Comparative Analysis of Effective AI-Based 3D Multi-Object Detection and Tracking Methods for Autonomous Driving

Submitted: 22nd November 2023; accepted: 17th September 2024

Dheepika P.S., Umadevi V.

DOI: 10.14313/jamris-2026-014

Abstract:
Object detection is a crucial task for autonomous driving, and different autonomous vehicles have varying perceptions. The advancements of object detection paved the way for 3D object detection, which is considered to be the central component of perception systems that predict obstacles, vehicles, pedestrians and other key features of the environmental backgorund. Generally, various sensors and cameras are used in autonomous driving producing accurate pereddiction of objects. Several algorithms have been employed in object detection, but they have not produced effective outcomes.

Thus, the present study implements HDL-MODT (hybrid deep learning based multi-object detection and tracking) using a sensor fusion approach. It uses solid-state LiDAR, pseudo-LiDAR and an RGB camera to capture objects and provide effective tracking abilities. Initially, the pre-processing methods involved noise removal using an A-Fuzzy (adaptive fuzzy) filter. Contrast enhancement is then performed using the MSO (moth swarm optimization) algorithm, and feature segmentation is done by LGAN (lightweight general adversial networks), where both channel and position attention mechanisms provide precise segmentation. The YOLOv4 approach is deployed for detection of objects such as ground, vehicles, pedstrians and obstacles. Finally, the tracking of objects is performed using IUKF (improved unscented Kalman filter). The simulation of the proposed method is demonstrated by using MATLAB R2020 simulation tool; the performance of the proposed method is also predicted by comparing the results with exisitng algorithms.

Keywords: *Autonomous Driving, Object Detection, Tracking, One Stage Object Detector, Two Stage Object Detector*

1. Introduction

With the development of autonomous driving, accurate object detection is a huge challenge in decision-making controls to ensure safe driving.

Operating a vehicle with little or no effort from human involvement is complex, and the number of accidents has rapidly increased in recent years. This can be mitigated using object detection techniques, which tend to perceive surrounding passengers, traffic lights, vehicles and even unknown objects. Object detection is considered one of the most important tasks for avoiding traffic accidents and a variety of other human errors [1]. Automatic driving techniques can improve their safety by accounting for human manipulation of the vehicle and prediction of road conditions. Obstacles in autonomous driving might include motorcycles, bicycles, trucks, cars, pedestrians or other objects within visual range. Several object detection approaches have been utilized in this field of image processing, but they have not produced efficient feature representation methods. Moreover, object detection algorithms have mostly relied on the development of manual features and traditional approaches involving DPM (deformable parts model), HOG (histogram of oriented gradients) and others [2]. Consequently, 3D object detection is considered to be the dominant component of perception systems in the case of autonomous driving that projects points onto a single prescribed assessment of feature learning. Generally, 3D object detection techniques suffer from inaccurate depth estimation, which causes reduced accuracy. In recent years, 3D object detection based on stereo matching [3], LiDAR [4] and radar perception [5, 6] have achieved more impressive performance in detection accuracy. To mitigate the limitations of 3D object detection, an effective methodology for autonomous driving should thus be developed.

Concurrently, rapid advancements in AI (artificial intelligence) computer vision have led to the development of efficient outcomes. Hazarika et al. [7] implemented a multi-camera solution in which 3D bounding boxes and weighted-box selection methods are used for object detection. A DCNN (deep convolutional neural network) is employed to map 2D bounding boxes to their 3D counterparts in order to identify an object's dimension and orientation. Additionally, a ViT (Vision Transformer) is addressed for accurately detecting depth and occlusion using a self-attentive module.

The comprehensive simulation is performed after fusing outcomes from multiple cameras. KITTI standard data is analyzed using existing similar camera-based techniques.

Similarly, MTL (multi-task learning) plays a significant role in the growing field of autonomous vehicles. Orchestrating multiple tasks with sensor data has proven to be a complex task, and MTL meets these challenges by training a single model to do multiple tasks simultaneously. The study in [8] involved a scalable MTL for object detection, which is used to develop an MTL network with varied shapes and scales.

It also deploys the extended version of Mask-RCNN (Region-CNN) to overcome the limitation of learning several objects in multi-label learning. Performance was evaluated using the Berkeley Deep Drive 100KBDD100k dataset, and the outcomes produced by the study achieved a mAP (mean average precision) of 50 .

However, single-stage approaches exemplified object detection with better accuracy and possessed the ability to predict bounding box coordinates and object classes from single evaluation of the network. The hybrid approach in [9] approach incorporated both Faster R-CNN and YOLO, combining the boundary-box assortment ability of YOLO with the RoI (region of interest) pooling of Faster R-CNN. Analysis found that the study produced an improved accuracy of 74.3%, with a processing time of 52 ms . Despite several benefits, image resolution and object detection of varying sizes under challenging driving conditions is an important task for single-stage detection algorithms.

To address the issues presented by single-stage approaches and low-resolution images, the proposed method involves 3D multi-object detection and tracking for autonomous driving. This approach is intended to leverage the strengths of the single-stage paradigm. The YOLOv4 approach is used for faster and more accurate 3D multi-object detection. The proposed method incorporates YOLOv4's effective object detection for bounding box assortment for classification.

