OPEN ACCESS

Kashf Journal of Multidisciplinary Research
Vol: 02 - Issue 4 (2025)
P-ISSN: 3007-1992 E-ISSN: 3007-200X
https://kjmr.com.pk

# SAFE AND EFFICIENT PEDESTRIAN DETECTION FOR AUTONOMOUS VEHICLES THROUGH ADVANCED 3D CNN-BASED SOLUTIONS

***Rabia Tariq***
*Institute of Computing, Muhammad Nawaz Shareef University of Agriculture, Multan.*
***Sadia Latif****
*Department of Computer Science, Bahauddin Zakaria University, Multan, Pakistan.*
***Rana Muhammad Nadeem***
*Department of Computer Science, Govt. Graduate College Burewala, Pakistan.*
***Muhammad Ans Khalid***
*Department of Computer Science, University of southern Punjab , Multan, Pakistan.*
***Hafiz Muhammad Ijaz***
*Department of Computer Science, University of southern Punjab , Multan, Pakistan.*

***Corresponding author: Sadia Latif (sadialatifbzu@gmail.com)***

## Article Info

## Abstract

Pedestrian detection is another significant special application of object detection in autonomous vehicles. In contrast to universal object detection, it has similarities and special traits. Nevertheless, there are some difficulties that influence pedestrian detection performance, namely (i) occlusion and deformation (ii) low-quality and multispectral images consisting primarily of lighting conditions, small-scale detection, and target detection extensively, and (iii) true-false pedestrians. Deep learning (DL) methods are a class of artificial intelligence method that can solve the issues mentioned above of pedestrian detection. This paper initially gives an elaborate description of pedestrian detection, difficulties in pedestrian detection, and latest advancements in solving them using the assistance of DL methods with informative discussions, aiming to provide insights to the readers. (2) A new pedestrian detection algorithm (PDA) of true/false pedestrian is suggested here, in which a new YOLO-3D CNN model is applied to reject true/false pedestrian. The primary purpose is to evaluate the performance of the existing 3D CNN taking into consideration the problem of rejecting true false pedestrians based on images captured using the car's onboard cameras and light detection and ranging (LiDAR) sensors. PDA initially utilizes YOLOv3 to capture the entire image for training detector model capable of real-time forecasting. Next, as a feature extractor, it utilizes the MobileNet-SSD that provides great accuracy as well as good trading uptime. PDA then implements the Faster R-CNN method to detect different parts of the object, over the convolutional layer. Lastly, data augmentation techniques are applied in PDA to augment the data coverage by fully exploiting available training data. Simulation results indicate that the proposed pedestrian detection model and PDA improve the accuracy of real and false pedestrians while maintaining real-time requirements.

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license https://creativecommons.org/licenses/by/4.0

**Keywords:**
*Pedestrian Detection, Autonomous Vehicles, Deep Learning, Faster R-CNN, YOLOv3, 3D CNN, Data Augmentation.*

## 1. Introduction

Pedestrian detection is the primary function in autonomous vehicles. The primary feature of autonomous vehicles is carrying people or goods to a specific destination without any human controlling the vehicles. Hence, there is a need to detect pedestrians, cars, etc. correctly in real-time to build the correct control decision making to maintain safety. Autonomous vehicles are gradually becoming the transport of the future. Nonetheless, to achieve fully automated operation, several challenges still need to be tackled. One of the main goals that are currently being pursued is a very accurate scene understanding and pedestrian detection. Pedestrian detection is a computer vision method and the most significant task in autonomous vehicles utilized for the detection of human movement in the way, useful in the safety of the individuals, identification and tracking of the offender in the crowd, prevention of accident and collision of moving vehicles and objects. These recognitions can be performed with the assistance of sophisticated integration of sensors such as Radar, Camera, and LiDAR. During the past few years, a system called ADAS (Advanced Driving Assistance System) has been proposed which assists in the prevention of the occurrence of unforeseen accidents. This system supports numerous aspects to substructure various tasks such as commuter safety, environment, and drivers. Pedestrian detection is one of the proven features in it. Later, engineers incorporated this feature in autonomous vehicles. Still, with this feature pedestrian detection is plagued with many problems that should be tackled. With various innovations, numerous researchers attempt to address these problems. These hard issues include bad obstacle detection under various states of lightning such as clear vision problems during nighttime hours, occlusion scenarios, low resolution, occurrence of small sizes, tracking and identification of pedestrian [13], [14], etc. These issues sort out with the assistance of various techniques, Figure 1-5 illustrates the quantity of papers on pedestrian detection from 2020 to 2024.

