YOLOv4Tiny: Bearing Angle Based Pose Estimation and Semantic Segmentation For 3D Object Detection From LiDAR Point Cloud & RGB-D Data

Ramana Rajendran (ramanarajphd@gmail.com)
Kalasalingam Academy of Research and Education
Murugan B.S.
Kalasalingam Academy of Research and Education

Research Article

Keywords: 3D object detection,noise removal,semantic segmentation, pose estimation Improved Mask R-CNN, AYOLO V4 tiny, 3D bounding box

Posted Date: July 20th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1864654/v1

YOLOv4Tiny: Bearing Angle Based Pose Estimation and Semantic Segmentation For 3D Object Detection From LiDAR Point Cloud & RGB-D Data

R.Ramana
Computer Science and Engineering
Kalasalingam Academy of Research and Education
TamilNadu, India.
{ramana@klu.ac.in, rramanabtechcse.21@gmail.com}

B.S.Murugan
Computer Science and Engineering
Kalasalingam Academy Of Research and Education
TamilNadu, India.
{b.s.murugan@klu.ac.in}

***Abstract* –3D object detection is a computer vision task that is used for various applications. However, it has several drawbacks due to occlusion, noise, inefficient field of view, etc. In addition, the recent LiDAR point clouds-based 3D object detection methods lack sparsity and homogeneity which affects the detection accuracy. In this paper, we detect 3D objects based on static and dynamic states by performing preprocessing, segmentation, extraction of features, classification, and regression considering RGB-D images, and LiDAR point clouds. Firstly, the quality of the image is enhanced by performing preprocessing which consists of two levels. In the first level, noise removal is performed for both RGB-D images and LiDAR cloud points by implementing Thresholding based Adaptive Median Filtering (TAMF) algorithm. In the second level, we perform point to voxel conversion for the LiDAR point clouds and fuse both RGB-D images and 3D LiDAR point clouds for better segmentation. Secondly, object segmentation is performed using Improved Mask R-CNN algorithm which creates masked instant bounding boxes by performing semantic segmentation considering both point and local-based information. Estimating pose for the segmented objects using bearing angle-based affine transformation method to extract features efficiently. Thirdly, Feature extraction is performed from the segmented 3D objects by considering high and low-level features using Attention-based YOLO V4 Tiny network and fusing both features for reducing the feature redundancy. Finally, based on the extracted feature the objects are classified as static and dynamic, and 3D bounding boxes are generated to improve the 3D object detection accuracy. The proposed work is evaluated by KITTI dataset. MATLAB R2020a simulation tool is used to perform simulation for the proposed work. The performance of the 3D object detection is evaluated in terms of several performance metrics such as accuracy, precision, recall, F-measure, computation time, and ROC-AUC curve compared to existing methods.**

Index Terms –3D object detection,noise removal,semantic segmentation, pose estimation Improved Mask R-CNN, A-YOLO V4 tiny, 3D bounding box.

I. INTRODUCTION

3D object detection from point cloud is a recent research topic in image processing. The point clouds are acquired from the digital cameras which are equipped with effective sensors [1]. However, recognition and detection of 3D objects from point clouds. However, the point clouds generated from photogrammetry are limited with poor precision which affects the object detection accuracy [2]. To overcome the issue of point clouds, LiDAR (Light Detection and Ranging) sensor is introduced which directly acquires the point clouds from the object and shows promising results in terms of accuracy, and provides faster results [3]. However, LiDARis susceptible to environmental noise, sparsity, and lack of depth information. The objects such as bicycle, vehicles, and pedestrians are represented by sparse point that makes complexity when using only point clouds and LiDAR point clouds [4], [5]. To solve this issue, researchers used fusion technology of RGB-D images and LiDAR point cloud for multiclass object detection [6]. The environment consists of multiple sensors for identifying moving objects, static objects, and obstacles. However, the RGB-D images and LiDAR images have many noises due to environmental factors which lead to image degradation hence it needs to be removed for obtaining high-quality images and cloud points for object detection [7], [8]. To reduce the noise due to environmental factors, many of the existing works adopt filtering methods and threshold methods. The noises in the images occluded the detected object which makes it infeasible to detect. The existing methods are lacks computational complexity during feature extraction and poor performance during objection tasks [9].

Object pose detection is one of the major challenging tasks in object detection existing works makes annotations manually. The manual annotations make them with increased computational complexity as there are multiple poses of an object [10]. Multiple features are extracted from the input images such as low-level features (Colour, Shape, and Texture) and high-level features (Semantic, Spatial, and Temporal) for feature learning and bounding box generation [11], [12]. Region proposal Network (RPN) is used to generate bounding box for object classification and regression. However, the RPNs are limited with high latency [13]. The bounding box is generated based on the center point, height, width, and angle. However static bounding boxes lead to loss of the viewport (Perspective view) of the images and increase false-positive rate [14]; hence the dynamic bounding boxes are

preferred for obtaining better perspective view of the images [15].

Many types of object detection algorithms are proposed for object detection such as point-based algorithms, voxel-based algorithms, and point-based and voxel-based algorithms. The point-based algorithm obtains the point-based information for object detection since the pose information of the object is lost. Voxel-based algorithms divide the point cloud into voxel that overcomes the sparsity problem of point cloud. A point-based and voxel-based algorithm provides the advantages of both point-based and voxel-based algorithms; therefore it provides fast and efficient results in object detection [16]-[18]. Many deep learning algorithms are proposed for 3D object detection such as Convolutional Neural Network (CNN), Region-based Convolutional Neural Network (RCNN), Faster RCNN, Single Shot Detector (SSD), and You Only Look Once (YOLO) which achieve better performance in object detection. However, the existing algorithms are limitedbyoverfitting, high time consumption, and less accuracy [19], [20]. Therefore, the precise solution for 3D object detection is yet to be proposed. The proposed work addresses the existing challenges and provides an efficient solution.

A. Motivation& Objectives

The main aim of this research is to detect and classify the objects in an accurate manner using RGB-D and LiDAR cloud point images. This research addresses the problems of cloud point sparsity, high information loss, high false-positive rate, and so on. The detection and classification of 3D objects are carried out from the RGB-D images and LiDAR cloud points however the existing approaches provide higher false detection rate that degrades the accuracy of 3D object detection system. In addition, 3D object detection faced many problems with state-of-the-art methods. We are motivated by these problems which are sorted as follows,

- **Multi-Feature Extraction:** The detection of 3D objects is based on both low level (Color, Shape, and Texture) and high-level features (Spatial, Temporal, and Semantic) but some researchers take only low-level features or high-level features that lead to inaccurate results during object detection that increases false alarm rate.
- **Static Bounding Boxes:** In existing works, generate a bounding box for 3D object detection however the bounding boxes are generated statically does not change dynamically which reduces the viewport of the bounding box hence it does not focus the perspective views of the images which reduces object detection accuracy.
- **High False Positive:** Pose estimation is one of the important challenges in 3D object detection because of various poses of the objects. However, the existing works do not concentrate on object pose estimation which loses the viewpoint of the images leading to high false-positive rate and less accuracy during object detection.
- **High Information Loss:** In 3D object detection most researchers are used region-based proposals for object detection and classification. However, it leads to high information loss because the regions get a chance to miss the small objects in the images which also reduces the detection accuracy.

The major objective of this research is to detect the 3D objects using RGB-D and LiDAR cloud points with high accuracy and low false-positive rate. The other objectives of this research are listed as follows,

- To increase the quality of RGB-D image and LiDAR cloud points by performing preprocessing which eliminates the noise from the images enhance the quality of the images and provide accurate result in object detection.
- To predict the viewport of the image by performing semantic segmentation and pose estimation which accurately detects the poses of the images and reduces the information loss of small objects and their pose.
- To increase detection and classification accuracy of 3D static and moving objects by performing multi-feature extraction which extracts both low level and high-level features from the images to increase accuracy and reduce false-positive rate.
- To increase the perspective view of the image by generating dynamic bounding boxes which provide better perspective view of the image and also increase accuracy of 3D object detection.

B. Research Contribution

This research mainly focuses on detecting the 3D objects and classifying the objects as static and dynamic. The major contributions of this work are sorted as follows,

- The noise in both RGB-D and LiDAR point clouds is removed by performing threshold-based adaptive median filtering algorithm and converting the points to voxels to enhance the quality of the image for improving the segmentation and 3D object detection accuracy.
- The segmentation of 3D objects from the pre-processed image is performed to reduce the classifier burden by employing improved Mask R-CNN which utilizes enhanced ROI (Region of Interest) alignment for extracting semantic information and estimating the pose of the objects using bearing angle to increase the detection accuracy.
- Implementing A-YOLO V4 tiny network for performing feature extraction from the segmented 3D objects by channel and spatial attention layers and

fusing both high and low-level features to improve the 3D object classification accuracy.

- The classification is done based on the feature extracted. The objects are classified as static and dynamic classes. The accuracy of the 3D object detection is enhanced by generating refined 3D bounding boxes for the 3D objects.

C. *Paper Organization*

The remaining sections of this paper are organized below. The state-of-the-art works on 3D object detection and its research gaps are summarized in section II. Section III describes the major problems which are experienced by the existing research methods. The proposed research methodology with procedure, algorithm, mathematical representations, and pseudocode are described in section IV. Section V represents the simulation and evaluation results of the proposed work and is compared with existing methods. Section VI concludes the proposed work and provides research directions for this work in future.

II. LITERATURE SURVEY

This section deals with literature on the existing approaches in which various research gaps are analyzed and mentioned. This section is further sub-divided into three sub-sections as follows,

A. *3D Point Cloud-based Approaches*

Point Cloud-based 3D object detection was performed using relational graphs [21]. The proposed work contains two modules namely 3D object proposal generation module and 3D relation module. In first module, the point clouds were taken as input, By using Multilayer Perceptron (MLP) semantic features are extracted from the input point clouds. From the semantic features, geometric centers, and pointed direction vectors are extracted and associated. The extracted features are sent to 3D object generation proposal for bounding box regression and classification. In second module, each proposal from bounding boxes is sent to point attention pooling, where it converts the extracted feature vectors into uniform vectors. The uniform vectors are sent to 3d object relation graph where it generates a graph. The graph shows the 3d object-object relationship between them. The Non-Maximum Suppression (NMS) algorithm is used to avoid the overlapping of 3d bound boxes. Here, MLP is used for feature extraction. It is time-consuming and the computations are difficult, this leads to high latency in our proposed work. Authors in [22], introduced a moving object classification based on 3D point clouds for urban traffic environments. The proposed work includes three processes such as point cloud pre-processing, feature extraction, and classification. In first process, ground points are removed from the point cloud and non-ground points are clustered for obtaining better results. In second process, three types of features are extracted from the pre-processed point cloud such as point based features, shape based features, and statistical features. Gini index criterion is used for selecting the optimal features from the extracted features. Based on the extracted features Support Vector Machine (SVM) is used for object classification. The simulation result demonstrates that the proposed work achieves better performance in terms of accuracy compared to existing works. This work considered only high-level features for object detection which is not enough for accurately classifying the object which reduces the accuracy of this research.

