

Real-Time Vehicle Detection Using YOLOv8-Nano for Intelligent Transportation Systems

Murat Bakirci 

Unmanned/Intelligent Systems Lab, Faculty of Aeronautics and Astronautics, Tarsus University, Mersin 33400, Turkey

Corresponding Author Email: muratbakirci@tarsus.edu.tr

Copyright: ©2024 The author. This article is published by IIETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ts.410407>

ABSTRACT

Received: 8 December 2023

Revised: 16 April 2024

Accepted: 14 May 2024

Available online: 31 August 2024

Keywords:

vehicle detection, YOLOv8, aerial monitoring, intelligent transportation systems, UAV

In recent years, deep learning models have seen extensive use in various domains, with the YOLO algorithm family emerging as a prominent player. YOLOv5, known for its real-time object detection capabilities and high accuracy, has been widely embraced in transportation-related research. However, the introduction of YOLOv8 in early 2023 signifies a significant leap forward in object detection technology. Despite its potential, the literature on YOLOv8 remains relatively scarce, leaving room for exploration and adoption in research. This study pioneers real-time vehicle detection using the YOLOv8 algorithm. An in-depth analysis of YOLOv8n, the smallest scale model within the YOLOv8 series, was conducted to assess its suitability for real-time scenarios, particularly in Intelligent Transportation Systems (ITS). To reinforce its real-time capabilities, a parametric analysis covering image processing time, detection sensitivity, and input image characteristics was performed. To optimize model performance, a training dataset was created through flight tests using a custom autonomous drone, encompassing various vehicle variations. This ensures that the model excels in recognizing diverse motor vehicle configurations. The results reveal that even this compact sub-model achieves an impressive detection accuracy rate exceeding 80%. The study establishes that YOLOv8n, evaluated for the first time in ITS applications, effectively serves as an object detector for real-time smart traffic management.

1. INTRODUCTION

Intelligent Transportation Systems (ITS) have assumed a pivotal role in contemporary transportation for a multitude of compelling reasons. These ITS innovations, encompassing traffic management systems, collision avoidance systems, and automated vehicle technologies, serve as prominent leaders in the pursuit of enhancing road safety, demonstrating a distinguished track record of reducing accidents and mitigating their severity [1-3]. They possess an uncanny ability to swiftly detect and respond to perilous conditions, outpacing human drivers and thereby contributing to accident prevention and the preservation of precious lives. In the ongoing battle against traffic congestion, ITS emerges as a formidable ally. Armed with the prowess of real-time data analysis, adaptive signal control, and dynamic route guidance, ITS champions the cause of optimized traffic flow [4]. This not only translates into reduced travel durations for commuters but also manifests as a tangible reduction in fuel consumption and the associated specter of air pollution, an affliction often linked to traffic bottlenecks. ITS technologies further underscore their significance by rendering the transportation system more operationally efficient. Their adept optimization of vehicle routes, judicious reduction of idling time, and minimization of fuel consumption all conspire to yield palpable cost savings for both individuals and businesses alike [5]. The environmental stakes are high, and ITS rises to the occasion by leveraging the reduction of greenhouse gas

emissions and air pollution. The complementing infrastructure for electric and hybrid vehicles, intricately interwoven with ITS, assumes the mantle of further reducing the ecological footprint of our transportation networks.

In an age emphasizing inclusivity, ITS demonstrates its commitment by providing real-time accessibility information. It becomes an enabler for individuals with disabilities, offering vital insights into accessible routes, facilities, and transit alternatives, thereby fostering a more inclusive and accommodating transportation system. Economic growth finds a promoter in the efficient transportation systems buttressed by ITS. By facilitating the fluid movement of goods and people, these systems bestow the gift of reduced transportation costs upon businesses [6]. Simultaneously, they entice investment by virtue of improved transportation infrastructure, leading to the creation of lucrative job opportunities. The data-driven essence of ITS emerges as a valuable asset, generating a wealth of information for the discerning eye of policymakers [7]. This treasure trove of data informs sound decisions concerning transportation policy formulation, infrastructure investments, as well as the ongoing tasks of planning, maintenance, and improvement. In times of crisis, ITS assumes a pivotal role by providing real-time traffic intelligence. It optimizes emergency vehicle routes and augments communication protocols during critical moments, thereby accelerating response times and enhancing disaster management capabilities [8, 9]. In the pursuit of environmental sustainability, ITS technologies, including eco-

driving systems and adaptive cruise control, ascend to the forefront [10]. Their judicious deployment optimizes fuel efficiency and contributes indispensably to the noble cause of curtailing vehicle emissions. Additionally, the overall travel experience undergoes a transformation under the benevolent gaze of ITS. By offering real-time insights into traffic conditions, public transit schedules, and available parking spaces, ITS elevates the convenience quotient and significantly mitigates the stress typically associated with daily commuting.

Traffic monitoring stands as an integral pillar within the domain of ITS, bearing significant responsibility for the management and optimization of traffic dynamics on our roadways. This facet of ITS brings together an array of technologies and systems to perform the essential tasks of collecting, analyzing, and disseminating real-time traffic data [11]. At its core, ITS employs a diverse range of sensors and data collection devices that read the intricate rhythms of traffic conditions. Picture cameras, radar, lidar, and the inconspicuous inductive loop detectors nestled within the road surface. Even GPS-equipped vehicles and the ubiquity of mobile phone data contribute to this process of data collection [12-14]. These instruments continually gather information about vehicle speeds, traffic volume, density, and other pertinent parameters. Once gathered, this data embarks on a journey of real-time processing and analysis. Advanced algorithms and software work tirelessly to transform raw data into meaningful insights. Their discerning gaze identifies traffic congestion, pinpoints incidents such as accidents or road closures, and unveils any unusual traffic patterns that might emerge. Traffic monitoring maintains a close relationship with traffic management systems. In response to real-time data, these systems take action, deftly adjusting traffic signal timings, activating dynamic message signs to inform drivers, and overseeing ramp meters to regulate the flow of vehicles entering highways [15].

A vital function of traffic monitoring is incident detection. When accidents or other incidents occur, the system springs into action, instantly recognizing the anomaly and initiating appropriate responses. This could involve notifying emergency services, fine-tuning traffic signals, or relaying real-time traffic updates to drivers through variable message signs and smartphone apps [16, 17]. The treasure trove of traffic information harvested through monitoring is then shared with a variety of stakeholders. Traffic management centers, law enforcement agencies, and the general public all benefit from this wealth of information. Websites, mobile apps, and electronic highway signs serve as conduits, equipping drivers with the knowledge they need to make informed decisions about their routes. Some ITS systems elevate their game by employing predictive analytics. By scrutinizing historical data and current trends, they offer glimpses into the future, forecasting traffic congestion and estimated travel times. This forecasting empowers drivers to chart more efficient routes. In terms of long-term transportation planning, traffic monitoring data is a priceless asset. By maintaining extensive records of historical data, transportation agencies uncover valuable trends, plan infrastructure enhancements, and evaluate the ramifications of policy changes. Finally, traffic monitoring synergizes with other ITS components, creating a seamless and interconnected transportation ecosystem. Vehicles equipped with advanced driver assistance systems (ADAS), like adaptive cruise control, become receptive to real-time traffic intelligence [18]. They adapt their

speed and routes accordingly, enhancing both safety and efficiency on the roadways. In essence, traffic monitoring within ITS exemplifies the marriage of cutting-edge technology with the complex choreography of traffic management, all aimed at making our journeys safer, smoother, and more informed.

The application of Unmanned Aerial Vehicles (UAVs) and image processing in traffic monitoring represents an innovative approach to the administration and analysis of traffic dynamics, aimed at mitigating congestion, bolstering safety, and enhancing transportation infrastructure [19-21]. This cutting-edge technology capitalizes on the capabilities of UAVs, frequently equipped with an array of cameras and additional sensors, to capture real-time aerial imagery. Subsequently, these assets are harnessed in conjunction with advanced image processing techniques to derive valuable traffic-related insights. UAVs, more colloquially known as drones, are strategically deployed in designated areas necessitating traffic oversight. These drones can operate autonomously or under remote control, facilitating flexible deployment strategies. They adeptly acquire high-resolution aerial images or videos of the traffic milieu, with the choice of cameras on the UAVs ranging from standard RGB cameras to specialized sensors such as thermal cameras, LiDAR, or multispectral cameras. The selection of sensors is contingent upon the precise objectives of the monitoring operation.

