International Journal of Transport Development and Integration
Vol. 7, No. 4, December, 2023, pp. 341-351
Journal homepage: http://iieta.org/journals/ijtdi

# Enhanced SSD Algorithm-Based Object Detection and Depth Estimation for Autonomous Vehicle Navigation

Vaibhav Saini[1], MVV Prasad Kantipudi[1*], Pramoda Meduri[2]

[1] Symbiosis Institute of Technology (SIT), Symbiosis International (Deemed University) (SIU), Pune 412115, India
[2] Verolt Engineering Pvt. Ltd, Pune 411001, India

Corresponding Author Email: mvvprasad.kantipudi@gmail.com

https://doi.org/10.18280/ijtdi.070408

**Received:** 21 September 2023
**Revised:** 28 November 2023
**Accepted:** 8 December 2023
**Available online:** 28 December 2023

***Keywords:***
*autonomous vehicle, safe driving performance, perception & vision, object recognition, camera, LiDAR, RADAR, SSD, YOLO, ZED-Camera*

**ABSTRACT**

Autonomous vehicles necessitate robust stability and safety mechanisms for effective navigation, relying heavily upon advanced perception and precise environmental awareness. This study addresses the object detection challenge intrinsic to autonomous navigation, with a focus on the system architecture and the integration of cutting-edge hardware and software technologies. The efficacy of various object recognition algorithms, notably the Single Shot Detector (SSD) and You Only Look Once (YOLO), is rigorously compared. Prior research has indicated that SSD, when augmented with depth estimation techniques, demonstrates superior performance in real-time applications within complex environments. Consequently, this research proposes an optimized SSD algorithm paired with a Zed camera system. Through this integration, a notable improvement in detection accuracy is achieved, with a precision increase to 87%. This advancement marks a significant step towards resolving the critical challenges faced by autonomous vehicles in object detection and distance estimation, thereby enhancing their operational safety and reliability.

## 1. INTRODUCTION

Recent advancements in autonomous vehicles (AVs) have garnered noteworthy attention, with a commensurate increase in research dedicated to this domain [1]. A critical component of AV technology is the object detection mechanism, which incorporates artificial intelligence and sensor-based methodologies to ensure driver safety [2]. Autonomous vehicles offer the promise to enhance driving comfort and reduce incidents resulting from vehicle collisions. These vehicles are engineered to sense and navigate their environment on highways autonomously, without human intervention [3, 4].

The suite of sensors distributed throughout the vehicle is integral to its functionality. An array of sensors, including LiDARs, radars, and cameras, is employed to survey and interpret the surrounding milieu [5]. The process of environmental sensing, or perception, encompasses several sub-tasks: object detection, object classification, 3D position estimation, and simultaneous localization and mapping (SLAM). Object detection itself involves localization—determining an object's position within an image—and classification—assigning a category to each detected object, such as a traffic light, vehicle, or pedestrian [6].

In autonomous driving systems, object detection is deemed one of the most crucial processes for safe navigation. It is essential for enabling the vehicle's controller to anticipate and maneuver around potential obstacles [7]. Therefore, the employment of precise object detection algorithms is imperative. The complexity of the requisite system architecture is necessitated by the need to process a multitude of features within the vehicle [8].

In the present study, the objective is to refine object detection accuracy using robust tools such as the Zed Camera in conjunction with algorithms like SSD, which have demonstrated superior performance in real-time scenarios. The Zed Camera, in particular, has proven to be an invaluable sensor for the collection of depth data, especially in challenging and dynamic environments. A robust perception system, integrating multiple sensors and sophisticated algorithms, such as the proposed SSD Algorithm, is requisite for AVs to achieve accurate object recognition and informed decision-making. To enhance the vehicles' perceptual capabilities, reliability, and safety, a synthesis of various sensors and algorithms, including the Zed Camera, is often pursued by researchers in the field of autonomous vehicles.

### 1.1 Self-driving vehicles

The conceptualization of autonomous vehicles has undergone a remarkable evolution since the 1920s. Historical accounts reveal that in the 1980s, a self-navigating vehicle capable of achieving speeds up to 31 km/h was engineered. Progressing through the eras, the propulsion methods transitioned from steam to the combustion of gasoline and diesel, leading to the current paradigm shift towards electric propulsion. This industry has witnessed transformative advances over decades, setting the stage for the manufacture of vehicles that are not only faster but also embedded with utilitarian features.

In the accelerated pace of today's vehicular traffic, which has regrettably led to a rise in traffic incidents, the human driver has been frequently identified as the critical failure point in vehicular accidents. It has been posited that the theoretical elimination of human error through the deployment of autonomous vehicles could serve as a panacea to this issue [9]. Despite the availability of numerous tools and features that augment human capabilities, it is recognized that human oversight remains a pivotal element in the realm of automation. Systems such as cruise control, object detection, depth estimation, and the implementation of autopilot in vehicles exemplify technological advancements that support human decision-making processes [10].

**Table 1.** Level of automation in autonomous vehicle

| Levels | Types of Automation | Vehicle Operating Condition | Driver Monitor System |
|---|---|---|---|
| 0 | "No Automation, Manual Only" | Everything On | Human Driver Involvement |
| 1 | "Driver Assistance" | Hand On, Eyes On | |
| 2 | "Partial Driving Automation" | Feet Off | |
| 3 | "Conditional Driving Automation" | Hands Off | ADAS System (Driver-less) |
| 4 | "High Driving Automation" | Eyes Off | |
| 5 | "Full Driving Automation" | Mind Off | |

According to the standard shown in Table 1, autonomous vehicles are divided into six tiers LOA (Level of Automation) according to the amount of automation that is supported. Level 5 vehicles are always capable of operating without human intervention [11]. As for better functionality of autonomous vehicle it is required to have excellent system architecture with additional features, a quick overview of the development of system architecture is explained in the next section.

### 1.2 System architecture of autonomous vehicle

The architectures of Electrical and Electronic (E&E) systems in vehicles exhibit a spectrum of complexity, ranging from the rudimentary to the highly intricate, and may be bifurcated into hardware and software components. The hardware itself is further stratified into three distinct classifications: distributed, domain-based, and vehicle-computer based systems. It is posited that the design of Autonomous Vehicles (AVs) is intrinsically aimed at diminishing temporal and spatial demands, curtailing fuel consumption, mitigating collision risks, alleviating traffic congestion, and augmenting mobility, particularly for populations such as the elderly and individuals with disabilities. The architecture employed within AVs is regarded as a universal standard, underpinning the operation of both automated and non-automated vehicles alike. In the context of AVs, these architectures consist of logical or functional blocks that are meticulously architected to align with the sequence of information flow and processing tasks, extending from data acquisition to vehicle control, inclusive of internal system monitoring [12, 13].

