

Hybrid Motion Metric Framework for Video-Based Vehicle Speed Estimation

Miss. Gayatri S. Chaudhari, Prof. Pragati B. Chandane, Prof. Jagruti R. Mahajan, Dr. Hemantkumar B. Jadhav, Dr. Pradeep M. Patil

*Dept. of Computer Engineering
Adsul's Technical Campus, Chas,
Ahilyanagar, Maharashtra, India
chaudharig2002@gmail.com*

*Dept. of Computer Engineering
Adsul's Technical Campus, Chas,
Ahilyanagar, Maharashtra, India
pragatichandane3@gmail.com*

*Dept. of Computer Engineering
Adsul's Technical Campus, Chas,
Ahilyanagar, Maharashtra, India
mahajanjagruti@gmail.com*

*Dept. of Computer Engineering
Adsul's Technical Campus, Chas,
Ahilyanagar, Maharashtra, India
hem3577@gmail.com*

*Dept. of Computer Engineering
Adsul's Technical Campus, Chas,
Ahilyanagar, Maharashtra, India
drpmp66@gmail.com*

Abstract - The demand for safety and sustainable transportation brings innovation in vehicle technology. Bicycling is one of the oldest and fittest ways of transportation. A lot of research is going on self-balanced and autonomous two-wheeler vehicles to attract riders. An attempt is made here to propose the design components of autonomous self-balancing bicycles. Gyroscopes, widely used in stabilizing systems, emerge as a vital component in the development of self-balancing bicycles. This study focuses on use of gyroscopes and specifically designed flywheel, aimed at facilitating autonomous motion along both straight trajectories and curved paths. This paper illustrates how gyroscopes and flywheel maintain stability in self-balancing bicycles during tilting. Autonomous capability includes use of various sensors and camera to detect objects for navigation and use computer vision efficiently with the help of machine learning algorithms. The proposed design framework will help in a transformative era of innovation in urban mobility, promising safer and more sustainable ways of getting around in cities. With the rapid growth of intelligent transportation systems, monitoring vehicle speed efficiently and accurately has become essential. Traditional speed measurement techniques rely on intrusive sensors or expensive radar-based systems, which are difficult to install and maintain. This paper proposes a vision-based vehicle speed estimation system that uses traffic surveillance cameras without requiring additional hardware sensors. The proposed system employs two speed estimation models: Distance-based model

Time-based model.

Advanced computer vision techniques including object detection (YOLO), tracking (DeepSORT), and optical flow are integrated to extract motion parameters. Experimental evaluation shows that the system achieves Mean Absolute Error (MAE) below 4 km/h, making it suitable for real-world applications.

Keywords: Vehicle Speed Estimation, Computer Vision, YOLO, DeepSORT, Optical Flow, Machine Learning, Traffic Monitoring

1.INTRODUCTION

Vehicle speed monitoring is a critical component of road safety, traffic management, and law enforcement. Conventional approaches are broadly classified into:

- Intrusive methods (loop detectors, piezo sensors)
- Non-intrusive methods (radar, laser guns)

Although effective, these systems involve high installation costs and maintenance complexity.

With the widespread deployment of CCTV cameras, there is an opportunity to utilize existing infrastructure for speed estimation. Vision-based systems provide:

- Cost efficiency
- Scalability
- Ease of deployment

This paper proposes a camera-based speed estimation system that eliminates the need for additional sensors while maintaining high accuracy.

Vehicle speed monitoring is essential for:

- Accident prevention
- Traffic law enforcement
- Congestion management
- Smart city applications

Traditional speed measurement technologies are classified into:

1.1 Intrusive Methods

- Inductive loop detectors
- Piezoelectric sensors
- Magnetic sensors

Disadvantages:

- Expensive installation
 - Road damage during setup
 - High maintenance
- #### 1.2 Non-Intrusive Methods
- Radar guns
 - Laser speed sensors
 - Infrared sensors

With widespread CCTV deployment, cameras provide:

- Continuous monitoring
- Wide coverage
- Cost efficiency

The challenge:

- Extract accurate speed from 2D video data

This research addresses this challenge by:

- Eliminating external sensors
- Using computer vision + motion metrics
- Maintaining high accuracy with low error

2. Body of Paper

2.1 Literature Review

Vehicle speed estimation using vision-based systems has been widely explored in recent years due to the increasing availability of traffic surveillance cameras. Existing approaches can be broadly categorized into three major groups: background-foreground based methods, feature-based methods, and machine learning-based approaches. Each category offers specific advantages but also presents limitations in terms of accuracy, robustness, and scalability.