In practice, the captured imagery is transmitted expeditiously, either in real-time or near-real-time, to a ground station or a cloud-based server for processing. This instantaneous data transfer affords authorities immediate access to the pertinent information. The crux of this endeavor hinges on the adept application of advanced image processing techniques, a process that begets invaluable traffic insights. These insights encompass an array of functions, including vehicle detection and tracking, traffic flow analysis, incident detection, counting and classification, license plate recognition, and pedestrian detection [22-25]. The efficacy of these algorithms is underscored by their proficiency in identifying vehicles, precisely tracking their trajectories, and gauging their velocity and heading. The analysis of vehicle positions and speeds serves as the basis for monitoring traffic congestion and discerning flow patterns, both pivotal facets of traffic management. Furthermore, this system excels in promptly identifying anomalies such as accidents or road obstructions, facilitating swift response and resolution. Additionally, this technological apparatus is adept at quantifying the volume of vehicles, classifying them by type (e.g., automobiles, trucks, motorcycles), and employing Optical Character Recognition (OCR) to decipher license plate information for enforcement or tracking purposes [26]. Moreover, pedestrian detection capabilities extend to the identification and monitoring of pedestrian movement in proximity to roadways and intersections. To present the amassed data in a user-friendly and actionable format, it is routinely transmuted into intuitive representations, including traffic heatmaps, graphical displays, or real-time dashboards. This facilitates informed traffic management strategies and informed decision-making processes.

In the ever-evolving landscape of object detection, traditional methodologies offer a well-trodden path. However, it is increasingly evident that deep learning-based techniques have surged to the forefront, consistently demonstrating their prowess in delivering superior results [27]. This paradigm shift has not been limited to a singular domain but has rather

permeated numerous sectors, with transportation-related applications being no exception. The demand for image processing in transportation has catalyzed the ascendancy of deep learning approaches, mirroring their success in various other fields. One of the hallmark applications in this trajectory is the identification of vehicles amidst the bustling traffic ecosystem [28]. Deep learning models have proven to be particularly adept at discerning the complex shapes, sizes, and orientations of vehicles, even in challenging real-world conditions. Moreover, these models extend their capabilities to encompass vehicle tracking, enabling the continuous monitoring of individual vehicles as they traverse through intricate traffic scenarios [29]. This tracking functionality is instrumental in applications like traffic management, security surveillance, and the optimization of transport logistics. Beyond mere identification and tracking, deep learning-based techniques have unlocked the potential for more nuanced insights. Estimating vehicle speeds, for instance, has become increasingly accurate and reliable, thanks to the granular analysis enabled by these models [30]. This capability is invaluable for traffic flow analysis, accident reconstruction, and the enhancement of road safety measures. Moreover, the ability to precisely count vehicles in various contexts, whether it be for traffic management or congestion analysis, has been significantly enhanced by deep learning methods [31]. These models can distinguish between different vehicle types, from sedans to trucks and motorcycles, offering a comprehensive view of traffic composition.

Among the plethora of deep learning models available, the You Only Look Once (YOLO) algorithm family has garnered considerable attention [32, 33]. Notably, YOLOv5 iteration has enjoyed widespread favoritism across a spectrum of transportation-related research [34, 35]. Its real-time object detection capabilities, coupled with high accuracy, make it an attractive choice for applications requiring timely responses and precision. Nonetheless, the most recent iteration of the YOLO algorithm, YOLOv8, was introduced to the research community in early 2023 [36]. This represents a significant advancement in the field of object detection. However, the academic corpus pertaining to YOLOv8 remains relatively sparse at this juncture. Consequently, while its potential in vehicle detection is promising, it has yet to firmly establish itself as a mainstream choice within the literature.

The research objectives within the scope of this study on real-time vehicle detection using the YOLOv8 algorithm in ITS can be summarized as follows: One of the primary objectives of this study is to comprehensively evaluate the performance of the YOLOv8 algorithm, the latest iteration of YOLO, in real-time vehicle detection within the framework of intelligent transportation systems. The aim is to meticulously analyze the accuracy, speed, and efficiency of the algorithm in detecting vehicles across diverse environmental conditions and traffic scenarios. Another key goal of the research is to explore the practical applications of real-time vehicle detection in ITS. This involves assessing how YOLOv8 can contribute to critical tasks such as traffic monitoring, congestion management, vehicle tracking, and enhancing safety within transportation systems. Moreover, the study endeavors to optimize YOLOv8 to enhance its performance in real-world ITS applications. This optimization primarily focuses on refining training strategies rather than solely fine-tuning the network architecture, aiming to achieve superior accuracy and reliability in vehicle detection. Additionally, the research aims to identify and address the challenges and

limitations associated with deploying YOLOv8 for real-time vehicle detection in ITS. This encompasses addressing issues such as occlusions, varying lighting conditions, instrument diversity, and computational constraints, while also proposing potential solutions to mitigate these challenges effectively.

2. METHODOLOGY

2.1 YOLOv8 network architecture

YOLOv8 (or v8) represents a significant advancement in object detection, particularly in terms of its anchor-free methodology. Unlike its earlier iterations, which relied on anchor boxes to predict bounding boxes for objects, v8 eliminates the need for these anchors, streamlining the detection process. By adopting a center point detection approach, it can more accurately localize objects within an image without being constrained by predefined anchor boxes. This anchor-free methodology offers several advantages, including improved adaptability to varying object scales and aspect ratios, as well as enhanced robustness to occlusion and cluttered scenes. Furthermore, v8's anchor-free design simplifies the model architecture and training process, resulting in faster inference speeds and reduced computational complexity. The backbone of v8, often regarded as the architectural foundation, plays a pivotal role in the model's overall performance. It employs a variant of the Cross Stage Partial Network (CSPNet) backbone architecture, which integrates Cross Stage Connection (CSC) modules and Partial Convolutional layers. This backbone design enhances feature extraction capabilities while maintaining computational efficiency. The CSPNet backbone facilitates effective information flow across network stages, enabling the model to capture both low-level and high-level features with greater accuracy. Additionally, v8's backbone architecture incorporates various backbone sizes, allowing for flexibility in balancing between computational cost and detection performance. With its innovative backbone design, v8 achieves remarkable accuracy and speed in object detection tasks. Moreover, the efficacy of the v8 is further enhanced through the utilization of an upgraded loss function designed to maximize efficiency across diverse computational platforms, encompassing both central processing units (CPUs) and graphics processing units (GPUs). This refined loss function not only optimizes the model's performance on hardware accelerators like GPUs, which excel in parallel processing, but also ensures seamless operation on CPUs, offering versatility in deployment across different computing environments. By tailoring the loss function to accommodate the distinct characteristics of CPUs and GPUs, v8 can deliver consistent and reliable performance across a wide range of hardware configurations, thereby enhancing its applicability and accessibility in various real-world scenarios.

YOLOv8 introduces various model sizes, each tailored to different application scenarios and computational constraints. The YOLOv8 family includes variants such as v8n (nano), v8s (small), v8m (medium), v8l (large), and v8x (extra-large), offering a spectrum of options to suit varying requirements. The different model sizes vary in terms of their depth, width, and number of parameters, with smaller models like v8n featuring fewer layers and parameters, while larger models like v8x are more complex and computationally intensive. This range of model sizes provides flexibility in balancing

between detection accuracy and computational resources. A significant divergence from earlier versions of the algorithm is the substitution of the objectness segment, marking a fundamental shift in the methodology employed. In this updated approach, the algorithm strategically divides the object detection task into distinct classification and regression branches, each focusing on specific aspects of the detection process. This separation of tasks allows for a more specialized treatment of classification and regression, optimizing the algorithm's ability to discern objects accurately. Notably, within the regression branch, there is a notable integration of integral form notation, a concept pioneered in the Distribution Focal Loss (DFL) framework. This utilization of integral form notation represents a sophisticated mathematical technique that enhances the algorithm's capacity to precisely localize objects within images.

2.2 Network's compatibility with real-time constraints

Embarking on the exploration of v8 marks the initiation of a multifaceted exploration, where the path forward hinges upon a pivotal decision: the careful selection of the most suitable network size amidst a rich tapestry of architectural choices. Within the v8 framework lies a vibrant spectrum of five pre-trained networks, crafted to cater to an extensive array of applications spanning domains from object detection in autonomous vehicles to surveillance systems in smart cities. These networks unfold like chapters in a compelling narrative, each offering a unique blend of characteristics and capabilities tailored to meet the demands of diverse scenarios. At one end of this spectrum resides the sleek and nimble v8n, designed for efficiency and swift inference, ideal for resource-constrained environments or real-time applications where agility is paramount. Conversely, at the other end stands the formidable v8x, boasting an impressive array of parameters and computational prowess, engineered to handle complex scenes with unparalleled precision and robustness. The magnitude of these models extends beyond mere numbers, encapsulating a wealth of nuanced features and architectural intricacies that define their essence. Each parameter, contributes to the overarching performance profile of the model, shaping its ability to discern objects with clarity amidst varying conditions and environmental contexts. As the exploration unfolds, the significance of this decision becomes increasingly apparent, as the chosen network size sets the stage for the ensuing exploration and experimentation. It dictates not only the computational footprint and inference speed, but also the scope and fidelity of the insights gleaned from the data. In this dynamic landscape of detection and inference, the choice of network size serves as a compass, guiding the expedition towards the realization of ambitious objectives and the attainment of unparalleled insights.