Initially, the images from the 3D LiDAR and stereo RGB cameras are used in pre-processing, a method that includes the removal of noise using an A-Fuzzy filter, followed by contrast enhancement on noiseremoved images using MSO. Voxelization is used enhance the perceptiveness of solid LiDAR points. Both contrast-enhanced images and voxelized point clouds are integrated to generate highquality images. The segmentation process combines LGAN with preprocessed fused images in order to reduce the complexity of classification and tracking. The channel and position attention mechanism are applied for, ensuring improved accuracy in segmentation. In order to reduce complexity, VGG-16 is deployed for better feature extraction, in which feature vectors are formed. These feature vectors are selected for object detection using YOLOv4, which creates four classes: ground, vehicles, pedestrians and obstacles. After classification, the detected objects are tracked by using IUKF. Time-based mapping is performed for vehicles by considering the RFID, velocity and location of the detected objects. The MATLAB R2020a simulation tool is used to simulate the proposed methodology. Further assessment of the proposed algorithm is performed using existing approaches with different performance metrics to examine the efficacy of the proposed system.

The main contributions of the paper are as follows:

- To apply pre-processing to input images, using A-fuzzy for noise removal and MSO algorithm for contrast enhancement;
- To process segmentation using LGAN and feature extraction using VGG-16 algorithm, extracting feature vectors while avoiding system complexity;
- To implement object detection using YOLOv4 to classify objects such as ground, vehicles, pedestrians, and obstacles; and
- To assess the efficiency of the proposed model by comparing its outcomes with those from existing methods.

1.1. Paper Organization

The paper is organized based on the most efficient methods deployed in the object detection of vehicles. It analyzes conventional methods used in similar applications with the varying approaches discussed in Section 2. The elaborated procedure of the proposed method is shown in Section 3. The results obtained by the implemented approach are then deliberated in Section 4. Finally, Section 5 presents the conclusion, with suggestions for future work using the proposed method.

2. Related Works

Vision-based driving assistance systems mainly rely on the concept of object detection, which has become increasingly attractive in smart transportation systems. However, it is complex to produce energy-efficient and cost-saving autonomous vehicle systems. So, [10] implemented edge-cloud assistance based object detection, along with a reconstructive CNN known as edge YOLO. This study avoided the extreme reliance on computational power and irregular spread of CC (cloud computing) assets, projecting a lightweight object-detection system that combined a compression feature fusion network with a pruning feature extraction network to improve the efficiency of multi-scale identification. An automatic driving platform with NVIDIA Jetson was also involved for system-level evaluation. Our analysis found that the study produced mAP (accuracy) of 47.3% at a speed of 26.6 FPS (frames per second). Though the study generated satisfactory results, the system reduces downprocessing at the edge, and the transmission pressure produced by CC increases latency. . Edge clouds significantly reduce computational burden and latency; however, their performance is affected by dynamic channel conditions.

To address the issue of edge clouds, [11] deployed an EODF (edge network-assisted real-time object detection framework). When using the EODF algorithm, an autonomous vehicle must capture the image and extract the RoI when the channel quality is not good enough for real-time object detection. For this reason, the extracted images transfer the compressed RoI to edge clouds, thus reducing the transmission latency of the system. The results in [11] showed an accuracy value of 82% for all categories such as tram, truck, van and car, and a mAP of 84%. However, increases in compression ratio caused decreases in information loss and degradation in object detection performance.

The autonomous driving system uses the surrounding environment to make driving decisions by modelling scenery data gained from the sensors. The FPN (feature pyramid network) generates high-level semantic feature pyramids and thus identifies objects of varying scales. Hence, [12] applied enhanced FPN (EFPN) to develop an adaptive parallel detection subnet and an improved feature extraction subnet. The adaptive parallel detection subnet employed PDB (parallel detection branch) and ACE (adaptive context expansion). Meanwhile, to enhance the pyramid features, the enhanced feature extraction subnet employed an FWM (feature weight module). The study evaluated the system on cityscapes' datasets along with the KITTI dataset, and produced improved object detection. However, the average precision of cityscapes has been found to be lacking due to an increase in small and occluded objects. Small objects at the pixel level have even vanished after different down-samplings. The complexity of traffic environments, lighting situations and variances in image quality have also decreased precision.

Similarly, the automatic driving can be enabled by identifying missed and false detections of small and occluded objects. Hence, Zhou et al. [13] employed an Automatic Drive-Faster-RCNN algorithm for issues related to small scale object detection, denseness and occlusion. The Resnet-50 and partial attention mechanism were used to improve the feature extraction of small objects, and the feature pyramid structure has also been employed to decrease feature loss in the fusion process. Moreover, three cascade detectors - namely, sideaware boundary localization, threshold mismatch and IOU (intersection-over-union) - have been adapted to perform frame regression. The study produced better outcomes, but with shortcomings; the positioning information is not transmitted, and the sematic informatics contained were diluted during information fusion.

The low accuracy and interference speed in recognition of objects is considered as the main hindrance in the development of automated vehicle systems. For this reason, [14] deployed MCS-YOLO (multi-scale small object detection), along with a coordinate attention mechanism, to detect multi-scale dense small objects. The attention mechanism was used to aggregate the cross-channel information and feature map's spatial coordinates. Additionally, a Swim Transformer structure was utilized to increase the focus of the network on contextual spatial information. The study produced better outcomes, but was not able to solve issues related to MOT (multiple object tracking). The end-to-end delays in object detection in real-time circumstances also present other safety concerns in autonomous driving.

In [15], three optimization approaches were implemented: on-demand capture, zero-slack pipeline and contention-free pipeline. The Darknet YOLO was used to optimize object detection dependent on OpenCv library for capturing realtime images and achieve a queue for buffering image frames.

The study was executed and evaluated in Nvidia AGX Xavier CPU-GPU heterogeneous computing platform. Analysis indicates that the study has generated better results with lower detection quality, but the system did not focus on a sensor fusion system for complex application scenarios. Multi-scale object detection for autonomous driving has been employed using a YOLOX-based network model under complex scenarios. A CBAM-G (channel-based attention module-grouping) approach has been integrated to alter the height and width of the convolutional kernel of the spatial attention module [16]. An object context-based feature fusion module has been utilized to produce additional semantic data and increase the observation of multi-scale objects. These experiments, conducted on BDD100k and KITTI datasets, produced better mAP values.

