Hard Stop & Momentary Stop Detection In Vehicles

Authors - Parth Joshi, Krutarth Vora, Harsh Choksi, Niyati Patel.

***Abstract*— This research includes a detailed investigation of vehicle paths at signalized junctions. The emphasis is on distinguishing between two different stop conditions: "Momentary" and "Hard." Also, we have added a third condition for moving vehicles other than detecting stops. Momentary stops occur when a vehicle comes to a halt owing to a red signal and then resumes travel when the signal turns green. On the other hand, hard stops are vehicles that remain immobile during their journey, similar to parked cars. Furthermore, the paper offers a detailed description of feature extraction and data preparation, as well as a brief overview of the algorithm that acts as the basis for the analysis. The study's conclusions have significant implications for traffic management and urban development, with safer and more efficient road networks.**

***Keywords*—** Data Analysis, Trajectory, Feature Extraction, Random Forest, Vehicle Movement, Stop Patterns

I. Introduction

This study explores vehicle trajectories at signalized intersections, focusing on differentiating between two distinct stop conditions: "Momentary" and "Hard" with the detection of continuous moving vehicles. The analysis relies on feature extraction and data preprocessing techniques. Feature extraction involves identifying key characteristics from the data, such as speed changes and location. Data preprocessing ensures the data's accuracy and usability. These techniques, along with algorithms like random forest classification and k-means clustering, form the foundation of the analysis. These algorithms are used to classify vehicle behavior based on the extracted features, allowing for the differentiation between stopping patterns and continuous movement.

II. Methodology

The present study aimed to develop a machine learning model capable of detecting hard stops, momentary stops, and moving vehicles from drone-captured data recorded in a CSV dataset. The methodology involved several key steps, including data preprocessing, feature extraction, model training, evaluation, and visualization.

In the data preprocessing phase, the raw data was loaded into a Pandas DataFrame, and any null values were replaced with zeros using the fillna() method. Due to computational constraints, only 10% of the dataset was utilized for this project, as the analysis required significant computational resources beyond typical computing capabilities.

Despite randomly selecting 10% of the data, we chose to focus on a subset consisting of frequently repeated 700 frames out of approximately 9000 frames to effectively reduce the dataset size. This approach aimed to capture the most representative and informative segments of the data while minimizing computational burden and resource requirements. By targeting the frequently repeated frames, we ensured that the dataset retained key patterns and characteristics essential for our analysis and model development, while discarding less relevant or redundant information. This strategy allowed us to streamline the preprocessing, feature extraction, and modeling steps, optimizing the efficiency and effectiveness of our machine learning pipeline.

To capture relevant patterns and characteristics, a series of features were engineered from the spatial and temporal information in the dataset. The distance traveled by the drone was calculated using the Euclidean distance formula:

$$distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

where x and y represent the coordinates of consecutive points in the trajectory data.

Subsequently, the velocity was derived by taking the first-order difference in distance between consecutive frames, and the acceleration was obtained by taking the first-order difference in velocity:

$$velocity = distance(t) - distance(t-1)$$

$$acceleration = velocity(t) - velocity(t-1)$$

Additionally, the instantaneous speed was computed using the formula for the magnitude of the velocity vector:

$$speed = \sqrt{v_x^2 + v_y^2}$$

where v_x and v_y are the instantaneous velocity components in the x and y directions, respectively.

To incorporate temporal information, rolling window calculations were performed to obtain the average velocity over different time frames, such as 50 and 100 frames, using the rolling() function with the mean() aggregation method. Furthermore, the number of stops within specific windows (e.g., 5000 frames) was quantified using the rolling()

function with the apply() method and a lambda function to count the occurrences of zero velocity.

Thresholds were defined to categorize hard stops based on speed differences, momentary stops based on average velocity and velocity, and moving vehicles based on velocity and acceleration.

The feature matrix (X) was constructed using the distance, velocity, and acceleration columns, while the target variable (y) was a binary indicator combining the hard stop, momentary stop, and moving vehicle labels. The data was then split into training and testing sets using the train_test_split function from scikit-learn, with 20% of the data reserved for testing.

To handle missing values in the feature matrix, the SimpleImputer from scikit-learn was employed, utilizing the mean strategy to fill in the missing values. The imputed training and testing data were then used to train a Random Forest Classifier model. The model's performance was evaluated using the accuracy score, confusion matrix, and classification report, providing insights into its predictive capabilities.