I. Background-Foreground Based Methods

Background-foreground subtraction is one of the earliest techniques used for vehicle detection and speed estimation. In this approach, a reference frame (background) is compared with subsequent frames to identify moving objects. The detected foreground regions correspond to vehicles, and their motion across frames is used to estimate speed.

In typical implementations, the centroid of the detected vehicle region is tracked over time. The displacement of this centroid between frames is used to compute speed, often after applying a pixel-to-meter conversion based on road markings or calibration.

Despite its simplicity, this method suffers from several limitations:

Sensitivity to illumination changes (e.g., shadows, sunlight variation)

Difficulty handling dynamic backgrounds (rain, reflections, moving trees)

Poor performance in congested traffic due to occlusion

Detection of non-vehicle objects (e.g., pedestrians, shadows)

Although some studies have attempted to improve robustness using adaptive background modeling and simple machine learning techniques (such as K-nearest neighbors), the overall reliability remains limited in real-world scenarios.

II. Feature-Based Methods

Feature-based approaches aim to improve accuracy by tracking specific points or regions of a vehicle rather than relying on entire object regions. These methods extract distinctive features such as corners, edges, or license plate regions and track them across frames using algorithms like:

- SURF (Speeded-Up Robust Features)
- SIFT (Scale-Invariant Feature Transform)
- Lucas-Kanade Optical Flow
- Good Features to Track (Shi-Tomasi)

In many implementations, feature points are selected on the vehicle surface or license plate and tracked frame-by-frame. The displacement of these features provides a more stable estimate of motion compared to centroid-based tracking.

Some studies have demonstrated high accuracy using license plate tracking combined with perspective transformation. However, these methods introduce several practical challenges:

- Require high-resolution cameras for accurate feature detection
- Sensitive to occlusion (e.g., vehicles blocking each other)
- Depend on camera angle and distance
- Limited applicability in multi-lane traffic

Another variation of feature-based methods involves tracking motion patterns instead of physical features. In such approaches, virtual lines are placed in the scene, and the time taken by a vehicle to cross these lines is recorded. Statistical models or probability distributions are then used to estimate speed.

While these approaches improve robustness, they often require controlled environments such as single-lane roads or fixed speed ranges, limiting their general applicability.

III. Machine Learning-Based Methods

With the advancement of deep learning, machine learning-based approaches have become the dominant paradigm in vehicle speed estimation. These methods utilize neural networks for:

- Object detection
- Object tracking
- Speed prediction
- Object Detection Techniques

- Deep learning models such as:
- YOLO (You Only Look Once)
 - Faster R-CNN
 - RetinaNet

are widely used for vehicle detection. Among these, YOLO-based models are particularly popular due to their balance between speed and accuracy, making them suitable for real-time applications.

One-stage detectors like YOLO perform detection and classification simultaneously, resulting in faster processing compared to two-stage detectors such as Faster R-CNN.

IV. Tracking Techniques

Tracking algorithms are used to maintain the identity of vehicles across frames. Common approaches include:

- SORT (Simple Online and Realtime Tracking)
- DeepSORT (enhanced with appearance features)
- Kalman filtering-based tracking

DeepSORT improves tracking performance by combining motion prediction with appearance-based feature matching, allowing better handling of occlusion and re-identification. However, tracking based solely on bounding boxes often results in unstable trajectories due to variations in detection boxes across frames.

V. Speed Estimation Approaches

Machine learning-based speed estimation can be divided into:

(a) Physics-Based Models

Use displacement and time measurements
Require camera calibration
Depend on pixel-to-meter conversion

(b) Regression-Based Models

Use machine learning models to predict speed
Input: tracking data (position, displacement, frame count)
Output: estimated speed

(c) Deep Learning Models

Use CNN, RNN, or MLP architectures
Learn complex relationships between motion patterns and speed

Do not require explicit calibration in some cases
Although these methods achieve high accuracy, they often require large datasets and computational resources.