However, the study did not concentrate on the lightweight multi-scale object detection that must be applied under practical applications.

In autonomous driving applications, Lidar-point cloud-based 3D object detection plays a significant part, and has proven challenging due to uneven distribution of data points. Hence, [17] proposed a transformed approach known as TCT (temporalchannel transformer) to develop the spatialtemporal and channel-domain relationships. The information encoded in the encoder was varied with the decoder, and the spatial decoder of the transformer decoded the information for each location of the present frame. In contrast, the temporal-channel encoder of the transformer has been specifically modelled to encode the data of the frames, as well as several channels. This study has produced better outcomes.

To increase the detection accuracy and the robustness of the perception system in autonomous driving systems, [18] imposed DMIQADNN (dual-modal image quality aware deep neural network). An analysis of the early, middle, late and score stages of fusion architectures was performed to predict detection accuracy and speed. An IQAN (image quality assessment network) was also used in the analysis of the RGB image quality score. The fusion weights for the LiDAR and RGB sub-network were allocated by applying a fusion weight assignment function. Further, the study found scores of 27 on a modified KITTI benchmark and 39.1 on an AP (average precision) benchmark. LiDAR-camera-based 3D object detectors were used to extract the specific features and adjacent 3D data known as point clouds. The camera captured high-resolution RGB images and combined the features of the RGB images and point cloud.

An early fusion module was employed to exploit camera and LiDAR for faster 3D object detection [19]. A feature fusion model has been utilized to extract point-wise features from raw RGB images and then fuse them to the equivalent point clouds. Initially, the system voxelizes a point cloud into 3D voxel grid, and then uses two methods to decrease of information loss while performing voxelization.

The results have been applied in a KITTI benchmark dataset to evaluate their speed and accuracy.

2.1. Problem Identification

The main concerns identified through the analysis of existing algorithms are conferred in this section.

- A hybrid approach incorporating Faster RCNN and YOLO [9]. This study could be improved by utilizing larger and more diverse datasets to predict the efficacy of the approach in various self-directed vehicle platforms. Enhancing the execution and discovering interactions with other AI-based techniques can help optimize the efficiency and safety of self-driving vehicles.
- Using EODF for object detection, with compressed images transmitted to edge clouds [11]. However, the compression ratio led to information loss and reduction in objectdetection performance. Thus, the compression ratio should be set to the maximum value to evaluate the average precision of the system.

3. System Framework

Increased potential of providing road safety, effective decision making and decreased traffic congestion can be achieved by using autonomous driving systems. They must observe, recognize, design, adopt and perform decisions within an uncontrolled, complicated real world. This is a more challenging goal, as a small fault in understanding the surrounding background and decision may lead to mortal effects. A reliable and effective autonomous system is thus required to eliminate errors and make correct decisions according to changing situations. The recognition system - namely, the 3D object detection method - supports automatic interpretation the vehicle's driving. The 3D sensor, known as LiDAR (light detection and ranging), works with RGB-D cameras to produce 3D information about the environment, such as speed and distance estimations. The LiDAR sensor is efficient under a variety of weather conditions, but its major drawback is that it tends to struggle in detecting close- and far-distance objects. Different approaches have been developed to implement the LiDAR point cloud data, but they provide less texture and color information. To overcome the limitations of existing methods, this study's method involves the enhancement of input images to provide effective outcomes in 3D object detection.

3.1. Data Collection

In the concept of 3D object detection, the datasets are categorized as "indoors" and "outdoors" based on their applications. Different studies that used 3D object detection for intelligent driving systems mostly rely on outdoor datasets. The most wellknown datasets use LiDAR and RGB cameras to predict various types of data. Moreover, these datasets contain a mass of 3D annotation bounding boxes, multiple object classifications, and selfdriving scenes. Table 1 shows a summary of 3D object detection datasets for autonomous driving systems.

The study uses the KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute) 3D object detection benchmark dataset for training and effective autonomous driving system. It is considered as a largest computer vision algorithm evaluation dataset for automatic driving scenarios. The dataset comprises 52,979 labelled objects, of which 7,481 are training and 7,518 testing images. All the images present in the dataset are colored and saved in .png format. One image in the KITTI dataset might consist of up to 30 pedestrians and 15 cars. The 3D object detection system has labels for nine categories: car, pedestrian, cyclist, van, truck, sitting person, tram, miscellaneous, and "don't care" objects.

3.2. Data Pre-Processing

Data pre-processing is the method of changing the raw data into a clean dataset. This important process prepares the data in the most meaningful and understandable way for the model to able to easily analyze. Data are cleaned, transformed and integrated in order to make them ready for analysis. The input dataset is also pre-processed to verify missing values, noisy data and other inconsistencies before executing the algorithm. Noisy data are the result of data entry errors and faulty data collection that are represented as meaningless data. Generally, noisy data can be handled by using methods such as clustering, regression and binning. Clustering is a technique in which the similar data points are grouped together to form "clusters." The main objective of clustering is to predict patterns present in the data and group based on similar data points, and to separate dissimilar data points into different groups. Regression, on the other hand, is the process of making data smooth in order to fit it to a regression function. The regression used can be either linear, with one independent variable, or multiple, with several independent variables. Binning is a technique that works on sorted data in which the whole dataset is categorized into segments of equal size. Each segment is handled separately and can be replaced by its boundary or mean values to complete a specific task.

