Team Name : **Artemis**

Team Members :

1. Rajesh Yepa (Team Leader): An undergraduate student in the stream of Computer Science and Engineering from Kakatiya Institute of Technology and Science, interested in exploring the field of Artificial Intelligence and Machine Learning.
2. Rishitha Reddy Gopu: An undergraduate student in the stream of Computer Science and Engineering from Kakatiya Institute of Technology and Science interested in the field of Networks, Machine Learning.

Problem Statement:

Bengaluru, often referred to as the "Silicon Valley of India," is a bustling metropolis known for its thriving tech industry, vibrant culture, and pleasant climate. However, along with its rapid growth and urbanization, Bengaluru has become infamous for its severe traffic congestion. The city's road network, originally designed for a much smaller population, now struggles to accommodate the ever-increasing number of vehicles.

The traffic situation in Bengaluru is characterized by long commute times, frequent traffic jams, and slow-moving traffic, especially during peak hours. Major roads and junctions often experience gridlock, with vehicles inching along at frustratingly low speeds. The city's narrow roads, combined with the high volume of cars, bikes, buses, and trucks, contribute to the congestion. Additionally, ongoing construction projects and inadequate public transportation options exacerbate the problem.

This hackathon aims to tackle this issue by encouraging innovative solutions for better traffic management in the city. The event is co-sponsored by the Bengaluru Traffic Police, the Centre for Data for Public Good, and the Indian Institute of Science (IISc). Participants will work with live camera feeds from 23 cameras in northern Bengaluru, near the IISc campus. Their challenge is to predict vehicle counts and turning patterns at specific road junctions up to 30 minutes in advance, even at locations not directly covered by the cameras.

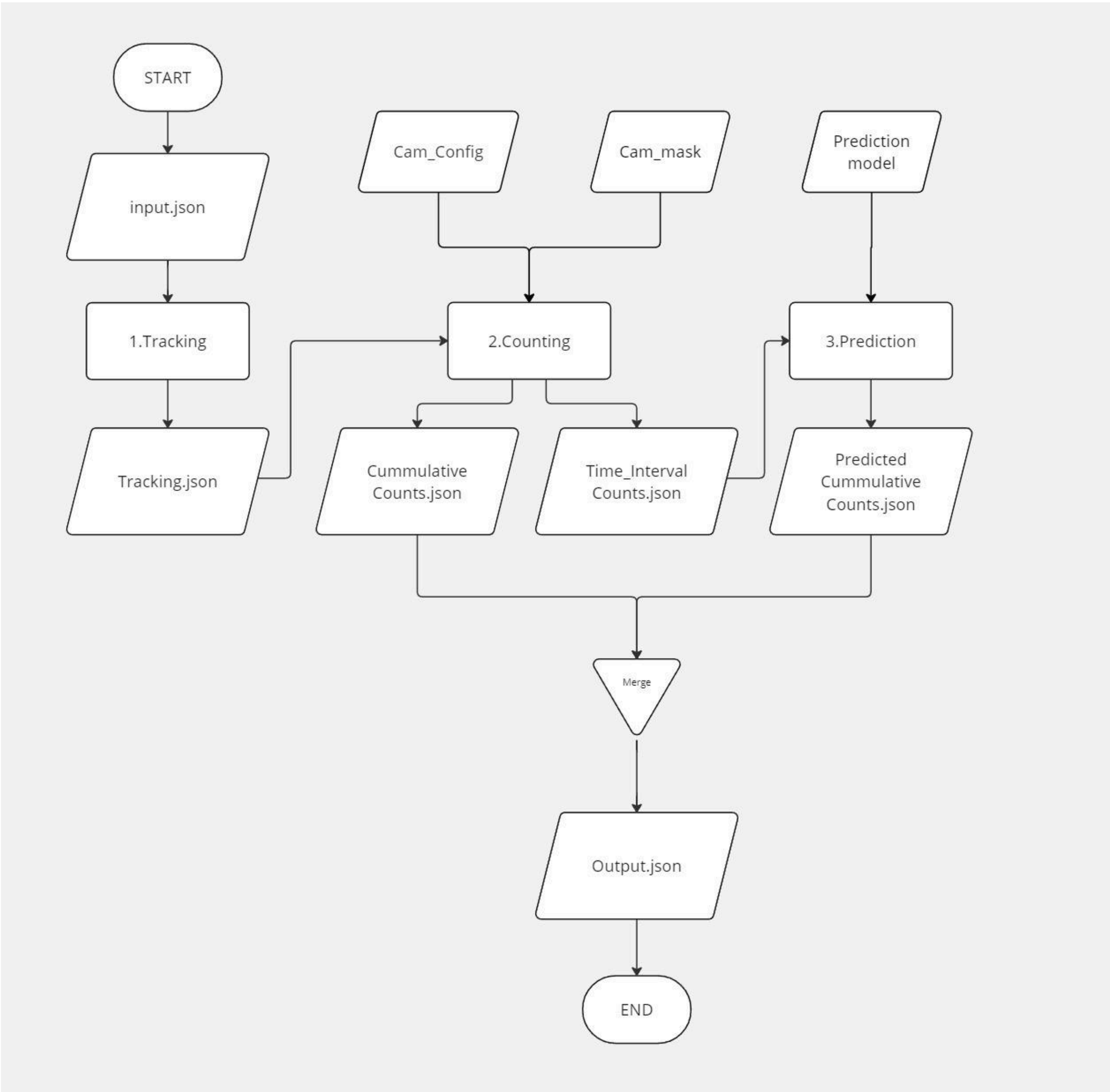
The challenge lies not only in making these predictions for the immediate locations covered by the cameras but also in extrapolating this information to predict traffic at nearby locations that are not directly monitored. These predictions could play a crucial role in optimizing traffic flow, reducing congestion, and improving the overall efficiency of the city's traffic management systems.