A parametric approach was employed to assess the algorithm's real-time compatibility. In summary, multiple tests were conducted, with detailed descriptions provided in subsequent paragraphs. Initially, various images of identical dimensions were fed into all sub-models of YOLOv8, and the inference times of each sub-model were compared. This facilitated a clearer understanding of both the similarities and differences between the sub-models. Following this, a single image was input into all sub-models, and IoU thresholds were parametrically compared. Essentially, this comparison quantified the extent of overlap between predicted and actual bounding boxes across all sub-models in object detection tasks.

The resulting mAP values corresponding to these comparisons were analyzed to ascertain the performance of each sub-model. Having identified YOLOv8 as the most suitable sub-model for real-time applications, further tests involved resizing a single image to various dimensions and comparing the inference times of this sub-model for each size. This analysis is crucial in real-world scenarios to determine the optimal image sizes for instantaneous processing. Additionally, a numerical comparison was conducted by calculating mAP values for input images generated parametrically at different sizes, providing further insights into the algorithm's performance across varying image dimensions.

In addition to these factors, a parametric approach was employed during data collection using the drone, capturing images from all conceivable orientations and viewing angles, a crucial aspect in training the algorithm. The drone utilized in the experiments is the DJI Mavic, a commercial vertical take-off and landing (VTOL) drone equipped with four rotors. Renowned for its high-resolution and wide-angle camera, it proves to be an ideal choice for aerial imaging. Operating with an advanced control system utilizing Pixhawk, the foremost control computer in its class, the drone ensures precise flight control owing to its sophisticated IMU sensors and GPS system. Furthermore, its programmable nature allows for autonomous control, enhancing the consistency of the data collection process. The drone's advanced telemetry system greatly facilitates data transmission to the ground station, boasting a range exceeding 25 kilometers. Notably, its extended flight time of over 40 minutes enables prolonged, uninterrupted data collection. Throughout the experiments, the drone was operated in various flight modes, capturing images from nearly every available spatial and visual perspective. This systematic approach ensured the acquisition of images at diverse altitudes, orientations, and locations, following a parametric methodology rather than a random one. The specifics and outcomes of each test are elaborated upon in the subsequent paragraphs.

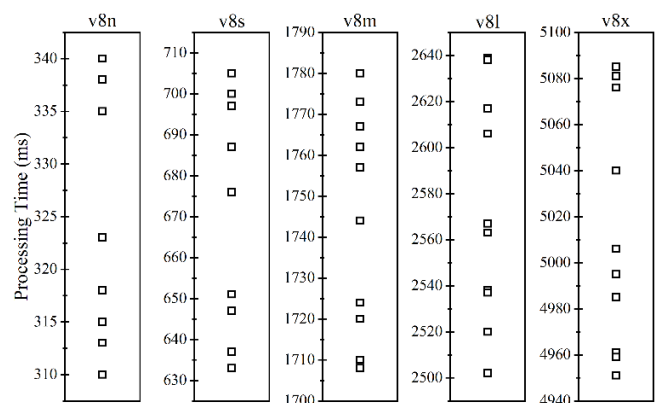


Figure 1. Exploring the time efficiency across different YOLOv8 model variants for image processing

In the experimental scrutiny, a comprehensive battery of tests was executed to gauge the temporal intricacies of image processing, from the genesis of image capture to the precise placement of bounding boxes. A series of trials were conducted employing a standardized protocol, leveraging images each boasting dimensions of 360×360 pixels, serving as the empirical foundation upon which insights were gleaned. The salient outcomes of these empirical inquiries have been visualized in Figure 1, providing a visual narrative of the

discerned trends and nuances. Amidst the intricate tapestry of findings, a notable revelation emerges: within dynamic and fluid environments, the diminutive stature of v8's v8n variant emerges as the paragon of real-time processing prowess. Its nimble architecture and judicious utilization of computational resources render it uniquely adept at navigating the ever-shifting terrain of rapidly evolving scenes, where the exigencies of time demand swift and decisive action. This proclivity positions v8n as the quintessential choice in scenarios where temporal exigencies reign supreme, offering a seamless fusion of speed and precision that is indispensable in the face of dynamicity.

However, amidst this narrative of real-time exigencies, it is imperative to acknowledge the multifaceted landscape of practicality and resource constraints. While the v8n variant emerges as the torchbearer of real-time proficiency, its brethren within the v8 lineage harbor their own unique virtues and capabilities. In resource-constrained milieus, where the exigencies of time may be tempered by pragmatic considerations, alternate models within the v8 pantheon offer viable avenues of exploration. Though they may not boast the instantaneous responsiveness of their v8n counterpart, these models nonetheless proffer a reservoir of computational acumen, capable of navigating less dynamic terrains with commendable accuracy and efficacy. Thus, within the labyrinthine expanse of computational vision, the choice of v8 model transcends mere performance metrics, embracing a holistic understanding of contextual exigencies and operational imperatives.

Performance metrics play a crucial role in evaluating the efficacy and reliability of machine learning models, particularly in tasks such as object detection, classification, and segmentation. Among the myriad metrics available, Precision (P), Recall (R), mean Average Precision (mAP), and F1 score stand out as fundamental measures that provide deep insights into the performance of a model. Precision, often referred to as positive predictive value, measures the proportion of true positive predictions among all positive predictions made by the model. It is calculated as the ratio of true positives (TP) to the sum of true positives and false positives (FP) as indicated in (1). In other words, precision quantifies the accuracy of positive predictions, emphasizing the model's ability to avoid false alarms. A high precision indicates that the model makes few false positive predictions, making it particularly valuable in applications where false positives can have significant consequences, such as medical diagnosis or security surveillance.

$$P = \frac{TP}{TP+FP} \quad (1)$$

On the other hand, recall, also known as sensitivity, measures the proportion of true positives that are correctly identified by the model out of all actual positive instances in the dataset. It is calculated as the ratio of true positives to the sum of true positives and false negatives as given in (2). Recall gauges the model's ability to capture all relevant instances of the target class, irrespective of false alarms. High recall is essential in scenarios where missing even a single positive instance can be detrimental, such as disease detection or anomaly identification in industrial processes.

$$R = \frac{TP}{TP+FN} \quad (2)$$

While precision and recall provide valuable insights individually, they often trade-off against each other. A model can achieve high precision by being conservative in its predictions, but this may lead to a lower recall as it might miss some positive instances. Conversely, a model with high recall might generate more false positives, thus lowering its precision. This trade-off is encapsulated in the F1 score, which represents the harmonic mean of precision and recall. It is calculated as:

$$F1 = \frac{2PR}{P+R} \quad (3)$$

ensuring that both precision and recall contribute equally to the final score. The F1 score provides a balanced assessment of a model's performance, particularly in scenarios where achieving both high precision and high recall is desirable.

In object detection tasks, where models generate multiple bounding boxes or segmentation masks for each object instance, evaluating performance becomes more nuanced. This is where mAP comes into play. mAP is commonly used to assess the accuracy of object detection models by considering the precision-recall curve across different confidence thresholds. It calculates the average precision across all classes and then computes the mean of these average precisions as indicated below:

$$mAP = \frac{1}{n} \sum_{i=1}^n AP_i \quad (4)$$

mAP provides a comprehensive measure of a model's ability to detect objects across various classes with varying levels of confidence. Higher mAP values indicate better detection performance, reflecting the model's capacity to accurately localize and classify objects in complex scenes.

In object detection evaluation metrics, the concept of mAP@n serves as a pivotal indicator, delineating the performance of detection systems under specific conditions. This metric encapsulates the precision of object localization and classification by imposing a stringent criterion: a minimum overlap of n%, as delineated by the Intersection over Union (IoU), must be established between the bounding box predicted by the detection system and the ground truth bounding box for an object to be deemed correctly identified. In essence, mAP@n gauges the efficacy of detection algorithms in discerning objects with a requisite level of accuracy, taking into account the spatial alignment and coverage between predicted and ground truth bounding boxes. This evaluation metric is instrumental in assessing the robustness and reliability of object detection systems across diverse scenarios, providing insights into their ability to precisely delineate object boundaries and classify them with fidelity.