Authors in [23], introduced 3D object detection from point clouds using Point track Net. The Back Bone network was used for feature extraction. The two adjacent raw point clouds are taken as input. The proposed work contains four consecutive stages for object detection. In first stage point, wise feature extraction was done in which backbone network extracted the point wise features which are sent to Set Abstraction Module for down sampling and up sampling respectively. In second stage, probability filter was used to reserve the high probability foreground points. In third stage, merged features are sent to refinement module for calculating the final tracking results. In final stage, Trajectory generation module is used for giving the trajectory of the object. Here Backbone network is used for point based feature extraction hence it takes more computation time during object detection which increases high latency. Authors in [24], introduced GRNet to detect a 3D object using geometric relation network from point clouds. The proposed work used backbone network for feature extraction. The features are separated as intra object features and inter object features for extracting the relationship and difference between them for accuracy. Here, centralization module was used to point to the extracted features for indicating the center of the extracting features. After it reaches the centralization module feature pooling was done for better extraction. The object relation learning modules relate the extracted features and predict the bounding box parameters. Finally, object was detected within the 3D bounding box with detailed relationship between them. Here, raw point points are directly sent for feature extraction which leads to high computational complexity as the raw point clouds contain noise.

Authors in [25], introduced a semantic learning network to detect a 3D object by extracting multi view semantic features. The proposed work used Regional Proposal Network (RPN) for classification and prediction of the object. Using multiple view generator, four views BEV (Bird's Eye View), Rotated-Right view(R-RV), Rotated –Left View(R-LV), and Rotated Front view (R-FV) are captured and determine the features of low-level information. The determined features are sent to Spatial Recalibration Feature (SRF) for recalibration and enable the interaction between the features of different projections. The 3D Regional Proposal Network (RPN) formed regions from the calibrated features then the classification and 3D box prediction were done. The performance of proposed work was analyzed by KITTI dataset. Here, Regional Proposal Network was used for generating 3D boxes however it takes much time for training which is time a consuming hence it leads to high latency.

Authors in [26], introduced a joint edge and stixel based object detection using 3D point cloud. The main goal of this research is to segment and detect the objects in a given image by estimating the location and bounding box of the object. The object proposals are generated by edge box and stixel estimation for accurate object detection. The features are extracted from the bounding box, and then perform 3D point cloud matching and segmentation by matching the 3D sparse point cloud with scene point cloud. The matching process of 2D images and 3D cloud points is performed for object segmentation and detection however it provides inaccurate results due to lack of depth information of the image which reduces the accuracy of object detection.

The 3D object detection from point cloud using SC Net was introduced in [27]. In this proposed work One Shot Regional Proposal Network (RPN) was used for classification and regression of bounding boxes. The proposed work use point cloud as input. The inputs are sent to point rearrangement module where points are rearranged and sub divided into sub grids. The subgrids are the sent to Stixel feature extractor; a set of rectangular sticks which compress the information about the obstacles generates the sparse feature map. From the sparse feature map, bounding box classification and regression were using one shot RPN. The performance of proposed work was analyzed by KITTI dataset. The one shot RPN was used which generates region proposals based on the region hence it leads to high information loss of smaller objects which reduces the accuracy of object detection.

B. LiDAR Point Cloud based Approaches

Authors in [28], introduced an object detection method for detecting a 3D object from LIDAR point clouds. The proposed work used Multilayer Neural Network (MNN) for feature extraction and classification. Raw point clouds from LIDAR sensors are taken as input. By using height threshold unnecessary ground is removed which can occupy space and leads to wastage of computation. From raw point clouds, multiple geometric features are extracted from voxels continuously. Before classifying the multiple geometric features, the feature vectors extracted from the voxel are filtered by using scale filter. The filtered vectors are given a standard size that was normalized by object feature matrix for convenient classifying of features. The normalized matrices containing multiple geometric features are sent to Multilayer Neural Network (MNN) for classification of objects. This work considered only geometric features which was not enough for 3D object detection and classification hence it leads to high false-positive rate and less accuracy.

Authors in [29], uses LiDAR point based method for 3D object detection. The LiDAR points were acquired from the KITTI benchmark dataset this work utilizes sparse convolutions instead of general convolution layer which reduce the overfiitng problem. Initially, the LiDAR point cloud was acquired from the environment. From the acquired point clouds, feature map of sparse nature is coded. The coded feature map was the provided to backbone network which consist of multiple sparse convolution layers for feature extraction. Based on the extracted features, the output layer segments and detects the objects. The objects represented as multiple bins 2D bounding boxes which enhance the overall accuracy. The validation of proposed work was evaluated in several metrics and shows optimal results. Here, the feature was extracted directly from the raw point clouds which degrade the model precision rate.

The rapid motion segmentation was done using LiDAR point clouds based on combination of Probabilistic and Evidential Approaches [30]. The classification was done using Fast Motion Point Segmentation Algorithm whether it is static, dynamic, or unknown. The inputs are sent to Point motion segmentation module where they are subjected to two approaches namely probabilistic motion approach and evidential approach. From these approaches, objects are classified whether the objects are static dynamic, or unknown (occluded objects).

Authors in [31], introduced a method of cooperative perception for 3D object detection using infrastructure sensors for driving scenarios. Initially, the LiDAR sensor data are pre-processed for obtaining better results in object detection. Then the data are fused by fusion scheme algorithm in central fusion system. Based on the fused data the proposed work detects 3D objects. The proposed 3D object detection model includes three blocks for object detection as feature learning network, multiple convolutional middle layers, and region proposal network.

This paper introduced a novel method namely SARPNET for object detection using LiDAR cloud points [32]. The proposed work includes three components such as voxel generator, feature extraction, and shape attention region proposal network. The voxel generator is used for point to voxel conversion. Voxel feature encoding is used for extracting the voxel based features from the cloud points. Finally shape attention RPN block is used for generating region proposal for object detection. Based on the attention value the RPN classifies the object from point cloud. The LiDAR point cloud is considered as input hence it has non-homogenous and sparsity problems that lead to less accuracy in object detection.

C. Image Fusion based Approaches

Authors in [33], introduced a novel method to detect 3D object pose by RGB-D images for bin picking using semantic part segmentation method. The proposed work used Mask Regional Convolutional Neural Network (Mask-RCNN) algorithm for segmentation of semantic information. By Convolutional Neural Network (CNN) semantic features are extracted simultaneously and sent to Regional Proposal Network (RPN). Regional Proposal Network (RPN) scores object poses orientation and predicts box parameters. For each proposal, feature maps are extracted and they are sent to three different branches. From the data, the point features are extracted using Semantic Point Pair Feature Method. After

extraction of semantic features, voting procedure was done using Iterative Closest point for estimating the pose of the object. The performance of proposed work was analyzed by KITTI dataset. The Mask RCNN is used for object segmentation which leads to misalignment problems during masking process that reduces the performance of object segmentation and detection.

Authors in [34], introduced a method to detect a 3D object with the features of joint camera and 3D LiDAR features by using 3D cross-view spatial feature fusion (3D-CVF). The proposed work used Convolutional Neural Network (CNN) algorithm for feature extraction and classification of objects. The image is sent through Convolutional Neural Network (CNN) for feature extraction where the multi-view camera features are extracted by auto-calibrated projection unnecessary noise is filtered out. The filtered 2D image was mapped using cross-view feature mapping. Parallelly, LiDAR point clouds are voxelized and then sent to 3D sparse convolution network which contains six grids for better extraction of features. Then, the 2D feature and 3D LiDAR features are aggregated by adaptive gated fusion layer as joint camera-LiDAR features. The joint camera-LiDAR features are sent to Regional Proposal Network (RPN) for detecting regions containing objects. Then, camera feature, LiDAR feature, and joint camera-LiDAR are pooled in ROI based proposal network for 3D box refinement.CNN is used for object detection hence it does not extract the pose and orientation information from the object which leads to less accuracy in 3D object detection.

Authors in [35], used Dense Residual Fusion Network (DRF-Net) for 3D object detection. The proposed work uses DRF-Net for local feature extraction. Three inputs are taken into account they are 3D point clouds, RGB-D image, and Bearing Angle (BA) image. Objection segmentation was done by extracting features from point clouds and noises are removed. From RGB-D image Region of convergence (ROC) was selected. Both are combined and transformed into 2D BA images. By using BA image, Depth image and RGB image local features were extracted by using DRF-Net. The DRF-Net contains Dense Residual blocks, where the feature extraction was done. All the three extracted local features were combined and sent to Convolutional Neural Network (CNN) for high-level feature extraction and classification. Table I denotes the survey of existing research works and corresponding research gaps.

TABLE I
SUMMARY OF LITERATURE SURVEY

Approaches	References	Objective	Algorithm/model used	Limitations
3D Point Cloud-based Approaches	[21]	3D object was detected using Relation Graph Network	MLP and NMS algorithms	• High time consumption • Poor accuracy
	[22]	3D point clouds for moving object detection	SVM algorithm	• High latency • Considers only limited features
	[23]	Point cloud-based 3D object detection and tracking	Point track Net	• High computation time • Poor accuracy
	[24]	GRNet based 3D object detection using point clouds	Geometric relational network	• Poor performance during feature extraction and classification
	[25]	To detect a 3D object with multi views using point clouds	RPN algorithm	• More training time needed • Raw points extracted lead to environmental distortion
	[26]	Location-based 3D object detection based on edge and stixel.	Segment and detect the object by its location	• Not considering the depth of information of the image leads to inaccurate results.
	[27]	SCNet-based 3D object detection using point clouds.	One shot RPN and stixel feature extractor	• Susceptible to environmental noises.
LiDAR Point Cloud-based Approaches	[28]	Multi-feature extraction based 3D object detection using LiDAR point clouds	MNN algorithm	• High false-positive rate • Considers only limited features
	[29]	3D object detection from foreground segmentation using LiDAR point clouds	Sparse convolutional network	• Susceptible to high noise • Limited features leading to in accurate classification results.
	[30]	Probabilistic and evidential based vehicle motion segmentation using LiDAR	Fast motion point segmentation algorithm, Probabilistic and Evidential methods	• High time consumption • High complexity
	[31]	Cooperative 3D object detection using LiDAR data	Fusion scheme algorithm	• Feature redundancy • Less accuracy
	[32]	RPN based 3D object detection using LiDAR	RPN, and SARPNET	• Sparsity problem • Reduced efficiency
Image Fusion based Approaches	[33]	RGB-D images for segmentation and estimation of pose	Mask-RCNN, and CNN	• Highly vulnerable to misalignment
	[34]	Fusion of LiDAR and camera for 3D object detection	CNN, and RPN	• Over fitting • High time consumption
	[35]	RGB-D, point clouds and bearing angle based 3D objection detection	DRF-Net	• High false positives • Limited feature extraction

III. PROBLEM STATEMENT

The major problem statement of this research work is high false alarm rate, insufficient features, and low accuracy in 3D object detection. This section represents the addressing specific problemsfrom the existing researches which are described as follows,

Authors in [36], proposed multilayer spatial structure for performing 3D object detection using CNN.Voxalization of point clouds, feature extraction, feature map generation, ROI pooling, classification, and prediction were performed in this work using KITTI dataset. The major problems of this method are sorted as follows,

- Here, CNN algorithm is used for object classification however the traditional drawback of CNN is cannot detect the pose and orientation of the images hence it reduces the accuracy of object detection.
- This work only considers spatial information of the images but it does not enough for providing accurate results in 3D object detection which reduces the detection and classification accuracy.
- The proposed work considers only LiDAR point cloud as an input which leads to non-homogeneous and sparsity problems hence it provides inaccurate results. In addition, the raw point clouds are used for object detection which reduces the quality of the images because it has much noise due to sunlight and other factors that need to be removed otherwise it reduces the quality of the points thus reducing the accuracy.
- Here, RPN is used for generating the bounding boxes based on the regions which lead to increased high information loss of the smaller objects in the images also increase false-positive rate, in addition, the bounding boxes are generated in a static manner that reduces perspective view of the images thus reduces the performance of this work.