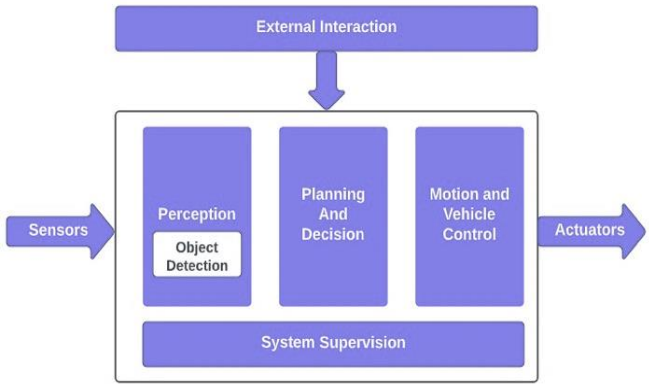


**Figure 1.** Functional block of autonomous vehicle

Depicted in Figure 1 is the functional block diagram of an Autonomous Vehicle. The architecture of each autonomous system is compartmentalized into four essential blocks. The Perception block is tasked with the assimilation of sensory information; the Planning and Decision-Making block synthesizes the acquired data, orchestrating all planning activities and making pivotal decisions; the Motion and Vehicle Control block implements the directives formulated in the preceding phase; and the System Supervision block is responsible for the ongoing surveillance of operational activities, addressing any anomalies, and instituting feature enhancements or modifications. External interactions, including rule definitions, user interfaces, and environmental data acquisition, are also integral to the system's functionality [14].

### 1.3 Challenges in autonomous vehicle

The acquisition of robust training datasets constitutes a significant impediment in the advancement of object detection systems within autonomous vehicles, with the accuracy of such systems being contingent upon the caliber of the training data. A comprehensive training dataset must encompass a vast array of objects that a vehicle is likely to encounter in various driving scenarios, including but not limited to street signs, lane markings, pedestrians, edifices, and other vehicles. Furthermore, the integrity of the dataset is paramount. During the processing stage, the quality of the images may influence the efficacy and interpretation of the algorithm. Factors such as image blur, the presence of visually similar but distinct objects requiring differentiated decision methodologies, or the ambient lighting conditions under which the images were captured can all pose challenges. For instance, the algorithm's ability to maintain detection accuracy irrespective of lighting conditions is a concern—whether an object bathed in excessive light can be equally discerned in dimmer conditions.

In addition to these challenges, the state of the roadway itself can affect the precision of object detection algorithms. Road conditions are known to fluctuate markedly, ranging from smooth surfaces with well-defined lane demarcations to degraded paths devoid of such markings. Moreover, the operational efficacy of autonomous vehicles is expected to remain consistent across a spectrum of meteorological conditions, be it under clear skies, amidst precipitation, or enveloped by fog.

The development of proficient object detection models for autonomous vehicles necessitates the procurement and annotation of an extensive corpus of high-fidelity data, which

is instrumental in augmenting perception and decision-making capabilities. In the context of this research, a Zed Camera was mounted on a vehicle to amass a dataset, with particular regard to the intricacies inherent in real-time outdoor environments, such as the labeling of the ground, walls, and various objects.

Within the realm of object detection algorithms, Convolutional Neural Networks (CNNs) stand as a noteworthy example [15]. However, CNNs exhibit limitations, particularly in scenarios involving multiple objects within a single frame, where the propensity to overlook certain objects is a documented shortcoming. Herein, the utility of the sliding window technique becomes evident [16]. The focus of this scholarly article is the refinement of the Single Shot MultiBox Detector (SSD) model, with the aim of enhancing overall system accuracy and reliability. Object detection encompasses two principal stages: image classification and image localization. In this study, images are primarily employed for the identification and categorization of objects [17]. Subsequently, the SSD algorithm is utilized to ascertain the distance of an object from the vehicle and to determine its positional relationship thereto.

Figure 2 displays the data flow process, from the acquisition by multiple sensors to the advanced sensor fusion core, also referred to as the Advanced Driver Assistance System (ADAS), situated within the vehicle. A sensor, in the form of a camera affixed to the automobile, transmits raw point data of the observed environment to the sensor fusion ADAS core Electronic Control Unit (ECU) via an Ethernet connection [18]. The data concerning objects, extracted from the raw sensor input, is processed within the sensor fusion core to generate a compilation of monitored object tracklets, which then serve as a dataset for perception and other advanced vehicular functions.

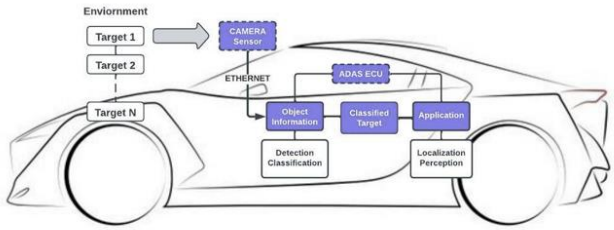


**Figure 2.** Constructed autonomous vehicle general system architecture for object detection

## 2. RELATED WORK

Object identification is one of the most researched topics in computer vision and self-driving vehicles. The process of object detection often begins with the extraction of features from the input picture using several algorithms, including RCNN, SSD, and YOLO. During the training phase, the CNN learns the feature in the object and detect the object. Localization, which includes locating an item inside an image, and classification, which entails giving the object a class (such as "pedestrian," "vehicle," or "obstacle"), are the two sub-tasks that make-up object detection.

Carranza-García et al. [19] contrasted single-stage like Yolo V3 and two-stage detectors like Faster R-CNN. Before deep diving into object detection algorithm let us understand the taxonomy includes in the process which is explained in the next section.

### 2.1 Taxonomy: Object detector

Taxonomy of object detection is classified as type of network and type of data (Figure 3). Data for object detection is of two types; it can be of 2-Dimensional and 3-Dimensional. It can depend on the application, what type of object are being used for algorithm.

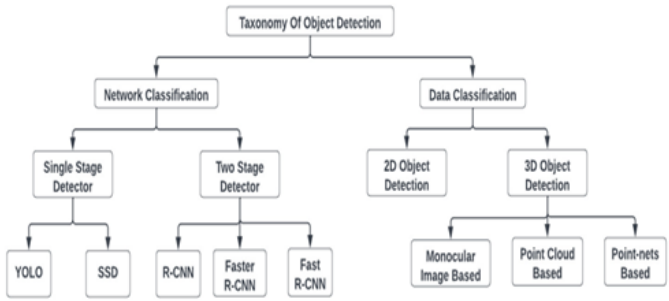


**Figure 3.** Taxonomy of object detection

### 2.2 Comparison of object detectors: Two-stage vs single stage

The two main steps of the 2-stage neural network-based object identification approach for autonomous vehicles are region recommendations and object categorization. In an input image provided by a pair of stereo cameras, the object detector generates a large number of Regions of Interest (ROIs) that are likely to include objects that are important during the region's proposal stage [20]. The second stage involves selecting the most promising ROIs, classifying the items included inside them, and discarding the other ROIs. RCNN, Fast R-CNN, and Faster R-CNN are examples of common two-stage detectors. In contrast, one-stage object detectors classify items in the same stage and construct bounding boxes utilizing just one neural network with feed-forward function. These kinds of detectors are often less accurate even if they are faster than two-stage detectors. One-stage detectors include YOLO, SSD, MobileNet, and RetinaNet, which are all well-known in terms of accuracy and scalability etc.