Figure 2 provides a comprehensive exploration of the performance landscape, exploring the efficacy of different detection model configurations through the lens of both mAP@50 and mAP@50-95 metrics. These metrics offer a nuanced assessment by averaging performance evaluations across a spectrum of IoU thresholds, capturing the model's ability to accurately detect objects across varying degrees of overlap between predicted and ground truth bounding boxes. The analysis of the results unravels a notable trend: a discernible decrease in performance accompanies the adoption of smaller model architectures. While compact models with fewer parameters may offer computational advantages and

expedited inference times, they inherently sacrifice detection accuracy. This trade-off becomes particularly pronounced when scrutinizing performance measurements across diverse IoU thresholds, underscoring the importance of balancing computational efficiency with detection precision in model selection processes.

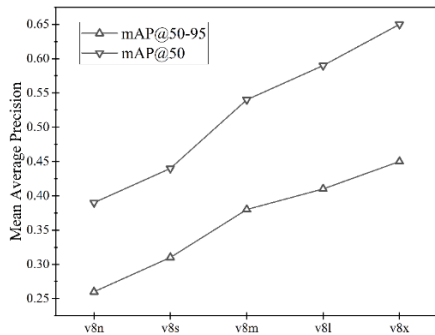


Figure 2. Exploring the variability in mean average precision (mAP) across different YOLOv8 model variants

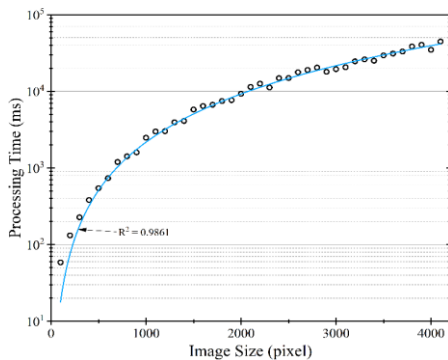


Figure 3. Analyzing the relationship between input image size and processing speed of YOLOv8n model

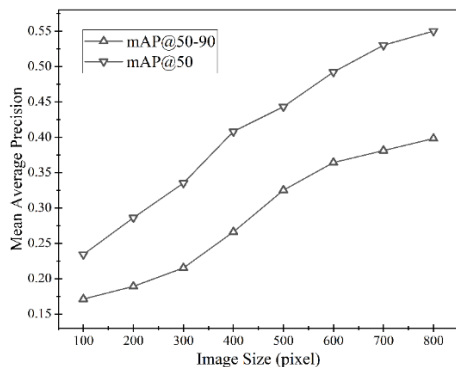


Figure 4. Impact of input image size on mean average precision (mAP) performance

In the pursuit of achieving real-time capability, it becomes essential to explore additional enhancements when deploying the leanest YOLOv8 model, v8n. Apart from merely modifying the size of the network, an alternative strategy entails decreasing the size of the source image, since it immediately impacts the inference duration. To delve deeper into this aspect, a rigorous experiment was conducted wherein a snapshot was arranged of various dimensions, with the width/height ratio consistently adjusted to one across each variation. The primary objective was to gauge the rendering time of the v8n model across these diverse input image sizes.

The findings of this comprehensive investigation, elucidating the correlation between inference times and the image dimensions, are succinctly portrayed in Figure 3. These results unearth a discernible pattern, showcasing an almost exponential scaling in inference durations as the image dimensions increase. Notably, this empirical observation highlights the tantalizing prospect of achieving rapid sensing outcomes in 325 milliseconds or less by harnessing the power of diminutive image sizes. However, it's imperative to acknowledge that opting for smaller input image sizes may entail certain trade-offs in the model's general proficiency. Figure 4 explores the dynamics between the dimensions of image and the mAP, offering a compelling investigation into how adjustments in input size could profoundly influence the precision and efficacy of the detection algorithm. Thus, while the reduction in image size presents a promising avenue for expediting inference durations, it is paramount to maintain a delicate consistency between inference time and capability to effectively address the specific exigencies of the context at hand.

Timing benchmarks were conducted to evaluate the real-time performance of v8n. The benchmarks took place on a workstation equipped with an NVIDIA GeForce RTX 2080 Ti GPU, selected for its high computational throughput crucial for real-time object detection tasks. To assess v8n's real-time capabilities, a diverse set of images representing various object densities, sizes, and environmental conditions were used. Each image underwent object detection using the v8n model, and the inference time was recorded. The results demonstrated v8n's impressive real-time performance, with an average inference time of approximately 12 milliseconds per frame on the specified platform, equivalent to around 68 frames per second. In comparison to other YOLO architecture variants like v5 and v7, v8n exhibited superior real-time performance while maintaining competitive accuracy. Its lightweight design enables efficient inference without compromising detection quality, making it well-suited for resource-constrained environments and applications requiring real-time processing. These findings highlight v8n's practical utility in various domains, including surveillance, autonomous driving, robotics, and ITS applications.

Understanding the computational resource requirements of v8n is crucial for determining the necessary hardware infrastructure for effective algorithm deployment. These requirements were evaluated on the same platform used for timing benchmarks. During the evaluation, the utilization of computational resources, including GPU, CPU, and memory, was monitored. These metrics provide key insights into the algorithm's efficiency and scalability, facilitating performance optimization and resource allocation. v8n harnesses GPU acceleration to accelerate the inference process. The experiments revealed efficient utilization of available GPU resources, resulting in high GPU utilization rates during inference. This optimized utilization ensures near-maximum GPU capacity utilization, enhancing throughput and minimizing latency. While GPU acceleration primarily supports inference, the CPU also contributes to preprocessing tasks and system-level operations. The evaluation indicated moderate CPU utilization during inference with v8n, with efficient handling of preprocessing tasks without causing bottlenecks.

Latency, on the other hand, another critical factor in assessing the real-time performance of object detection algorithms, was encountered during the evaluation of v8n.

These latency issues, which could potentially impede real-time responsiveness, stem from various factors such as computational overhead, data transfer latency, and system contention for resources. To address these latency issues and improve real-time performance, several optimization strategies were tailored to v8n. These included algorithmic optimizations and hardware acceleration techniques. The v8n algorithm underwent optimization to reduce computational complexity and streamline the inference process. This entailed fine-tuning model parameters, optimizing network architecture, and implementing efficient data processing techniques. By leveraging hardware acceleration, such as GPU acceleration, inference latency can be significantly reduced by delegating computation-intensive tasks to specialized hardware. v8n was optimized to efficiently utilize GPU resources, thereby decreasing inference latency and enhancing real-time responsiveness. To assess the effectiveness of these optimization strategies in reducing latency, latency benchmarking experiments were conducted. These experiments involved measuring the end-to-end latency of v8n under various workload scenarios and comparing it against baseline performance metrics.

2.3 Data crafting and preparing the dataset for model learning

The capability of any evaluated algorithm is intricately linked to the caliber and diversity of the set of images employed in its training process. This nexus between dataset quality and model efficacy hinges on several critical factors that collectively shape the learning process and subsequent inference capabilities. Foremost among these factors is the overall quality of the dataset itself, which encompasses aspects such as data cleanliness, annotation accuracy, and class balance. A high-quality dataset ensures that the model is exposed to accurate and representative examples of the objects it is tasked with detecting, thereby laying a robust foundation for learning. Moreover, the inclusion of diverse image variations within the dataset is paramount, as it enables the model to generalize effectively across a spectrum of real-world scenarios, encompassing variations in lighting conditions, viewpoints, occlusions, and object poses. By training on a rich and varied dataset, the model becomes adept at recognizing objects under diverse circumstances, enhancing its adaptability and robustness in real-world applications. Additionally, the abundance of samples available for training plays a crucial role in model performance, as it affords the model ample opportunities to learn and generalize patterns from the data. A larger dataset facilitates more comprehensive coverage of the object space, mitigating the risk of overfitting

and enabling the model to discern subtle nuances and variations inherent in real-world data. Thus, by prioritizing dataset quality, diversity, and abundance, significant enhancements can be made to the model's performance and generalization capabilities, thereby paving the way for the development of more reliable and effective solutions.

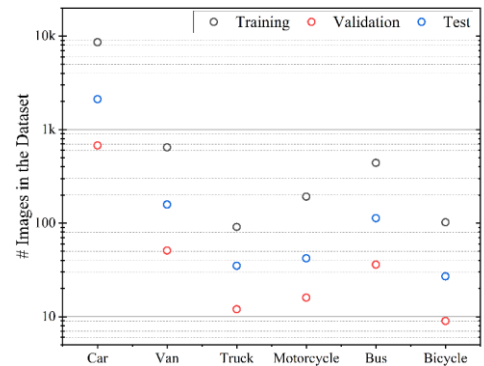


Figure 5. The diversity of vehicle classes within the dataset

In the preparation for model training and testing, a comprehensive dataset featuring a multitude of motor vehicle images was curated. This involved a rigorous process initiated by a series of planned flight tests utilizing a drone system. During these flights, the drone systematically captured an extensive array of images, traversing various terrains and environmental conditions to ensure diversity in the dataset. Upon completing the flights, the recorded video footage underwent post-processing, during which it was divided into specific time intervals, resulting in frames being extracted from each video. This extraction process ensured a rich and varied dataset, capturing a wide array of vehicle instances in different contexts. Subsequently, the acquired dataset underwent careful segregation, with distinct subsets earmarked for training, validation, and testing purposes. The allocation of data into these subsets was adjusted to ensure balanced representation across all classes and to prevent bias in model evaluation. Providing a visual insight into the dataset's composition, Figure 5 offers a detailed depiction of the distribution of vehicle classes, showcasing the diversity and breadth of images encompassed within. It's important to highlight that this dataset not only boasts quantity but also encompasses a diverse range of images, reflecting various vehicle types, sizes, colors, and environmental settings. This diversity serves as a cornerstone for robust model training, enabling the developed algorithms to generalize effectively across real-world scenarios and achieve superior performance in deployment.