The 3D object detection using MLP algorithm was performed in [37].Drosophila-based elementary motion detection point net approach was used in this work to voxalize the LiDAR point clouds and remove the background noise. Finally, objects and their motion were detected using ML approach. The main problems of this research are as follows,

- Here, raw cloud points are converted into voxel then the background noises are eliminated which increases complexity because the LiDAR cloud points have many noises that are needs to remove from the cloud points before voxelization otherwise it does not provide accurate cloud points thus leading to poor accuracy in 3D moving object detection.
- The proposed work considers only geometrics features for segmentation and 3D object detection which are not enough for accurate moving object detection to reduce the accuracy of detection and increase the false alarm rate.
- MLP is used for 3D box generation however perceptron trained adaptively for detecting optimal solutions thus increasing latency during object detection. In addition, it only classifies the linearly separable vectors however objects are always not present in a linear manner hence it does not detect 3D objects accurately thus reducing the detection performance of this work.

Authors in [38],used RGB-D images with novel deep learning network to detect 3D objects. Map generation, reverse mapping, and voxalizing 2D images were performed in this method. KITTI dataset was used to validate the proposed work. The problems addressed in this work are sorted as follows,

- Here, 3D raw cloud points are considered for object detection which provides inaccurate results due to the presence of noise that reduces the performance of the proposed work.
- This work achieves less accuracy for 3D object detection due to lack of features because this work considers only low-level features for both 2D and 3D images which is not enough for accurate 3D object detection.
- Hough voting is used for object classification which mostly depends upon the inputs, hence the input has noise then the result of Hough voting is also inaccurate which reduces the accuracy. It calculates the voting for many features during object classification thus increasing high complexity that leads to high latency.

The random sample consensus algorithm (RANSAC) was proposed in [39] for performing segmentation for both 3D LiDAR and 2D images. Noise removal and feature fusion were performed in this work in which YOLO was used to perform classification. KITTI dataset is used to evaluate the proposed method. Several problems addressed in this research are listed as follows,

- Here, RANSAC algorithm is proposed for ground segmentation. However, if the number of iterations is limited then the RANSAC does not provide an optimal result in segmentation thus reducing the classification accuracy.
- YOLO algorithm is proposed for object classification which does not detect smaller objects in the image which leads to less accuracy and increased false alarm rate thus reducing the performance of object classification.
- YOLO generates 2D bounding boxes for object classification it does not consider the orientation of the bounding box thus increasing misclassification rate and the bounding boxes are not changed dynamically with

respect to the object position which reduces the perspective view of the object thus leading to inaccurate result.

- Here, 2D images are considered for object detection however it provides inaccurate results due to the presence of noise and absence of depth information thus reducing the quality and perspective views of the image.

Authors in [40], proposed regional-based convolutional network (RCNN) to detect 3D objects. Voxalization, feature extraction, and 3D box generation were performed to detect the 3D objects. KITTI dataset was used to evaluate the performance of this work. The major problems in this work are described as follows,

- Here, RCNN is used for 3D object detection however it takes much time for training because of generating multiple region proposals per image which leads to high latency during object detection and classification, in addition, it generates the bounding boxes based on the regions it causes high information loss of the smaller objects.
- Here, raw cloud points are considered for object detection which leads to inaccurate results due to the presence of the noise. In addition, this research only considers point clouds for object detection which leads to non-homogenous and misorder points problem that reduces the quality of point clouds.
- The detection of objects is greatly influenced by the pose of the objects in the training dataset. The lack of estimation of object pose results in increased false-positive rate.

Research Solutions: In order to overcome the problems experienced by the state-of-the-art methods, we perform 3D object detection using RGB-D and LiDAR cloud points which overcome the sparsity and misorder problems by considering the depth information of the images. Two-level preprocessing is performed to remove the noise from LiDAR point clouds and RGB-D images using TAMF algorithm and perform point to voxel conversion which enhances the image quality that increases the accuracy and decreases the false positive rate. Semantic segmentation and pose estimation are performed to increase the accuracy of 3D object detection in which Mask RCNN algorithm is used to perform semantic segmentation and affine transformation method is used to perform pose estimation. Implementation of A-YOLO V4 tiny algorithmfor performing object classification and regression by extracting both high level and low-level features with low latency. Generating 3D bounding boxes dynamically, based on the pose of the objects and orientation enhances the viewport of the images.

IV. PROPOSED WORK

In this research work, we have focused on accurate detection of 3D objects from LiDAR cloud points and RGB-D images. The accuracy of the detection is also determined based on the segmentation and feature extraction from cloud points. The main goal of this research is to enhance detection accuracy in segmentation and classification. For 3D object detection, we have used KITTI datasetThe proposed work includes four consecutive phases such as,

- Two-level preprocessing
- Improved Mask R-CNN For Semantic Segmentation and Pose Estimation
- Multi-Feature Extraction and Feature Fusion
- 3D Object Classification and Regression

A. Two Level Pre-Processing

In this research, we have taken two inputs such as RGB-D images and LiDAR cloud points. Initially, we perform two-level of preprocessing, in first level noises are removed from both 3D LiDAR cloud points and RGB-D images since the raw LiDAR cloud points have much noise such as sunlight shot noise and photoresponse non-uniformity noise that are needed to eliminate for increasing the quality of the cloud points. In other words, the height of the cloud points affects the quality; hence we need to solve this problem for obtaining high accuracy in 3D object detection. For that purpose, we have divided the LiDAR cloud points into groups such as noise points (ground points) and non-noise points. Then the noise points are eliminated based on the threshold. The noise in the RGB-D image is also removed to enhance the image quality. For that purpose, we proposed Thresholding based Adaptive median filter (TAMF) which eliminates the noise that obtains high-quality LiDAR point clouds.

Let $\check{K}(\text{ঌ})$ be the noisy RGB-D image and $\check{K}(\text{ɾ})$ be the noisy LiDAR cloud points. The formulation of these inputs is represented as,

$$\check{K}(\text{ঌ}) = S(\text{ฑ}) + N(8) \quad (1)$$

$$\check{K}(\text{ɾ}) = S(\text{ฑ}) + N(8) \quad (2)$$

Where $S(\text{ฑ})$ represents the original information and the $N(8)$ denotes the noise. The RGB-D images are converted into greyscale before performing noise removal. Applying a sliding window to the median filtering with dimension $(\partial = v\ x\ v)$. The pixel $\check{K}(\text{ʃ})$ of the RGB-D image which is arranged from small to large based on its grey value. The pixel median value is represented as $\check{K}_{med}$.

Assuming, $\check{K}_{min}$ as the window's minimum grey value, $\check{K}_{max}$ are maximum grey value and the center pixel's grey value is represented as $\check{K}_{\zeta}$. If $\check{K}_{\zeta}$ is not equal to $\check{K}_{min}$ or $\check{K}_{max}$ then it is the original information of the image ζ, otherwise, it is considered noise. Replacing $\check{K}_{\zeta}$, with $\check{K}_{med}$ to perform denoising.

The size of the window is adjusted adaptively to increase the filtering template's size. The value of a pixel in this filter window is represented as $\check{K}(\xi)$. The adaptive median filtering has 2 steps which are defined as follows,

Step 1: If $\check{K}_{min} < \check{K}_{med} < \check{K}_{max}$ shift to step 2 otherwise, adjust the window as $v = v + 2$. When $\partial > \partial_{max}$, then the output is $\check{K}_{med}$ otherwise, repeat step 1.

Where, ∂_{max} denotes the window's upper limit.

Step 2: If $\check{K}_{min} < \check{K}_{med} < \check{K}_{max}$ then the output is $\check{K}(\xi)$otherwise $\check{K}_{med}$.

Step 2 may reduce the window's upper limit and it does not preserve edges and other important details for smoothening the image. The improved wavelet threshold function is implemented to eliminate the remaining noise with preserving the edge and other important information. This method is used for both RGB-D and LiDAR cloud points. This threshold function has two types of frequency coefficients such as low and high-frequency coefficients in which the image information is in the low-frequency coefficients whereas the noise and edge information are in the high-frequency coefficients. In order to preserve useful information and eliminate noise, the wavelet threshold function is applied to high-frequency coefficients. Two characters of threshold function are considered such as soft and hard threshold function. The hard threshold function is represented as,

$$h_{th} = \begin{cases} h & |h| \geq t \\ 0 & |h| < t \end{cases} \quad (3)$$

Where h_t denotes the wavelet threshold. The soft threshold function is represented as,

$$h_{ts} = \begin{cases} sign(h)(|h| - t) & |h| \geq t \\ 0 & |h| < t \end{cases} \quad (4)$$

The wavelet threshold function is improved which is continuous at $|h| = t$ and acquire the overall definition domain with various kinds of slopes during $|h| < t$and $|h| \geq t$. The formulation of the improved wavelet threshold function is expressed as,

$$h_{ti} = \begin{cases} sign(h)(|h| - t) & |h| \geq t \\ (1 - \lambda)\,|h| & |h| < t \end{cases} \quad (5)$$

Where λ denotes the adjustable factor. This improved threshold function is having the adjustment factor $\lambda(0 < \lambda < 1)$ determines the threshold function is not equal to zero during $|h| < t$. This shows that the wavelet coefficients are less compressed than soft threshold function which eliminates the remaining noise in RGB-D images and noise points of the 3D LiDAR point clouds. Fig 1 illustrates the proposed 3D-YOLOv4 model.

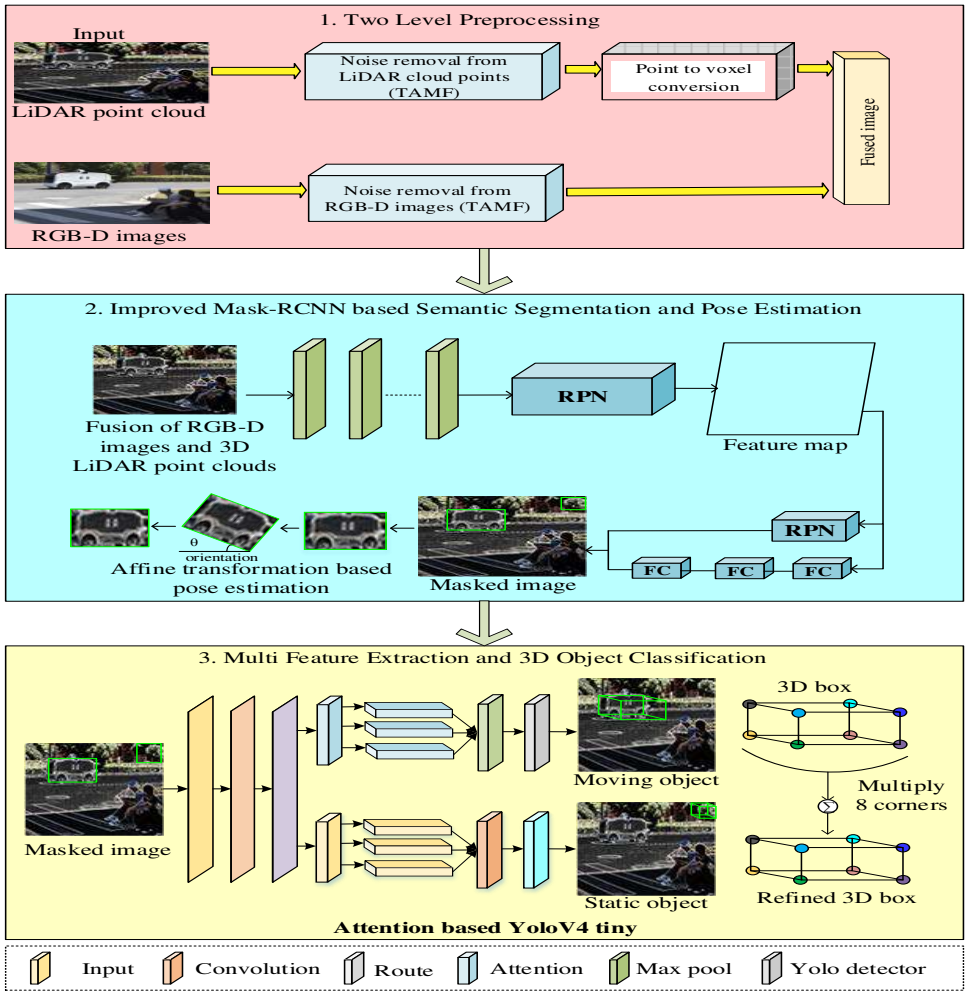


Fig. 1 Proposed 3D-YOLOv4 Model

In second level, we perform points to voxel conversion by achieving 3D LiDAR cloud points for better perception view which helps improve the accuracy of 3D object detection. Each cloud points in the LiDAR with three coordinates (X, Y, Z) having three coordinate value pairs such as $([x_{min}, x_{max}], [y_{min}, y_{max}], [z_{min}, z_{max}])$. All the point cloud data in the LiDAR are initially bounding based on Euclidean space s^3 which is divided into voxel subsets based on Cartesian coordinate system and each voxel subset is represented in terms of index $U(d, f, c)$.