Noise Removal

This study uses RGB-D images and LiDAR cloud points for object detection. These images tend to generate more noise, which reduces the quality of the image. Here, pre-processing involves noise removal from input data using the A-Fuzzy (Adaptive Fuzzy filter) performed under two stages. The A-Fuzzy rules and member functions are used to determine whether the pixel under consideration is noisy. To differentiate the noisy pixels, the difference between the gradients are calculated, and it is verified whether they are small or large. The a_1, a_2, a_3, a_4 are the four adaptive fuzzy rules and are defined as:

$$a_1 = \text{Small}\,(F_1, \mu_1, \mu_2)\,.\,\text{Small}\,(F_2, \mu_1, \mu_2)$$

$$a_2 = \text{Small}\,(F_1, \mu_1, \mu_2)\,.\,\text{Large}\,(F_2, \mu_1, \mu_2)$$

$$a_3 = \text{Small}\,(F_1, \mu_1, \mu_2)\,.\,\text{Small}\,(F_2, \mu_1, \mu_2)$$

$$a_4 = \text{Small}\,(F_1, \mu_1, \mu_2)\,.\,\text{Large}\,(F_2, \mu_1, \mu_2)$$

Table 1. Summary of 3D Object Detection Datasets

Types of Datasets	Sensors	3D Boxes	Classes	Number of Scenes	Annotated Frames
Waymo	RGB+Li DAR	12M	4	1 k	200k
ApolloSc ape	RGB+Li DAR	70K	35	Nil	140K
nuScene s	RGB+Li DAR	1.4 M	23	1 k	40 k
A*3D	RGB+Li DAR	230 K	7	Nil	39k
H3D	RGB+Li DAR	1.1 M	8	160	27k
Lyft Level 5	RGB+Li DAR	1.3 M	9	366	46k
KITTI	RGB+Li DAR	200 K	8	22	15k

These terms are represented as adaptive fuzzy sets. The adaptive fuzzy membership functions "Small" and "Large" are given by the equations (1) and (2):

$$\text{Small}\,(\mathrm{F}_1,\mu_1,\mu_2)=\begin{cases}1, & \mathrm{F}_1<\mu_1\\ \left(\frac{\mathrm{F}_1-\mu_2}{\mu_1-\mu_2}\right), & \mu_1\le \mathrm{F1}<\mu_2\\ 0 & \mathrm{F}_1\ge\mu_2\end{cases} \tag{1}$$

$$\text{Large}\,(\mathrm{F}_1,\mu_1,\mu_2)=\begin{cases}0, & \mathrm{F}_1<\mu_1\\ \left(\frac{\mathrm{F}_1-\mu_1}{\mu_2-\mu_1}\right), & \mu_1\le \mathrm{F1}<\mu_2\\ 1 & \mathrm{F}_1\ge\mu_2\end{cases} \tag{2}$$

Where μ_1 and μ_2 are represented as the threshold parameters. After applying the adaptive fuzzy rules, the adaptive membership degrees is denoted as in equation (3),

$$\omega_{\text{degree}}=\text{Maximum}\,(\mathrm{a}_1,\mathrm{a}_2,\mathrm{a}_3,\mathrm{a}_4) \tag{3}$$

The membership degree plays a significant role in the adaptive filtering phase, in which pixels with membership degree (ω_{degree}) $= a_1$ are noisy pixels. When $\omega_{\text{degree}} = a_2$ or a_3, it is treated as a noisy pixel that is filtered using the A-fuzzy technique. Pixels with $\omega_{\text{degree}} = a_4$ signify noise-free pixels.

Contrast Enhancement

Meta-heuristic-based algorithms used for contrast enhancement yield different pixel intensity redistribution patterns compared to traditional histogram equalization (HE). Some image contrast enhancement techniques include FFA (firefly algorithm), CS (cuckoo search algorithm), GA (genetic algorithm), ABC (artificial bee colony) and others. Though these algorithms have produced better results, they also have certain flaws, such as an inability to maintain population diversity and premature convergence to local optima. These effects cannot be improved unless noise, irrelevant visual information and small sets of pixels are removed.

The proposed work thus employs an MSO (moth swarm optimization) algorithm to overcome these impacts and to generate more effective image quality. The MSO is a meta-heuristic algorithm inspired by the alignment of moths towards the moonlight that analyses three explicit subpopulations. Depending on the sub-population, the individual performs by implementing varying evolutionary operations.

This process is inspired by the real-life behavior of moths, and the integration of operators in the search method tends to mitigate critical issues like premature convergence and improper explorationexploitation balance. The moth swarm algorithm in the study is used to predict the best redistribution to assemble the improved image. Here, the search features of the algorithm allow for discovery of the solution space. In MSO, the swarm population is segregated into three parts: pathfinders, prospectors and onlookers. The steps involved are as follows:

Step 1: Initialization

In the initialization process, the random positions are assumed by search agents based on the model given in equation(3.2),

$$mso_{pq}=random(0,1)\cdot\left(mos_q^{maximum}-mos_q^{minimum}\right)+mos_q^{minimum},\ \mathrm{p}\in(1,2,3,\ldots\mathrm{i})\ \text{and}\ \mathrm{q}\in(1,2,3,\ldots\mathrm{j})$$

Where $mos_q^{minimum}$ and $mos_q^{maximum}$ signify the lower and upper limits of the search space.