Figure 6. Sample images from the training dataset

In Figure 6, a selection of images from the curated dataset is showcased, offering a glimpse into the diverse array of motor vehicle instances captured during the drone flights. These images encapsulate a spectrum of scenarios, ranging from bustling urban streets to serene countryside roads, each presenting unique challenges and characteristics for object detection algorithms. Through this visual representation, the richness and complexity of the dataset can be seen, witnessing firsthand the variability in vehicle types, sizes, colors, and environmental conditions. This diverse collection serves as evidence of the thoroughness of the dataset preparation process and underscores the importance of capturing real-world variability to ensure the robustness and efficacy of the trained models.

2.4 Training of the network

Training a detection algorithm is an iterative process that entails feeding a model with labeled data to enable it to learn and recognize patterns in images or videos. This process typically begins with the preparation of a comprehensive dataset containing images or videos annotated with bounding boxes or segmentation masks indicating the location and class of objects of interest. Once the dataset is curated, the training commences by feeding batches of data into the model iteratively. During each iteration, the model adjusts its internal parameters through backpropagation, gradually optimizing its performance based on the provided data and the defined objective function, such as minimizing classification errors or maximizing object localization accuracy. This iterative training process continues until the model converges to a satisfactory level of performance, as measured by predefined evaluation metrics. Throughout training, it is essential to monitor the model's performance on a separate validation set to prevent overfitting and ensure generalization to unseen data. Hyperparameter tuning, such as adjusting learning rates or network architectures, may also be performed iteratively to enhance model performance further.

The process of training the algorithm was carried out using the flexible platform named Google Colaboratory, commonly known as Colab. Google Colab provides a cloud-based environment that allows researchers and developers to run Python code in a Jupyter notebook format without the need for expensive hardware or local installations. Leveraging the power of Google's infrastructure, Colab offers free access to GPUs and TPUs, enabling accelerated training of deep learning models without the burden of managing hardware resources. This cloud-based platform facilitates seamless collaboration and sharing of code and notebooks, making it an ideal choice for training detection algorithms, especially for those working with limited computational resources. Additionally, Colab provides pre-installed libraries and frameworks commonly used in machine learning and deep learning, streamlining the setup process and allowing users to focus on model development and testing.

During the thorough training and validation phases lasting for 300 epochs, the model underwent a significant development, demonstrating a gradual refinement in its performance metrics and capabilities. The results of this are visually depicted in Figure 7 and Figure 8, where the evolution of key performance indicators is graphically presented. The figures provide a comprehensive overview of the model's progression over time, showcasing trends in metrics such as accuracy, loss, and convergence rates. From the initial epochs

characterized by fluctuations and uncertainties to the latter stages marked by stability and convergence, the graphical representations offer key insights into the efficacy of the training regimen and the model's adaptability to the dataset. The model's performance evaluation involved assessing three core metrics: box loss, object loss, and classification loss. These indicators serve as vital indicators of the model's proficiency in object detection tasks, each encapsulating distinct aspects of performance related to localization accuracy, object presence detection, and class prediction precision. Box loss quantifies the disparity between predicted bounding boxes and ground truth annotations, reflecting the model's ability to accurately localize objects within an image. Object loss measures the model's capability to detect the presence of objects within the scene, penalizing false negatives and rewarding true positives. Classification loss evaluates the model's accuracy in predicting the correct class label for detected objects, reflecting its proficiency in object recognition and classification.

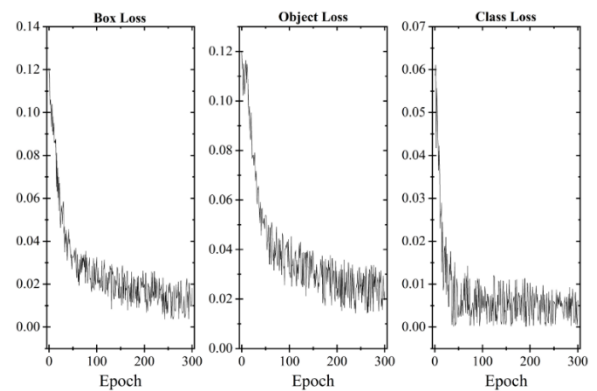


Figure 7. Evolution of loss metrics across training epochs for the model

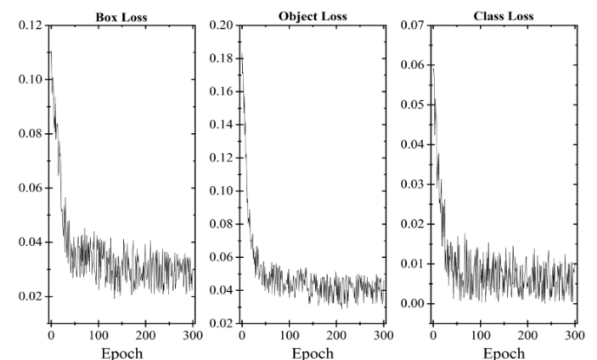


Figure 8. Evolution of loss metrics across validation epochs for the model

2.5 Dataset spectrum and spatial distribution

In the pursuit of optimizing algorithmic performance, the constitution of objects within the dataset utilized for training detection algorithms assumes paramount importance. Beyond merely ensuring the presence of objects, the quality and diversity of object instances play a pivotal role in shaping the algorithm's efficacy. It is imperative that the dataset encompasses a broad spectrum of object categories, encompassing various classes, sizes, orientations, and contextual settings. This rich diversity ensures that the algorithm is exposed to a comprehensive range of real-world

scenarios, fostering adaptability and resilience in its learning process. Furthermore, the spatial distribution of objects within the dataset also warrants careful consideration. It is essential not only for objects to be evenly distributed across the dataset but also for their spatial arrangement to reflect real-world conditions. For instance, in a dataset focused on traffic surveillance, vehicles should be represented in varying densities and spatial configurations, mirroring the complexities of urban and rural road networks. This holistic approach to dataset curation ensures that the algorithm is trained on a representative sample of real-world scenarios, enhancing its ability to generalize effectively and perform robustly in diverse environments.

However, deviations from this ideal scenario can introduce biases into the model, potentially compromising its effectiveness and reliability. Biases can arise from various sources, including imbalances in the dataset composition, annotation errors, or systemic biases in data collection methods. Moreover, biases can also manifest in the form of label noise, where inaccuracies or inconsistencies in annotation labels affect the model's learning process. These biases not only undermine the fairness and equity of the algorithm but also pose ethical and societal concerns, particularly in applications where algorithmic decisions can have significant real-world implications. To address these challenges, it is imperative to curate datasets with a balanced representation of object classes, ensuring that each category is adequately sampled to mitigate biases and promote equitable learning across all classes. This may involve employing sampling techniques such as stratified sampling or data augmentation to address imbalances in the dataset. Additionally, rigorous quality control measures can be implemented to identify and rectify annotation errors, ensuring the integrity and reliability of the dataset.

Moreover, fostering adaptability within the model architecture serves as a crucial mechanism to mitigate the risks associated with dataset biases. An adaptable model can effectively learn from the variability present in the dataset, adjusting its internal representations and decision boundaries to accommodate diverse object distributions. This adaptability is often achieved through the use of techniques such as regularization, dropout, and ensemble learning, which help prevent the model from becoming overly reliant on specific features or patterns present in the training data. By promoting adaptability, the algorithm's capacity can be enhanced to generalize across diverse scenarios, thereby improving its efficacy and reliability in real-world applications. Additionally, techniques such as transfer learning and domain adaptation can further bolster the model's adaptability by leveraging knowledge gained from related tasks or domains to improve performance on the target task or domain. Furthermore, interpretability and explainability techniques can help shed light on the decision-making process of the model, allowing stakeholders to identify and mitigate biases that may arise during training.

