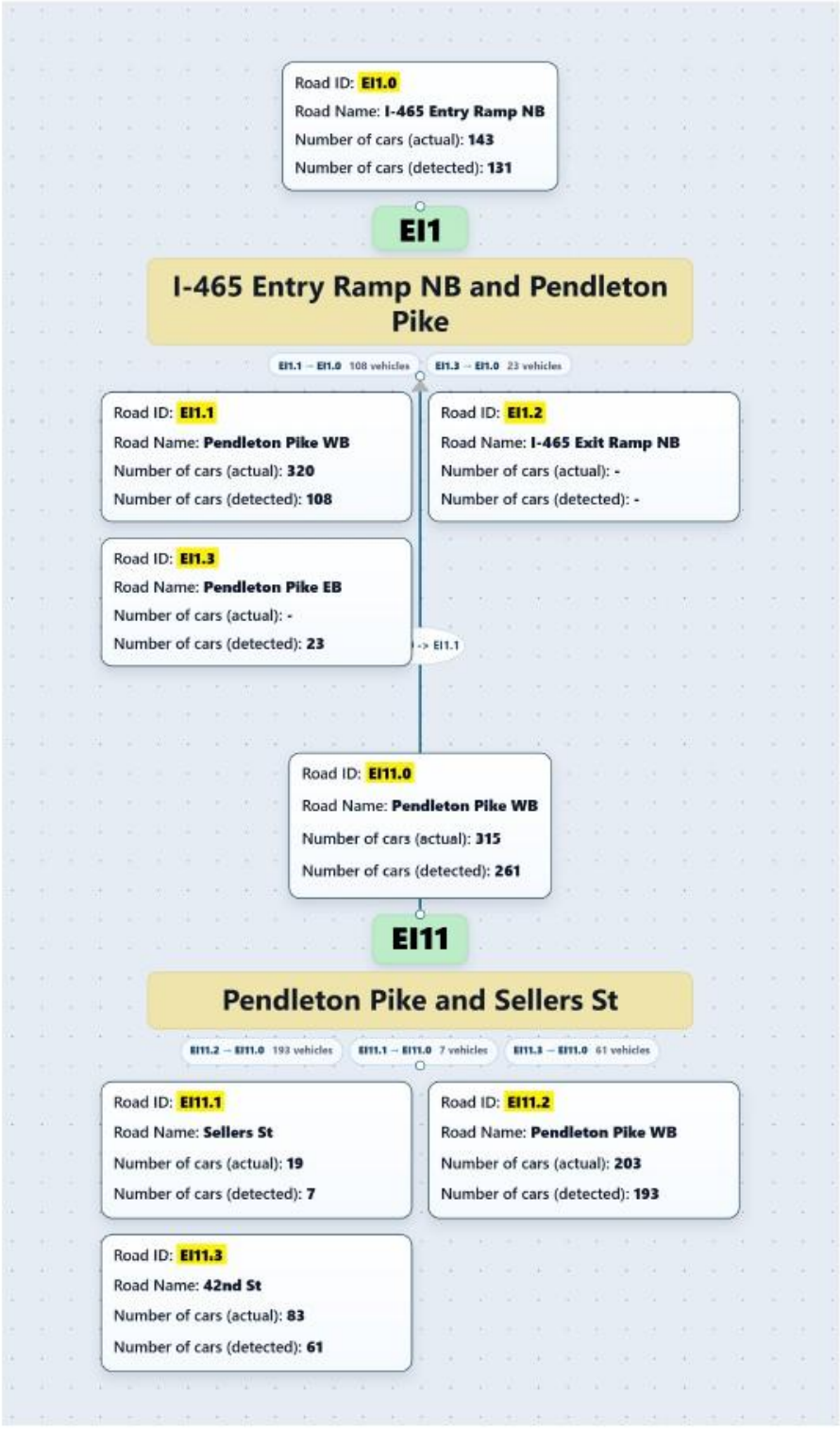


Intersection 2: Ei11 - Pendleton Pike and Sellers St



5. Intersection Linking

The scenario is presented in a tree structure diagram in which each node corresponds to an intersection.



6. Accuracy Analysis

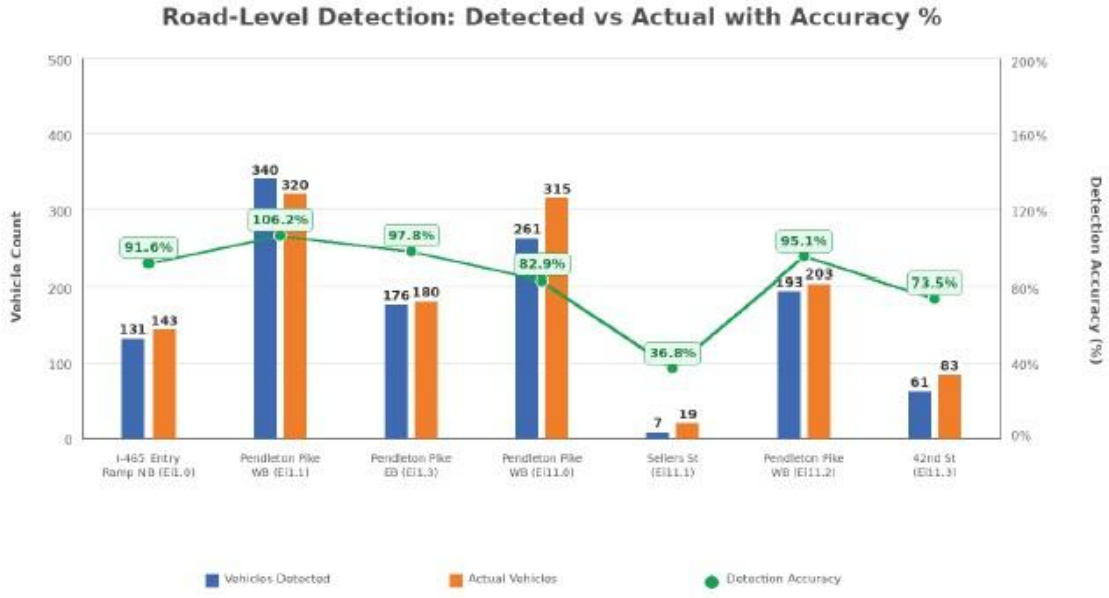
This section evaluates the performance of the vehicle detection and tracking system used in the weaving analysis.