Pathak [21] discussed a deep learning system for spotting objects which is based on CNN. To increase the complexity while decreasing the number of parameters, CNN employs different types of pooling layers, such as max pooling, average pooling, deformation pooling etc.

**Table 2.** Comparison of Faster R-CNN vs SSD algorithm

| Factors | Faster R-CNN | SSD |
|---|---|---|
| Mean Average Precision (mAP) | It performs worse for real-time processing than SSD. | It has the greatest mAP for real-time processing when compared to Faster R-CNN. |
| IoU Thresholds | 0.3 and 0.7 | 0.5 |
| Memory Usage | It utilizes highest Memory. | Lowest memory utilization. |

In Table 2, factors comparison for SSD, Faster R-CNN is listed in terms of mAP, Accuracy and Memory usage. In this paper we have used SSD algorithm which has better coverage on the location, scale and the aspect ratio which is very crucial for an autonomous vehicle. And also, by removing the delegated region proposal and using lower resolution images, the model can run at real-time speed and still beats the accuracy of the state-of-the-art Faster R-CNN.

### 2.3 Divergence and comparison: 2D vs 3D object detection

Object Detectors come in two functionalities: 2D and 3D. 2D Image data is generally used by 2D object detectors to find the item. Four-degree-of-freedom bounding boxes are provided by 2D object detectors (DOF). The most popular technique for encoding bounding boxes in 2D is [x, y, height, width], whereas for 3D, the method is ["xmin", "ymin", "xmax", "ymax"] [22]. However, the position of an item in 2D can only be revealed via 2D object detection; it cannot reveal the item's depth. To increase performance in various autonomous tasks like path following and collision avoidance, depth of the object is crucial in predicting its size, shape, and location.

Han et al. [23] presented a revolutionary "Wasserstein loss" approach for detecting objects from two layers. Level 1 distinguishes objects from vehicles and people, whereas level 2 distinguishes objects for detailed framework. Researchers further expressly suggest categorizing things for improved performance and reducing the degree of misperception for self-driving with increased Wasserstein loss.

Xu et al. [24] proposed AutoFPN to look for a more efficient and effective detection system design. Their Auto-Fusion may be done naturally on multiple feature structures and even on a single stage detector on a specific dataset. They used their Auto Fusion module on top of SSDs and searched for VOC and COCO. With greater mAP, they received superior performance outcomes.

Li et al. [25] implemented a hybrid framework to create a deep learning model. They targeted object detection with precision and quickness. Detection accuracy on a dataset is used to test many conventional detectors with cutting-edge algorithm performance. They recognized several objects with the Yolo V4 model, which has less parameters and can provide quicker processing speed and greater detection accuracy than the original. They employed the RISE (Randomized Input Sampling for Explanation) technique to explain the categorization results by creating a saliency mAP for each picture.

### 2.4 Various technological approaches in object detection

There are many technologies in object detection for self-driving but mainly three technologies are now in use and in development for autonomous driving functions: image recognition with camera systems, radar detection (RADAR), and light detection & ranging (LiDAR). Ultrasonic sensors (USS) are only utilised for close-range observation due to their narrow operational range, also there are more challenges of using USS. LiDAR is now used in research vehicles, whilst image recognition and radar technologies are used in mass-produced automobiles worldwide. Although it represents the costliest technology, it has immense potential [26].

**Image Recognition:** Image Recognition using camera is one of the most popular ways of detecting the object. Also, Industry 4.0 uses the same technology for the upcoming products and research. The fundamental capabilities of utilizing CAMERA with computer vision, also known as machine vision, are utilised for object & motion detection, distance estimation, and the identification of certain (predefined) features or object properties, such as edges and corners. In this manner, it is used to identify lane markers, road boundaries, and the overall location of other cars or obstructions on the road or close by the study [27]. Enhanced computer vision capabilities enable more precise item detection, but they also call for a different, more involved strategy that incorporates machine learning techniques to train an AI to identify and categorize certain things.

**Zed Camera:** Stereo depth cameras that can perceive 3D depth in high quality are called ZED cameras, created by Stereolabs. This sensor can be incorporated into the suite of sensors used by autonomous cars to detect obstacles. The ZED camera sensor can be used with solutions for obstacle detection like:

**Depth Sensing:** To detect depth in its surroundings, the ZED camera uses stereo vision. It estimates depth by calculating the differences between corresponding locations in images taken with two lenses spaced apart by a baseline. For the purpose of identifying impediments and calculating their distance from the vehicle, this depth information is essential for the algorithm for better outcomes.

**Obstacle Detection:** The ZED camera can detect obstacles in the route of the vehicle by using its real-time depth map creation capability. Road barriers, cyclists, pedestrians, and other items which can obstruct the vehicle's path could all be considered obstacles.

**Deep Learning Integration:** Appropriate algorithms for obstacle tracking and classification can be combined with depth data from the ZED camera. Through model training using data gathered from the ZED camera, the system is able to identify and anticipate the movements of various obstacles.

Angesh et al. [28] talked about the use of numerous cameras for object recognition to enhance results. They state that the tracking system for each camera configuration is solely rated based on the ground truth track seen in that camera arrangement. Tracker appears to perform quite well on all measures, regardless of the number of cameras utilized. A taxonomy of modern deep learning-based object detectors is shown in Figure 3. In this part, there is a classification of these object detectors and the numerous ways used to locate the item.

**RADAR:** It is widely used in autonomous vehicle, but it is very expensive. Radar sensors can detect distances and speed relative to objects with great accuracy. An important advantage of radar-based technology is their consistent and reliable operation in a range of conditions related to the environment, including rain, dust, and pollution. The method uses fewer radar lobes than camera-based systems, which increases costs and leads to a less accurate representation of things [29]. Radar sensor systems and camera-based image recognition are frequently coupled to maximize the benefits of each technology. A good example is the employment of relative velocity measurement and obstacle recognition using radar sensors, which convey information about the kind and class of objects via a combined 2D camera system. This is an illustration of how to recognize pedestrians, identify traffic signs, or recognize traffic lanes.