In conjunction with addressing dataset biases and fostering model adaptability, the collection of image data via the drone has significantly contributed to enhancing the diversity and spatial arrangement of the dataset. Leveraging the drone for image acquisition has provided unparalleled versatility and flexibility, enabling the capture of images from vantage points and perspectives that would otherwise be inaccessible or impractical using traditional ground-based methods. This aerial perspective not only facilitated the collection of data

across vast geographical areas but also allowed for the capture of images under varying environmental conditions, including urban, rural, and natural landscapes. Furthermore, the dynamic nature of drone flight has enabled the capture of images with diverse spatial arrangements, encompassing varying distances, angles, and elevations relative to the objects of interest. Consequently, the dataset has benefited from a rich variety of image compositions, thereby reflecting the complexities and intricacies inherent in real-world scenarios.

3. RESULTS

Following an extensive training and validation process, the performance of v8n was assessed using the test dataset. While this dataset featured similar proportions of vehicle images compared to the other two datasets, it generally comprised somewhat more challenging images. It encompassed a diverse range of variations, including scenarios with heavy traffic, low-light conditions, and narrow-angle images, aligning closely with real-world situations. This approach allowed to scrutinize v8n's adequacy in handling these complex scenarios and gauge its ability to generalize within the context of ITS. A thorough analysis of the results was conducted, aiming to identify the model's weaknesses, limitations, and instances of superior performance. The performance evaluation encompassed a thorough examination of various key metrics to gauge the efficacy of v8n. Within this framework, precision, recall, mean Average Precision (mAP), and the F1 score were assessed. Beyond these fundamental metrics, the analysis also included an evaluation of the detection speed to assess how efficiently the algorithm processes and identifies motor vehicles.

Figure 9 displays sample detection results illustrating instances where v8n faced challenges, made incorrect detections, or failed to detect certain vehicles. These results can be summarized as follows: i) The model erroneously identifies two roadside objects as motorcycles. Additionally, an object situated in the median between inbound and outbound lanes is misclassified as a motorcycle. This is attributed to the image being captured from a high altitude, where small vehicles like motorcycles can easily blend with the background. ii) In this image taken under low-light conditions, numerous vehicles elude detection by the model. Nevertheless, it successfully detects most of the vehicles in the image. iii) A motorcycle in the left lane remains undetected due to its compact size. Furthermore, the trams visible in the image, which were not part of the model's training data, are incorrectly classified as buses. iv) While small-scale vehicles present in this image may not be detected, the model encounters difficulty in identifying vans positioned along the roadside, partially obscured by trees and similar objects. v) In the image captured under low-light conditions, one car, despite its proximity relative to other vehicles, remains undetected. This highlights the significance of vehicle color in the detection criteria. vi) This daytime image exemplifies the model's performance in the presence of shadows cast on side streets. Many vehicles experiencing reduced visibility due to pronounced shadows go unnoticed by the model. vii) In this image portraying what could be considered heavy traffic, some vehicles, although adequately lit, evade detection due to their diminutive size. Moreover, an object situated by the roadside, unrelated to a vehicle, is mistakenly identified as a car. viii) In this instance, captured at a high altitude and under low lighting

conditions, a traffic sign on the road is inaccurately recognized as motorcycles. ix) Within an image taken under low-light conditions and an inclined camera angle, the model fails to detect two vehicles. Notably, despite the poor visibility of the darker vehicle, the other vehicle by the roadside also goes unnoticed. x) In the final example, two motorcycles in an image taken at a high altitude escape detection due to their diminutive size. Furthermore, an object unrelated to a vehicle by the roadside is erroneously identified as a car.

Table 1 presents an overview of the model's performance measurements on the test dataset. The images, captured under varying conditions, were categorized to assess v8n's performance under each specific scenario. Under adequate lighting conditions, v8n exhibited notably successful outcomes in terms of precision (P), recall (R), mAP and F1 scores. These high values, particularly in the detection of cars and buses, can be attributed to the distinct topological characteristics of these vehicle types. Conversely, metrics yielded relatively lower results for vans due to their diverse configurations. Trucks, on the other hand, displayed smaller performance metrics because of the wide range of trailer variations. Notably, in trucks with open trailers, the model occasionally made erroneous identifications, often mistaking the trailer's contents as part of the background. The detection of motorcycles and bicycles presented challenges for the model, given their smaller size compared to other vehicles. Even under adequate lighting conditions, the model occasionally struggled to detect them. In scenarios with

reduced lighting, a predictable decrease in performance metrics was observed across all vehicle classes. These performance values remained consistent across different classes, maintaining self-consistency. Images captured at higher altitudes demonstrated a significant drop in performance, with motorcycles and bicycles exhibiting particularly pronounced reductions. However, it's noteworthy that, in the case of images taken from inclined angles, apart from cars, all other classes displayed the highest mAP and F1 values. The elevated performance in such conditions can be attributed to several factors. Vans, at times, resemble either trucks or cars in bird's-eye views. However, in inclined images, they can be more distinctively differentiated from other vehicles. In the case of trucks, reduced trailer variety, where the trailer's interior isn't visible, contributed to enhanced detection capabilities. Buses maintained a detection rate similar to adequate lighting conditions, primarily due to their distinct topological features. The most substantial improvement in detection accuracy was observed for motorcycles and bicycles. This improvement stemmed from the clarity in distinguishing riders on these vehicles in images captured at inclined angles, mitigating uncertainty in bird's-eye view or high-altitude images. However, it's worth noting that motorcycle and bicycle classes could sometimes be mistaken for each other. In addition to these, Figure 10 presents the precision/recall curve, offering a comprehensive framework.



Figure 9. Examples of cases where v8n faces detection challenges

Table 1. Performance metrics of v8n for each object class

Image Category	Indicator	Class					
		Car	Van	Truck	Bus	Motor	Bicycle
Adequate Lighting	P	92.6	71.7	72.5	85.2	68.8	47.3
	R	73.4	55.7	62.0	76.6	35.5	16.8
	mAP	87.4	65.5	67.3	83.1	62.7	43.6
Low Lighting	F1	81.9	62.7	66.8	80.6	46.8	24.8
	P	85.6	66.9	70.2	83.3	57.4	36.2
	R	67.5	54.3	56.7	70.6	31.2	14.0
High Altitude	mAP	81.5	62.2	65.1	79.8	55.7	31.5
	F1	75.5	59.9	62.7	76.4	40.4	20.2
	P	85.1	66.8	66.9	84.1	50.2	30.3
Inclined	R	64.8	51.9	61.3	73.7	28.0	10.1
	mAP	79.6	60.1	65.9	81.4	44.4	23.7
	F1	73.6	58.4	64.0	78.5	35.9	15.1
Inclined	P	88.7	72.4	79.8	86.4	70.0	63.3
	R	67.7	58.7	65.9	75.4	40.5	32.6
	mAP	83.3	67.5	74.8	83.4	68.4	59.9
	F1	76.8	64.8	72.1	80.5	51.3	43.0

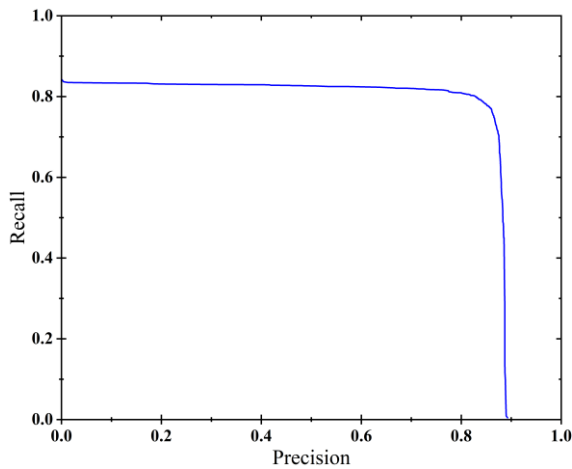


Figure 10. Precision/recall curve for YOLOv8n

Predicted	Ba	0.11	0.08	0.17	0.03	0.45	0.68	0
	Bi	0	0	0	0	0.16	0.13	0.37
	M	0	0	0	0	0.37	0.16	0.21
	Bu	0.01	0.05	0.06	0.82	0	0	0
	T	0	0.04	0.71	0.11	0	0	0.13
	V	0.04	0.68	0.05	0.02	0.01	0.01	0.04
	C	0.84	0.15	0.01	0.02	0.01	0.02	0.25
	True	C	V	T	Bu	M	Bi	Ba

Figure 11. Confusion matrix illustrating the detection accuracy of YOLOv8n across vehicle classes

The success and misclassification rates for detecting vehicle classes are presented in greater detail within the confusion matrix depicted in Figure 11. In addition to the previously mentioned information, it's worth noting that the detection of cars exhibits a low misclassification rate, with only 11% going undetected. One notable observation is that 15% of vans are misclassified as cars, and they can also be erroneously categorized as trucks or buses depending on their specific configurations. The challenge of detecting trucks as background, due to the variability of their trailers, is evident with a misclassification rate of 17%. Furthermore, 11% of buses are misclassified as trucks despite their distinct geometry, indicating an area for potential improvement. The matrix values also highlight the difficulties in detecting motorcycles and bicycles, primarily because of their small size. Consequently, these classes are often incorrectly classified among themselves.