6.1 Road Level Vehicle Detection

This table compares the number of vehicles detected by the automated system against manually validated counts for each road segment.

Road-Level Detection: Detected vs Actual with Accuracy %

Road ID	Road Name	Vehicles Detected	Actual Vehicles (Testing Case Only)	Detection Accuracy (%)
Ei1.0	I-465 Entry Ramp NB	131	143	91.6%
Ei1.1	Pendleton Pike WB	340	320	106.2%
Ei1.3	Pendleton Pike EB	176	180	97.8%
Ei11.0	Pendleton Pike WB	261	315	82.9%
Ei11.1	Sellers St	7	19	36.8%
Ei11.2	Pendleton Pike WB	193	203	95.1%
Ei11.3	42nd St	61	83	73.5%

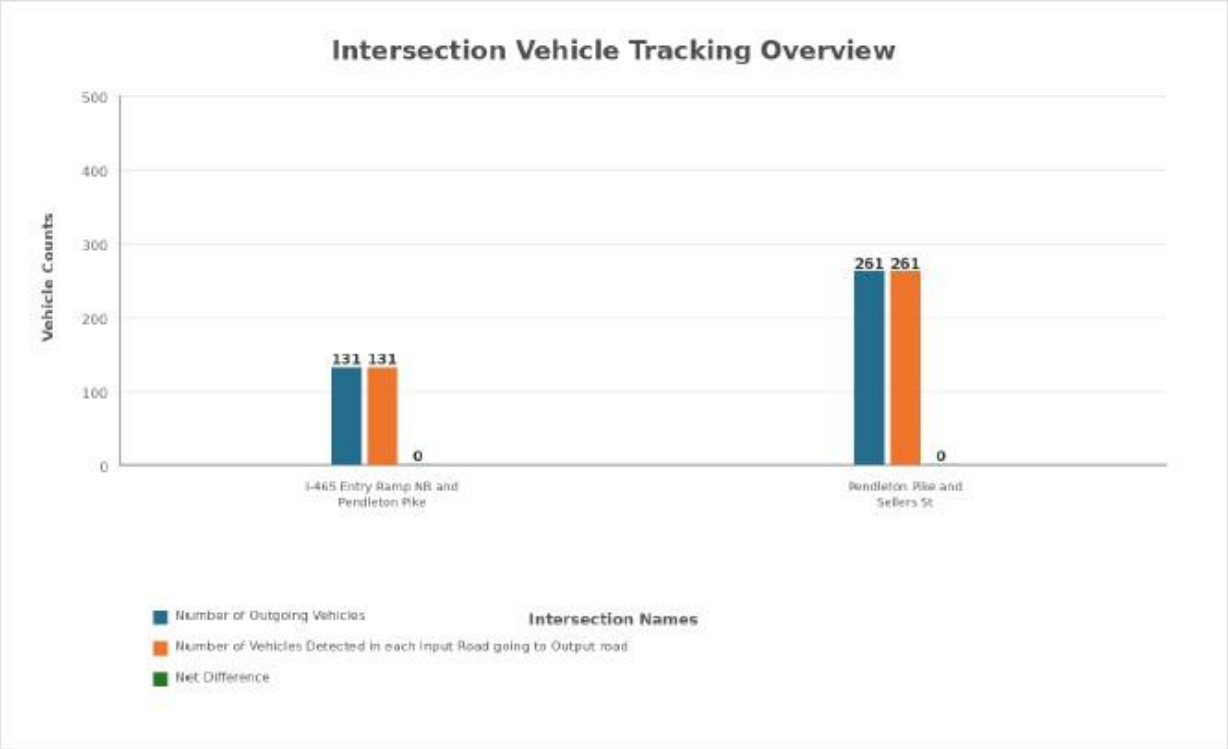


6.2 Intersection Vehicle Tracking Detection (Within Intersection)

This table verifies vehicle tracking continuity by comparing the total number of vehicles entering each intersection with the total number of vehicles tracked exiting the corresponding roads for Entry scenarios.

Table: Intersection Vehicle Tracking Detection (Entry Scenario)

Intersection ID	Intersection Name	Number of Outgoing Vehicles	Number of Vehicles Detected in each Input Road going to Output road (Within Intersection)	Net Difference
Ei1	I-465 Entry Ramp NB and Pendleton Pike	131	131	0
Ei11	Pendleton Pike and Sellers St	261	261	0



6.3 Intersection Summary

This table evaluates detection accuracy for every input road to the scenario output road.