Wei et al. [30] constructed MmWave RADAR, however the most difficult aspect of using MmWave RADAR is the

scarcity of radar characteristics. MmWave radar provides relatively limited information and cannot significantly increase performance when compared to visual image recognition methods (Camera). However, for improved results and identification, our concept includes Camera technology for object detection.

**LiDAR:** To realize the notion of self-driving cars while maintaining safety, LiDAR must be incorporated into the design process. A LiDAR system creates 3D maps of its surroundings by using laser pulses. For the untrained sight, these lasers are invisible. With its capacity to look in all directions and determine precise distances, LiDAR has many more capabilities than a pair of human eyes. LiDAR allows autonomous vehicles to make precise judgements without human error, which reduces their likelihood of crashes, with security being of the biggest significance [31]. The function of the technology is not impacted by ambient illumination conditions, as opposed to camera-based systems, because LiDAR produces laser light. Cameras, on the other hand, offer a better resolution and can distinguish between colors. LiDAR systems cost many tens of thousands of dollars currently, however new technological developments, such as stationary LiDAR without internal moving parts, can considerably reduce prices. Combining LiDAR with camera-based systems for self-driving automobiles offers the greatest potential for object and environment detection and distinction.

In Table 3, Sensor's technology is listed with their challenges and comparison. According to survey, most used technologies is Camera due to its reliable and accuracy factors.

According to literature review, there are 3 popular approaches for object detection i.e., Faster R-CNN, Yolo, SSD. The comparison criteria are speed, accuracy, and simplicity of implementation. In terms of speed, how quickly can the model produce results. For accuracy, results are correct or not. The ease with which we can apply these concepts and get started.

**Table 3.** Challenges of different sensor technologies

| Sensor Technology | Challenge 1 | Challenge 2 | Challenge 3 |
|---|---|---|---|
| Ultrasonic Sensor | Very low range support i.e., up to 2m | It cannot be used when vehicle is used for high-speed applications. | Resolution is very low |
| RADAR | Range is between 5m to 200m | Results more false alarms due to metal detection | Images captured are of low resolution compared to LiDAR and Camera |
| LiDAR | Limited with maximum range i.e., 200m | False results in Bad weather conditions | Very expensive as compared to RADAR |
| Camera | Range depends upon the lens of the camera. | Combination of excess sensors data integration takes long time for high end applications | Cannot be integrated with other sensors |

**Table 4.** Different algorithm outcomes

| Algorithm | Speed | Accuracy | Ease of Implementation |
|---|---|---|---|
| FR-CNN | Bad | Good | Average |
| YOLO | Good | Good | Good |
| SSD | Good | Best | Good |

Table 4 illustrates the comparison between three models YOLO SSD and Faster RCNN. The first aspect is speed of inference, or how quickly a model can provide results. In this scenario, quicker R-CNN is clearly the loser. It is important to recall that faster R-CNN is derived from the R-CNN family, which consists of two short detectors. The algorithms in these detectors examine the image twice. One for obtaining backbone network properties and the other for estimating recent suggestions. Even though the speedier RCNN approaches the problem differently than previous models, it is still sluggish. SSD, on the other hand, have single shot detectors; these will look at the image once and deliver the results.

Both the YOLO and SSD algorithms were meant to function in real time and on smaller devices such as mobile phones and IoT, hence their performance is relatively high when compared to the quicker R-CNN. Accuracy - In this regard, all three models are comparable, but one thing stands out: each of these models has its unique set of issues. SSD and YOLO are thought to provide faster speeds. However, they have limitations in terms of precision. SSD, for example, is ineffective in detecting very small objects. YOLO also has a trouble identifying items in images when they are close together. In terms of simplicity of implementation. It essentially refers to and includes two things.

1) Framework or package required to use the model.

2) The number of lines of codes that is required to write the smallest program.

With SSD, we may utilize any hardware, but in other models, there is some reliance. In SSD, the number of lines is lower than in other systems. It simplifies system implementation by utilizing all open-source systems operating on a general-purpose CPU, eliminating the need for an embedded engineer to learn how to design specialist hardware. It's completely Linux-based, with an AI inference running on top of it.

### 2.5 Strength and limitations of proposed SSD Algorithm

One common object recognition technique that is well-known for its real-time performance and efficacy in identifying things in images is the SSD (Single Shot Multibox Detector) algorithm. Knowing its advantages and disadvantages is essential when considering using it in autonomous vehicles:

A. Strengths of SSD Algorithm for Autonomous Vehicles:

- *Real-Time Performance:* Real-time object recognition is made possible by SSD's renowned for speedy envision processing. This ability is critical for safe navigation in autonomous vehicles, as quick and precise object identification in the surrounding environment is critical.
- *Multi-Scale Feature Extraction:* SSD uses convolutional neural networks (CNNs) to carry out multi-scale feature extraction, which enables it to identify objects at various dimensions and levels within of a single network architecture. This feature is useful for identifying various-sized items on the road.

- *Single-Shot Detection:* SSD is statistically more efficient than two-stage detectors since it can detect objects in a single forward run of the network. Autonomous driving and other real-time applications benefit from this efficiency.
- *High Accuracy*: When it comes to object detection tasks, SSD can attain a comparatively high accuracy. Autonomous cars benefit from its ability to recognize several things in a picture with good accuracy, since it helps distinguish different objects in the surrounding environment.

B. Limitations of SSD Algorithm for Autonomous Vehicles:

- *Detection Difficulty:* SSD's standard anchoring box dimensions and ratios of aspect may make it difficult for it to precisely identify very small items in images. This restriction may have an impact on autonomous vehicles' ability to identify traffic signs and tiny obstructions.
- *Localization Accuracy for Overlapping items:* SSD may have trouble correctly localizing and differentiating each item when several objects overlap in an image, which could result in mistakes in object borders and categorization.
- *Vulnerability to Object Aspect Ratios:* It may not be possible to reliably detect objects with extreme aspect ratios due to SSD's default anchor box design's lack of optimization for all object aspect ratios.
- *Minimal Contextual Data:* SSD processes images on their own, without taking temporal sequences or more comprehensive contextual information into account. Its inability to comprehend context may impair its capacity to manage intricate scenarios or dynamic events while driving.

## 3. METHODOLOGY

Object detection in Autonomous Vehicle is very important feature to make the vehicle more advanced. Multiple things will need to be recognized in a single image. Multiple item detection in an image and distance estimation are some difficult issues, but with our work applied, it is possible to do so accurately and in real time [32]. We have implemented improved SSD ("Single shot detector") in our Algorithm model to have accurate and reliable results. SSD is a well-liked object detection method that has become known for its accuracy and speed in real time. By utilizing both the camera's precise depth data and the algorithm's object detection abilities, we combined the SSD with stereo depth information from the ZED camera, potentially improving object detection capabilities.