METHODOLOGY

1. Vehicle Detection

The first step in the process involves detecting vehicles in video footage. For this task, we utilized the YOLOv10x model from Ultralytics, which is well-suited for real-time object detection tasks. The YOLOv10x model was trained using a custom dataset created with the Roboflow tool, which provided annotated images of various vehicle classes, including Bicycle, Bus, Car, Light Commercial Vehicle (LCV), Three-Wheeler, Truck, and

Two-Wheeler. The robust training of the model allowed for high precision and recall in detecting these vehicle classes in the input video frames.



Flowchart of the proposed model

2. Vehicle Tracking

Once the vehicles were detected, the next step involved tracking them across the video frames. For this, we employed the BoTSORT algorithm, which is designed for high-performance multi-object tracking. The BoTSORT algorithm was used in conjunction with the Clip-Vehicle.pt weights from the BoxMOT library, which are specifically optimized for vehicle re-identification tasks. This combination enabled us to maintain consistent tracking of each vehicle across the entire duration of the video, even in cases where the vehicles temporarily exited the camera's field of view or were occluded by other objects.

3. Vehicle Movement Assignment

After tracking the vehicles, the next step was to assign their movements to predefined typical trajectories. These trajectories were defined in camera configuration files, which were created based on historical tracking data from the same camera. The trajectories represent the common paths that vehicles follow as they move through the intersection, including various turning patterns.

To match the tracked vehicle trajectories with the predefined trajectories, we employed the Hausdorff distance, which measures the degree of similarity between two sets of points. Additionally, we considered the direction and angle of vehicle movement to ensure accurate assignment to the correct turning pattern. The vehicle movements were then categorized according to their assigned turning patterns.

4. Vehicle Counting

Vehicle counting was performed based on the assigned trajectories. For each vehicle class and turning pattern, we maintained a cumulative count of vehicles that passed through the intersection during the video. The counting was done in time intervals of 15 seconds, which allowed us to capture fine-grained temporal variations in vehicle traffic. The results were compiled into a dataset that recorded the number of vehicles for each class and turning pattern at each 15-second interval.

5. Training a Spatio-Temporal Graph Neural Network

With the vehicle counts aggregated into time intervals, the next step was to predict future vehicle counts. For this, we trained a spatio-temporal graph neural network (GNN). The GNN model was designed to take the vehicle counts from the previous 30 minutes and predict the cumulative vehicle counts for the next 30 minutes.

The dataset for training the GNN was created by processing 6 hours of video data, which was divided into 15-second intervals, resulting in a total of 1440 intervals. Each interval contained the vehicle counts for each class and turning pattern, which were used as the input features for the GNN model. The GNN model architecture included Graph Convolutional Network (GCN) layers to capture spatial dependencies between the different trajectories and a Closed-form Continuous-time (CfC) layer, which replaced traditional LSTM layers to model temporal dependencies more effectively. The CfC layer, inspired by liquid neural networks, allowed the model to handle the continuous nature of vehicle movements.

6. Predicting Future Vehicle Counts

The trained GNN model was then used to predict the future vehicle counts. The predictions were made for each vehicle class and turning pattern over the next 30 minutes, based on the historical data from the previous 30 minutes. The predictions were then aggregated to obtain the cumulative predicted counts for each vehicle class and turning pattern.