Step 2: Prediction Phase

Two operators - crossover and Levy perturbation - are employed for each individual solution. The formula for the crossover point is given by the equation (5),

$$\omega_q^p=\frac{\sqrt{\frac{1}{i_j}\Sigma_{k=1}^{i_j}\left(mos_{kq}^p-mos_q^p\right)^2}}{mos_q^p};\quad mos_q^p=\frac{1}{i_j}\sum_{k=1}^{i_j}mos_{kq}^p \tag{5}$$

Where i_j is represented as the total number of pathfinder moths. Thus, the new solution is shown as in equation (6):

$$\mu^p=\frac{1}{j}\sum_{q=1}^{j}\omega_q^p \tag{6}$$

The MSO algorithm implies Levy perturbations in order to generate random steps. Lev_p is the random sample generated. and is calculated as in equation (7),

$$Lev_p\approx scale*L(\theta)\approx 0.01\frac{u}{|a|^{\frac{1}{\theta}}} \tag{7}$$

Where scale denotes the dispersion size, $\theta \epsilon [0, 2]$, entry-wise multiplication is performed with *, and the two normal distributions are $u = M(0, \omega_u)$ and $a = M(0, \omega_a)$.

Further, in adaptive crossover, each pathfinder updates its position by integrating the mutated variables and crossover operators. After completing adaptive crossover, the fitness value for entire trail solution is evaluated and is equated with its corresponding host solution. Additionally, a set of solutions are chosen based on the roulette approach.

Step 3: Transversal Flight

For the next iteration level, a cluster of elements comprising the best luminescence concentrations are represented as prospectors. The number of prospectors is reduced during the optimization process.

Step 4: Celestial Navigation

With the reduction in the number of prospectors, the onlookers number increases, which causes a large increase in the convergence rate. Onlookers are considered to be the moths producing less luminescent causes in the swarm. In the celestial navigation step, the onlookers search, following the prospectors. Then the onlookers are further separated into two phases, namely Gaussian walks and ALIM (associative learning mechanisms with immediate memory). Then, the fitness of the onlookers is evaluated and global best is updated until the criteria are reached. Therefore, in contrast improvement of the input image, the MSO algorithm is deployed to adjust the pixel concentrations, and thus the quality of the image is enhanced.

Voxelization is performed to enhance the perceptiveness of the image. Owing to the increased variable density of LiDAR point clouds, it is a complex task to predict the information loss and perceptiveness, and achieve balance a between these and the speed of point cloud processing. Voxelization is the technique in which the grouping of points into voxels is performed based on their corresponding spatial proximity. The depth of information in the resulting voxel grid is determined depending on the voxel size. With the contrast-improved image, voxel point clouds are thus integrated to generate high-quality images. After the pre-processed fused images are obtained, the instance segmentation using is performed an LGAN approach to reduce complexity of the tracking and classification.

3.3. Segmentation

In the case of LGAN-based instance segmentation, pre-processed RGB-D images are combined with the voxelized LiDAR point clouds. The angles of the images are varied from each other, reducing detection accuracy; thus, they are changed to 10°, 90°, 180° and 270° to increase their precision. After altering the angles of the images, instance segmentation is performed for fused images using the LGAN method. This segmentation is a method of interpreting visual data associated with an entity while also considering spatial information.

In the proposed study, LGAN is used to perform instance segmentation, and it comprises two models - a generator and a discriminator. Here, the generator captures the distribution of input data and generates fake samples of data. The study implements three losses: adversarial loss, L_1 loss and Jaccard loss. The adversarial loss slows down the learning process; L_1 loss stores the object boundaries; and Jaccard loss enhances the correlation between the original and segmented images. The generator is trained while the discriminator is stable, and the parts that trains the generator are as follows:

- Input with noisy vector
- Generator network that transforms the random input into data instances
- Discriminator network that classifies produced data
- Generator loss

On the other hand, the discriminator is an NN (neural network), which is used to predict real data using the fake data produced by the generator. The discriminator consists of four layers: a convolutional, position attention, channel attention, and an activation layer. The training data for the discriminator is done using:

- Real data instances utilized by the discriminator as positive samples during training
- Fake data instances produced by the generator, signified as negative samples

The attention layers in the encoder and decoder are used to learn both low- and high-level features. In this way, the discriminator categorizes real and fake data produced from the generator by effectively using the position and channel attention layers. Thus, both generator and discriminator operate simultaneously to learn and train complex data.

3.4. Feature Extraction

The segmented image produced from the LGAN is processed for feature extraction to increase detection accuracy. Here, feature extraction is performed using the VGG-16 approach. The VGG16 architecture is composed of 41 layers, of which 13 are convolutional layers, 16 are weighted layers, and three are fully-connected layers. Input at 224 × 224 is provided with RGB channels. To improve outcomes, the size of input image is reduced for each pixel. The algorithm shows the steps involved in extracting features using VGG-16.

Using VGG-16, high-level features are extracted, allowing for more accurate classification of the images.

3.5. Object Classification and Tracking

After the feature extraction phase, object classification is performed by implementing YOLOv4, a one-stage object detection network. YOLOv4 uses anchor boxes to detect classes of objects in the input image and identifies three attributes: (Intersection over Union), anchor box offsets, and class probability.

The YOLOv4 is composed of three parts backbone, neck and head - as shown in Figure 1.