Table A - Intersection Ei1 Summary

From Road ID	From Road Name	To Road ID	To Road Name	Vehicles Detected	Actual Vehicles (Testing Case Only)	Detection Accuracy (%)	Share of Intersection Flow (%)
Ei1.1	Pendleton Pike WB	Ei1.0	I-465 Entry Ramp NB	340	320	106.2%	65.9%
Ei1.3	Pendleton Pike EB	Ei1.0	I-465 Entry Ramp NB	176	180	97.8%	34.1%

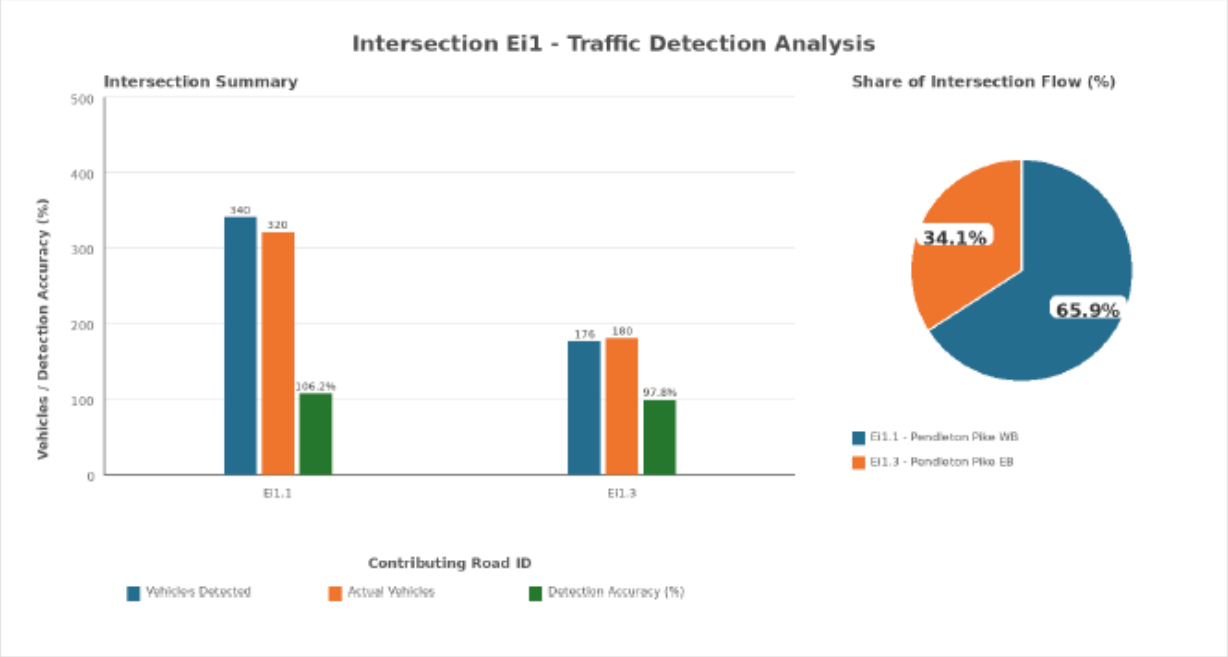
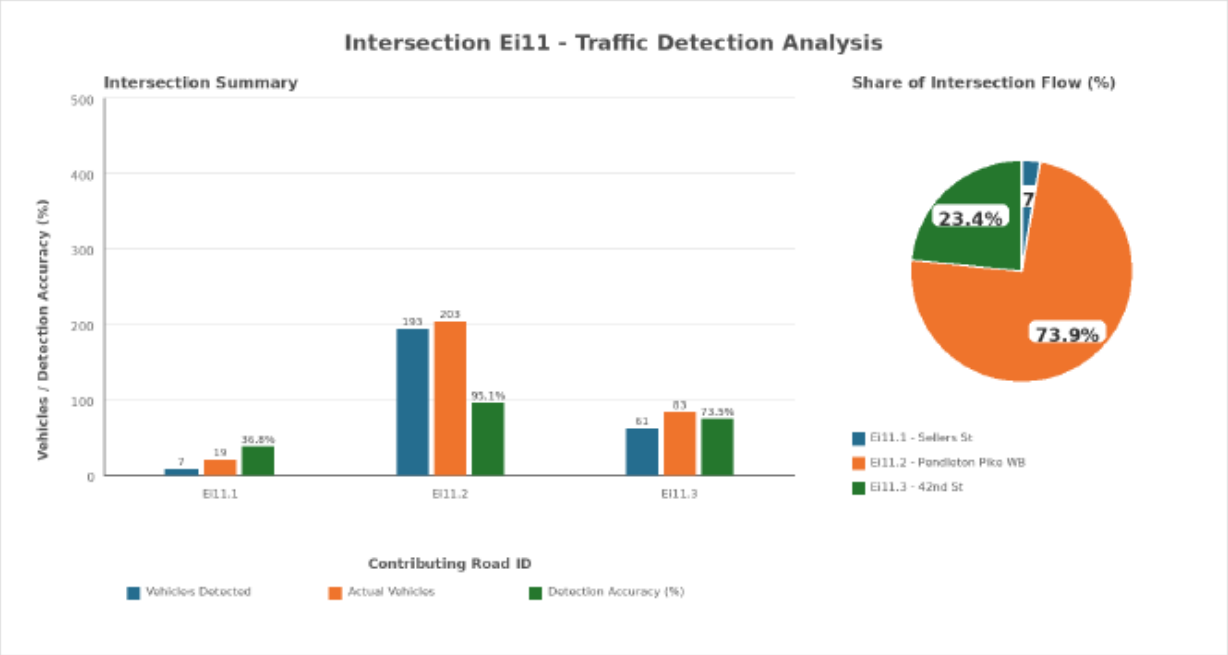