Where $d \in [0; L_x - 1], f \in [0; L_y - 1]$, and $c \in [0; L_z - 1]$. According to the individual voxels' dimensions $(\Delta_x, \Delta_y, \Delta_z)$, the count of voxels (L_x, L_y, L_z) at every direction are expressed as,

$$L_x = \frac{(x_{max} - x_{min})}{\Delta_x} + 1 \quad (6)$$

$$L_y = \frac{(y_{max} - y_{min})}{\Delta_y} + 1 \quad (7)$$

$$L_z = \frac{(z_{max} - z_{min})}{\Delta_z} + 1 \quad (8)$$

Where, $\Delta_x, \Delta_y, \Delta_z$ are denoted as size of the voxel and L_x, L_y, L_z represents the voxels count in each direction. A brief explanation of noise removal is provided by describing the pseudo code below.

Pseudo code
Noise Removal Initialize{ɬ, ɾ, ξ} Initialize $\check{K}_{min}, \check{K}_{med}, \check{K}_{max}$ **For**ɬ, ɾ ← m**do** **If** the step 1 condition is not met **If**$\partial > \partial_{max}$ **then** Repeat step 1 **Else** Shift to step 2 **End if** Computeh_{th}, h_{ts}using (3), and (4) Compute $h_{ti}$and adjust threshold using (5) Return denoisedɬ, ɾ **End for**

B. *Improved Mask RCNN for Segmentation and Pose Estimation*

After completing preprocessing, pose estimation and segmentation process are initialized. For that, the preprocessed RGB-D image and 3D LiDAR point clouds are fused for obtaining better results in 3D object detection. The fused images are considered for semantic segmentation. Here, we consider both point-based and local-based semantic information for semantic segmentation by generating region boxes using Improved Mask R-CNN (IMR-CNN) which helps to achieve high accuracy of pose estimation by solving the misalign problem of traditional Mask RCNN.

The classification head performs classification of ROI and regression of bounding box in traditional Mask R-CNN. The Improved Mask-RCNN is a model of instance segmentation which is expanded based on faster RCNN. The mask-RCNN is improved by modifying the head block of the mask to identify the boundaries in a precise manner. Initially, learnable upsampling is performed for feature maps by adding decoder layer which provides spatial resolution efficiently. Second, ROI alignment is performed by the ROI align block with skip connections to generate features with high resolution which is further used by altered mask head.

a) Improved Mask R-CNN Network Structure

The improved mask-RCNN algorithm is used to segment the fused image to enhance the accuracy of 3D object detection. It consists of input block, backbone network,region proposal network (RPN), feature pyramid network (FPN), two types of heads (i.e., class and mask heads),and ROIAlign for alignment. The RPN is used to identify the locations and objects, the class head is used to perform bounding box classification, and the mask head is implemented for masks extraction based on pixel-wise from the respective cropped features which is acquired from the bounding box.

b) RPN Network

The obtained features from the image are passed as input for the RPN block to identify every class of the objects and for bounding box calculation around the specific objects. Extraction of large objects is performed from the feature maps with low-resolution whereas, feature maps with high resolution are used to extract smaller objects.

c) Backbone Network

In the backbone network, a hybrid deep encoder model is deployed as ResNet101 with FPN. This hybrid backbone network is implemented in Mast R-CNN for feature map extraction for the input images in various resolutions and scales. Multi-scale semantic features (i.e., point and local) are extracted based on adding FPN with ResNet through lateral connections, top-down pathways, and bottom-up pathways. Assume, the residual building block to perform residual mapping which is represented as,

$$ɘ = \mathrm{F}(ϒ, \{\Psi_i\}) + ϒ \quad (9)$$

Whereɘ denotes the input vector, ϒ represents the output vector and$\mathrm{F}(ϒ, \{\Psi_i\})$shows the residual mapping which will be learned.The dimensions of F and ϒ must be same in the above equation otherwise, linear projection Ψ_s is performed for dimensions matching by shortcut connections which are represented as,

$$ɘ = \mathrm{F}(ϒ, \{\Psi_i\}) + \Psi_s ϒ \quad (10)$$

Then the feature pyramid is viewed based on image pyramid. In general, ROI assigning with ɦ x ɯ to ɣ$_q$ of the FPN which is represented as,

$$q = [q_0 + log_2 \left(\frac{\sqrt{wh}}{360 \text{ x } 640}\right)] \quad (11)$$

Where ɣ denotes the feature pyramid, q_0 represents the target level of ROI with $h\,x\,w$ which are mapped and $360\ x\ 640$ represents the height and width of the image size.

d) ROI Alignment

In Fast R-CNN, the ROI pooling layer block is used in the ROI bounding box to acquire the feature map from the backbone network which is small in size. The ROI features are resized by the Fast R-CNN into small stable spatial extent of ℏ x ш feature map using max-pooling layer. Where, ℏ X ш represents the hyperparameters (i.e., height and width) of the layers. The segmentation misalignment occurs during feature extraction based on spatial quantization method. The ROI Align block with (14 X 14) pixelscancels the rounding operation to overcome these issues when ROI resizing. High-resolution feature generation is performed by the other ROI Align layer of (56 X 56) pixels for modified mask head. The bilinear interpolation(ω) in the RPN is carried and generation of anchor points with Ж numbers.Let us assume anchor points with four numbers which are generated to find the pixel of objects and it is expressed as,

$$\omega = \{A_1, A_2, A_3, A_4\} \quad (12)$$

Where, the base points are represented as ω which are expressed as follows,

$$ROI(Ⱶ, ɲ) \approx \frac{ROI(A_1)}{(Ⱶ_2 - Ⱶ_1)(ɲ_2 - ɲ_1)} * (Ⱶ_2 - ɲ)(ɲ_2 - ɲ) + \frac{ROI(A_2)}{(Ⱶ_2 - Ⱶ_1)(ɲ_2 - ɲ_1)} * (Ⱶ_2 - ɲ)(ɲ_2 - ɲ) + \frac{ROI(A_3)}{(Ⱶ_2 - Ⱶ_1)(ɲ_2 - ɲ_1)} * (Ⱶ_2 - ɲ)(ɲ_2 - ɲ) + \frac{ROI(A_4)}{(Ⱶ_2 - Ⱶ_1)(ɲ_2 - ɲ_1)} * (Ⱶ_2 - ɲ)(ɲ_2 - ɲ) \quad (13)$$

Where,Ⱶ and ɲ denote the directions (i.e., axis).

e) Loss Function

In Mask-RCNN loss is occurred due to multiple task learning for ROI sampling. This results in various types of losses such as mask loss accumulation, classification loss, and bounding box loss which is expressed as,

$$£ = £_ŋ + £_ć + £_ɓ \quad (14)$$

Where, $£_ŋ$ denotes the loss of mask prediction during segmentation,$£_ć$ denotes the loss of class-label prediction, and $£_ɓ$ shows the loss of bounding-box refinement. From the above equation, the segmentation of objects is performed with efficient generalization performance.

After segmenting the objects, pose estimation is performed. Object pose estimation is challenging process in 3D object detection due to occlusion, and the pose of the images are varied from one another since we need to perform pose estimation in 3D object detection. The affine transformation is used for estimating the pose based on bearing angle by rotating and translating the boxes with respect to the orientation and position of the box. The detection of 3D objects having close relationship between the past and the present consecutive frames.The computation of bearing angle by finding the objects' centroid to estimate the position.The object movement is calculated which is represented as,

$$ж_{m,n} = \sum_ш \sum_ы ш^m ы^n I(m,n) \quad (15)$$

Where$m, n \in \{0.1\}$and $I(\ , ы)$ represent the intensity of the pixel at its location $(ш, ы)$. The centroid of the 3D object is calculated as follows,

$$\{ш', ы'\} = \left\{\frac{ж_{10}}{ж_{00}}, \frac{ж_{01}}{ж_{00}}\right\} \quad (16)$$

The bearing angle □ is computed based on the above centroid equation which is represented as,

$$□' = tan^{-1}\frac{ш'}{□} \quad (17)$$

Where □ denotes the focal length. The approximate position of the object is determined in the present frame based on bounding box generated by the improved Mask R-CNN during segmentation. The positions of the objects are sampled by performing sampling using affine transformation. This samples the bounding box of the object with high accuracy by sampling the effective region of the object. The pose estimation using affine transformation has several steps which as described below.

Step 1: The bounding box of the objects' initial value of frame ⍵is set by initializing Ƭ = 1in which the initial value is same as the object bounding box of previous frame ⍵ − 1, which is$ֆ_{⍵-1} = [ր_1, ր_2, ր_3, ր_4, ր_5, ր_6]$. Where, $ր_1$ to $ր_6$ represents the 6-dimensional vectors ranges between $-1,1)$ with ₥ samples. The dimensional vectors $ր \in -1 < ր_i < 1, i = 1,2,3,4,5,6)$.

Step 2: Standardize the vectors which are represented as,

$$ր' = \frac{ր}{\sqrt{\sum_{i=1}^{6} ր_i{}^2}} \quad (18)$$

Step 3: The velocityա computation of the object with velocity vector$ա_{⍵-1}$ from the previous frame⍵ − 1 which is expressed as,

$$ա^Ƭ_{⍵-1} = ४ա_{⍵-1} + ր' \quad (19)$$

Step 4: Let Ƭ = Ƭ + 1. Repeat steps 1 to 3, until Ƭ = ₥. Then the velocity is expressed as,

$$ա_⍵ = [ա^1_⍵, ա^2_⍵, \ldots ա^{₥}_⍵] \quad (20)$$

Step 5: According to velocity, the ₥ samples with parameters of affine transformation are generated as $\{ֆ^1_⍵, ֆ^2_⍵, \ldots ֆ^{₥}_⍵\}$ based on Riemannian manifold design and its corresponding tangent space which is expressed as,

$$ֆ_⍵ = ֆ_{⍵-1}\exp(ա_⍵) \quad (21)$$

After obtaining all affine samples, draw the object region based on the current frame of the 3D image. Then normalize the size of the object and estimate the pose of the 3D objects based on bearing angles.Fig 2 represents the segmentation and pose estimation of 3D objects with improved Mask R-CNN network.