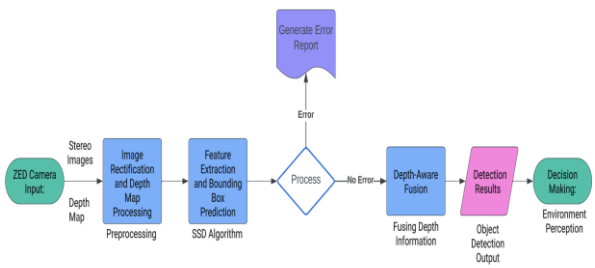


**Figure 4.** Block diagram representing integration of SSD and ZED camera

A high-level block diagram shows in Figure 4 explaining the general steps involved in merging the ZED camera and the SSD algorithm is shown below:

1. ZED Camera Input:

- *Stereo Images:* The ZED camera provides left and right image inputs by employing its two lenses to capture stereo images.
- *Depth Map*: The ZED camera uses stereo vision to determine depth details and produce a corresponding depth map in addition to the stereo envision pair.

2. Preprocessing:

- *Image Rectification:* To make sure that corresponding spots in the left and right images align correctly for stereo vision algorithms, the stereo images from the ZED camera may need to be rectified.
- *Processing of the Depth Map:* To improve its quality and prepare it for fusion with object identification algorithms, the depth map may go through preprocessing operations like normalization or filtering.

3. Improved SSD Algorithm:

- *Feature Extraction:* Using the stereo pictures that were acquired from the ZED camera, the SSD algorithm extracts features in order to identify things. In order to extract pertinent information at various scales, convolutional neural networks (CNNs) are used to process the images.
- *Boundary Box Prediction:* SSD creates bounding boxes around things it detects and makes predictions about the positions and class probabilities of those objects inside the image.

4. Depth-Aware Fusion:

- *Combining Depth Data:* The bounding box data produced by SSD is integrated with the depth map acquired from the ZED camera.
- *Depth-Aware Object Localization:* Depth information can be integrated with SSD to enable depth-aware object localization, which increases the precision of determining the size and distance of objects that are identified.

5. Object Recognition Results:

- *Findings for Detection:* For each object recognized in the scene, the integrated system returns object detection results that include bounding boxes, class labels, and depth-related data.

6. Decision Making:

- *Environment Perception:* By combining data, the autonomous system is able to gain a better awareness of its surroundings and make judgments based on the items it has observed and their connections to one another.

Figure 5 shows the flowchart of methodology used to implement object detection in Autonomous Vehicle. The initial step in object detection is regarding the hardware installation. First the camera is installed inside the vehicle for which stereo cameras has been used.

Stereo cameras calculate detailed information, and the data is examined to see whether it is adequate or need more input. If the data is not sufficient the process will not work further. If it's sufficient then data analysis takes place. There are two targets. The target 1 is object detection by designing the SSD algorithms. We implement these algorithms into our model. Python script has been used for estimating the distance between the objects.

Improved SSD Algorithm: By altering the basis network MobileNet, we were able to enhance feature extraction and object localization while maintaining the original SSD Algorithm. Improvements in precisely localizing objects, lowering false positives, and improving bounding box predictions are also achieved by making adjustments to the

algorithm to speed up inference without sacrificing accuracy—a necessity for autonomous vehicles.

Our model, which comprises of many layers for categorizing supplied objects into one of the stated classes, was developed using convolutional neural networks. With the use of higher resolution feature maps, recent advancements in deep learning techniques for image processing have rendered it possible to identify these objects. "MobileNet SSD object identification" has been implemented in our model, which takes the input picture to compute the output bounding box and object class. This "Single Shot Detector" (SSD) object detection model, which can quickly detect objects, leverages MobileNet as its structural support [33]. In order to determine the final bounding box and object class, an object detection model known as MobileNet SSD utilises the image that is input. Considering the use of MobileNet as a backbone, this "Single Shot Detector" (SSD) object detection technology could provide fast object detection.

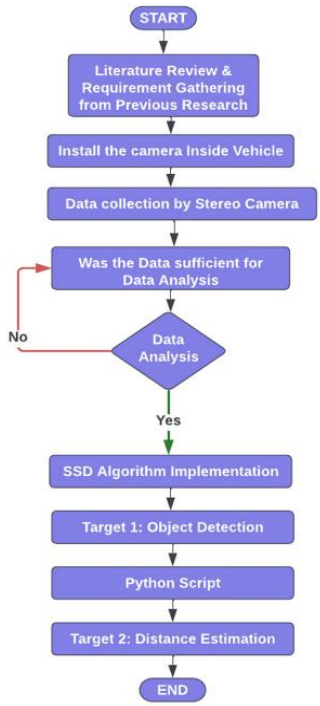


**Figure 5.** Flowchart of implemented methodology

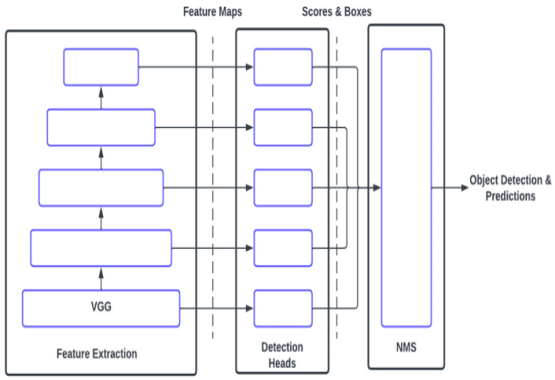


**Figure 6.** Constructed SSD layered architecture

Figure 6 shows the Layered architecture of SSD (Single shot detector). As opposed to RPN-based systems like the R-CNN series, which required two shots, one for developing region proposals and the other for identifying the target of each proposal regional proposal network (RPN) based techniques just need one shot. we have used SSD implementation for our model because it only need one shot to detect multiple objects within the image. Therefore, SSD is significantly quicker as compared to other object detecting strategies. For feature extraction in our model, SSD employs an auxiliary network. This also goes by the name base network [34]. For detection, further convolution layers are added, and the intermediate tensors are retained. It concludes in a stack of feature maps of various sizes. As a result, there are k alternative bounding boxes, each with a probability score, for each location of the items that have been located.

SSD uses stepwise methodology for the implementation, designed steps are:

*Step 1:* Firstly, Image is processed via several convolutional layers, which extract feature maps at various locations across the model.

*Step 2*: Each location in every one of those feature maps makes use of a filter in order to evaluate a small, low default box in the provided image.

*Step 3*: Determine the bounding box offset for each box with boundaries.

*Step 4*: Class possibilities for every box of boundaries should be predicted.

*Step 5*: Employing IOU, the actual boxes are compared to the anticipated boxes.