Table B - Intersection Ei11 Summary

From Road ID	From Road Name	To Road ID	To Road Name	Vehicles Detected	Actual Vehicles (Testing Case Only)	Detection Accuracy (%)	Share of Intersection Flow (%)
Ei11.1	Sellers St	Ei11.0	Pendleton Pike WB	7	19	36.8%	2.7%
Ei11.2	Pendleton Pike WB	Ei11.0	Pendleton Pike WB	193	203	95.1%	73.9%
Ei11.3	42nd St	Ei11.0	Pendleton Pike WB	61	83	73.5%	23.4%



7. Overall Road Connection Summary

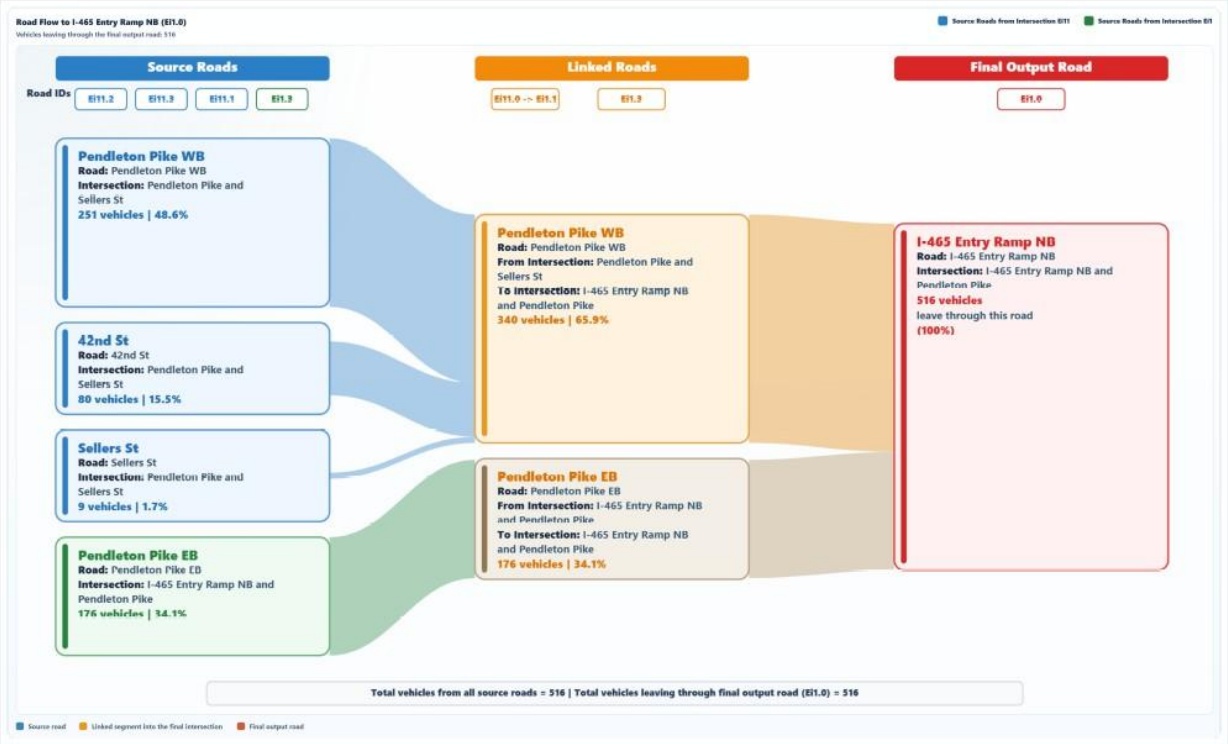
This section provides a map overview of the road connections included in the scenario.

Map Thumbnail



8. Sankey Diagram

This diagram shows vehicle movement flow visually. Wider paths represent larger vehicle counts.



Appendix B. The OD-Proportion Analysis Backend System

The backend architecture of the Origin–Destination Proportion Analysis system was developed to support a complete, file-based, machine-learning-assisted traffic analysis workflow. The backend functions as the system integration layer that connects the web-based user interface, shared scenario storage, machine learning processing pipeline, and analytical reporting components. The design emphasizes structured data exchange, scenario-level traceability, persistent file management, and reproducible processing across the full analysis lifecycle.

1. Backend System Overview and Component Interaction

The backend system is implemented as a Python-based API service that coordinates data exchange between the user browser, NGINX web server, application server, NAS-based file storage, and machine learning processing server. Through REST API requests, the frontend sends scenario metadata, uploaded videos, map thumbnails, intersection images, and region-of-interest annotations to the backend. The backend validates these inputs and stores them within a scenario-specific workspace on the shared NAS file server.

The major backend architecture components include:

- **User browser:** Provides browser-based access for users to create scenarios, upload videos, annotate regions of interest, review scenario configurations, visualize analysis results, and generate reports.
- **NGINX web server:** Functions as the reverse proxy and web server. It serves the React frontend build files and routes backend API requests to the internal backend programs running on the application server.
- **Application Server – TASI02:** Hosts the web root directory, React frontend build files, and backend programs for the OD Proportion Analysis system, vehicle verification system, and ground truth application.
- **Backend API programs:** Handle scenario creation, file uploads, JSON generation, scenario retrieval, processing status updates, output data access, and communication with the shared file storage environment.
- **File Server – TASI-VRU3:** Provides NAS-based storage for scenario workspaces, including input files, uploaded assets, thumbnails, intersection videos, processing status files, output JSON files, tree and Sankey results, and report-related files.
- **Machine Learning Server – TASI22:** Reads the finalized scenario input JSON and uploaded videos from the NAS file server, performs vehicle detection, tracking, and matching, and writes processed outputs back to the scenario workspace.