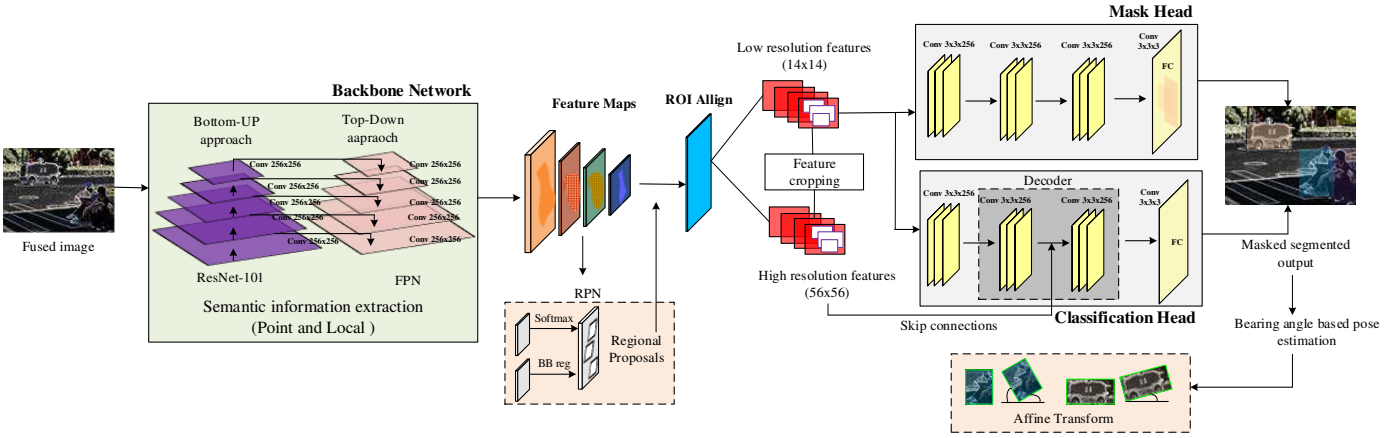


Fig. 2 3D Object Segmentation and Pose Estimation

C. *Multi Feature Extraction and Feature Fusion*

After completing segmentation, we perform multi-feature extraction and feature fusion process. From the segmented part, we extract the features for object classification. In this research, we have extracted both low level (Color, shape, texture) and high level (temporal, spatial, semantic, and geometric features) features which increase detection accuracy of 3D objects. For that, we proposed Attention-based YOLO V4 Tiny (A-YOLO V4 tiny) which performs well compared to YOLO V3 tiny and YOLO V4. The A-YOLO V4 tiny extracts multiple features from the segmented part at the backbone network (cross stage partial dense network) for accurate 3D object classification. Feature mapping is performed at YOLO V4 neck (path aggregation network). The feature pooling and 3D box generation are performed at YOLO V4 head. Then feature fusion is performed for reducing the redundant features and extracting the important feature for 3D object classification.

The CSP block module in the YOLO V4 tiny method used activation function which is LeakyReLU function and it is represented as,

$$\acute{\omega} = \begin{cases} \int_{bi} \int_{bi} \geq 0 \\ \frac{\int_i}{\Pi_i} \int_{bi} < 0 \end{cases} \qquad (22)$$

Where, $\acute{\omega}$ and $\int$ denote the output and input of the module, and $\Pi_i \in (1, +\infty)$ denotes the constant parameters. The CSP block module is then replaced with ResBlock-D module for increasing the speed with slight accuracy impact. This block uses dual paths in the network for the input feature map.Path one contains dual layers of 1x1 and 3x3 convolutions with 1x1 convolutions and 2 strides. Path two contains average poolings of 2x2 with 1x1 convolutions and 2 strides for two layers. These paths in the ResBlock-D module decrease the computation. The redundancy of the bounding box is reduced by evaluating its confidence score which is represented as,

$$\square_a^b = \square_{a,b} * IOU_p^t \qquad (23)$$

Where $\square_a^b$ denotes the confidence score of the bounding box $b$in grid a, $\square_{a,b}$ denotes the object's function, p denotes the prediction, and t denotes the truth.The floating-point operation is performed for the ResBlock-D module to compute the complexity during computation which is expressed as,

$$\varepsilon_{\square} = \sum_{\mathfrak{I}=1}^{\mathfrak{S}} \mathfrak{z}_{\mathfrak{I}}^2 . \epsilon_{\mathfrak{I}}^2 . \Omega_{\mathfrak{I}-1} . \Omega_{\mathfrak{I}} \qquad (24)$$

Where $\mathfrak{S}$ denotes sum of convolution layers, $\mathfrak{z}_{\mathfrak{I}}^2$ denotes output size of the feature map in $\mathfrak{I}^{th}$ layer of convolution, $\epsilon_{\mathfrak{I}}^2$ denotes the kernel size numbers, $\Omega_{\mathfrak{I}-1}$ and $\Omega_{\mathfrak{I}}$ denotes the input and output channel numbers respectively.

Design and add two blocks of auxiliary network to the ResBlock-D to increase the 3D object detection accuracy. Extracting both high-level and low-level features from the object and fusing those features to detect the 3D objects with high accuracy. Implementing two types of attention modules such as channel attention and spatial attention in the attention layer.The channel attention module deals with the important features of the 3D object whereas the spatial attention module deals with the position of the 3D object. The convolutional block attention module (CBAM) is used to simultaneously deals with both channel and spatial attention modules in which the CBAM is represented as,

$$\square' = \square_{ch}(\square) \otimes \square,$$
$$\square'' = \square_{sp}(\square') \otimes \square' \qquad (25)$$

Where $\square$ denotes the feature map of the input, $\otimes$ denotes multiplication based on element-wise, $\square''$ represents the final feature map output, $\square_{ch}()$ and $\square_{sp}()$ are attention map of channel and spatial module.

The loss function of A-YOLO V4 tiny consists of three major parts which can be represented as follows,

$$\mathbb{L} = \mathbb{L}_1 + \mathbb{L}_2 + \mathbb{L}_3 \qquad (26)$$

Where $\mathbb{L}_1$ denotes the confidence loss function, $\mathbb{L}_2$ represents the loss function of classification, and $\mathbb{L}_3$ represents the loss function of bound box regression. The confidence loss

function is due to negative transfer problem and it is solved by entropy which is applied in the network and can be expressed as,

$$\mathbb{L}_1 = -\sum_{a=0}^{G^2}\sum_{b=0}^{\dot{\alpha}} R_{ab}^{o}\left[\square'^{b}_{a}\log(\square_a^b) + (1 - \square'^{b}_{a})\log(1 - \square_a^b)\right] - \tilde{\eta}\sum_{a=0}^{G^2}\sum_{b=0}^{\dot{\alpha}}(1 - R_{ab}^{o})\left[\square'^{b}_{a}\log(\square_a^b) + (1 - \square'^{b}_{a})\log(1 - \square_a^b)\right] \quad (27)$$

Where G^2 denotes the grid number in the input image, $\dot{\alpha}$ denotes the bounding box numbers in a grid, R_{ab}^{o} denotes the object function, If the current object detection is performed bythe bounding box b of grid a, $R_{ab}^{o} = 1$, otherwise $R_{ab}^{o} = 0$.The $\square_a^b$ denotes the predicted box confidence score, $\square'^{b}_{a}$ denotes the truth box confidence score, and $\tilde{\eta}$ denotes the weight parameter. The loss function of classification is expressed as,

$$\mathbb{L}_2 = -\sum_{a=0}^{G^2}\sum_{b=0}^{\dot{\alpha}} R_{ab}^{o}\sum_{u=1}^{U}[P'^{b}_{a}(c)\log\left(P_a^b(c)\right) - 1-P'abc\log 1-Pabc] \quad (28)$$

Where $P_a^b(c)$ denotes the probability of prediction, P'^{b}_{a} denotes the probability of truth of the object which belongs to the classification c in bounding box b of the grid a. The loss function of bounding box regression is represented as,

$$\mathbb{L}_3 = 1 - IOU + \frac{\dot{r}^2(n,n^g)}{y^2} + \frac{16}{\pi^4}\frac{\left(arctan\ \frac{\acute{o}^g}{\underline{d}^g} - arctan\ \frac{\acute{o}}{\underline{d}}\right)^4}{1-IOU+\frac{4}{\pi^2}\left(arctan\ \frac{\acute{o}^g}{\underline{d}^g} - arctan\ \frac{\acute{o}}{\underline{d}}\right)^2} \quad (29)$$

Where IOU denotes the intersection over union in between the predicted and truth bounding boxes, $\acute{o}^g$ denotes the truth width of bounding box, $\underline{d}^g$ represents the truth bounding box height, $\acute{o}$ denotes the predicted width of the bounding box, $\underline{d}$ denotes the predicted height of the bounding box, $\dot{r}^2(n,n^g)$ represents the Euclidean distance between the predicted and truth bounding box center points,y denotes the least diagonal distance between predicted and truth bounding box.

D. 3D Object Classification and Regression

After completing feature extraction, we have performed classification and regression. We have classified the objects into two classes as static and moving objects using fused features by A-YOLO V4 tiny which has two types of attention layers for static and moving object features, where each type of attention layer includes multiple layers that are adaptively changed based on the features. Based on the attention layer the object is detected whether it is static or moving. The $\square_{ch}(\square)$ and $\square_{sp}(\square')$ are represented as,

$$\square_{ch}(\square) = \sigma(\square\left(Avg(\square)\right) + \square\left(Max(\square)\right)) \quad (30)$$

$$\square_{sp}(\square') = \sigma(\dot{\alpha}^{7x7}[Max(\square'); Avg(\square')]) \quad (31)$$

Where $Avg()$ and $Max()$ denote the average and max pooling operation, $\square$ represents the multi-layer perceptron (MLP) network, $\sigma()$ shows the sigmoid function, $\dot{\alpha}^{7x7}$ denotes the convolution operation with 7 X 7 Kernal size, (;) denotes the concatenate operation.