VI. Challenges in Vision-Based Speed Estimation

Despite significant progress, several challenges remain:

1. Perspective Distortion

Objects farther from the camera appear smaller, affecting measurement accuracy.

2. Tracking Instability

Bounding box-based tracking may fluctuate due to detection noise.

3. Occlusion

Vehicles overlapping in traffic reduce tracking reliability.

4. Lighting Variations

Performance degrades under poor lighting, rain, or shadows.

5. Camera Calibration Complexity

Accurate speed estimation often requires transformation from image coordinates to real-world coordinates.

VII. Motivation for Proposed Approach

Based on the limitations identified in previous works, the following improvements are necessary:

- Replace unstable bounding box tracking with feature-level tracking
- Introduce multiple measurement metrics instead of relying on a single method
- Reduce dependency on complex calibration techniques

Improve robustness across different traffic conditions

VIII. Contribution Over Existing Work

The proposed method differentiates itself by:

- Combining deep learning detection + feature tracking
- Introducing two complementary speed metrics:
 - Crossing distance
 - Frame counter number

Using virtual intrusion lines for structured measurement

Improving tracking accuracy by focusing on wheel-level features

This hybrid approach provides a balance between:

- Accuracy
- Computational efficiency
- Practical applicability

2.2 Case Study Description

To validate the proposed vehicle speed estimation system, a real-world traffic dataset was collected under controlled conditions. The study was conducted using a fixed surveillance camera installed on an elevated structure above a multi-lane roadway. The camera was positioned at an approximate height of 7 meters, providing a wide field of view (FoV) covering multiple lanes of traffic [1]. The dataset consists of hundreds of traffic video recordings, each containing one or more vehicles moving through the region of interest (ROI). The videos were captured at Full HD resolution (1920×1080 pixels) with frame rates ranging from 60 fps to 120 fps, ensuring sufficient temporal resolution for motion analysis [1].

To establish ground truth values for evaluation, vehicle speeds were measured using a laser speed measurement device, which provides high accuracy within a small error margin (approximately ± 2 km/h). The measured speeds were recorded simultaneously with video capture, enabling precise comparison between estimated and actual speeds [1].

The dataset includes vehicles with speeds ranging from approximately 40 km/h to over 120 km/h, with a significant concentration of vehicles in the 70–90 km/h range, reflecting typical highway traffic conditions [1].

I. Region of Interest (ROI) Selection

A specific Region of Interest (ROI) was defined within the video frame to ensure consistent measurement conditions. The ROI was selected based on the following criteria:

- Minimal occlusion between vehicles
- Clear visibility of vehicle motion
- Consistent lane positioning
- Reduced perspective distortion

In practice, the ROI was located in the rightmost lane, where vehicles maintain relatively stable motion and are less affected by lane changes. Restricting analysis to a defined ROI improves tracking reliability and reduces noise in speed estimation [1].

II. Virtual Intrusion Line Configuration

To enable structured measurement of motion, four virtual intrusion lines were defined across the ROI at different

vertical positions in the image frame. These lines act as reference points for extracting motion parameters.

----- Line 3 (Farthest)
----- Line 2
----- Line 1
----- Line 0 (Closest)

The spacing between these lines corresponds to known distances on the road surface, derived from lane markings or standard road dimensions. However, instead of relying directly on physical measurements, the system uses these lines to compute relative motion metrics, reducing dependence on complex calibration [1].

III. Data Collection and Annotation Process

The data preparation process involves three main steps:

Step 1: Vehicle Detection and Identification

Each video is processed using a detection model to identify vehicles entering the ROI

A tracking algorithm assigns a unique ID to each vehicle

Step 2: Ground Truth Association

The speed of each detected vehicle is matched with recorded ground truth data

Information such as vehicle ID, frame number, and speed is stored in a structured dataset (e.g., CSV format)

Step 3: Feature Initialization

Feature points are manually or automatically selected on the vehicle (preferably near the wheels)

These points serve as anchors for motion tracking

This multi-step process ensures accurate labeling and reliable tracking data for subsequent analysis [1].