*Step 6*: The outcome leverages the best-assured loss for each default box rather than all the Negative cases.

Evaluation Metrics:

The performance measures that are used to assess how well the modified SSD algorithm predicts boundary boxes and truth boxes for object classification [35]. For future aspects, data optimization can also be integrated for optimizing the overall vehicle system as suggested in research [36]. Eq. (1) below is used to compute the accuracy, which might be more accurate than the original dataset. The average precision (AP) of all classes divided by the mean is called the mAP. In (2), the mathematical equations are displayed, with N(C) representing the class numbers. Recall (R) and precision (P) both affect AP. The mathematical formulas in (3), (4), and (5) are displayed, with FP and TP denoting the proportion of True Positive and False Positive, respectively. False Negative (FN) is the quantity. By dividing the number of images by the image detection time, the frame rate (FPS) is obtained.

$$Accuracy = \frac{Object\ (O\ Correct)}{Total\ object\ (T\ obj)} \tag{1}$$

$$mAP = \frac{\sum AP}{N(C)} \tag{2}$$

$$AP = \int_0^1 P(R)dR \tag{3}$$

$$R = \frac{TP}{FN + TP} \tag{4}$$

$$P = \frac{TP}{FP + TP} \tag{5}$$

(O correct) in the equation stands for the number of properly identified objects, and (T obj) for the overall number of images dataset.

## 4. RESULTS AND DISCUSSION

We have examined the implementation part in this section and the results that are derived from the analysis. We can divide the implementation part into various categories and is defined below:

*Input Data*:

The main task in deep learning is the construction of the algorithm that can learn from the data or to make predictions on this data. This SSD algorithm is used for data driven predictions. For our implementation, ZED camera has been used in the vehicle to capture the images. The camera is installed at the front of the vehicle so that it can capture the images appearing in the front. For this application we have taken both color and depth images that can be seen in Figures 7 and 8. The advantage of using the depth image is to calculate the distance of the object from the vehicle.

**Figure 7.** Depth image (input)

**Figure 8.** RGB image (input)

In the above images, it can be clearly seen that depth image and color image also called RGB-D Images taken as input for object detection and predicting the distance from vehicle to object. The depth of the image tells us the amount of color information contained in a pixel. It is important as the vehicle doesn't collide into the objects. This information is very critical for an autonomous vehicle so that it can be safe.

*Performance Evaluation:*

This experiment evaluates the revised model's performance using a customized dataset gathered by the real-time outdoor environment and compares it to Fast R-CNN, Faster R-CNN, SSD, and YOLO to verify the validity of the changed approach. Table 5 presents the comparative end outcomes of different networks.

**Table 5.** Comparative end outcomes of different networks

| Methods | Backbone | mAP | FPS | Environment |
|---|---|---|---|---|
| Fast RCNN | VGG | 71 | 9 | Nvidia GPU |
| Faster RCNN | VGG | 72.6 | 11 | Nvidia GPU |
| SSD | MobileNet | 75.2 | 18 | Nvidia GPU |
| YOLO | VGG | 74 | 17 | Nvidia GPU |
| Ours | MobileNet | 78.4 | 20 | Nvidia GPU |

*Labelling Of Images*:

This is the process where the image is labelled with various details of the image. This step is important as we have used supervised machine learning models. In supervised methods, for each image we should have the information of the target feature. The labelling has been completed using the tool to specify the target objects. These labels have been saved in the json files. These files contain the target object name, their coordinates and the image name and shape.

The metadata for one of the images is explained as below:

*{*
*"Object_Name ": "CAR",*
*"lat_Cordinates": "2.9",*
*"long_Cordinates": "3.4",*
*"Image_name ":"CAR_1",*
*"ROI_shape ": "0:720,415:865"*
*}*

*Json data visualizer:*

JSON is a relatively simple data standard that can express extremely complex datasets using layered data structures. In Figure 9, we can view the complicated data of our image json file in a fully structured manner.

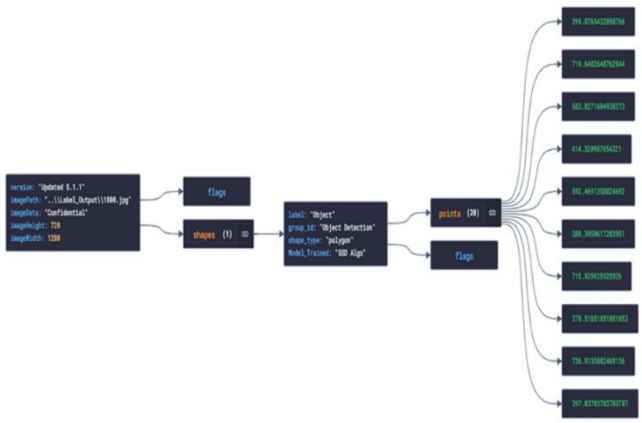

**Figure 9.** Structured data with Json visualizer

*Training and Testing*:

Training in machine learning methods is to learn important and useful information from the data. Training consists of different steps: First is collecting the required dataset. In our case we have taken the images, but this depends on the problem statement that we are trying to solve. Next step is to prepare the data called preprocessing. Brightness correction, Noise reduction, grey scale, Translation, Blur removal etc. are the steps that we have applied on the dataset. To train the model, we have taken the images captured by the zed camera and fed into the pre-trained SSD model. The total dataset has been divided in to 70:30 ratio for training and testing respectively. For distance estimation, depth image dataset has been used. Hyperparameter tuning has been also applied on the model. We tried with multiple epoch values, stochastic

optimization methods called Adam, RMSprop, Nesterov etc.

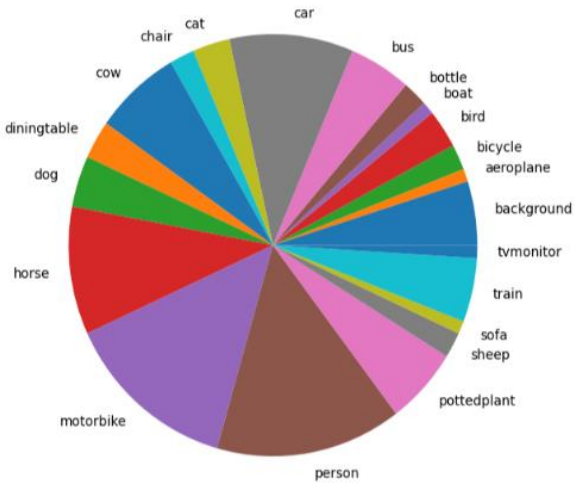


**Figure 10.** Pie chart for trained model classes

During training the model, the learning rate is very crucial and important. This determines the convergence of the training. If it sets too low, the convergence will be slow and will take too much time to train the model. If it sets too high, it might overshoot the optimum value. For this implementation we have taken 0.9 learning rate, we started with 0.001 with batch size of 10. The total dataset is 1500 real images collected in outdoor environment. Data augmentation has also applied on those collected images. mAP (Mean Average Precision) has been used as a loss function. This function calculated the average precision of each class of the dataset. When every class is detected and evaluated, the mean of all the average precision is taken as the result. For the validation of the models, the model has tested on the unknown input data and that is called Test Set. Training loss indicates how well the model can learn from the data and the validation loss is how well the model is able to fit the new data. Is the trained model capable of predicting the test set correctly? For a good fit, validation loss and training loss both should decrease to a point of stability and should have small gap between them. Our proposed model has been trained for 21 classes and Figure 10 illustrates pie graph which depicts the contribution of each data class to the overall picture. A pie chart is a circular statistical visual that is broken into slices to show the numerical fraction of each class for which the model has been trained.

