Whitepaper

Optimizing Vehicle Speed Estimation Through Classical Image Processing in Resource-Limited Systems

Authored by:

Rounaq Choudhuri,
Somsuvra Chatterjee,
Vishwanathan Raman,
LTIMindtree

Executive summary

This whitepaper introduces an innovative approach to vehicle speed estimation, a critical use case within Intelligent Traffic Management Systems (ITMS), particularly designed for environments with limited computational resources, such as construction sites. Traditional methods often rely on state-of-the-art single-shot detectors (SSDs) and high-performance GPUs, which are not only resource-intensive but also costly and difficult to deploy in temporary or resource-constrained settings.

Our proposed solution leverages classical image processing techniques, bypassing the need for advanced SSDs and GPUs, thereby offering a cost-effective, scalable alternative. This method employs motion detection and object masking to accurately estimate vehicle speeds in real time, even in challenging conditions with noisy backgrounds, varying lighting, and environmental factors.

The solution was successfully deployed at a construction site to monitor various types of vehicles, from standard passenger cars to heavy construction machinery. Despite the complexities involved, the approach delivered superior performance, achieving real-time processing speeds significantly faster than conventional methods while maintaining high accuracy.

Key advantages of this approach include its independence from specific object properties, field of view variations, and environmental factors, as well as its ability to operate effectively with minimal computational power. The results demonstrate the viability of this method as a robust alternative to traditional SSD-based solutions.

This whitepaper provides a detailed discussion of the methodology, its advantages, and areas for further improvement, particularly in enhancing the quality of camera inputs and leveraging more powerful computational resources. The findings underscore the potential for classical image processing algorithms to drive innovation in ITMS and related applications, offering a practical, efficient solution for real-time vehicle speed estimation.

Introduction

In a time when CV applications play a crucial role in smart cities, intelligent traffic management systems (ITMSs) are one of the most sought-after solutions. Automatic number plate recognition (ANPR) and vehicle type detection are the most common use cases in an ITMS project. In this whitepaper, we will discuss vehicle speed estimation, which has emerged as an important use case in recent times. However, an ITMS is required to process frames at a high rate from the incoming video feed. This is achieved by employing high-performing CPUs coupled with GPUs. A generalized approach to the solution uses a SOTA SSD to detect the vehicle within the ROI. Furthermore, the detection outcomes throughout the ROI are used to track and estimate the speed. However, an SOTA SSD is resource intensive and requires installing additional drivers for the GPUs. This disrupts the eventual cost of the use case. Hence, we will be discussing the proposed methodology. An intuitive approach to the vehicle speed estimation use case also circumvents the use of such resource-intensive processors.

This solution was deployed at a construction site to detect vehicles (viz., transit mixer, cranes, multi-axel trucks, etc.). A custom solution for the vehicles in the use case was devised per the requirement. An off-the-shelf SOTA SSD does not detect such vehicles. It requires model retraining with a large custom dataset. Since such a custom dataset was unavailable, it provided another impetus for developing a naive but scalable approach. The details of our approach are discussed in the following sections, where an explanation of the approach and its advantages are provided.

Problem statement

A leading company in the construction industry wanted to employ Intelligent Surveillance Systems (ISSs) at their construction sites. They aimed to integrate an ISS within their ecosystem to promote construction safety and enforce regulations. One of the use cases was vehicle speed estimation within the construction site. These vehicles ranged from standard passenger vehicles to heavy construction vehicles. Moreover, the company could not install high-performance GPU servers at the temporary construction site.

Therefore, the following were the challenges in deploying such a solution:

- Detection of construction vehicles with noisy backgrounds
- Processing of frames with minimal computational power
- Providing real-time solutions

To make this solution more robust, the company wanted to generate alerts (emails and SMS) whenever a vehicle was deemed overspeeding at the site. The same alerts must be available with an image of the vehicle on a dashboard. Event and alert generation are beyond the scope of this article. Therefore, we will not be discussing the same.