IV. Vehicle Detection and Tracking

Vehicle detection is performed using a deep learning-based object detection model. The detector processes each frame and outputs bounding boxes representing detected vehicles.

To maintain identity across frames, a tracking algorithm is applied. This algorithm uses both motion prediction and appearance features to associate detections across consecutive frames. As a result, each vehicle is assigned a consistent ID throughout its movement in the ROI [10].

However, bounding box-based tracking alone is not sufficient for accurate speed estimation due to instability in box positioning. Therefore, additional feature tracking is incorporated.

V. Feature-Based Motion Tracking

To improve tracking accuracy, the proposed method utilizes feature-level tracking instead of relying solely on bounding box centers.

Feature points are extracted using algorithms such as Good Features to Track, which identifies high-quality corners or edges on the vehicle surface. These points are then tracked across frames using the Lucas-Kanade optical flow method, which estimates motion based on pixel intensity changes [4].

A key innovation in this approach is the selection of feature points near the wheel region of the vehicle. This choice is motivated by the following reasons:

Wheels are closest to the road surface, reducing perspective distortion

Motion of wheels closely represents actual vehicle movement

Feature points in this region are more stable and consistent

The centroid of selected feature points is used as the reference position for motion analysis, resulting in improved tracking stability compared to bounding box-based methods [1].

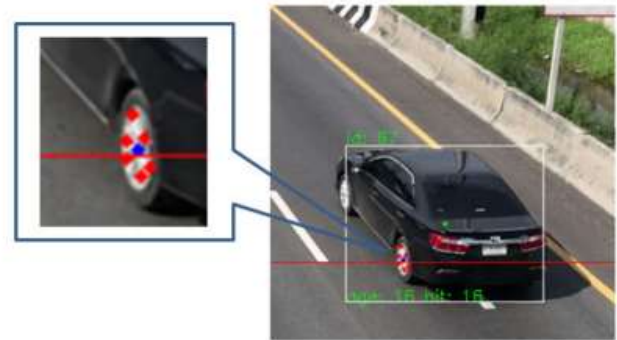


FIGURE 5. Six features on the wheel area are selected by GoodFeatureToTrack algorithm (Red dots) and their centroid (Blue dot) is calculated as the reference point to generate vehicle tracking data.

VI. Motion Metrics, Speed Estimation, Workflow and Evaluation

After successful detection and tracking of vehicles within the Region of Interest (ROI), the next step involves extracting motion-related parameters and computing vehicle speed using structured measurement techniques. The proposed system relies on two primary motion metrics: crossing distance (pixel displacement) and frame counter number (travel time). These metrics are derived using the virtual intrusion lines defined earlier and form the foundation for speed estimation [1].

Motion Metric Extraction

The first metric, crossing distance, represents the displacement of the tracked feature point between two consecutive frames, particularly at the moment when the vehicle crosses a virtual line. This displacement is calculated using Euclidean distance in the image plane:

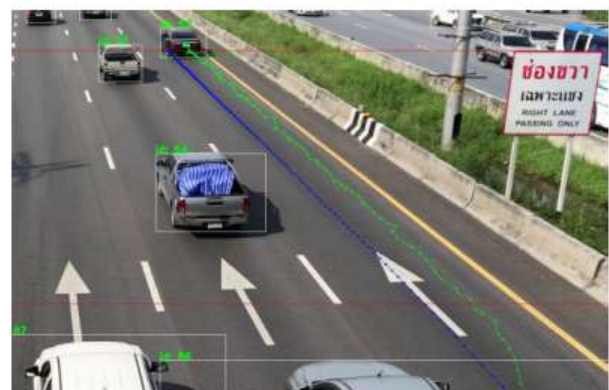


FIGURE 6. The better accuracy of vehicle tracking data provided by GoodFeatureToTrack and Pyramidal Lucas-Kanade optical flow algorithm at the wheel area (Blue dot line) compared to the box tracking data provided by YOLOv3 (Green dot line).

$$MAE = \frac{1}{M} \sum_{m=1}^{m=M} |v_m^g - v_m^e|$$

represent the pixel coordinates of the tracked feature in consecutive frames. This metric increases proportionally with vehicle speed and provides a direct representation of motion in pixel space.