This architecture separates the visualization layer, web server layer, backend service layer, shared storage layer, and machine learning processing layer. The front end remains focused on

user interaction and visualization, while the backend manages persistent data storage, file organization, scenario retrieval, processing status, JSON-based data exchange, and output access.

A key backend function is the creation of the finalized scenario input JSON file, which serves as the formal data contract between the web application and the machine learning service. After the JSON file is finalized, the machine learning service reads the JSON and associated videos from the NAS file server, performs detection, tracking, and matching, and writes the resulting outputs back to the scenario workspace. The backend then retrieves these outputs and provides them to the frontend for visualization, verification, and report generation. Figure B.1 shows the detailed architecture of the system and illustrates the interaction among the frontend, web server, application server, shared NAS file storage, machine learning service, and analytical output modules.

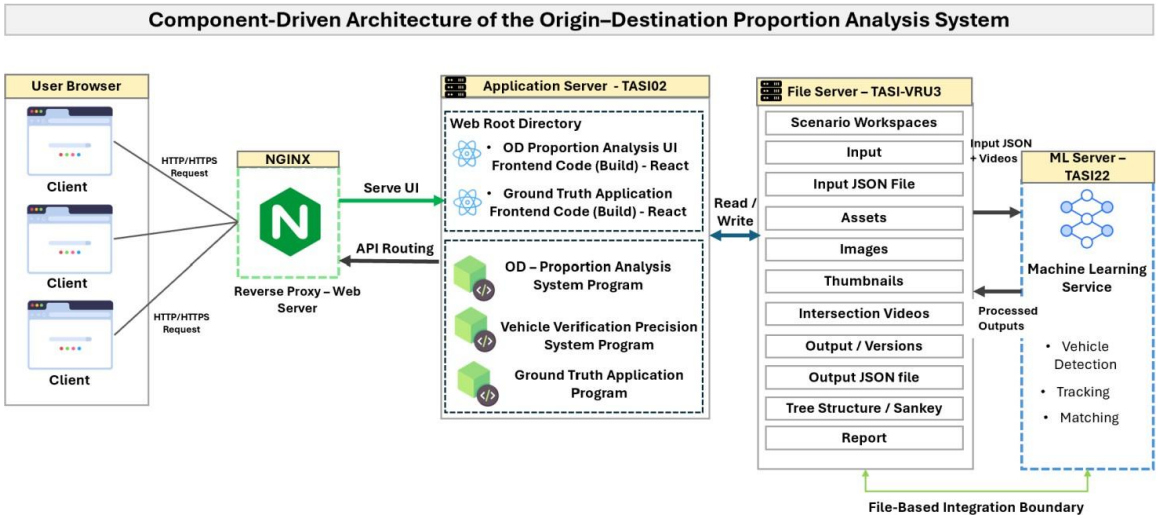


Figure B.1 Component-Driven Architecture of the Origin–Destination Proportion Analysis System.

2. Scenario Workspace and File Server Structure

The OD Proportion Analysis system uses a scenario-based file organization structure to support persistent storage, traceability, and reproducible analysis. All scenario-related data are stored within the main project data repository, JTRP_5019_Data_Store, located on the shared NAS file server. Within this repository, each scenario is assigned an independent workspace folder using a scenario identifier, timestamp, and scenario type. This structure ensures that user inputs, uploaded assets, processing status files, ground truth data, machine learning outputs, and report files remain organized under the corresponding scenario (Figure B.2).

The scenario workspace is designed to separate raw inputs, user-generated annotations, intermediate processing records, and final analytical outputs. This separation is important because the system involves multiple processing stages, including frontend scenario configuration, video upload, region-of-interest annotation, machine learning processing, ground truth review, vehicle

verification, and report generation. By maintaining these files in a structured workspace, the backend can reliably retrieve scenario information, monitor processing status, and return the required outputs to the frontend interface. The top-level structure of the scenario data store is organized as follows:

The scenario workspace is organized using a standardized naming convention to support traceability, processing control, validation, and report generation. Each scenario is stored under the main project data repository using a folder name that includes the scenario identifier, creation date, creation time, and scenario movement type. The folder naming format follows the structure **[SCENARIO_ID]_[YYYYMMDD]_[HHMM]_[ENTRY_OR_EXIT],** where **SCENARIO_ID** represents the unique scenario number, YYYYMMDD records the creation date, HHMM records the creation time, and ENTRY_OR_EXIT identifies the scenario type. In this system, Entry represents a merge-type scenario, while Exit represents a split-type scenario. For example, a merge scenario created on June 14, 2026, at 2:30 PM may be stored as **SC0001_20260614_1430_Entry**, while a split scenario created later may be stored as **SC0002_20260614_1510_Exit.**