After detecting the objects, 3D bounding boxes are generated by considering center, angle, width, and height. Then eight corners of the boundary boxes are multiplied and added for generating refined 3D boxes that obtain high perceptional view of the object thus increasing high detection accuracy in 3D object detection.For each bounding box corner aggregation centroids $\mathbb{C}^{(\dot{y})}$, the $e_{\dot{y}}$ neighbors with $r_{(k)}$ radius are aggregated to form a point set $\mathbb{S}^{(\dot{y})}$ which is represented as,

$$\mathbb{S}_i^{(\dot{y})} = \{\mathbb{C}_j^{\dot{y}-1} - \mathbb{C}_i^{\dot{y}})\mathbb{C}_j^{\dot{y}-1}\} \quad (32)$$

Where $\left\|\mathbb{C}_j^{\dot{y}-1} - \mathbb{C}_i^{\dot{y}}\right\| < r_{(k)}$ and $\left(\mathbb{C}_j^{\dot{y}-1}, \mathbb{C}_j^{\dot{y}-1}\right) \in \mathbb{C}^{(\dot{y})}$. Ineach $\mathbb{S}^{(\dot{y})}$, applying $MLP_3^{\dot{y}}$ for feature $F_i^{\dot{y}}$ transform from previous$ch^{\dot{y}-1}$ to current$ch^{\dot{y}}$ channels which are represented as,

$$F_i^{\dot{y}} = MAX\left(MLP_3^{\dot{y}}\left(\mathbb{S}_i^{(\dot{y})}\right)\right) \in \mathbb{R}^{1\ X\ ch^{\dot{y}}} \quad (33)$$

The transformed features aggregation is performed by the max-pooling with point axis to the point$\mathbb{C}^{(\dot{y})}$ is represented as,

$$P^{\dot{y}} = \{\left(\mathbb{C}_i^{\dot{y}}, F_i^{\dot{y}}\right): i = 1, \dots., N^{\dot{y}}\} \quad (34)$$

Aggregating the features in $P^{\dot{y}}$ of eight corners perspective is represented as,

$$F^{cp} = [f_1^{\dot{y}-1}, \dots., f_1^{\dot{y}-1}] \in \mathbb{R}^{8\ X\ ch^{\dot{y}+1}} \quad (35)$$

The aggregated corners from the features are then multiplied based on element-wise multiplication which is represented as,

$$F^{cp'} = Å \odot F^{cp} \quad (36)$$

Where $F^{cp'}$ denotes the reweighted feature, Å denotes the important information based on channel and perspective-wise dimensions. Finally, summation of multiplied eight sub-features (i.e., corners) to create 3D bounding box which is represented as,

$$F^{b} = \sum_{i=1}^{8} F_i^{cp'} \quad (37)$$

Fig 3 illustrates the feature extraction and classification and it is clearly explained by the pseudocode which is described as follows,

Pseudocode

```
# Feature Extraction and Classification
Initialize Features {F = f1, f2, f3, ... fN}
Initialize A-YOLO V4 tiny
A-YOLO v=V4 tiny ← Input training data for extracting features and classification
   For j ← 0 to N do
      Extract F from the segmented 3D object
Extract features by □ch
      Extract features by □sp
      Combining the extracted features
      Perform 3D object classification using (30), and (31)
   Class ← {static, dynamic}
End for
```

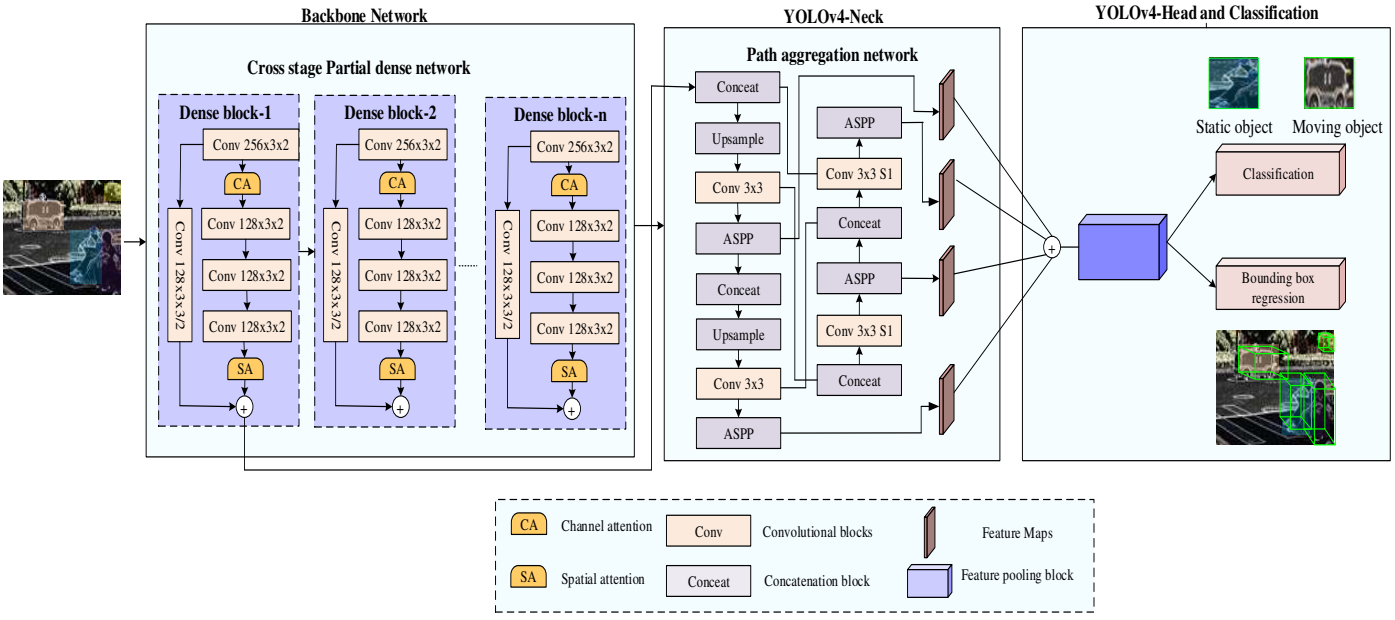


Fig. 3 A-YOLO V4 Tiny Network based 3D Object Detection

V. Experimental Results

This section deals with experimental results of the proposed 3D-YOLOv4 tiny approach. Further, this section is validated with sub sections such as dataset description, simulation step up, comparative analysis, and research summary which are described below.

A. Dataset Description

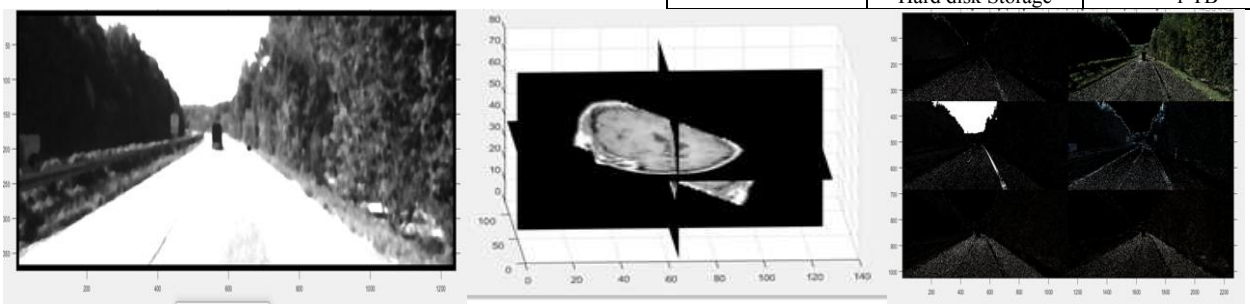
The images are acquired from the Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) dataset. The KITTI is one of the prominent datasets for computer vision and 3D object detection tasks. In our work, we have mainly focused on 3D object detection (moving and static). The image from the dataset was acquired by placing the many monochrome RGB cameras, laser scanners, inertial measurement, and localization units on the roof of the car which captures the LiDAR and RGB-D data. There are eight labels employed in the dataset such as pedestrians, cyclists, persons, trucks, vans, trams, and, Segway. The considered dataset consists of one hundred and seventy training sequences and forty-six testing sequences. The proposed work utilizes this dataset for performing operations such as semantic segmentation, detection of depth, and 3D object detection. We have trained the proposed classifier by the KITTI dataset based on the different objects' size, shape, and pose information.

B. Simulation Setup

The simulation of the proposed 3D-YOLOv4 tiny approach was executed using MATLAB R2020a. The MATLAB in image processing domain enables as a best supportive a tool that can enhance the performance of simulation results. The algorithms in the proposed work can be easily implemented in the MATLAB simulation tool in which the algorithm can independently work to their nature. Table II provides the system configuration where the proposed work is simulated. The simulation results of the proposed processes such as noise removal, 3D voxel conversion, semantic segmentation, feature extraction results, and object detection are shown in fig 4,

Table II
System Configuration

Soft Ware Specifications	Operating System	Windows 10
	MATLAB	R2020a
Hard Ware Specifications	Processor	Intel(R) Core(TM) i5-4590S
	Random Access Memory	6 GB
	Central Processing Unit	3.00 GHz
	Hard disk Storage	1 TB



(a) (b) (c)

(d) (e) (f)

Fig. 4 (a) Noise removal results, (b) 3D voxel conversion results, (c) Semantic segmentation results, (d) Pose estimation results, (e) Multi feature extraction results, (f) Classification results

C. *Comparative analysis*

This section provides the comparative analysis of the proposed 3D-YOLOv4 in which the proposed work is compared with existing methods such as P2V-CNN [40], Hough-3D [38], and RANSAC [39]. The performance of the proposed work is compared with several metrics such as accuracy (%), precision (%), recall (%), f-measure (%), computational time (ms), and ROC-AUC curve.

a. *Impact of Accuracy*

Accuracy is calculated based on the final classification accuracy results of the proposed work. The accuracy is defined as the ratio of sum of true positive and negative samples to the sum of overall samples respectively which can be formulated as,

$$\mathbb{A} = \frac{t_p + t_v}{t_p + t_v + \not{f}_p + \not{f}_v} \qquad (38)$$

Where, t_p, t_v are truly positive and negative rates while $\not{f}_p, \not{f}_v$ are false positive and negative rates respectively. From the fig 5, it is shown that when the distance increases accuracy also increases. Among that, our proposed 3D-YOLOv4 achieves higher accuracy than the existing works.

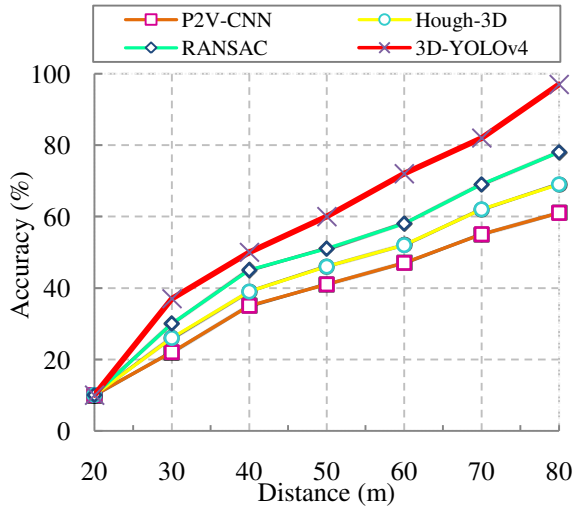


Fig. 5 Distance vs. Accuracy

Normally, accuracy is affected by inefficient feature extraction and classification that leads to high false positive rates which are overcome by extracting both low level and high level features using attention based YOLOv4 classifier. The proposed classifier has two attention layers for classifying the features of static and moving objects separately for detecting the static and moving objects whereas the existing work RANSAC used YOLO classifier for feature extraction and classification respectively which extracts only limited features and is also susceptible to localization error thereby leading to less accuracy. The proposed work achieves higher accuracy when the distance increases to 80m are 97% while the existing works P2V-CNN, Hough-3D, and RANSAC achieve accuracy of 61%, 69%, and 78% respectively. The numerical results show that the proposed work attains better accuracy than existing works. The table III represents the average accuracy of the proposed and existing methods.