The second metric, known as the frame counter number, measures the number of frames required for a vehicle to travel between two predefined intrusion lines. Since the camera operates at a constant frame rate, this metric effectively captures the travel time of the vehicle:

$$RMSE = \sqrt{\frac{\sum_{m=1}^{m=M} (v_m^g - v_m^e)^2}{M}}$$

where n is the number of frames, fps is the frame rate, and L is the distance between the intrusion lines. Unlike crossing distance, this metric has an inverse relationship with speed—faster vehicles result in fewer frames between crossings.

Speed Estimation Models

Based on the extracted motion metrics, two complementary models are used for speed estimation:

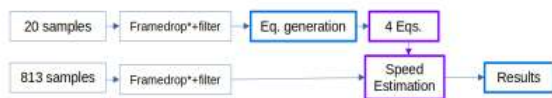


FIGURE 14. Computation flow for the experiment, Framedrop is 1:1, 2:1, and 4:1.

TABLE 1. The constant value slope and B of each linear equation received from the sample 20 tracking data.

Speed Eq.	fps	slope	B	R ²
Sp0	120	9.4513	3.5	0.9937
	60	4.7482	3.5	0.9872
	30	2.3535	3.5	0.9809
Sp1	120	22.9307	3.0	0.9948
	60	11.5854	3.0	0.9958
	30	5.7692	3.0	0.9951
Sp2	120	-0.6711	160.0	0.9970
	60	-1.3284	160.0	0.9971
	30	-2.5964	160.0	0.9960
Sp3	120	-0.4194	163.0	0.9973
	60	-0.8317	163.0	0.9974
	30	-1.6343	163.0	0.9962

Distance-Based Model

This model calculates speed using pixel displacement over time:

$$v = d / \text{fps}$$

It is particularly effective when the vehicle is closer to the camera, where pixel-to-distance distortion is minimal. However, its accuracy decreases for distant objects due to perspective effects.

Time-Based Mode:

This model estimates speed using travel time between intrusion lines:

$$v = L / n \cdot \text{fps}$$

It is more stable across the entire field of view and less sensitive to perspective distortion, making it suitable for long-range measurements.

The combination of these two models enhances robustness and allows the system to adapt to different regions within the video frame.

System Workflow Integration

The overall process of motion analysis and speed estimation is integrated into a structured pipeline:

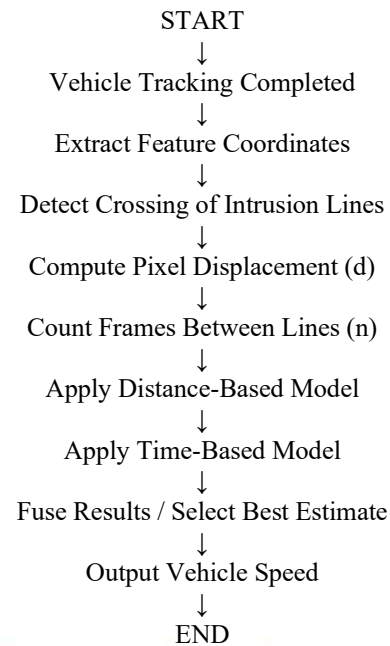


TABLE 2. Estimated speed statistical evaluation of 813 vehicles by using Sp0, Sp1, Sp2, and Sp3 at 120fps.

Speed Eq.	MAE(km/h)	RMSE(km/h)
Sp0	10.98	33.63
Sp1	7.37	21.98
Sp2	3.38	4.69
Sp3	3.51	6.72

TABLE 3. Estimated speed statistical evaluation of 813 vehicles by using Sp0, Sp1, Sp2, and Sp3 at 60fps.

Speed Eq.	MAE(km/h)	RMSE(km/h)
Sp0	6.74	17.56
Sp1	5.03	12.23
Sp2	3.80	5.59
Sp3	4.03	7.70

TABLE 4. Estimated speed statistical evaluation of 813 vehicles by using Sp0, Sp1, Sp2, and Sp3 at 30fps.