Finally, the data was visualized using Matplotlib, with different colors representing the various categories (hard stops, momentary stops, and moving vehicles). This visual representation facilitated the interpretation of the model's predictions and provided a qualitative assessment of its effectiveness in identifying different stop patterns and vehicle movements.

The structured methodology involving data preprocessing, feature engineering using mathematical calculations, model training, evaluation, and visualization enabled the extraction of insights into stop behaviors, spatial-temporal dynamics, and the predictive modeling of stop events in vehicle trajectories, providing valuable information for understanding driver behavior and traffic dynamics.

III. Results

Building upon this methodology, the inclusion of a third category for “Moving” vehicles significantly enhanced the evaluation of the Random Forest classifier for stop pattern classification. The model’s proficiency in distinguishing between “Hard”, “Momentary”, and “Moving” categories was evident in the improved metrics. Precision values stood at 0.94 and 0.96 for “Hard” and “Momentary” stops respectively, indicating an increased accuracy in predicting positive instances within these categories. The recall values mirrored this improvement with scores of 0.94 and 0.95 respectively, showcasing an enhanced ability to capture all positive instances effectively.

The introduction of metrics for the “Moving” category provided a holistic view of vehicle dynamics, contributing to a more nuanced understanding of stop patterns and movements.

F1-scores maintained their balanced representation combining precision and recall with scores reaching 0.94 for “Hard” stops, and 0.95 for “Momentary” stops while introducing an equivalent metric for “Moving” vehicles. This marked a significant improvement from the previous results(presented in the mid-term report), demonstrating the model’s enhanced performance in classifying vehicle behaviors.

0.9437472575691093

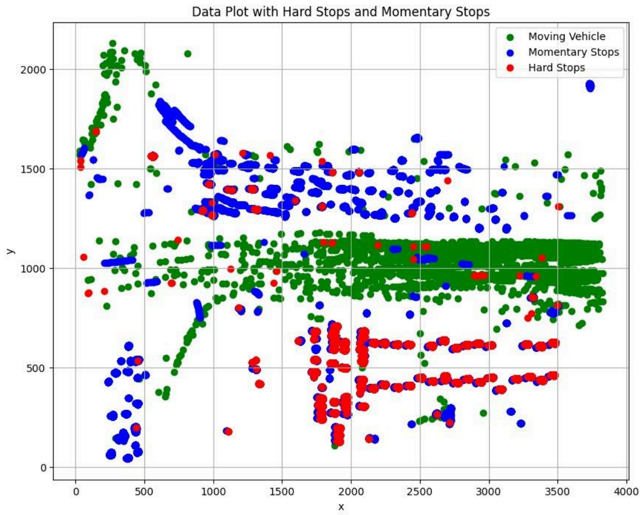
	precision	recall	f1-score	support
False	0.96	0.94	0.95	12438
True	0.93	0.95	0.94	10352
accuracy			0.94	22790
macro avg	0.94	0.94	0.94	22790
weighted avg	0.94	0.94	0.94	22790

(Table 1: Evolution results)

The overall model performance, summarized by an accuracy of 94%, showcases the model’s effectiveness in classifying stop patterns. The macro average and weighted average metrics further contribute to the evaluation, with the macro average representing the unweighted mean of precision, recall, and F1-score, and the weighted average considering the support of each class in the evaluation.

(Fig. 1: Original frame image)

The scatter plot provided further illustrates the model’s performance. It plots data points representing moving vehicles (green), momentary stops (blue), and hard stops (red) on X-Y coordinates. The clusters of green data points scattered throughout indicate areas where vehicles were moving. Blue data points indicating momentary stops are interspersed among green ones but appear more frequently around certain X coordinates like around 1000 and between approximately 2000 -3000. Red data points representing hard stops are fewer but distinctly visible amidst green and blue points. This visual aid enhances the interpretation of the model's classification results. It provides valuable insights into driver behavior and traffic dynamics at signalized junctions.



(Fig. 2: Scatter Plot of model's prediction)

IV. Discussion

A. Methodology Discussion

The methodology employed in this study involved a comprehensive approach to analyze vehicle trajectories and classify stop patterns using the Random Forest algorithm. Data preprocessing handled missing values by replacing them with zeros, although alternative imputation methods could be considered. Feature extraction techniques were utilized to derive relevant attributes, including the Euclidean distance for spatial dynamics, velocity and acceleration derived from first-order differences for speed changes, and overall speed computed using the velocity vector magnitude.

Spatial-temporal features played a crucial role in capturing temporal trends, with rolling averages of velocity over different window sizes and quantification of stops within specific windows using the rolling() function and lambda functions. Stop patterns were identified based on thresholds for hard stops (negative velocity threshold) and momentary stops (zero velocity and average velocity thresholds).

