Functional & Performance Testing

1. Test Cases Executed

To ensure the robustness and correctness of the system, multiple test cases were designed and executed across data handling, model prediction, and output generation. The following scenarios were tested:

* Data Ingestion: Verified the correct loading of traffic datasets from various formats such as CSV and Excel.

* Data Cleaning: Ensured that the system accurately handled missing or null values by removing or imputing them.

* Model Training and Prediction: Confirmed that machine learning models could be trained with clean datasets and accurately predict traffic volume.

* Input Handling: Tested the system's ability to process both valid and invalid inputs gracefully, returning either accurate results or meaningful error messages.

* Visualization: Checked the generation of various plots and charts (traffic volume trends, time-based analysis) for correctness and readability.

* Result Exporting: Tested the export functionality to ensure that analysis results could be saved in formats like CSV or PDF.

All critical test cases were successfully passed, demonstrating the functional stability of the system.

---

2. Bug Fixes and Improvements

During development and testing, several bugs and performance issues were identified and resolved:

*Encoding Issues in Dataset: Certain datasets failed to load due to incompatible encodings. This was resolved by specifying the correct file encoding (UTF-8 or ISO-8859-1).

* Low Model Accuracy: Initially, the model performed poorly on edge cases. Improvements were made by enhancing the feature set, including time-of-day and weekday/weekend flags.

* Visualization Overlap: Some charts appeared cluttered. Adjustments were made to figure dimensions and label formatting.

*Slow Frame Processing: High-resolution videos caused performance delays. The resolution was reduced, and lightweight object detection models (e.g., MobileNet) were used.

*Error Handling: The system initially crashed on receiving null or invalid inputs. Exception handling was implemented to manage these gracefully.

---

3. Final Validation

The final product was evaluated against the initial project goals and requirements:

* Accurate Traffic Volume Estimation: Achieved through the implementation of machine learning models trained on real or simulated traffic datasets.

* Use of Advanced Tools and Technologies: Python was used along with relevant libraries such as Scikit-learn, Pandas, OpenCV, and Flask/FastAPI.

* Visualization and Reporting: Clear, informative graphs and summaries were generated for analysis and presentation purposes.

* Usability and Modularity: The system architecture is modular, scalable, and easy to maintain or extend.

* Optional Real-Time Features: Real-time processing through video analysis and API endpoints was prototyped successfully.

Overall, TrafficTelligence meets and exceeds the initially defined functional and technical requirements, demonstrating a viable solution for intelligent traffic volume estimation using machine learning.