TABLE III
ANALYSIS OF ACCURACY

Methods	Accuracy
3D-YOLOv4	58.25±0.5
RANSAC	48.71 ±0.3
Hough-3D	43.42 ±0.2
P2V-CNN	38.71±0.1

b. *Impact of Precision*

Precision is calculated based on the segmentation and pose estimation results of the proposed work. The precision is defined as, ratio of true positive rate to the sum of true positive and false positive rates respectively. Mathematically, the precision can be formulated as,

$$\mathbb{P} = \frac{t_p}{t_p + \not{f}_p} \qquad (39)$$

Fig 6 represents the comparison of precision rate to distance of the object with proposed and existing works respectively. From the figure, when the distance increases precision rate also increases. Among that our proposed work achieves high precision rate even though the distance increases. Generally, the precision rate is affected by ineffective segmentation a result which is due to lack of

metrics considered for segmentation. The proposed work 3D-YOLOv4 fuses the RGB-D image and LiDAR point clouds and performs semantic segmentation using IMRCNN algorithm which segments the desired part based on the point and local based sematic information. Based on the semantic information, the pre-defined bound box is provided to the segmented object. While, the existing work P2V-CNN, considers only point based information which limits them tofewer precision results. The proposed work achieves high precision rate of 95% when the distance increases to 80m whereas the existing works P2V-CNN, Hough-3D, and RANSAC achieve precision of 61%, 68%, and 80% respectively. From the numerical results, the proposed works outperform the existing works. Table IV represents the average precision rate of the proposed and existing methods.

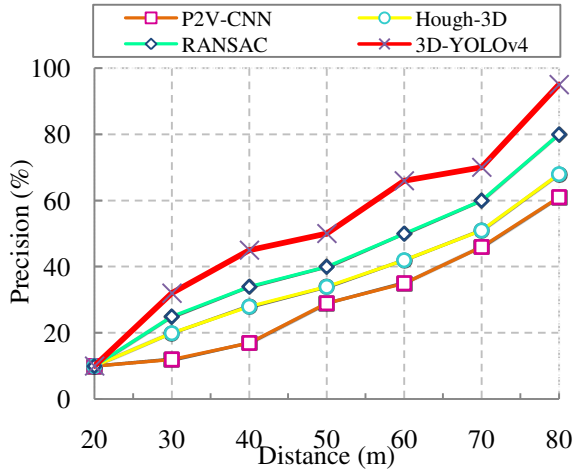


Fig. 6 Distance vs. Precision

TABLE IV
ANALYSIS OF ACCURACY

Methods	Precision
3D-YOLOv4	52.57±0.5
RANSAC	42.71±0.3
Hough-3D	36.14±0.2
P2V-CNN	30±0.1

c. Impact of Recall

Recall is calculated based on sensitivity. Recall is defined as the ratio of true positive to the sum of true positive and false negative respectively. The mathematical representation of recall rate can be formulated as,

$$\mathbb{R} = \frac{t_p}{t_p + \not{f}_n} \qquad (40)$$

Fig 7 represents the recall rate comparison of proposed work and the existing work. When the distance increases, recall rate (sensitivity) also increases from which the proposed work achieves high recall rate. The reason for achieving high recall rate is due to two-level preprocessing technique for both the input images (RGB-D image and LiDAR point cloud) using TAMF method. The proposed method reduces the noise from the environmental factor and enhance the quality of image thereby improving the recall rate (i.e. highly sensitive) whereas, the existing work RANSAC employs matching process to remove the unnecessary points and data from the camera image and LiDAR image respectively which was not efficient as the environmental factors were not considered. This would lead to less precision rate (i.e. less sensitivity to noise). The proposed work attains precision rate of 90% when the object reaches the maximum distance of 80m which is higher than the existing approaches such as P2V-CNN, Hough-3D, and RANSAC of 57%, 69%, and 77% respectively. Table V represents the average recall rate of the proposed and existing methods.

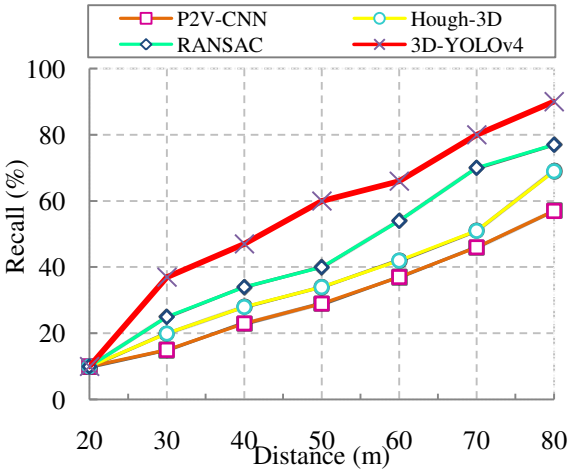


Fig. 7 Distance vs. Recall

TABLE V
ANALYSIS OF ACCURACY

Methods	Recall
3D-YOLOv4	55.71±0.5
RANSAC	42.28±0.3
Hough-3D	36.28±0.2
P2V-CNN	31±0.1

d. Impact of F-Measure

The F-measure is defined as the harmonic mean of the precision and recall which shows how precise and sensitive the result is. The F-measure can be formulated as,

$$\mathbb{F}m = 2.\frac{\mathbb{P}\times\gamma}{\mathbb{P}+\gamma} \qquad (41)$$

Where $\mathbb{P}\ and\ \gamma$ denotes the precision and recall rate respectively. Fig 8 illustrates the comparison of F-measure in terms of distance. From the figure, when the distance of the object increases the f-measure also gradually increases. From that, the proposed 3D-YOLOv4 achieves high f-measure rate even though the distance of the object gets increased. The reason for achieving high f-measure is due to two levels of pre-processing, and improved mask RCNN based segmentation. The two-level pre-processing increase the recall rate (noise sensitive rate) by performing noise removal for both inputs (RGB-D and LiDAR data) using TAMF method. The improved mask-RCNN based semantic segmentation precisely segments the objects masks and improves the precision results by considering both point and local based semantic information. While the existing works P2V-CNN

and RANSAC limits with less f-measure because the RANSAC lacks poor noise sensitivity and P2V-CNN lacks poor segmentation results. The proposed work achieves 94% of f-measure when the distance of objects increases to 80m which is higher than the existing works such as P2V-CNN, Hough-3D, and RANSAC of 60%, 69%, and 82% respectively. The table VI represents the average F-measure rate of the proposed and existing methods

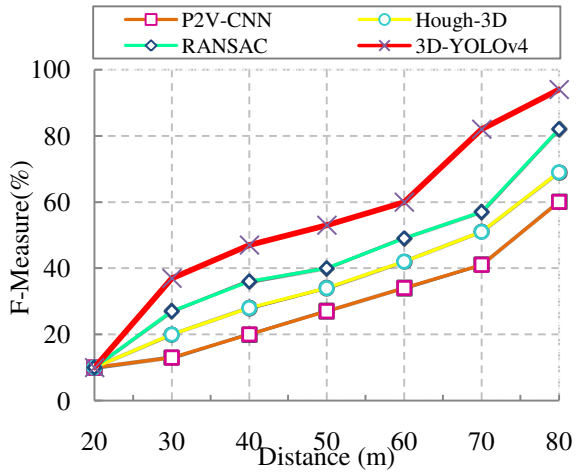


Fig. 8 Distance vs. F-Measure

TABLE VI
ANALYSIS OF ACCURACY

Methods	F-Measure
3D-YOLOv4	54.71±0.5
RANSAC	43±0.3
Hough-3D	36.28±0.2
P2V-CNN	29.28±0.1

e. Impact of Computation Time

Computation time is defined as the amount of time taken to complete the given task. A good framework must have low computation time.

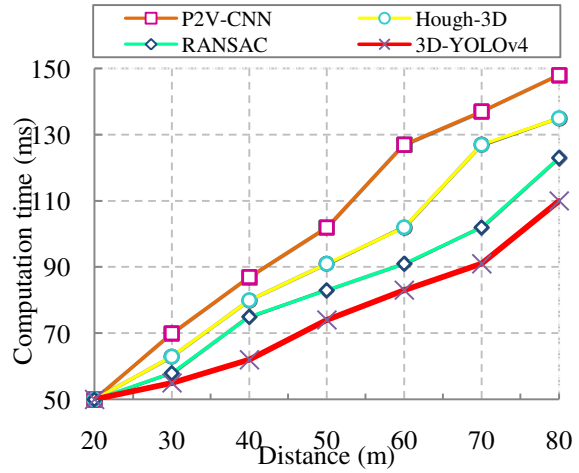


Fig. 9 Distance vs. Computation time

Fig 9 represents the computation time comparison of proposed and existing works. From the figure, it is shown that the proposed work achieves lesser computation time than the existing works even though the distance of the objects gets increased. The reason for achieving lesser computational time is due to better classification and regression methods. The proposed work achieves faster classification using attention based YOLOv4 tiny model which extracts multi features and performs classification over a short period of time than the existing works and based on the faster computational results the proposed uses attentive corner aggregation method which also reduces the computational time. The computation time of existing work P2V-CNN utilizes Mask-RCNN which consumes more time for feature extraction and classification which leads to increased computational time. The proposed work achieves a computation time of 110ms when the distance increases to 80m whereas the existing works P2V-CNN, Hough-3D, and RANSAC of 148ms, 135ms, and 123ms respectively. The above numerical results show that the proposed work outperforms the existing works. Table VII represents the average computation time of the proposed and existing methods.

TABLE VII
ANALYSIS OF ACCURACY

Methods	Computation Time
3D-YOLOv4	75±0.1
RANSAC	83.14±0.2
Hough-3D	92.57±0.3
P2V-CNN	103±0.5

f. ROC-AUC Curve

The Receiver Operating Curve (ROC) is used to represent the system's effectiveness in terms of true positive and false positive rates. The true positive rate is defined as the overall measure of correct prediction predicted by the system while the false positive rate is defined as the overall measure of wrong predictions. The accuracy (classification and segmentation) of the proposed work is determined by the ROC curve. The Area Under Curve (AUC) is used to define the area under the ROC which is a scale invariant. When ROC of the system increases, AUC also increases which provided better results.

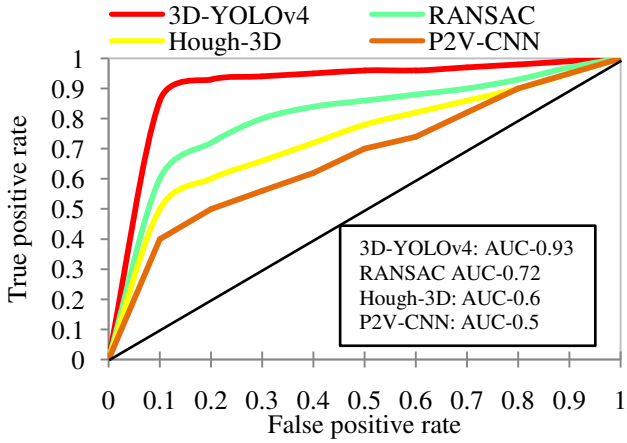


Fig. 10 ROC-AUC Curve

From the fig 10, the proposed work achieves less false positive rate with high true positive rate due to consideration of effective multiple features. The features employed in the proposed work are low level (Color, shape, texture) and high level (temporal, spatial, semantic, and geometric features) which enhances the detection accuracy thereby increasing the true positive rate and reducing the false positive rate respectively while the existing works consider only limited features (i.e. low level or high level features) for feature extraction which lacks with low true positive rate and high false positive rates. The proposed achieves better AUC under ROC of 0.86 whereas the existing works P2V-CNN, Hough-3D, and RANSAC attain lesser AUC of 0.6, 0.5, and 0.4 respectively.