Solution outline

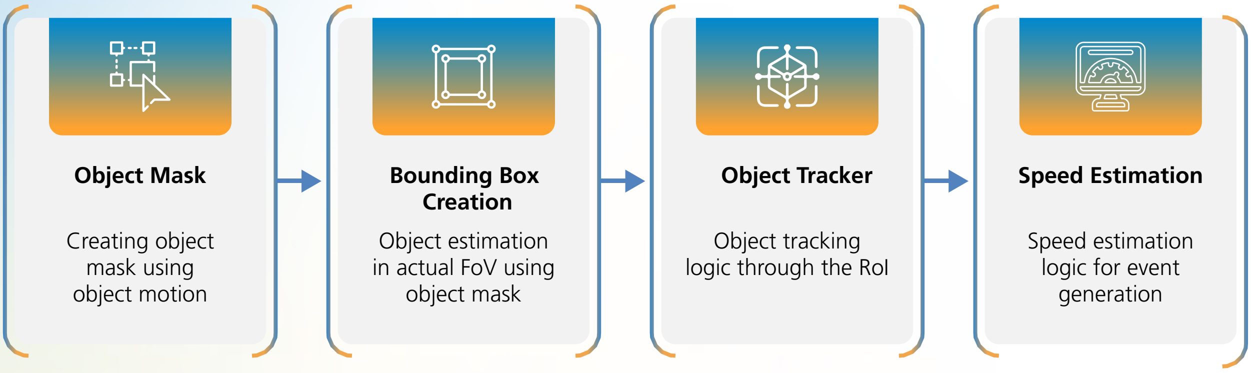


Figure 1: The workflow

In this section, we will highlight the proposed approach's intuitiveness. Figure 1 describes the workflow of the approach, whereas Figure 2 provides details of the approach flow with an example. At first, an object mask was created to estimate the object in the frame and a bounding box. The bounding box information of the object across the ROI was used to create a tracker. Creating an object mask eliminates the possible detection of background noise. Furthermore, the masking image provides a more accurate estimation of the object's position within a field of view (FoV). The estimated position of the object helps us to draw a bounding box, which is relayed onto the object tracker. The tracker information is further used to estimate the speed of the vehicle. In further sections, we will discuss object mask creation and bounding box estimation. However, we will not be discussing object tracking and the business logic for the speed estimation since it is beyond the scope of this article.