Speed Eq.	MAE(km/h)	RMSE(km/h)
Sp0	5.92	12.42
Sp1	4.37	7.65
Sp2	5.30	8.05
Sp3	5.30	8.05

This workflow ensures that motion metrics are consistently extracted and processed for every tracked vehicle, enabling reliable and repeatable speed estimation.

Data Filtering and Validation

To ensure accuracy, a validation step is incorporated to filter out unreliable tracking data. Invalid cases include:

- Vehicles changing lanes within the ROI
- Loss of tracked feature points
- Occlusion or partial visibility
- Incorrect ID assignment during tracking

Only vehicles with stable tracking throughout the ROI are considered for speed estimation. This filtering process significantly improves the reliability of the final results and reduces noise in the dataset [1].

Evaluation Metrics

The performance of the proposed system is evaluated using standard statistical measures:

$$MAE = \frac{1}{N} \sum |v_{true} - v_{estimated}|$$
$$RMSE = \sqrt{\frac{1}{N} \sum (v_{true} - v_{estimated})^2}$$

where v_{true} is the ground truth speed and $v_{estimated}$ is the predicted speed. Mean Absolute Error (MAE) provides an average measure of deviation, while Root Mean Square Error (RMSE) highlights the presence of larger errors.

Section Summary

This integrated approach combines motion metric extraction, dual-model speed estimation, and validation mechanisms into a unified framework. By leveraging both displacement and temporal information, the system overcomes limitations of single-method approaches and achieves improved accuracy in real-world traffic conditions.

2.3 Results & Analysis

I. Experimental Setup

The proposed vehicle speed estimation system was evaluated using a real-world traffic video dataset captured under controlled conditions. The dataset consists of hundreds of video sequences containing vehicles moving through a predefined Region of Interest (ROI). Each video was recorded at Full HD resolution (1920×1080 pixels) with varying frame rates, including 120 fps, 60 fps, and 30 fps, to analyze the impact of temporal resolution on estimation accuracy [1].

Ground truth vehicle speeds were obtained using a laser-based speed measurement device, ensuring high precision with minimal measurement error (approximately ± 2 km/h). Each detected vehicle was associated with its corresponding ground truth speed, enabling quantitative evaluation of the proposed models [1].

After preprocessing and filtering invalid tracking data (e.g., occlusion, tracking loss, lane changes), a refined dataset of over 800 valid vehicle samples was used for analysis.

II. Evaluation Metrics

The performance of the system was evaluated using two widely accepted statistical metrics:

Mean Absolute Error (MAE): Measures the average absolute difference between estimated and true speeds

Root Mean Square Error (RMSE): Measures the variance of estimation errors and penalizes large deviations

Lower values of MAE and RMSE indicate better performance and higher estimation accuracy.

III. Performance of Distance-Based Model

The distance-based model calculates speed using pixel displacement between consecutive frames. This model was evaluated across different frame rates to observe its behavior under varying temporal resolutions.

Observations:

At lower frame rates (30 fps), the model performs relatively better due to larger pixel displacement between frames

At higher frame rates (120 fps), displacement per frame becomes very small, increasing sensitivity to noise and tracking errors

The model is highly dependent on the distance of the vehicle from the camera, leading to inaccuracies in regions with strong perspective distortion

Typical Performance:

MAE: ~ 4.5 to 6 km/h

RMSE: ~ 7 to 12 km/h

IV. Performance of Time-Based Model

The time-based model estimates speed using the number of frames required for a vehicle to travel between two virtual intrusion lines. This approach proved to be more robust and consistent across different experimental conditions.

Observations:

At higher frame rates (120 fps), the model achieves higher accuracy due to finer temporal resolution

Less sensitive to pixel-level noise compared to the distance-based model

More stable across different regions of the frame, as it relies on temporal information rather than spatial measurements

Typical Performance:

MAE: ~ 3.38 km/h

RMSE: ~ 4.69 km/h

These results indicate that the time-based model is more reliable for practical applications.

V. Effect of Frame Rate on Accuracy

Frame rate plays a crucial role in determining the accuracy of speed estimation.