4. DISCUSSION

In particular, when compared to previous iterations of YOLO utilized in ITS studies, the superior performance of YOLOv8, particularly v8n, becomes apparent. The results demonstrate that v8n exhibits significant improvements over v3 in terms of accuracy and speed. While v3 introduced notable enhancements in object detection, v8n surpasses it

with superior precision and recall rates, particularly under challenging conditions. Moreover, v8n showcases improved efficiency, enabling higher inference speeds without compromising accuracy. On the other hand, v8n demonstrates noteworthy advancements over v4 in both accuracy and efficiency. While v4 introduced cutting-edge features such as CSPDarknet53 and PANet, v8n further refines these innovations with its ELAN structure, resulting in higher detection accuracy and faster processing speeds. Additionally, v8n's optimized architecture and inference pipeline contribute to improved performance metrics compared to v4. Furthermore, v8n surpasses v5, which has been extensively adopted in ITS studies, in terms of adaptability. While v5 presents advancements in model architecture and training techniques, v8n builds upon these foundations to achieve even higher precision and recall rates. Moreover, v8n's specialized optimizations for real-time vehicle detection within ITS applications demonstrate superior performance in challenging scenarios, especially concerning inference time. Similarly, v8n demonstrates superiority over v6 in accuracy, speed, and scalability. While v6 introduces improvements in model efficiency and training strategies, v8n advances these enhancements to achieve faster processing speeds. Additionally, v8n's architecture offers scalability and adaptability for various ITS applications. Finally, v8n exhibits superiority over v7 in terms of detection accuracy and overall efficiency. Its streamlined architecture and improved inference pipeline enable faster processing speeds and higher detection accuracy in real-time applications compared to v7.

While the study utilized a drone to gather vehicle images for real-time detection using v8n, it's essential to acknowledge potential biases in the dataset. It's probable that the dataset doesn't fully encompass all conceivable scenarios and variations encountered in real-world settings. For instance, certain vehicle types, colors, or orientations might be disproportionately represented, leading to bias in the algorithm's performance. Given that the images were captured from specific points in the city center, they may lack comprehensive coverage of all vehicle types. For example, while buses may be abundant in urban areas, the number of trucks could be limited, a distribution that might differ if the images were obtained from a highway. Additionally, environmental factors potentially influence UAV operation and consequently affect the quality of the collected data in specific instances. Variables such as weather conditions (e.g., wind, rain, fog), lighting conditions (e.g., bright sunlight, low light), and terrain variability (e.g., urban versus rural areas) can impact the UAV's stability, image quality, and maneuverability. These environmental factors introduce variability and uncertainty into the dataset, potentially influencing the vehicle detection algorithm's performance. While these factors were considered during data collection, variations in the dataset could arise due to diverse environmental conditions in different locations. For instance, collecting images with drones in urban areas may face constraints such as bridges and tall buildings along the flight path. Conversely, wind might pose limitations in rural areas due to open spaces, depending on the location. Additionally, since urban images typically feature more complex backgrounds compared to rural settings, the algorithm may encounter difficulties in vehicle detection. Moreover, despite tests demonstrating v8n's robustness and effectiveness as an object detection algorithm, it has inherent limitations. v8n, like its counterparts, may encounter challenges such as occlusions,

varying scales, and complex backgrounds. Furthermore, v8n's performance is directly influenced by factors such as image resolution, input size, and the quality of training data.

YOLOv8n presents a promising solution for numerous ITS applications, including traffic monitoring and surveillance, traffic flow optimization, infrastructure monitoring, and pedestrian and cyclist safety. Firstly, v8n can revolutionize traffic monitoring and surveillance by enabling real-time detection, tracking, and classification of vehicles on roadways. This capability facilitates the identification of traffic violations, monitoring of congestion levels, and enhancement of overall road safety through proactive incident detection and response. Furthermore, v8n's robust performance in object detection allows for accurate and efficient traffic flow optimization. By providing real-time data on vehicle movement patterns and identifying congestion hotspots, v8n can assist traffic management authorities in making informed decisions to alleviate traffic congestion and improve traffic efficiency. Moreover, v8n can play a crucial role in infrastructure monitoring by detecting anomalies and potential hazards in transportation infrastructure, such as bridges, tunnels, and roadways. By continuously monitoring infrastructure integrity, it contributes to proactive maintenance planning and ensures the safety and reliability of transportation networks. Additionally, v8n's capabilities extend to enhancing pedestrian and cyclist safety by detecting and alerting drivers to the presence of vulnerable road users at intersections, crosswalks, and bike lanes. Integrating v8n into existing ITS frameworks allows for the leveraging of its advanced object detection capabilities to create safer and more efficient urban environments for all road users.

5. CONCLUSION

After a comprehensive examination of all the sub-models within the YOLOv8 series, it has become evident that YOLOv8n is the most suitable model for traffic monitoring applications within Intelligent Transportation Systems (ITS). In this context, a series of performance tests were conducted to assess the processing speed of YOLOv8n and its dependence on input image size. While YOLOv8n's detection success may not reach the levels achieved by larger-scale models like YOLOv8l and YOLOv8x, it stands out as the only model capable of meeting the real-time requirement with its high-speed detection capabilities. However, it's important to acknowledge certain drawbacks associated with model size, especially concerning specific object detection. While vehicles with standard geometries like cars or buses can be relatively easily detected, the challenges intensify when it comes to vans and trucks, which can have varying configurations. Moreover, motorcycles and bicycles, due to their small size, pose difficulties in detection. Additionally, detecting vehicles partially obscured by various objects such as traffic signs and trees can be problematic.

The results of the performance tests affirm that YOLOv8n performs admirably in real-time applications. The algorithm demonstrates a high detection success rate in scenarios with adequate lighting but exhibits relatively lower performance in images captured in low-light conditions or at high altitudes. Notably, it excels in high-level detection of various vehicle classes, except for cars, particularly in images taken at medium and low altitudes and in inclined situations where the camera is positioned between 30 and 50 degrees. This success can be

attributed to the enhanced visibility of vehicle characteristics at this altitude and inclined perspective. The algorithm, boasting an impressive average detection speed of just 12 milliseconds, attains a detection accuracy exceeding 80% in scenarios with ample lighting conditions. This remarkable performance firmly establishes it as a leading contender among the existing object detection algorithms employed in real-time traffic imaging applications.

One of the key advancements in YOLOv8 seems to stem from the inclusion of the C2f module, which enhances the structural design to optimize data propagation through YOLOv8. This integration of the ELAN concept with the C3 module appears to enhance detection capabilities, resulting in an increased gradient information flow. Furthermore, the decoupled head structure in v8 shows potential for attaining superior precision in detection. By partitioning confidence and regression data, the model's efficacy in object localization and classification is bolstered, resulting in heightened accuracy and refined predictions.

Potential avenues for future research involving YOLOv8n could include further enhancing its accuracy and efficiency. It is imperative to explore methods that can advance the overall detection performance of v8n for real-time vehicle detection, particularly in complex urban environments and adverse weather conditions. Additionally, there are opportunities to investigate the integration of v8n with other sensor modalities like LiDAR and radar, aiming to enhance object detection performance and robustness across diverse traffic scenarios. Techniques for adapting v8n to dynamic traffic environments, characterized by fluctuating conditions such as construction zones, temporary road closures, and special events, warrant examination. Furthermore, addressing privacy and ethical concerns associated with the deployment of AI-based surveillance systems in public spaces is essential. Various strategies can be proposed to ensure the responsible and transparent utilization of v8n in ITS applications. By integrating these enhancements, a more comprehensive discussion on the potential ITS applications of v8n can be presented.