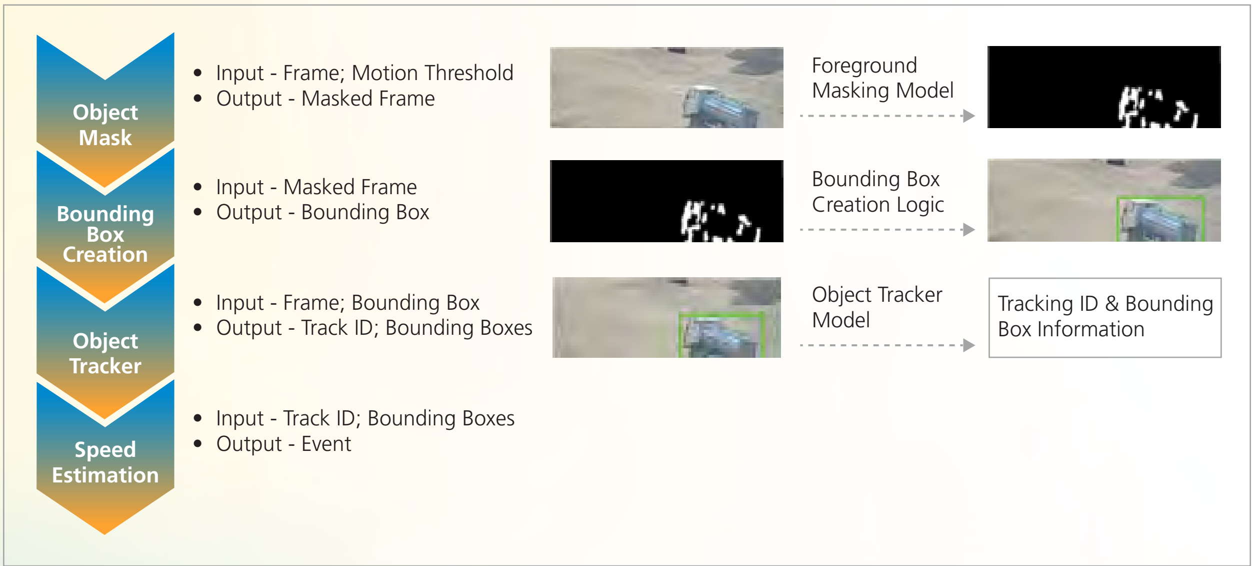


Figure 2: Detailed workflow with an example

A. Object mask creation

A motion detector curtails the processing of frames when there is no moving object within the FoV. A motion detector employs the following sub-methodologies:

a. Static background subtraction
b. Continuous background subtraction

Motion detection with static background subtraction captures the motion of a frame using a template background image. However, a major drawback is that this methodology does not account for lighting and climatic changes. Continuous background subtraction, however, overcomes this flaw by capturing motion with respect to the previous frame. This also eliminates the possibility of any background noise detection (like static vehicles within the FoV).

The latter was employed in this approach. Moreover, a motion detector serves a secondary purpose. It also helps to create a mask of the moving object, as shown in Figure 2. An object mask is a black-and-white image that simply highlights the moving object.

B. Bounding box creation

As shown in Figure 2, the mask image can be used to accurately estimate a virtual bounding box around the object. Using simple logic, the extreme corners of the mask can be located, from which an estimated box can be derived. These masked images, along with the original frame and the generated bounding box information, are tagged with a timestamp for further processing.