*Output*:

The accuracy of the model is 87% that is successfully able to capture the objects in the image and the distance from the vehicle as shown in Figure 11. But there are scenarios of false positives as well. By training the model on more data, it is possible to enhance it.

*Jupyter Notebook Output Window Result:*

This result shows the distance estimation from the vehicle, it is only possible with our customized SSD Model approach with Zed Camera implementation.

{

Object is 6.436 meters away.

}

**Figure 11.** Output image with detected object

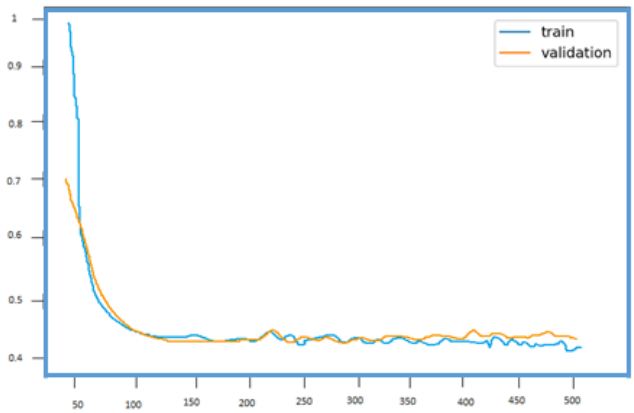


**Figure 12.** Training vs validation graph

Figure 12 shows the final testing and verification in real-time environment by considering all the required parameters. In most cases, model has a processing time of 20 FPS with high accuracy of 87% and a confidence rate close to 100%. This performance might be improved by utilizing much better GPUs.

## 5. CONCLUSION

In this research, we have studied about the Autonomous Vehicle and its system architecture. It has two parts one is hardware which includes various sensors such as Camera, LiDAR, RADAR which perceives the information from this hardware and then fed into the software part of the vehicle. The software architecture is the core of the entire system which has the operating system, algorithms which takes the input data from different sensors and apply logic for the decision-making. This logic's output is then taken by the control modules which regulates the acceleration, motion of the vehicle. Advanced technologies like machine learning computer vision are being applied for this process. There are various algorithms available like Convolutional Neural Networks (CNN), R-CNN, YOLO etc. but our customized SSD model is preferable for real time predictions and considerably has less localization errors, computationally inexpensive and require less storage & processing power for the obstacle detection. The object distance estimate algorithm was created using the mono-depth technique. The overall model has been trained on stereo data and draws inferences on monocular views. Also, we have tested the suggested software model and algorithm in real-time environment with Zed

Camera mounted on the vehicle which gives the outstanding results with accuracy of 87%. We may combine the object detection technique with the estimation distance to share the feature extraction layers, thereby improving its efficiency. The possible benefits of incorporating the SSD algorithm with the ZED camera in self-driving vehicles are demonstrated by applications such as the autonomous golf buggy in the golf course, load automobiles on construction sites, and for other autonomous industries. Such applications allow for improved perception, increased safety, and effective navigation in a variety of dynamic environments. Autonomous vehicles will be far more reliable if their algorithms can adjust to varied lighting situations, diverse surroundings, and different object orientations.

## REFERENCES


[1] Ahangar, M.N., Ahmed, Q.Z., Khan, F.A., Hafeez, M. (2021). A survey of autonomous vehicles: Enabling communication technologies and challenges. Sensors, 21(3): 706. https://doi.org/10.3390/s21030706

[2] Faisal, A., Kamruzzaman, M., Yigitcanlar, T., Currie, G. (2019). Understanding autonomous vehicles: A systematic literature review on capability, impact, planning and policy. Journal of Transport and Land Use, 12(1): 45-72.

[3] Balasubramaniam, A., Pasricha, S. (2022). Object detection in autonomous vehicles: Status and open challenges. arXiv preprint arXiv:2201.07706. https://doi.org/10.48550/arXiv.2201.07706

[4] Gomez, V.V., Cortes, A.S., Noguer, F.M. (2015). Object detection for autonomous driving using deep learning. In Meeting of the Universitat Politecnica de Catalunya, Spain.

[5] Li, Y., Ibanez-Guzman, J. (2020). Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. IEEE Signal Processing Magazine, 37(4): 50-61. https://doi.org/10.1109/MSP.2020.2973615

[6] Fernandes, D., Afonso, T., Girão, P., Gonzalez, D., Silva, A., Névoa, R., Novais, P., Monteiro, J., Melo-Pinto, P. (2021). Real-time 3D object detection and SLAM fusion in a low-cost LiDAR test vehicle setup. Sensors, 21(24): 8381. https://doi.org/10.3390/s21248381

[7] Benciolini, T., Wollherr, D., Leibold, M. (2023). Non-conservative trajectory planning for automated vehicles by estimating intentions of dynamic obstacles. IEEE Transactions on Intelligent Vehicles, 8(3): 2463-2481. https://doi.org/10.1109/TIV.2023.3234163

[8] Zhao, J.F., Liang, B.D., Chen, Q.X. (2018). The key technology toward the self-driving car. International Journal of Intelligent Unmanned Systems, 6(1): 2-20. https://doi.org/10.1108/IJIUS-08-2017-0008

[9] Eggers, F., Eggers, F. (2022). Drivers of autonomous vehicles—analyzing consumer preferences for self-driving car brand extensions. Marketing Letters, 33: 89-112. https://doi.org/10.1007/s11002-021-09571-x

[10] Chen, S.T., Jian, Z.Q., Huang, Y.H., Chen, Y., Zhou, Z.L., Zheng, N.N. (2019). Autonomous driving: cognitive construction and situation understanding. Science China Information Sciences, 62: 81101. https://doi.org/10.1007/s11432-018-9850-9