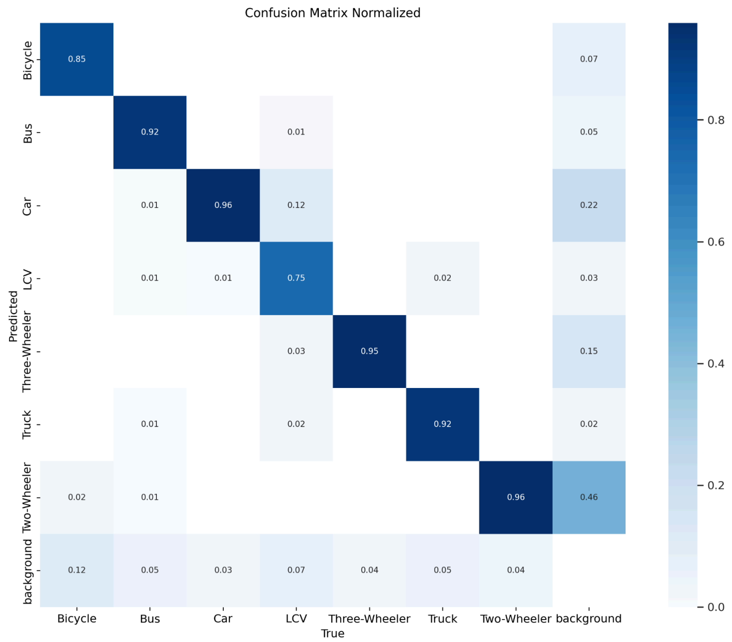
7. Compiling the Results

Finally, the cumulative counts and predicted counts were compiled into a single output JSON format. The input data structure consisted of two continuous 15-minute video segments, which were processed to extract the observed cumulative counts. The cumulative counts and the predicted counts for each vehicle class and turning pattern were then combined into the final output JSON structure.

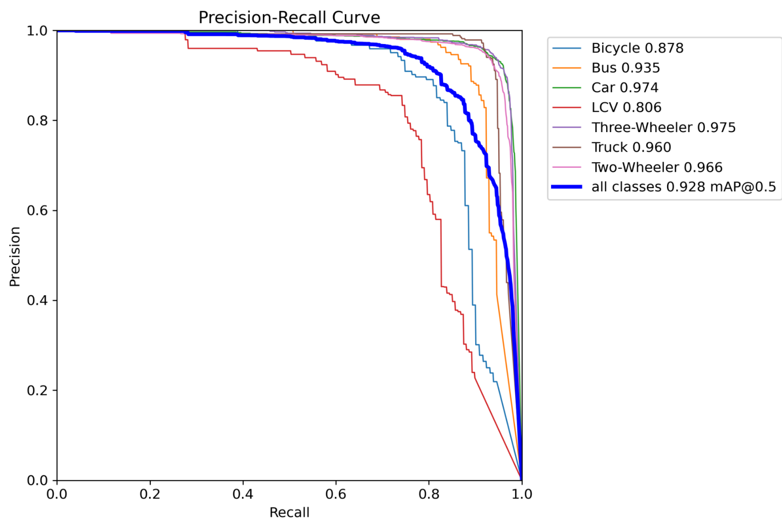
In summary, the methodology involved a comprehensive pipeline for vehicle detection, tracking, trajectory assignment, counting, and future count prediction, all integrated into a spatio-temporal GNN framework. The final output provided both historical and future vehicle counts, enabling a detailed analysis of traffic patterns at the intersection.

RESULTS

1.MODEL TRAINING



Confusion matrix



PR curve

CONCLUSIONS

Bengaluru's traffic congestion represents a significant challenge that affects the daily lives of its residents, the city's economy, and the environment. While various measures have been taken to address the issue, the scale and complexity of the problem require innovative and data-driven solutions. The hackathon focused on traffic prediction is a promising initiative that brings together technology, expertise, and creativity to develop practical solutions for managing Bengaluru's traffic more effectively. By harnessing the power of predictive analytics and collaborative problem-solving, there is hope for a future where Bengaluru's roads are safer, more efficient, and less congested, ultimately improving the quality of life in this vibrant city.

REFERENCES

[1] Lu, J., Xia, M., Gao, X., Yang, X., Tao, T., Meng, H., Zhang, W., Tan, X., Shi, Y., Li, G., & Ding, E. (2021). Robust and Online Vehicle Counting at Crowded Intersections. CVPR 2021 Workshop. https://doi.org/10.1109/cvprw53098.2021.00451

[2] C. Zhang, J. J. Q. Yu and Y. Liu, "Spatial-Temporal Graph Attention Networks: A Deep Learning Approach for Traffic Forecasting," in IEEE Access, vol. 7, pp. 166246–166256, 2019, doi: 10.1109/ACCESS.2019.2953888.