REFERENCES

- [1] Małeckı, K., Iwan, S., Kijewska, K. (2014). Influence of intelligent transportation systems on reduction of the environmental negative impact of urban freight transport based on Szczecin example. *Procedia - Social and Behavioral Sciences*, 151: 215-229. <https://doi.org/10.1016/j.sbspro.2014.10.021>
- [2] Lee, W.H., Tseng, S.S., Shieh, W.Y. (2010). Collaborative real-time traffic information generation and sharing framework for the intelligent transportation system, *Information Sciences*, 180(1): 62-70. <https://doi.org/10.1016/j.ins.2009.09.004>
- [3] Javed, M.A., Zeadally, S., Hamida, E.B. (2019). Data analytics for cooperative intelligent transport systems, *Vehicular Communications*, 15: 63-72. <https://doi.org/10.1016/j.vehcom.2018.10.004>
- [4] Iyer, L.S. (2021). AI enabled applications towards intelligent transportation. *Transportation Engineering*, 5: 100083. <https://doi.org/10.1016/j.treng.2021.100083>
- [5] Zhang, H., Lu, X. (2020). Vehicle communication network in intelligent transportation system based on

- Internet of Things. *Computer Communications*, 160: 799-806. <https://doi.org/10.1016/j.comcom.2020.03.041>
- [6] Olayode, I.O., Tartibu, L.K., Okwu, M.O., Uchechi, U.F. (2020). Intelligent transportation systems, un-signalized road intersections and traffic congestion in Johannesburg: A systematic review. *Procedia CIRP*, 91: 844-850. <https://doi.org/10.1016/j.procir.2020.04.137>
- [7] Ennehar, B.C., Samra, B. (2023). Master-slave convolutional deep architecture for vehicle identification and type classification. *Traitement du Signal*, 40(2): 619-627. <https://doi.org/10.18280/ts.400220>
- [8] Kumar, R.D., Rammohan, A. (2023). Revolutionizing intelligent transportation systems with cellular vehicle-to-everything (C-V2X) technology: Current trends, use cases, emerging technologies, standardization bodies, industry analytics and future directions. *Vehicular Communications*, 43: 100638. <https://doi.org/10.1016/j.vehcom.2023.100638>
- [9] Bakirci, M., Ozer, M.M. (2023). Post-disaster area monitoring with swarm UAV systems for effective search and rescue. In 10th International Conference on Recent Advances in Air and Space Technologies, Istanbul, Turkey, pp. 1-6. <https://doi.org/10.1109/RAST57548.2023.10198022>
- [10] Nie, Z., Farzaneh, H. (2022). Real-time dynamic predictive cruise control for enhancing eco-driving of electric vehicles, considering traffic constraints and signal phase and timing (SPaT) information, using artificial-neural-network-based energy consumption model. *Energy*, 241: 122888. <https://doi.org/10.1016/j.energy.2021.122888>
- [11] Yu, L., Qin, H.W., Zhang, C., Wang, J., Zou, J. (2023). Saliency object detection method based on real-time monitoring image information for intelligent driving. *Traitement du Signal*, 40(3): 1025-1033. <https://doi.org/10.18280/ts.400318>
- [12] An, S., Yang, H., Wang, J., Cui, N., Cui, J. (2016). Mining urban recurrent congestion evolution patterns from GPS-equipped vehicle mobility data. *Information Sciences*, 373: 515-526. <https://doi.org/10.1016/j.ins.2016.06.033>
- [13] Yu, L., Zhang, B.L., Li, R. (2020). Detection of unusual targets in traffic images based on one-class extreme machine learning. *Traitement du Signal*, 37(6): 1003-1008. <https://doi.org/10.18280/ts.370612>
- [14] Bakirci, M., Cetin, M. (2023). Reliability of MEMS accelerometers embedded in smart mobile devices for robotics applications. In: Garcia Marquez, F.P., Jamil, A., Eken, S., Hameed, A.A. (eds) *Computational Intelligence, Data Analytics and Applications. ICCIDA 2022. Lecture Notes in Networks and Systems*, vol 643, pp. 78-90. https://doi.org/10.1007/978-3-031-27099-4_7
- [15] Li, S.J. (2020). Forecast of traffic flow state during large competition events. *Revue d'Intelligence Artificielle*, 34(2): 227-232. <https://doi.org/10.18280/ria.340214>
- [16] Ali, F., Ali, A., Imran, M., Naqvi, R.A., Siddiqi, M.H., Kwak, K.S. (2021). Traffic accident detection and condition analysis based on social networking data. *Accident Analysis & Prevention*, 151: 105973. <https://doi.org/10.1016/j.aap.2021.105973>
- [17] Mercader, P., Haddad, J. (2020). Automatic incident detection on freeways based on bluetooth traffic monitoring. *Accident Analysis & Prevention*, 146: 105703. <https://doi.org/10.1016/j.aap.2020.105703>
- [18] Tikar, S.S., Patil, R.A. (2022). A novel fast responding driver assistance technique with efficient lane detection and collision avoidance using dynamic feature extraction in any environment. *Traitement du Signal*, 39(2): 459-468. <https://doi.org/10.18280/ts.390207>
- [19] Kumar, A., Krishnamurthi, R., Nayyar, A., Luhach, A.K., Khan, M.S., Singh, A. (2021). A novel Software-Defined Drone Network (SDDN)-based collision avoidance strategies for on-road traffic monitoring and management. *Vehicular Communications*, 28: 100313. <https://doi.org/10.1016/j.vehcom.2020.100313>
- [20] Wang, L., Chen, F., Yin, H. (2016). Detecting and tracking vehicles in traffic by unmanned aerial vehicles. *Automation in Construction*, 72(3): 294-308. <https://doi.org/10.1016/j.autcon.2016.05.008>
- [21] Wu, H., Sun, X.Y., Zhong, J., Cao, F.Y. (2022). A traffic parameter detection algorithm based on double coils. *Traitement du Signal*, 39(5): 1621-1629. <https://doi.org/10.18280/ts.390519>
- [22] Wang, Y. (2020). Moving vehicle detection and tracking based on video sequences. *Traitement du Signal*, 37(2): 325-331. <https://doi.org/10.18280/ts.370219>
- [23] Patel, H., Panchal, A. (2016). License plate detection using harris corner and character segmentation by integrated approach from an image. *Procedia Computer Science*, 79: 419-425. <https://doi.org/10.1016/j.procs.2016.03.054>
- [24] Trivedi, J.D., Mandalapu, S.D., Dave, D.H. (2022). Vision-based real-time vehicle detection and vehicle speed measurement using morphology and binary logical operation. *Journal of Industrial Information Integration*, 27: 100280. <https://doi.org/10.1016/j.jii.2021.100280>
- [25] Yang, Y. (2020). A vehicle recognition algorithm based on deep convolution neural network. *Traitement du Signal*, 37(4): 647-653. <https://doi.org/10.18280/ts.370414>
- [26] Abdellatif, M.M., Elshabasy, N.H., Elashmawy, A.E., AbdelRaheem, M. (2023). A low cost IoT-based Arabic license plate recognition model for smart parking systems. *Ain Shams Engineering Journal*, 14(6): 102178. <https://doi.org/10.1016/j.asej.2023.102178>
- [27] Loey, M., Manogaran, G., Taha, M.H.N., Khalifa, N.E.M. (2021). A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. *Measurement*, 167: 108288. <https://doi.org/10.1016/j.measurement.2020.108288>
- [28] Luo, X., Shen, R., Hu, J., Deng, J., Hu, L., Guan, Q. (2017). A deep convolution neural network model for vehicle recognition and face recognition. *Procedia Computer Science*, 107: 715-720. <https://doi.org/10.1016/j.procs.2017.03.153>
- [29] Sadeghian, P., Håkansson, J., Zhao, X. (2021). Review and evaluation of methods in transport mode detection based on GPS tracking data. *Journal of Traffic and Transportation Engineering (English Edition)*, 8(4): 467-482. <https://doi.org/10.1016/j.jtte.2021.04.004>
- [30] Zhai, Y., Zeng, W.J., Li, N. (2022). A novel detection method using YOLOv5 for vehicle target under complex situation. *Traitement du Signal*, 39(4): 1153-1158. <https://doi.org/10.18280/ts.390407>
- [31] Chen, X., Chen, H., Yang, Y., Wu, H., Zhang, W., Zhao, J., Xiong, Y. (2021). Traffic flow prediction by an ensemble framework with data denoising and deep

- learning model. *Physica A: Statistical Mechanics and its Applications*, 565: 125574. <https://doi.org/10.1016/j.physa.2020.125574>
- [32] Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E., Liang, Z. (2019). Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Computers and Electronics in Agriculture*, 157: 417-426. <https://doi.org/10.1016/j.compag.2019.01.012>
- [33] Shinde, S., Kothari, A., Gupta, V. (2018). YOLO based human action recognition and localization. *Procedia Computer Science*, 133: 831-838. <https://doi.org/10.1016/j.procs.2018.07.112>
- [34] Mahaur, B., Mishra, K.K. (2023). Small-object detection based on YOLOv5 in autonomous driving systems. *Pattern Recognition Letters*, 168: 115-122. <https://doi.org/10.1016/j.patrec.2023.03.009>
- [35] Rafique, S., Gul, S., Jan, K., Khan, G.M. (2023). Optimized real-time parking management framework using deep learning. *Expert Systems with Applications*, 220: 119686. <https://doi.org/10.1016/j.eswa.2023.119686>
- [36] GitHub Inc. GitHub Developer Platform. <https://github.com/ultralytics/>, accessed on Dec. 5, 2023.