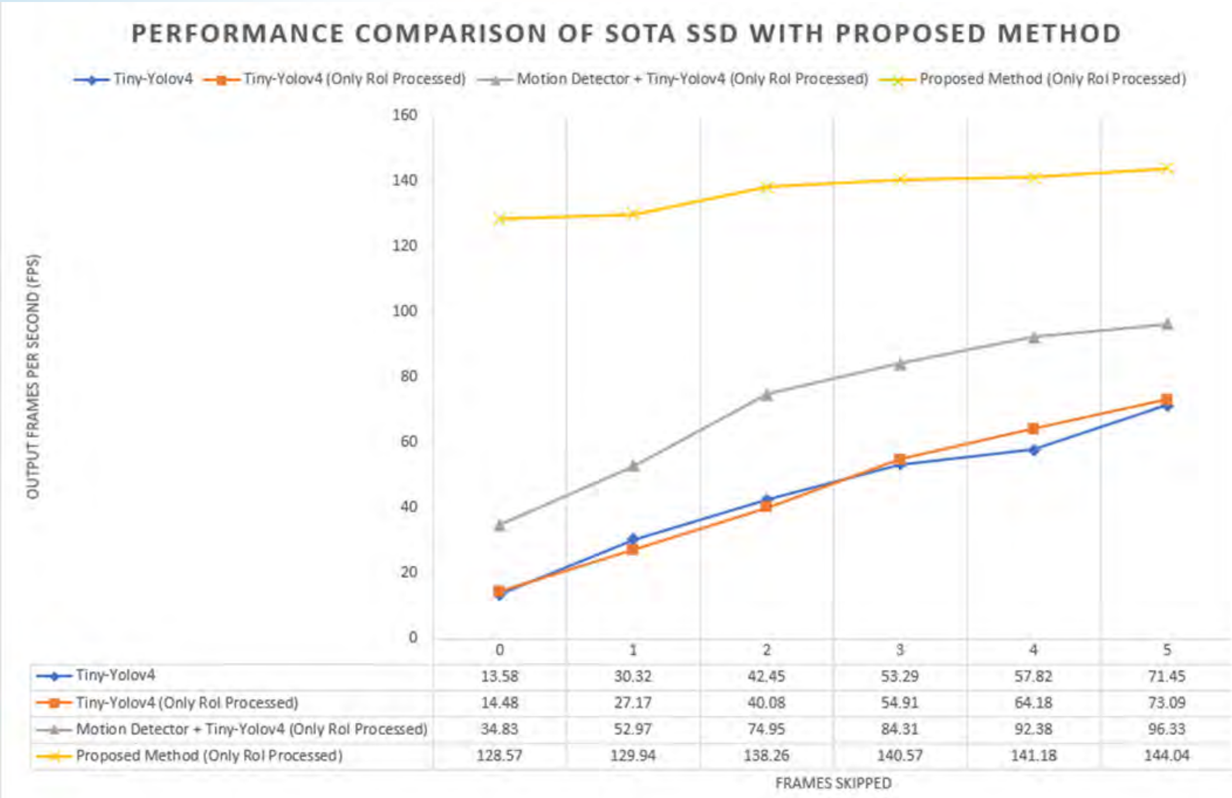
C. Object tracking

Frames and the bounding box information are fed into the object tracker. A wide variety of off-the-shelf object trackers are available, the most popular being Kalman Filter trackers. However, considering the approach's naivety, a box tracker was employed. Moreover, a box tracker is sufficient for this use case, considering the vehicle traffic at a construction site.

D. Speed estimation

The tracker information is relayed onto a set of business logic developed particularly for the use case. The business logic can also be configured externally to meet the demands for a custom speed limit within an ROI in a FoV.

Results



	0	1	2	3	4	5
Tiny-Yolov4	13.58	30.32	42.45	53.29	57.82	71.45
Tiny-Yolov4 (Only RoI Processed)	14.48	27.17	40.08	54.91	64.18	73.09
Motion Detector + Tiny-Yolov4 (Only RoI Processed)	34.83	52.97	74.95	84.31	92.38	96.33
Proposed Method (Only RoI Processed)	128.57	129.94	138.26	140.57	141.18	144.04

Figure 3: Performance Comparison

This section discusses an analysis of the effectiveness of the proposed method with respect to output frames per second (FPS). Figure 3 highlights the same when tested on a sample video clip containing 1149 frames. The figure also tabulates the output FPS. Output FPS is inversely proportional to the time required to process the total number of frames in the sample video clip. On the X-axis, we have the number of frames skipped while processing the sample video clip, whereas on the y-axis, we are measuring the output FPS.

For experimental purposes, we have selected Tiny-Yolov4 (SOTA SSD) as a baseline for our proposed method. It is observed that Tiny-Yolov4, without any frame skip, achieves almost real-time inference. A standard real-time inference is 20-25 output FPS. It is also observed that increasing the number of frames skip value to five can achieve the real-time inference speed thrice. However, considering that this use case had to be deployed across several sites, the objective was to achieve as much output FPS as possible. Therefore, the output FPS was recorded across all frame skip values when only the RoI within the FoV was processed. It is observed that processing only the FoV has an insignificant effect on the output FPS. Furthermore, a motion detector was employed before object detection to reduce the processing of static frames. Here, we see a significant increase in output FPS from the baseline. It is observed that we achieve almost four times the real-time inference speed. Finally, the proposed method is employed in the same experimental settings. It is observed that even if no frames are skipped, it can achieve at least five times the real-time inference speed. On gradually increasing the number of frames skipped the output FPS plateaus.

Therefore, it is safe to claim that the proposed method can achieve almost seven times the real-time inference speed when configured to process every sixth frame.

Discussions

Although the approach is naive, it can easily circumvent the use of SOTA SSDs in complex CV applications. In this section, we will review the advantages of the proposed methodology and suggest areas for improvement.

A. Advantages

a. Field of view independent

Since this approach to the solution does not require an SOTA SSD, it can be deployed at different sites with various configurable settings. An SSD requires contextual training to learn the FoV, which improves object detection compared to the off-the-shelf SOTA SSD.

b. Object property independent

Avoiding the usage of a SOTA SSD eliminates the effects of object properties during object detection. Object properties like dimensions and, in the case of vehicle type and colors, play a crucial role in the accuracy of an SSD. Since our approach relies on object masking, object properties do not play any role. This is especially desirable because construction vehicles are monitored, which differs from conventional passenger vehicles.