Algorithm I: Feature Extraction using VGG-16

Input Training images (T), Corresponding labels (L) pre-trained VGG 16 and
Output Features are extracted from input images
Arrange the VGG-16 to perform extraction of features from input image by eliminating the fullyconnected layers, module-VGG-16 = VGG-16-FC
For j = 1 to T :
Read image j
The image j is resized to 224×224×3 Features extracted: (j)=moduleVGG 16(j)
Flatten (j)
Transform the features extracted(j) from 3D feature stack to 1D array Flatten (j)=Features(j)
End for

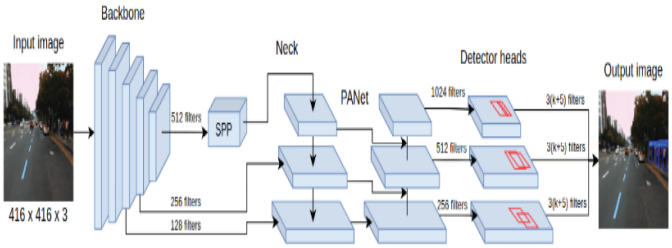

Figure 1. Architecture of YOLOv4 [20]

- The backbone is pre-trained using CSPDarkNet53 (Cross Scale Partial DarkNetwork-53) and acts as a feature extraction network, calculating feature maps from input images. The backbone consists of five residual block modules, and the feature map outcomes from residual blocks are combined together in the neck part.
- The neck part interlinks the backbone with the head of the network. It consists of PAN (Path Aggregation Network) and SPP (Spatial Pyramid Pooling). It concatenates the feature maps from several layers of backbone and transfers them as an input to the head network.
- The head operates the accumulated features, and thus identifies the bounding boxes and objectness scores, along with classification scores.

The loss of YOLOv4 such as object classification, localization and offset loss, are evaluated, and loss function is predicted. Hence, the use of YOLOv4 implies that the learning ability of the network is increased and thus the classification accuracy has been optimized. In this way, the tracking of moving objects is performed after classification based on RFID, unique ID, dimension and orientation. The IUKF (improved unscented Kalman filter) is used to track the position and velocity of the target and objects.

It is a recursive algorithm used to estimate the evolving state of process when measurements are done. Time-based mapping is employed by considering the preceding and present time and location from RFID to improve tracking reliability.

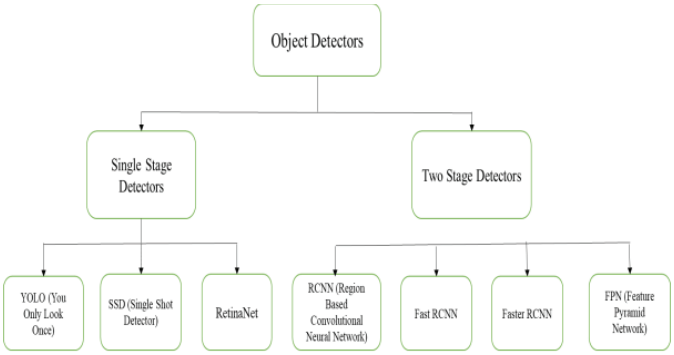

Figure 2. Architecture of Object Detectors

4. Comparative Analysis of AI-Based Techniques Involved in Object Detection

Visual object recognition is applicable to autonomous vehicles, which are able to sense and navigate the surrounding environment without human involvement. Prediction of obstacles for safe riding is a significant challenge in automatic driving systems. The only way to avoid road accidents is to consciously recognize obstacles and traffic lights.

Thus, DL (deep learning)-based object recognition techniques are used to detect objects with better accuracy. They are classified into two major classes: single-stage architecture and two-stage architecture. Two-stage object detectors usually achieve better detection performance, while the single-stage detectors are more efficient and are suitable for detecting objects with limited resources. Figure 2 shows the architecture of single-stage and two-stage object detectors.

4.1. Object Detection Using Single-Stage Architecture (Three Algorithms)

The single-stage object detectors require only a single pass through the neural network (NN) and predict entire bounding boxes in one attempt. This makes their approach to work faster and increases their speed of detection. LiDAR-based 3D object detection plays a crucial role in autonomous driving; however, its performance degrades under highly sparse point cloud conditions. In [21], a 3D object detector was implemented based on voxels, using PV-SSD (projection double branch feature extraction) to reduce information loss. Voxel features were fed as input containing local semantic features. In feature extraction phase, these semantic features were combined with projected features to decrease the local information loss from point cloud projection. A feature point-sampling algorithm with weight sampling was deployed to find the feature points that were most beneficial for the detection process. The MSSFA (multi-layer spatial semantic feature aggregation) method was applied to provide better detection accuracy.

The outcomes generated by the study were evaluated in KITTI dataset and were found to produce better results. When compared with the proposed HDL-MODT approach, this method is advantageous in terms of detection accuracy and tracking reliability. The study did not, however, involve the tracking of moving target objects, which would show the improvements in our proposed model.

Another method of object detection using YOLOv4 was applied using the backbone network CSPDarknet53_dcn (P) [22]. Feature fusion was performed using PAN++, along with five scale detection layers to enhance detection accuracy for small objects. The last layer of CSPDarknet53 was replaced with deformable convolution, and a pruning approach was introduced in the study to resolve issues created during the real-time performance of the system. A sparse scaling factor is implied in order to overcome these issues, as the parameter significance evaluation technique cannot differentiate the importance of the convolution kernel in a large-redundancy convolutional layer. The BN (batch normalization) layer avoids the ”internal covariate shift” issue, in which the activation input value is always maintained with a specific distance from the derivative saturation area and in sensitive areas. This resolves the vanishing gradient problem occurring in the backpropagation.

4.2. Object Detection Using Two-Stage Architecture (Three Algorithms)

Distant object prediction in autonomous driving can be addressed by using Faster RCNN algorithm [23]. With the development of DCNN, vision-based vehicle detection has achieved significant enhancements compared to traditional methods. However, heavy occlusion and large-scale vehicle variation has led to limitations in DCNN’s performance. The emergence of Faster RCNN thus has allowed for faster vehicle detection with an improved framework. Initially, the MobileNet structure was developed to construct the first level of the convolutional network in Faster RCNN. The NMS (non-maximum suppression) approach is then applied subsequently to the region proposal network, and the Faster RCNN is substituted using Soft-NMS in order to resolve fake proposals. Then, the context-aware Rol pooling layer is considered to regulate the proposals into its specific size without eliminating the essential contextual data. To construct the final phase of Faster RCNN, the assembly of depth-wise separable convolution in the MobileNet design is employed to develop the classifier. Then, proposals are classified and the bounding box is adjusted for each of the detected vehicles. The experimental outcomes produced in the LSVH dataset and KITTI dataset prove that this method attained better performance in terms of both inference time and detection accuracy.