[11] Peiris, S., Berecki-Gisolf, J., Chen, B., Fildes, B. (2020). Road trauma in regional and remote Australia and New Zealand in preparedness for ADAS technologies and autonomous vehicles. Sustainability, 12(11): 4347. https://doi.org/10.3390/su12114347

[12] Gopalswamy, S., Rathinam, S. (2018). Infrastructure enabled autonomy: A distributed intelligence architecture for autonomous vehicles. In 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, pp. 986-992. https://doi.org/10.1109/IVS.2018.8500436

[13] Bagschik, G., Nolte, M., Ernst, S., Maurer, M. (2018). A system's perspective towards an architecture framework for safe automated vehicles. In 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, pp. 2438-2445. https://doi.org/10.1109/ITSC.2018.8569398

[14] Yeong, D.J., Velasco-Hernandez, G., Barry, J., Walsh, J. (2021). Sensor and sensor fusion technology in autonomous vehicles: A review. Sensors, 21(6): 2140. https://doi.org/10.3390/s21062140

[15] Parekh, D., Poddar, N., Rajpurkar, A., Chahal, M., Kumar, N., Joshi, G.P., Cho, W. (2022). A review on autonomous vehicles: Progress, methods and challenges. Electronics, 11(14): 2162. https://doi.org/10.3390/electronics11142162

[16] Nguyen, N.D., Do, T., Ngo, T.D., Le, D.D. (2020). An evaluation of deep learning methods for small object detection. Journal of Electrical and Computer Engineering, 2020: 1-18. https://doi.org/10.1155/2020/3189691

[17] Chu, W.Q., Cai, D. (2018). Deep feature based contextual model for object detection. Neurocomputing, 275: 1035-1042. https://doi.org/10.1016/j.neucom.2017.09.048

[18] Changalvala, R., Malik, H. (2019). LiDAR data integrity verification for autonomous vehicle. IEEE Access, 7: 138018-138031. https://doi.org/10.1109/ACCESS.2019.2943207

[19] Carranza-García, M., Torres-Mateo, J., Lara-Benítez, P., García-Gutiérrez, J. (2020). On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data. Remote Sensing, 13(1): 89. https://doi.org/10.3390/rs13010089

[20] Zhao, X.M., Sun, P.P., Xu, Z.G., Min, H.G., Yu, H.K. (2020). Fusion of 3D LiDAR and camera data for object detection in autonomous vehicle applications. IEEE Sensors Journal, 20(9): 4901-4913. https://doi.org/10.1109/JSEN.2020.2966034

[21] Pathak, A.R., Pandey, M., Rautaray, S. (2018). Application of deep learning for object detection. Procedia Computer Science, 132: 1706-1717. https://doi.org/10.1016/j.procs.2018.05.144

[22] Kocur, V., Ftáčnik, M. (2020). Detection of 3D bounding boxes of vehicles using perspective transformation for accurate speed measurement. Machine Vision and Applications, 31: 62. https://doi.org/10.100710.1007/s00138-020-01117-x

[23] Han, Y.Z., Liu, X.F., Sheng, Z.F., Ren, Y.T., Han, X., You, J., Liu, R.S., Luo, Z.S. (2020). Wasserstein loss-based deep object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 998-999.

[24] Xu, H., Yao, L.W., Zhang, W., Liang, X.D., Li, Z.G. (2019). Auto-fpn: Automatic network architecture adaptation for object detection beyond classification. In

Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6649-6658.

[25] Li, Y.F., Wang, H.X., Dang, L.M., Nguyen, T.N., Han, D., Lee, A., Jang, I., Moon, H. (2020). A deep learning-based hybrid framework for object detection and recognition in autonomous driving. IEEE Access, 8: 194228-194239. https://doi.org/10.1109/ACCESS.2020.3033289

[26] Patole, S.M., Torlak, M., Wang, D., Ali, M. (2017). Automotive radars: A review of signal processing techniques. IEEE Signal Processing Magazine, 34(2): 22-35. https://doi.org/10.1109/MSP.2016.2628914

[27] Unlu, E., Zenou, E., Riviere, N., Dupouy, P.E. (2019). Deep learning-based strategies for the detection and tracking of drones using several cameras. IPSJ Transactions on Computer Vision and Applications, 11(1): 1-13. https://doi.org/10.1186/s41074-019-0059-x

[28] Angesh, A., Trivedi, M.M. (2019). No blind spots: Full-surround multi-object tracking for autonomous vehicles using cameras and lidars. IEEE Transactions on Intelligent Vehicles, 4(4): 588-599. https://doi.org/10.1109/TIV.2019.2938110

[29] Chaudhary, S., Wuttisittikulkij, L., Saadi, M., Sharma, A., Al Otaibi, S., Nebhen, J., Rodriguez, D.Z., Kumar, S., Sharma, V., Phanomchoeng, G., Chancharoen, R. (2021). Coherent detection-based photonic radar for autonomous vehicles under diverse weather conditions. PLoS ONE, 16(11): e0259438. https://doi.org/10.1371/journal.pone.0259438

[30] Wei, Z., Zhang, F., Chang, S., Liu, Y., Wu, H., Feng, Z. (2022). Mmwave radar and vision fusion for object detection in autonomous driving: A review. Sensors, 22(7): 2542. https://doi.org/10.3390/s22072542

[31] Manoharan, S. (2019). An improved safety algorithm for artificial intelligence enabled processors in self driving cars. Journal of Artificial Intelligence, 1(02): 95-104. https://doi.org/10.36548/jaicn.2019.2.005

[32] Zaarane, A., Slimani, I., Al Okaishi, W., Atouf, I., Hamdoun, A. (2020). Distance measurement system for autonomous vehicles using stereo camera. Array, 5: 100016. https://doi.org/10.1016/j.array.2020.100016

[33] Younis, A., Li, S.X., Jn, S., Hai, Z. (2020). Real-time object detection using pre-trained deep learning models MobileNet-SSD. In Proceedings of 2020 6th International Conference on Computing and Data Engineering, New York, USA, pp. 44-48. https://doi.org/10.1145/3379247.3379264

[34] Deng, J., Xuan, X.J., Wang, W.F., Li, Z., Yao, H.W., Wang, Z.Q. (2020). A review of research on object detection based on deep learning. In Journal of Physics: Conference Series: The 2020 International Seminar on Artificial Intelligence, Networking and Information Technology, Shanghai, China, 1684: 012028. https://doi.org/10.1088/1742-6596/1684/1/012028

[35] Ferdous, S.N., Mostofa, M., Nasrabadi, N.M. (2019). Super resolution-assisted deep aerial vehicle detection. Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, 11006: 432-443.

[36] Jain, A., Nandan, D., Meduri, P. (2023). Data export and optimization technique in connected vehicle. Ingénierie des Systèmes d'Information, 28(2): 517-525. https://doi.org/10.18280/isi.280229