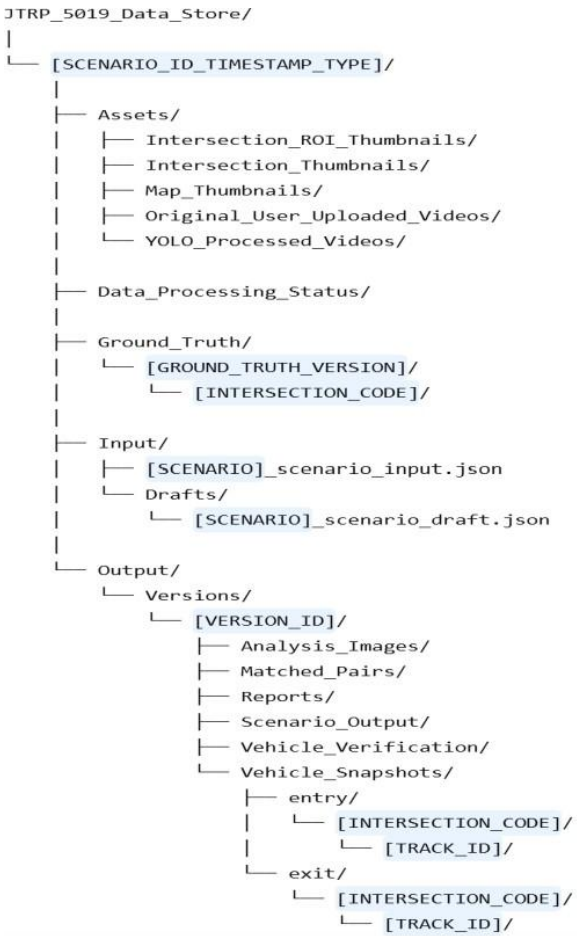


Figure B.2 Scenario Workspace and NAS File Server Directory Structure.

Within each scenario workspace, each directory and file has a defined role in the analysis workflow:

- **Assets:** Stores visual and video-based files used for scenario setup, review, and machine learning processing. This includes original uploaded videos, YOLO-processed videos, map thumbnails, intersection thumbnails, and ROI thumbnails.
- **Input:** Stores scenario configuration files. Draft JSON files preserve intermediate user progress, while the finalized scenario input JSON file serves as the formal input file used by the machine learning service. This file defines the scenario metadata, roadway structure, video references, intersection labels, and annotation information.
- **Data_Processing_Status:** Stores status JSON files that describe the current processing stage of the scenario. These files allow the backend and frontend to determine whether a scenario has been created, whether machine learning processing has started, whether processed videos are available, and whether final outputs have been generated.
- **Ground_Truth:** Stores human-reviewed annotation data organized by ground truth version and intersection code. This structure allows multiple review or validation cycles to be preserved for comparison with machine learning results.
- **Output:** Stores machine-learning-generated results and report-ready files. Output files are organized by version so that updated processing runs can be stored without overwriting previous results. This includes matched vehicle pairs, scenario output JSON files, verification results, analysis images, vehicle snapshots, and generated reports.

This folder and file organization provides a clear role for each stored item in the workflow. For researchers and INDOT reviewers, the structure makes it possible to trace final OD proportion results back to the original videos, scenario input JSON, user annotations, processing status records, ground truth files, and machine learning output files.

3. Backend Program Logic and API Workflow

The backend program serves as the operational control layer of the Origin–Destination Proportion Analysis system. Implemented using Python FastAPI, it connects the React frontend, NAS-based scenario file store, machine learning service, vehicle verification module, and report-generation workflow. Its primary role is to convert user actions into structured scenario files, JSON records, processing-status updates, analysis outputs, and report-ready results.

The backend is organized into two main services. The main backend supports scenario creation, file uploads, workspace creation, JSON storage, processing-status tracking, output retrieval, and report generation. The verification backend supports matched-pair review, manual verification, progress saving, and verification-result storage. In production, the main backend is routed through port 5011, while the verification backend is routed through port 5012.

4. Scenario Initialization and Workspace Creation

The backend workflow begins when the frontend requests a new scenario identifier. The backend scans the existing scenario data store, determines the highest existing scenario number, and returns the next available scenario ID. After the user confirms the scenario location, movement type, and related metadata, the backend creates a scenario-specific workspace on the NAS file server.

Each workspace follows the standardized folder naming convention described in Section 2, such as SC0005_20260604_1149_Entry. The workspace becomes the authoritative storage location for all files associated with the scenario, including inputs, assets, status records, ground truth files, machine learning outputs, and reports.

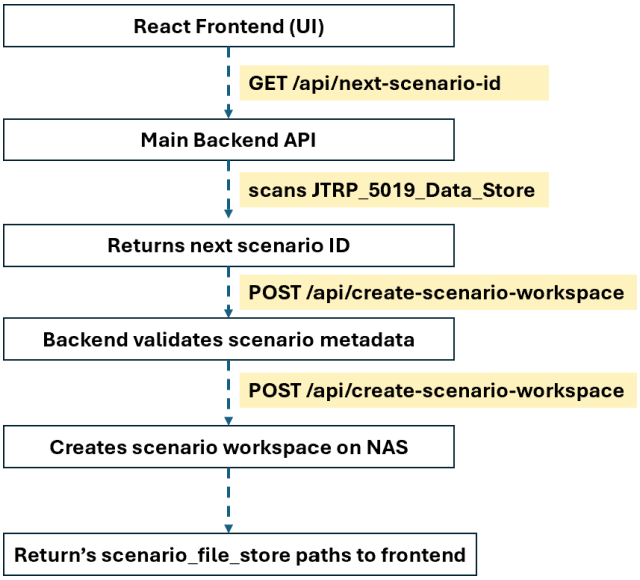


Figure B.3 Backend Workflow for Scenario Initialization and NAS (File Server) Workspace Creation.

5. JSON-Based Payload Storage and Data Exchange

The Origin-Destination Proportion Analysis system uses JSON files, or JSON payloads, as the primary file-based data storage method for quick data retrieval and structured data exchange. Instead of depending on a traditional relational database, the project stores scenario information, machine-learning outputs, verification data, and processing status records as JSON files within each scenario workspace.

This file-based JSON approach supports scenario-level traceability because each scenario contains its own input files, output files, status files, verification files, and report-ready data. The backend can quickly read and write these JSON files to retrieve scenario details, display processing progress, load analysis outputs, and prepare report data for the frontend.