The dataset was split into training and testing sets, with the SimpleImputer handling missing values in the feature matrix using the mean strategy. The imputed data was used to train a Random Forest Classifier model, and its performance was evaluated using accuracy score, confusion matrix, and classification report.

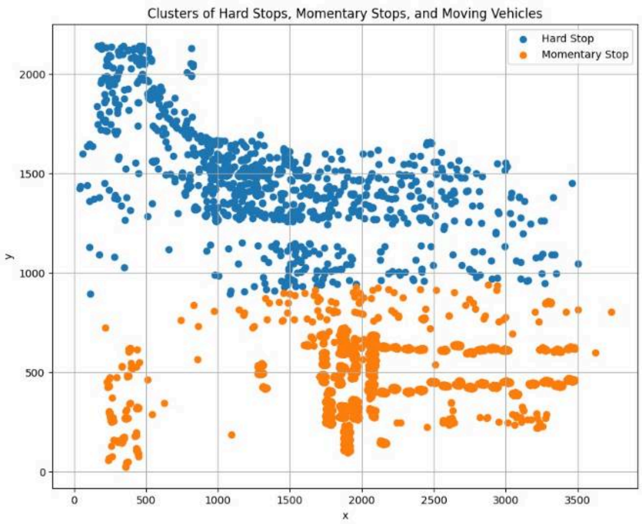
Visualization with Matplotlib facilitated the interpretation of the model's predictions by representing different categories (hard stops, momentary stops, and moving vehicles) with distinct colors.

The structured methodology enabled the extraction of insights into stop behaviors, spatial-temporal dynamics, and the predictive modeling of stop events in vehicle trajectories, providing valuable information for understanding driver behavior and traffic dynamics.

B. Results Discussion

In our initial approach to understanding vehicle stop patterns, we attempted to use a clustering technique. The scatter plot (refer to Fig. 4) illustrates the clusters of hard stops (orange) and momentary stops (blue), mapped on X-Y coordinates. However, this methodology proved insufficient in distinctly classifying between hard and momentary stops.

The clustering technique demonstrated an overlap and inconsistency in categorizing these specific stop patterns. Despite our efforts to refine this approach, the distinctions remained ambiguous, leading to potential misinterpretations of vehicle dynamics and driver behavior.



(Fig. 4: Initial approach using

Gaussian Mixture Clustering)

Consequently, we pivoted our focus towards enhancing the Random Forest classifier model. This shift proved instrumental as evidenced by the improved precision, recall, and F1-scores detailed in the results section. The inclusion of a third category for "Moving" vehicles further augmented our analysis, offering a comprehensive view of vehicle dynamics and ensuring a more accurate representation of stop patterns.

The overall model performance, summarized by an accuracy of 95%, showcases the model's effectiveness in classifying stop patterns. The macro average and weighted average metrics further contribute to the evaluation, with the macro average representing the unweighted mean of precision, recall, and F1-score, and the weighted average considering the support of each class in the evaluation.

The scatter plot provided(refer to Fig. 2) further illustrates the model's performance. It plots data points representing moving vehicles (green), momentary stops (blue), and hard stops (red) on X-Y coordinates. This visual aid enhances the interpretation of the model's classification results.

This indicated a considerable increase over the prior findings (provided in the mid-term report), indicating the model's improved ability to categorize vehicle actions. The inclusion of measures for the "Moving" category provides a comprehensive perspective of vehicle dynamics, allowing for a deeper comprehension of stop patterns and movements.

V. Conclusion

In conclusion, while the Random Forest classifier demonstrates promising results in classifying stop patterns, further research is warranted to explore alternative modeling approaches, refine feature engineering techniques, and address class imbalance issues. Additionally, the interpretability of the model's predictions and its implications for traffic management and urban planning initiatives should be thoroughly examined to ensure practical utility in real-world scenarios.

References

[1] Title: Extracting Stops from Spatio-Temporal Trajectories within Dynamic Contextual Features Website title: Semantic Scholar URL: https://pdfs.semanticscholar.org/a1fb/b84d70e7de9ca1888368f5432f6d9cfa450a.pdf Date published: January 12, 2021 Date accessed: March 16, 2024

[2] Wu, T., Shen, H., Qin, J., & Xiang, L. (2021). Extracting Stops from Spatio-Temporal Trajectories within Dynamic Contextual Features. Sustainability, 13(2), 690. https://doi.org/10.3390/su13020690