Initially, traditional methods were utilized as machine learning methods owing to their unprecedented performances between the years 2005 and 2015 but then between 2015 and 2017 scholars turned towards the new approaches called "hybrid" because of characteristics which weren't hand-picked by the feature extraction step, this approach also offers us the best output but yet again it's also affected by the same drawback of the previous one, i.e., the feature isn't hand-picked. Now deep learning (DL) is widely applied over the previous traditional algorithms because of the splendid performance results and the skill to be attained. M. Jones and Viola improve real-time detection capability and effectiveness with the world-renowned VJ infrared [15]. Romero and Antonio [16] have mainly described the DL algorithms but some of them were actually defined and they never presented the richness and clarity in the features of the design, for example, the process and databases utilized by it, behaviour problem and results obtained. Numerous algorithms were found to be identified by the pedestrian tasks; for example, Haar was presented in 2000 by Poggio and Papageorgiou. It can display the variation in the gray level of the image, which includes four types: border function, line function, central environment function, and special diagnostic line functions. Haar is the basis of pedestrian detection automation, which other than Haar and histograms of oriented gradients (HOGs) [17], started as this approach classified the target by getting functional information from the image by edge direction distributions [18].

**i.** Besides, classification is implemented with support vector machines (SVMs). Furthermore, Zhang et al. presented a new feature set based on the AdaBoost classifier named Shapelet for detecting pedestrians [19]. The original framework of the traditional approach is shown in Figure 1-6. Traditional detection approach has been used in artificial feature creation and classification. First, features must be extracted from the image, knowing the gray-scale, border, complexion, gradient histogram, and other information for the target. Then, the classifier's role is to identify what attributes are associated with the pedestrians. In addition, there are two ways in which traditional techniques address the core three pedestrian problems i.e. (i) occlusion; (ii) low-quality images and multi-spectral images issue; and (iii) cannot separate poster person and actual. First of all, the objects are divided into various components and the exposed component can calculate the locations of pedestrians. Second, pedestrians are educated on a particular general classifier to reduce the impact of disruption on daily life and estimate the pedestrian location with caution. But from the year 2022, the role of conventional approaches on different sets of data such as the Caltech dataset started declining just because of the advent of advanced technology such as DL and hybrid technology.This paper is directed towards designing and implementing actual pedestrian discovery through DL technology to detect pedestrian rejections.

**i.** In this work, our aim is to forecast the performance of the current 3D convolutional neural network in rejecting authentic false pedestrians from images taken by the onboard cameras and light detection and ranging (LiDAR) sensors of the vehicle.

**ii.** We evaluate the single-phase (YOLOv3 models) and two-phase (Faster R-CNN) DL meta-structure under distinct image resolutions and attribute extractors (MobileNet).

**iii.** To solve the issue, there is a need to employ the data augmentation method in order to improve the performance of the framework. In the observation of the performance, the techniques used are applied on the recent datasets.

**iv.** The proposed algorithm is tested using the KITTI and Waymo benchmarks, where 110 random samples were annotated to verify the real pedestrian with a learning rate of 0.5-0.9. Experimental results indicate that the proposed method/algorithm enhances the detection accuracy of true and false pedestrians, and also continues to be under real-time constraints.

## Related work

Pedestrian classification/perception is excellent research and technical topics in automotive industry and scientific community, since pedestrian level of newest pedestrian recognition taken in self-driving systems, advanced-driver assist system, and car shield systems. Within research [29] there is probable demonstration regarding the application to regard the integration of a four-component approach in pedestrian detection. These components are feature extraction, deformity processing, occlusion processing, classification etc. Then learning how to create their strongest strength is delivered by the co-operative process, which boosts pedestrian detection accuracy accuracy. Rajesh and Ragish [30] carried out another study. The complete overview includes the certain requirement for ADAS structure. To detect pedestrians, they shielded the deep learning (DL) and traditional approaches in which various matrices were assessed.