The **four** main JSON data files used by the system are:

- **User-Entered Input JSON File:** Generated by the UI Backend Program. This file stores the scenario details entered through the web interface, including location details, scenario type, road labels, intersection labels, video references, ROI data, and user notes. This file serves as the main input JSON file for the Machine Learning Program.
- **Machine Learning Output JSON File:** Generated by the Machine Learning Program. This file stores the processed analysis results, including vehicle counts, movement relationships, OD analysis outputs, and data used for the final analysis report.
- **Matched Vehicle Pair JSON File:** Generated by the Machine Learning Program. This file stores candidate matching vehicle pairs produced by the vehicle matching process. The Verification UI reads this file and displays the matched vehicle images for human review.
- **Verified_Vehicles JSON File:** Generated by the UI Backend Program after human review in the Verification UI. This file stores the verification results for matched vehicle pairs, including whether each pair is classified as a true positive match or false positive match.

In addition to these main data JSON files, the system also uses a Data Processing Status JSON File. This file is generated by the Machine Learning Program and is used by the Data Processing Status page to show the current processing stage, progress updates, completed steps, and output readiness. The backend reads this file and provides status updates to the frontend so users can monitor the processing workflow.

Using JSON payloads as the project's file-based data layer allows the system to remain lightweight, transparent, and easy to debug. It also allows researchers and developers to directly inspect scenario inputs, machine-learning outputs, verification results, and processing status records without requiring database queries.

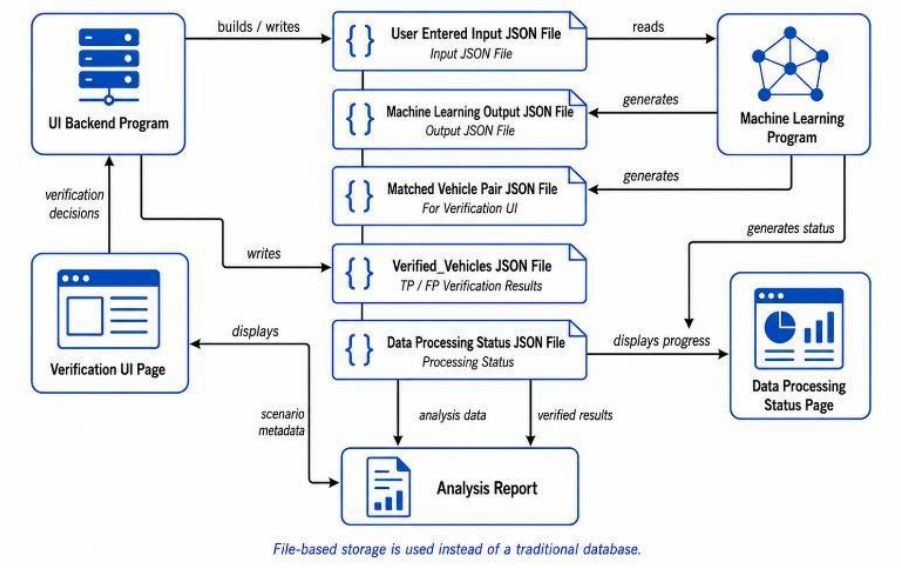


Figure B.4 JSON-Based Payload Files and System Associations.

6. Project Core Dependencies

The backend system was resolved and tested using Python 3.12. The project dependencies support API routing, file upload handling, image processing, report generation, and backend service execution.

The main direct runtime dependencies include:

- **FastAPI 0.115.14:** Provides the REST API framework used by the backend services.
- **Uvicorn 0.34.3:** Runs the FastAPI backend programs as ASGI web services.
- **aiofiles 25.1.0:** Supports asynchronous file reading and writing for scenario files and outputs.
- **python-multipart 0.0.21:** Enables file uploads from the frontend, including videos, thumbnails, and other assets.
- **Pillow 11.3.0:** Supports image processing and thumbnail handling.
- **Jinja2 3.1.6:** Supports template-based report generation.
- **WeasyPrint 68.1:** Converts HTML-based report content into PDF output.
- **xhtml2pdf 0.2.17:** Provides additional PDF generation support. Main PDF library.

The project also includes transitive dependencies required by these core packages, including libraries for request handling, data validation, HTML parsing, PDF processing, font handling, cryptography, timezone support, and HTTP communication. Examples include Starlette, Pydantic, ReportLab, pypdf, requests, urllib3, lxml, html5lib, tinycss2, fonttools, and cryptography. Maintaining these dependencies in the project requirements file allows the backend environment to be recreated consistently on the application server and helps reduce deployment issues caused by version conflicts.

7. Deployment Architecture Overview

The OD Proportion Analysis System is deployed as a distributed, multi-component software environment that integrates web-based user interfaces, backend services, machine-learning processing programs, and shared file storage. The deployment architecture is designed to support the complete analysis workflow, including scenario configuration, video upload, roadway and intersection labeling, region-of-interest annotation, machine-learning-based vehicle detection and tracking, vehicle matching, verification, ground-truth review, and report generation.

The system is organized into four major software programs:

- OD Proportion Analysis UI Program and Backend
- Ground Truth UI Backend Program
- Vehicle Verification UI Program
- Machine Learning Service

These programs are deployed across a Linux virtual machine, a GPU-enabled workstation, and a shared Synology NAS file server TASI-VRU3. The Linux virtual machine TASI02 hosts web-facing applications and backend services, while the Alienware Aurora workstation TASI22 hosts the computationally intensive Machine Learning Service. The Synology NAS file server provides the central storage and data-exchange layer used by all programs.