Both ADS (advanced driver assistance) and the ADAS (advanced driver assistance system) require effective detection of traffic signs. Though the FPN (feature pyramid network) has obtained better outcomes, [24] employed plug-and-play neck network IFA-FPN (integrated FPN with feature aggregation).

At first, the light operation is applied to efficiently use the system and enhance the inference speed of the system. Then IO (integrated operation) is deployed to resolve the imbalance issues with the RoI in pyramid stages. FA is then introduced to improve the feature representation capability of the feature maps, and this optimizes the robustness of the network. To signify data with large variances in size, the FA structure is applied, aggregating the multi-scale features to produce features with high representational ability. The experimental results of the study have projected that the system will produce better outcomes when applied in the TT100k (Tsinghua-Tencent 100k), STSD (Swedish Traffic Sign Dataset) and GTSDB (German Traffic Sign Detection Benchmark) datasets, but would lack improvement in detection accuracy and efficiency.

The comparison of one-stage and two-stage detectors denotes that the two-stage detectors use the proposal generator to generate a sparse set of proposals, then extract the features from each proposal. This is followed by region classifiers predicting the category of the proposal region. One-stage detectors directly perform categorical prediction of objects on each location of feature maps without the use of a cascaded region classification phase. This shows that the one-stage object detectors are effective for visualizing objects with optimized detection accuracy and reduced computational time.

5. Results and Discussion

To determine the performance of the proposed approach, the outcomes produced by the proposed study are compared with those from existing approaches. Table 1 represents the assessment of proposed method alongside conventional algorithms.

The conventional study employed a YOLO NMS fuzzy algorithm to simulate the driver’s reaction to obstacles with improved speed and accuracy. Object detection and tracking was accomplished using a hybrid of the fuzzy and NMS algorithms. The performance of the system is examined using KITTI dataset. According to the comparison of the algorithms shown in Table 1, however, a better outcome is gained through the proposed YOLOv4 model, with an AP of 98% and an MAP of 95%. This indicates increased speed and detection accuracy of objects with optimized time efficiency.

The existing method used a DL algorithm for active sensor fusion of the visible camera with corresponding sensors for autonomous driving. A skip connection allowed a feature-level sensor fusion approach to be applied, along with a thermal camera and millimeter-wave radar. Two networks called TV-Net and RV-Net were employed for performing sensor object detection, feature level fusion and specific feature extraction.

Table 2. Comparison of Proposed Model with Existing Models [25]

Architectures	Input	Output	Metrics	Outcome
YOLOv3 WCCS	RGB frame	Speed Detect	MAE MAP	78% 85%
YOLOV3 V3-Tiny	Speed RGB frame Position	Speed	Successful Episodes	84%
YOLOv3 FZ-NMS	Speed RGB frame IMU	Detect Speed	MAP AP	89% 93%
Proposed	RGB frame LIDAR	Speed Detect	AP MAP	98% 95%

Table 3. Comparison of Average Precision of Proposed Approach with Conventional Algorithms [26]

Models	AP (Average Precision)	Computational Time (ms)
Tiny YOLOv3	0.40	10
Late Fusion	0.40	14
RVNet	0.56	17
Proposed	0.74	14.85

Table 2 shows the average precision levels produced by the existing algorithms tiny YOLOv3, late fusion and RVNet, which are 0.40, 0.40 and 0.56 , with computational times of 10,14 and 17 ms , respectively. The proposed method, on the other hand, achieved an average precision of 0.74 with a computational time of 14.85 ms . This substantiates the better performance of proposed approach when compared with conventional methods.

6. Conclusion and Reflections on Future Work

Object detection approaches with improved speed and accuracy are essential for real-time control of automatic vehicles. Various approaches have investigated object detection using different techniques, but they were unable to rectify the issue of balancing speed with accuracy of detection. To address these problems, the present study implemented HDL-MODT to effectively detect and classify objects such as ground, vehicles, pedestrians, and obstacles. Noise removal and contrast enhancement was performed by A-Fuzzy and MSO techniques, which reduced the complexity of the system. The segmentation process involved in instance segmentation of the features was then used to increase the detection accuracy of the system. VGG-16 was used for feature extraction, and the extracted feature vectors were used for object detection with the YOLOv4 algorithm. Detected objects were tracked using IUKF, and the mapping of vehicles was exhibited, taking into account the RFID, velocity and location of objects. This work was applied in the MATLAB R2020 simulation tool to obtain the results, which showed improved outcomes. To evaluate performance, we also made a comparative analysis of existing studies with proposed method. The results projected that the proposed method outperformed other existing algorithms in both accuracy and speed. In the future, this proposed work could be extended by using an improved version of YOLO with detection of different objects to promote safer autonomous driving.

7. Declaration

- **Conflict of Interest:** The author reports there is no conflict of interest.
- **Funding:** None
- **Acknowledgement:** None

AUTHORS
Dheepika P.S.* – Department of Computer Science, The American College, Madurai Kamaraj University, Madurai, India, e-mail: psdheepika@gmail.com.
Umadevi V. – Department of Computer Science, Nehru Memorial College, Bharathidasan University, Tiruchirapalli, India, e-mail: umadevi@gmail.com.
*Corresponding author

References

[1] J. Mao, S. Shi, X. Wang, and H. Li, "3D Object Detection for Autonomous Driving: A Comprehensive Survey," *International Journal of Computer Vision*, pp. 1–55, 2023.