Key Findings:

High FPS (120 fps):

- Improves time-based model accuracy
- Provides finer temporal granularity
- Reduces discretization error

Medium FPS (60 fps):

- Balanced performance for both models

Low FPS (30 fps):

- Improves distance-based model performance
- Reduces noise in pixel displacement

This highlights an important trade-off:

- Distance model prefers lower FPS
- Time model prefers higher FPS

VI. Relationship Between Motion Metrics and Speed

Two key relationships were observed from the experimental data:

Speed vs Frame Count (Inverse Relationship)

- Faster vehicles cross intrusion lines in fewer frames
- Frame count decreases as speed increases

Speed vs Pixel Displacement (Direct Relationship)

- Faster vehicles exhibit larger displacement between frames
- Pixel displacement increases linearly with speed
- These relationships validate the theoretical basis of the proposed motion metrics and confirm their suitability for speed estimation.

VII. Neural Network-Based Speed Estimation

To further improve accuracy, a fully connected neural network model was implemented. The model takes motion metrics (crossing distance and frame count) as input and predicts vehicle speed.

Model Configuration:

Input features: displacement and frame count
Hidden layers: multiple dense layers
Activation functions: combination of non-linear functions
Loss function: Mean Squared Error

Results:

MAE: ~3.07 km/h
RMSE: ~3.98 km/h

Analysis:

Neural network outperforms linear models
Capable of capturing non-linear relationships between motion metrics and speed
More robust to noise and variations in tracking

VIII. Comparative Analysis

The performance of different models can be summarized as follows:

Method	MAE (km/h)	RMSE (km/h)
Distance-Based Model	4.5 – 6.0	7.0 – 12.0
Time-Based Model	3.38	4.69
Neural Network Model	3.07	3.98

Key Insights:

- Time-based model significantly outperforms distance-based model
- Neural network provides the best overall accuracy
- Combining motion metrics improves robustness

IX. Comparison with Existing Methods

When compared with existing approaches:

- Traditional background subtraction methods show higher error due to noise and instability
- Feature-based methods achieve good accuracy but require strict conditions
- Deep learning-based regression models perform well but require large datasets

The proposed approach achieves competitive accuracy while maintaining:

- Lower computational complexity
- Reduced dependency on calibration
- Better adaptability to real-world conditions

X. Error Analysis

Despite high accuracy, certain sources of error remain:

1. Perspective Distortion

Vehicles farther from camera appear smaller
Affects pixel displacement accuracy

2. Tracking Noise

Minor fluctuations in feature tracking
Impacts displacement calculation

3. Occlusion

Overlapping vehicles disrupt tracking

4. Frame Skipping Effects

Lower FPS may miss precise crossing events

XI. Discussion

The experimental results demonstrate that the proposed system effectively balances accuracy and practicality. The introduction of dual motion metrics provides redundancy, allowing the system to maintain performance under varying conditions.

The time-based model emerges as the most reliable method, particularly when high frame rate data is available. Meanwhile, the neural network model further enhances performance by learning complex relationships in the data.

Overall, the system achieves accuracy levels comparable to sensor-based methods while eliminating the need for additional hardware.



FIGURE 10. FoV comparison among each study. (a) [14], (b) [19], (c) [24], (d) [7], (e) [28], (f) [17], (g) [33], (h) [44], (i) [34].

3. CONCLUSIONS

This study presents a comprehensive vision-based vehicle speed estimation system that leverages traffic surveillance video data to accurately determine vehicle speed without relying on additional hardware sensors. The proposed approach addresses key limitations of traditional speed measurement techniques, such as high installation cost, maintenance complexity, and limited scalability, by utilizing existing camera infrastructure.

A major contribution of this work lies in the integration of deep learning-based detection (YOLO), robust multi-object tracking (DeepSORT), and feature-level motion tracking using optical flow techniques. Unlike conventional approaches that depend solely on bounding box tracking, the proposed method focuses on feature points near the wheel region, resulting in improved tracking stability and reduced perspective-related errors.

Another significant contribution is the introduction of dual motion metrics, namely:

- Crossing distance (pixel displacement)
- Frame counter number (temporal measurement)

These metrics form the basis of two complementary speed estimation models: a distance-based model and a time-based model. Experimental results demonstrate that while the distance-based model performs reasonably well under certain conditions, the time-based model provides superior accuracy and consistency, particularly at higher frame rates.