c. No missed and wrong detection

A motion detector must process every incoming frame. Therefore, a mask image is created for every frame. This eliminates the possibility of missing an object in the frame. SSDs tend to miss objects depending on the image resolution and configuration they are running.

d. Uniform bounding box estimation

While SSDs are accurate in detecting objects and generating bounding boxes, they tend to estimate a loose bounding box. It is imperative to mention that tightly bound boxes are desirable. A tight bounding box generates more accurate speed estimations.

e. Lighting and climatic conditions independent
Employing a continuous background subtraction approach while capturing motion helps us to circumvent all the effects that light and climate may have on the FoV. This is because such a motion detector also captures the change in the surroundings.

f. Background Noise Independent
If there are non-static objects, especially objects of interest within the FoV, they may affect the performance of an SSD. Detecting extra non-static objects, particularly in our use case, may hamper the performance of the complete speed estimation pipeline. It will also require more computational power to process those objects. Therefore, using motion mask images helps reduce the detection of such objects.

g. Intuitive and naive approach
Since we are employing classical image processing algorithms, it circumvents the re-training of an SOTA SSD. Moreover, as previously mentioned, in such use cases, it is also imperative to run a contextual training process along with a custom dataset. Furthermore, this approach solves the problem efficiently and with minimal computation power.

B. Areas for improvement

a. Real-time streaming protocol (RTSP) camera property
RTSP camera properties play a crucial role in any real-time solution for an ISS. In this use case, we dealt with non-static vehicles, for which better-quality cameras are desirable. The camera's resolution, along with the recording FPS, affects the detection accuracy. Camera resolution is directly proportional to the quality of the motion mask image; likewise, FPS is directly proportional to the accuracy of estimating the speed.

b. More computation power
The results discussed here have been computed using an Intel(R) Core(TM) i5-10210U CPU at 1.60GHz. If a more computationally powerful CPU is available, the current results can be surpassed.

Conclusion

In conclusion, this whitepaper has presented a novel approach to vehicle speed estimation within the realm of intelligent traffic management systems (ITMS), particularly tailored for construction sites with limited computational resources. By leveraging classical image processing techniques, we have demonstrated how to circumvent the need for resource-intensive state-of-the-art SSDs, achieving significant improvements in real-time processing speeds while maintaining high accuracy in challenging environments.

Our method, characterized by its intuitiveness and scalability, not only addresses the unique challenges of detecting and tracking construction vehicles but also offers a solution that is independent of object properties, field of view variations, and environmental factors such as lighting and climatic conditions. The simplicity and efficiency of this approach make it an attractive alternative for scenarios where computational power is limited and where conventional SSDs may falter due to the specificity of the use case.

While the proposed method shows promise, especially in terms of speed and adaptability, there are areas for further enhancement. The quality of the input from RTSP cameras and the computational power available are two key factors that could be optimized to push the boundaries of this solution even further.

Ultimately, this whitepaper underscores the potential of classical image processing algorithms in developing robust, cost-effective solutions for complex CV applications, providing a viable pathway for future innovations in the field of intelligent surveillance and traffic management.

About the Authors

Rounaq Choudhuri

Data Scientist, LTIMindtree

Rounaq is an IT professional with 6+ years of experience in R&D in the field of Artificial Intelligence, Machine Learning, Deep Learning and Computer Vision. He completed a bachelor's in computer science and engineering, with an interest in Computer Vision and Social Graph-based algorithms. He is currently engaged in developing solutions for various

Somsuvra Chatterjee

Director in Product Engineering, LTIMindtree

A committed and passionate data scientist with overall 15+ years of experience, 12+ in the field of Analytics and Data Science. Worked across diversified domain starting with CPG, Retail & Manufacturing, Energy, etc. Engaged in developing solutions for various Computer Vision and Image Processing based products and projects.

Vishwanathan Raman

Principal Director in Product Engineering, LTIMindtree

Having 26+ years of experience across Geographies, expertise in the field of Business Intelligence, Data Warehousing, Artificial Intelligence, Machine Learning, Deep Learning and Data Sciences. A Technology Leader, Technical Author, Storyteller, AI Strategist, Trainer, completed master's from BITS Pilani in Data Analytics.