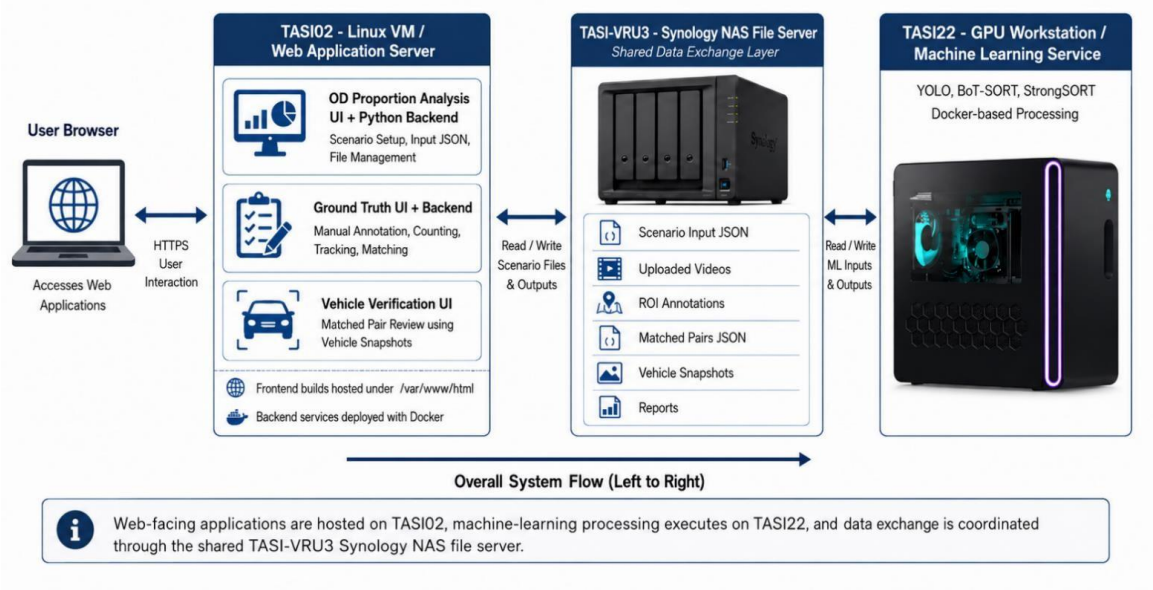


Figure B.5 OD Proportion Analysis System Deployment Architecture.

This deployment model separates web application responsibilities from machine-learning computation and long-term data storage. As a result, each component can be maintained, updated, and scaled independently while still participating in the same end-to-end OD Proportion Analysis workflow.

8. Web Application Server Environment

The web application components are hosted on the TASI02 Linux virtual machine. NGINX is used as a web server and reverse proxy layer. It serves the compiled frontend builds from the web directory /var/www/html. NGINX also routes API requests from the browser to the appropriate backend services running in Docker containers. This allows the frontend applications to remain browser-accessible while backend services operate in separate containerized environments.

9. Machine Learning Service Deployment

The Machine Learning Service is deployed on the TASI22 Alienware Aurora workstation, which includes an NVIDIA RTX GPU for accelerated video processing. This service contains the computer-vision workflow used for YOLO-based vehicle detection, vehicle tracking using algorithms such as BoT-SORT and StrongSORT, and cross-location vehicle matching. The Machine Learning Service is deployed using Docker to maintain a consistent runtime environment for model execution and video processing. During processing, it reads input files from the shared scenario workspace on TASI-VRU3 and writes processed videos, vehicle snapshots, matched-pair results, and analysis outputs back to the same shared storage environment.

10. Shared File Server and Data Exchange Layer

The TASI-VRU3 Synology NAS file server functions as the central data-exchange layer for the system. Instead of requiring direct communication between every program, the system uses a file-based integration approach where each component reads and writes standardized files within the scenario workspace.

11. Maintainability and Future Extension

The deployment design improves maintainability by separating the system into modular components with clear responsibilities. Computer TASI02 manages web hosting, API routing, and backend services. Computer TASI22 handles GPU-based machine-learning processing. TASI-VRU3 provides centralized storage and scenario-level data exchange. This separation allows future frontend features, backend APIs, machine-learning models, verification tools, and reporting functions to be added without redesigning the complete system architecture.

About the Joint Transportation Research Program (JTRP)

On March 11, 1937, the Indiana Legislature passed an act which authorized the Indiana State Highway Commission to cooperate with and assist Purdue University in developing the best methods of improving and maintaining the highways of the state and the respective counties thereof. That collaborative effort was called the Joint Highway Research Project (JHRP). In 1997 the collaborative venture was renamed as the Joint Transportation Research Program (JTRP) to reflect the state and national efforts to integrate the management and operation of various transportation modes.

The first studies of JHRP were concerned with Test Road No. 1—evaluation of the weathering characteristics of stabilized materials. After World War II, the JHRP program grew substantially and was regularly producing technical reports. Over 1,600 technical reports are now available, published as part of the JHRP and subsequently JTRP collaborative venture between Purdue University and what is now the Indiana Department of Transportation.

Free online access to all reports is provided through a unique collaboration between JTRP and Purdue Libraries. These are available at docs.lib.purdue.edu/jtrp/.

Further information about JTRP and its current research program is available at engineering.purdue.edu/JTRP.

About This Report

An open access version of this publication is available online. See the URL in the citation below.

Chien, S. Y.-P., Chen, Y., Kumar, S., & Desai, S. (2026). *Vehicle moving pattern analysis of road merge and split weaving areas involving highway and local streets* (Joint Transportation Research Program Publication No. FHWA/IN/JTRP-2026/27). West Lafayette, IN: Purdue University. https://doi.org/10.5703/1288284318711