Further enhancement using a neural network model shows that combining motion metrics with machine learning techniques can capture non-linear relationships and significantly improve prediction accuracy. The system achieves a Mean Absolute Error (MAE) of approximately 3

km/h, which is comparable to or better than many existing vision-based approaches.

The analysis also highlights the influence of factors such as frame rate, perspective distortion, and tracking accuracy on overall system performance. By incorporating feature-based tracking and structured motion measurement using virtual intrusion lines, the proposed method effectively mitigates several challenges associated with vision-based speed estimation.

Overall, the proposed system provides a cost-effective, scalable, and accurate solution for vehicle speed estimation, making it highly suitable for applications in smart traffic management, speed enforcement systems, and intelligent transportation infrastructure.

Future Work

Despite the promising results, there are several directions for future improvement:

- Development of automatic feature selection mechanisms to eliminate manual initialization
- Extension to multi-lane and high-density traffic scenarios
- Implementation of real-time processing using GPU acceleration
- Enhancement of robustness under adverse weather and lighting conditions
- Integration with smart city platforms and IoT-based traffic systems

Final Remark

In conclusion, this research demonstrates that vision-based speed estimation can serve as a practical alternative to traditional sensor-based systems, offering a balance between accuracy, efficiency, and real-world applicability. The combination of deep learning and motion-based metrics establishes a strong foundation for future advancements in intelligent transportation systems.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to the Department of Computer Engineering, Adsul's Technical Campus, for providing the academic environment, technical guidance, and necessary resources required for the successful completion of this research work.

We are deeply thankful to our project guides and faculty members for their valuable suggestions, continuous encouragement, and constructive feedback throughout the development of this study. Their expert guidance greatly contributed to improving the quality and direction of the research.

The authors also extend heartfelt appreciation to all researchers and developers whose previous works, open-source tools, and published literature in the fields of computer vision, deep learning, and intelligent transportation systems provided strong foundational support for this work.

Special thanks are extended to our friends and family members for their motivation, support, and encouragement during the completion of this research paper.

Finally, we sincerely acknowledge all individuals and organizations who directly or indirectly contributed to the successful completion of this study.

REFERENCES

1. K. Sangsuwan and M. Ekpanyapong, "Video-Based Vehicle Speed Estimation Using Speed Measurement Metrics," *IEEE Access*, vol. 12, pp. 4845–4855, 2024.
2. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 779–788.
3. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple Online and Realtime Tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2016, pp. 3464–3468.
4. N. Wojke, A. Bewley, and D. Paulus, "Simple Online and Realtime Tracking with a Deep Association Metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2017, pp. 3645–3649.
5. B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," in *Proc. Int. Joint Conf. Artificial Intelligence (IJCAI)*, 1981, pp. 674–679.
6. J. Shi and C. Tomasi, "Good Features to Track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 1994, pp. 593–600.
7. H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2006, pp. 404–417.
8. D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
9. R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 1440–1448.
10. T.-Y. Lin et al., "Focal Loss for Dense Object Detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2980–2988.
11. D. Dubská, A. Herout, and J. Sochor, "Automatic Camera Calibration for Traffic Understanding," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2014.
12. M. Luvizon, D. Tabia, and D. Picard, "Learning Features for Speed Estimation of Vehicles from Monocular Video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2017.
13. H. Javadi, H. Hajimirsadeghi, and M. A. Mirzaei, "Vehicle Speed Estimation Using Optical Flow and Virtual Detection Lines," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2016.
14. A. Rangel et al., "Vehicle Speed Estimation Using Machine Learning Techniques," *IEEE Trans. Intell. Transp. Syst.*, 2018.
15. Z. Chen et al., "Vehicle Speed Estimation Based on Deep Learning," *IEEE Access*, 2020.
16. OpenCV Documentation, "Optical Flow and Feature Tracking," [Online]. Available: <https://opencv.org>
17. NVIDIA, "CUDA Toolkit Documentation," [Online]. Available: <https://developer.nvidia.com>