[2] Z. Wei, et al., "Mmwave Radar And Vision Fusion For Object Detection In Autonomous Driving: A Review," *Sensors*, vol. 22, p. 2542, 2022.

[3] B. Pan, L. Zhang, and H. Wang, "Multi-Stage Feature Pyramid Stereo Network-Based Disparity Estimation Approach For Two To Three-Dimensional Video Conversion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, pp. 1862–1875, 2020.

[4] Y. Zhang, et al., "PMPF: Point-Cloud Multiple-Pixel Fusion-Based 3D Object Detection for Autonomous Driving," *Remote Sensing*, vol. 15, p. 1580, 2023.

[5] D. Feng, et al., "Deep Multi-Modal Object Detection And Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, pp. 1341–1360, 2020.

[6] S. Yao, et al., "Radar-Camera Fusion For Object Detection and Semantic Segmentation in Autonomous Driving: A Comprehensive Review," *arXiv preprint arXiv:2304.10410*, 2023.

[7] A. Hazarika, et al., "Multi-camera 3D Object Detection for Autonomous Driving Using Deep Learning and Self-Attention Mechanism," *IEEE Access*, 2023.

[8] S. Rinchen, B. Vaidya, and H. T. Mouftah, "Scalable Multi-Task Learning R-CNN for Object Detection in Autonomous Driving." *In 2023 International Wireless Communications and Mobile Computing (IWCMC)*, Marrakesh, 19-23 June 2023; pp. 518–523.

[9] S. A. Khan, H. J. Lee, and H. Lim, "Enhancing Object Detection in Self-Driving Cars Using a Hybrid Approach," *Electronics*, vol. 12, p. 2768, 2023.

[10] S. Liang, et al., "Edge YOLO: Real-Time Intelligent Object Detection System Based on Edge-Cloud Cooperation in Autonomous Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 25345–25360, 2022.

[11] S.-W. Kim, et al., "Edge-Network-Assisted Real-Time Object Detection Framework for Autonomous Driving," *IEEE Network*, vol. 35, pp. 177–183, 2021.

[12] Y. Wu, et al., "An Enhanced Feature Pyramid Object Detection Network for Autonomous Driving," *Applied Sciences*, vol. 9, p. 4363, 2019.

[13] Y. Zhou, et al., "Object Detection in Autonomous Driving Scenarios Based On An Improved Faster-RCNN," *Applied Sciences*, vol. 11, p. 11630, 2021.

[14] Y. Cao, C. Li, Y. Peng, and H. Ru, "MCSYOLO: A Multiscale Object Detection Method for Autonomous Driving Road Environment Recognition," *IEEE Access*, vol. 11, pp. 22342–22354, 2023.

[15] W. Jang, et al., "R-TOD: Real-time Object Detector With Minimized End-To-End Delay for Autonomous Driving," *in 2020 IEEE Real-Time Systems Symposium (RTSS)*, 1-4th December 2020, pp. 191–204.

[16] S. Wu, Y. Yan, and W. Wang, "CF-YOLOX: An Autonomous Driving Detection Model for Multi-Scale Object Detection," *Sensors*, vol. 23, p. 3794, 2023.

[17] Z. Yuan, et al., "Temporal-Channel Transformer for 3D Lidar-Based Video Object Detection for Autonomous Driving," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, pp. 2068–2078, 2021.

[18] K. Geng, G. Dong, and W. Huang, "Robust Dual-Modal Image Quality Assessment Aware Deep Learning Network for Traffic Targets Detection Of Autonomous Vehicles," *Multimedia Tools and Applications*, vol. 81, pp. 6801–6826, 2022.

[19] L.-H. Wen and K.-H. Jo, "Fast and Accurate 3D Object Detection For Lidar-CameraBased Autonomous Vehicles Using One Shared Voxel-Based Backbone," *IEEE Access*, vol. 9, pp. 22080–22089, 2021.

[20] K. Roszyk, M. R. Nowicki, and P. Skrzypczyński, "Adopting the Yolov4 Architecture for Low-Latency Multispectral Pedestrian Detection in Autonomous Driving," *Sensors*, vol. 22, p. 1082, 2022.

[21] Y. Shao, et al., "PV-SSD: A Projection and Voxel-based Double Branch Single-Stage 3D Object Detector," *arXiv preprint arXiv:2308.06791*, 2023.

[22] Y. Cai, et al., "YOLOv4-5D: An Effective And Efficient Object Detector for Autonomous Driving," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2021.

[23] H. Nguyen, "Improving Faster R-CNN Framework for Fast Vehicle Detection," *Mathematical Problems in Engineering*, vol. 2019, pp. 1–11, 2019.

[24] Q. Tang, G. Cao, and K.-H. Jo, "Integrated Feature Pyramid Network with Feature Aggregation for Traffic Sign Detection," *IEEE Access*, vol. 9, pp. 117784–117794, 2021.

[25] N. Zaghari, et al., "The Improvement in Obstacle Detection in Autonomous Vehicles Using YOLO Non-Maximum Suppression Fuzzy Algorithm," *The Journal of Supercomputing*, vol. 77, pp. 13421–13446, 2021.

[26] V. John and S. Mita, "Deep feature-level Sensor Fusion Using Skip Connections for Real-Time Object Detection in Autonomous Driving," *Electronics*, vol. 10, p. 424, 2021.