D. Research Summary

This sub-section illustrates the overall performance of the proposed 3D-YOLOv4 framework. Based on the results acquired from the previous sub-section, it is clearly shown that the proposed 3D-YOLOv4 framework performs better than the existing works in terms of accuracy (58.25%), precision (52.57%), recall (55.71%), F-measure (54.71%), computation time (75ms), and ROC under AUC (0.86). The attained results are obtained through proposed methods such as two-level pre-processing (i.e. pre-processing both RGB-D and LiDAR data), improved mask-RCNN based semantic segmentation, multi feature extraction and fusion, and attention based YOLOv4 classification and effective bounding box regression. All these processes co-operatively produce better results. Some of the specific highlights of the proposed work are provided below,

- For improving the quality of the inputs (RGB-D, 3D LiDAR cloud points) we perform two-level preprocessing which removes the noise from the input using thresholding based adaptive median filtering algorithm (TAMF) and perform point to voxel conversion for obtaining better results during object detection.
- For estimating the viewport and accurately detecting the objects we perform semantic segmentation and pose estimation using Improved Mask RCNN which accurately predicts the pose of the objects with the help of affine transformation that increases detection accuracy.
- For increasing 3D object detection accuracy we perform multi feature extraction and classification which detects accurately and reduces the false positive rate
- The perspective views of the objects are accurately estimated by generating dynamic bounding boxes that increase the efficiency of 3D object detection and classification.

VI. CONCLUSION AND FUTURE WORK

The proposed 3D-YOLOv4 method is designed for detecting the 3D objects accurately using RGB-D and LiDAR cloud points. Initially, both the RGB-D images and LiDAR cloud points are preprocessed by removing the noise to enhance the image quality using TAMF algorithm. Conversion ofLiDAR cloud points to voxel is performed for achieving 3D LiDAR cloud points and fused with RGB-D images. After preprocessing of images, 3D object segmentation is performed using Improved Mask R-CNN which increases the precision and accuracy of feature extraction. Affine transformation is used to estimate the pose of the segmented 3D objects based on bearing angle.After segmenting the 3D objects, feature extraction is performed using A-YOLO V4 tiny algorithm which extracts low and high-level features with two attention modules. Classification of 3D objects is performed based on the extracted features into two classes such as static and dynamic. Finally, 3D bounding box is generated to increase the 3D object detection accuracy. The MATLAB R2020a simulation tool us used to evaluate the performance of the proposed 3D-YOLOv4 method in terms of accuracy, precision, recall, F-measure, computation time, and ROC curve. The comparative results prove that the proposed 3D-YOLOv4 method achieved high performance when compared with other existing works. In future,we will plan to improve our work to track the 3D objects with high accuracy.

Conflict of Interest

- ✓ All authors have participated in (a) conception and design, or analysis and interpretation of the data; (b) drafting the article or revising it critically for important intellectual content; and (c) approval of the final version.
- ✓ This manuscript has not been submitted to, nor is under review at, another journal or other publishing venue.
- ✓ The authors have no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript
- ✓ The following authors have affiliations with organizations with direct or indirect financial interest in the subject matter discussed in the manuscript:

'The authors declares that there is no conflict of interest'

Declaration of Interest:

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability:

No associated data is available.

REFERENCES

1. Li, W., Cheng, H., & Zhang, X. (2021). Efficient 3D Object Recognition from Cluttered Point Cloud. Sensors (Basel, Switzerland), 21.
2. Khazari, A.E., Que, Y., Sung, T.L., & Lee, H.J. (2020). Deep Global Features for Point Cloud Alignment. Sensors (Basel, Switzerland), 20.
3. Hu, H., Zhu, M., Li, M., & Chan, K. (2022). Deep Learning-Based Monocular 3D Object Detection with Refinement of Depth Information. Sensors (Basel, Switzerland), 22.
4. Weng, X., &Kitani, K. (2019). Monocular 3D Object Detection with Pseudo-LiDAR Point Cloud. 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), 857-866.
5. Pang, S., Morris, D.D., &Radha, H. (2020). CLOCs: Camera-LiDAR Object Candidates Fusion for 3D Object Detection. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 10386-10393.
6. Wen, L., & Jo, K. (2021). Fast and Accurate 3D Object Detection for Lidar-Camera-Based Autonomous Vehicles Using One Shared Voxel-Based Backbone. IEEE Access, 9, 22080-22089.
7. Wang, Y., Yang, B., Hu, R., Liang, M., &Urtasun, R. (2021). PLUME: Efficient 3D Object Detection from Stereo Images. IROS.
8. Xie, Q., Lai, Y., Wu, J., Wang, Z., Zhang, Y., Xu, K., & Wang, J. (2021). Vote-Based 3D Object Detection with Context Modeling and SOB-3DNMS. Int. J. Comput. Vis., 129, 1857-1874.
9. Xia, C., Wei, P., Wei, W., &Zheng, N. (2021). A multilevel fusion network for 3D object detection. Neurocomputing, 437, 107-117.
10. Liu, S., Jiang, H., Xu, J., Liu, S., & Wang, X. (2021). Semi-Supervised 3D Hand-Object Poses Estimation with Interactions in Time. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 14682-14692.
11. Fang, J., Zhou, D., Song, X., & Zhang, L. (2021). MapFusion: A General Framework for 3D Object Detection with HDMaps. 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 3406-3413.
12. Mohapatra, S., Yogamani, S.K., Gotzig, H., Milz, S., &Mäder, P. (2021). BEVDetNet: Bird's Eye View LiDAR Point Cloud based Real-time 3D Object Detection for Autonomous Driving. 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), 2809-2815.
13. Liu, L., He, J., Ren, K., Xiao, Z., &Hou, Y. (2022). A LiDAR–Camera Fusion 3D Object Detection Algorithm. Information.
14. Carrillo, J., &Waslander, S.L. (2021). UrbanNet: Leveraging Urban Maps for Long Range 3D Object Detection. 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), 3799-3806.
15. Feng, D., Zhou, Y., Xu, C., Tomizuka, M., & Zhan, W. (2021). A Simple and Efficient Multi-task Network for 3D Object Detection and Road Understanding. 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 7067-7074.
16. Li, Y., Yang, S., Zheng, Y., & Lu, H. (2021). Improved Point-Voxel Region Convolutional Neural Network: 3D Object Detectors for Autonomous Driving. IEEE Transactions on Intelligent Transportation Systems, 1-7.
17. Kuang, H., Wang, B., An, J., Zhang, M., & Zhang, Z. (2020). Voxel-FPN: Multi-Scale Voxel Feature Aggregation for 3D Object Detection from LIDAR Point Clouds. Sensors (Basel, Switzerland), 20.
18. Wang, Y., Fathi, A., Kundu, A., Ross, D.A., Pantofaru, C., Funkhouser, T.A., & Solomon, J.M. (2020). Pillar-based Object Detection for Autonomous Driving. ECCV.
19. Shen, X., &Stamos, I. (2021). 3D Object Detection and Instance Segmentation from 3D Range and 2D Color Images †. Sensors (Basel, Switzerland), 21.
20. Yang, Z., Sun, Y., Liu, S., &Jia, J. (2020). 3DSSD: Point-Based 3D Single Stage Object Detector. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 11037-11045.
21. Feng, M., Gilani, S.Z., Wang, Y., Zhang, L., &Mian, A. (2021). Relation Graph Network for 3D Object Detection in Point Clouds. IEEE Transactions on Image Processing, 30, 92-107.
22. Zhang, M., Rui, F., Guo, Y., & Wang, L. (2020). Moving Object Classification Using 3D Point Cloud in Urban Traffic Environment. Journal of Advanced Transportation, 2020, 1-12..
23. Wang, S., Sun, Y., Liu, C., & Liu, M. (2020). PointTrackNet: An End-to-End Network For 3-D Object Detection and Tracking From Point Clouds. IEEE Robotics and Automation Letters, 5, 3206-3212.
24. Li, Y., Ma, L., Tan, W., Sun, C., Cao, D., & Li, J. (2020). GRNet: Geometric relation network for 3D object detection from point clouds. Isprs Journal of Photogrammetry and Remote Sensing, 165, 43-53.
25. Yang, Y., Chen, F., Wu, F., Zeng, D., Ji, Y., & Jing, X. (2020). Multi-view semantic learning network for point cloud based 3D object detection. Neurocomputing, 397, 477-485.
26. Hu, F., Yang, D., & Li, Y. (2019). Combined Edge- and Stixel-based Object Detection in 3D Point Cloud. Sensors (Basel, Switzerland), 19.
27. Wang, Z., Fu, H., Wang, L., Xiao, L., & Dai, B. (2019). SCNet: Subdivision Coding Network for Object Detection Based on 3D Point Cloud. IEEE Access, 7, 120449-120462.
28. Tian, Y., Song, W., Sun, S., Fong, S., &Zou, S. (2019). 3D object recognition method with multiple feature extraction from LiDAR point clouds. The Journal of Supercomputing, 1-13.
29. Wang, B., Zhu, M., Lu, Y., Wang, J., Gao, W., & Wei, H. (2021). Real-Time 3D Object Detection From Point Cloud Through Foreground Segmentation. IEEE Access, 9, 84886-84898.
30. Jo, K., Lee, S., Kim, C., &Sunwoo, M. (2019). Rapid Motion Segmentation of LiDAR Point Cloud Based on a Combination of Probabilistic and Evidential Approaches for Intelligent Vehicles. Sensors (Basel, Switzerland), 19.
31. Arnold, E., Dianati, M., & Temple, R.D. (2019). Cooperative Perception for 3D Object Detection in Driving Scenarios using Infrastructure Sensors. ArXiv, abs/1912.12147.
32. Ye, Y., Chen, H., Zhang, C., Hao, X., & Zhang, Z. (2020). SARPNET: Shape attention regional proposal network for liDAR-based 3D object detection. Neurocomputing, 379, 53-63.
33. Zhuang, C., Wang, Z., Zhao, H., & Ding, H. (2021). Semantic part segmentation method based 3D object pose estimation with RGB-D images for bin-picking. Robotics Comput. Integr. Manuf., 68, 102086.
34. Yoo, J.H., Kim, Y., Kim, J.S., & Choi, J. (2020). 3D-CVF: Generating Joint Camera and LiDAR Features Using Cross-View Spatial Feature Fusion for 3D Object Detection. ECCV.
35. Chiang, C., Kuo, C., Lin, C., & Chiang, H. (2020). 3D Point Cloud Classification for Autonomous Driving via Dense-Residual Fusion Network. IEEE Access, 8, 163775-163783.
36. Wang, Z., Xia, Qi., Du. J., Huang. Sha., Su, Jin., Junior, Jo., Li, Jo., Cai.Gu. (2021). 3D MSSD: A multilayer spatial structure 3D object detection network for mobile LiDAR point clouds. Elsevier.
37. Wang, L., Zhao, D., Wu, T., Fu, H., Wang, Z., Xiao, L., Xu, X., & Dai, B. (2020). Drosophila-Inspired 3D Moving Object Detection Based on Point Clouds. Inf. Sci., 534, 154-171.
38. Yan, M., Li, Z., Yu, X., & Jin, C. (2020). An End-to-End Deep Learning Network for 3D Object Detection From RGB-D Data Based on Hough Voting. IEEE Access, 8, 138810-138822.
39. Weon, I., Lee, S., &Ryu, J. (2020). Object Recognition Based Interpolation With 3D LIDAR and Vision for Autonomous Driving of an Intelligent Vehicle. IEEE Access, 8, 65599-65608.
40. Li, J., Sun, Y., Luo, S., Zhu, Z., Dai, H., Krylov, A., Ding, Y., & Shao, L. (2021). P2V-RCNN: Point to Voxel Feature Learning for 3D Object Detection From Point Clouds. IEEE Access, 9, 98249-98260.