For pedestrian algorithm detection, provide the trends and future work tips. However, the stated DL algorithms were not adequate such as Recurrent Neural Network (Long short-term memory), encoder and decoder framework, and objects were not stated. On CityScape and Caltech datasets the models were tested and trained. But with time passing Convolutional neural networks have achieved enormous success in detecting common objects on networks, i.e., on MS COCO datasets [31], Pascal datasets, and ImageNet [32]. Li et al., in 2018 introduced the situation analysis framework RCNN based on the perception hypopaper [33], which effectively improved the performance of pedestrian detection on different scales Now CNN extended further to explain the fundamental challenges of pedestrian such as rejecting true false, occlusion manipulate by label the different parts of body, low-resolution quality images and multispectral images such as color, RGB images, thermal images, simultaneous facts as a whole. Early application of studies, pedestrians detection rules are based on RCNN design on high-quality external suggestions to improve performance [34]. More recently, Faster-RCNN [35] is the de facto standard framework, this allows for end-to-end learning. In the meantime, something companies use non-standard architectures to provide good results, such as MS-CNN [36] and SA-Fast RCNN [37]. Once again, an appropriate adjustment of vanilla Foster RCNN [38] gives state-of-the-art pedestrian detection results. Thus, we adopt [39] and employ the modified Faster RCNN.

Kidono has pioneered a pedestrian detection system that relies on 3D LiDAR data, as detailed in reference [40]. Their algorithm begins by identifying 3D points of interest and filtering out ground-level traffic. For remaining 3D points, the system forms clusters, with each cluster ideally representing a single object, particularly when the object is further away. Potential targets that don't align with typical pedestrian height are then discarded. To further refine the identification process, nine distinct features, based on the characteristics and intensity of the 3D point clusters [41], are manually extracted. Subsequently, each clustered 3D object is classified as either a pedestrian or not using a Support Vector Machine (SVM) trained on these extracted features. In a related study [42], researchers explored pedestrian identification using data from different sensor modalities, specifically visible light and far-infrared (FIR) imagery, along with various types of information derived from this data, such as point density, depth information, and movement patterns.

The core of the research in [42] involves leveraging publicly available databases and extracting features from both visible light and far-infrared imagery, capturing spectral information from two distinct cameras. Within this framework, directional gradient histograms (HOG), local binary patterns (LBP), intrinsic shape signatures (ISS) for density similarity, and local gradient patterns (LGP) were employed to derive relevant object characteristics. This research is built upon both individual methodologies and a combined approach. Notably, the far-infrared (FIR) based pedestrian grouping method demonstrated superior performance compared to other techniques explored in the study.

The past decade has witnessed significant strides in the field of computer vision, largely driven by the rise of deep learning-based methodologies. The increasing computational power of modern Graphics Processing Units (GPUs) has enabled researchers to develop very deep convolutional neural networks, which have proven to be highly effective for extracting meaningful information from images. For the specific task of object detection, Convolutional Neural Networks (CNNs) have also become the dominant approach in academic literature. The field of Artificial Intelligence has recently experienced a substantial surge in progress, largely fueled by the development of sophisticated deep learning algorithms [43]. Deep neural networks, and particularly convolutional neural networks (CNNs), have become foundational for lower-level tasks such as object recognition and classification due to their ability to effectively utilize contextual information within the data.

Hamid and colleagues [44] investigated the impact of combining manually engineered features with features automatically learned by a Convolutional Neural Network (CNN) framework. Their findings indicated that this fusion led to improved performance on the JAAD database [45]. The extent of the improvement, however, was influenced by whether recordings were considered from before or immediately preceding a transition event. This approach achieved an accuracy of 91%.

Abughalieh et al. [46] developed an integrated system combining a CNN with an in-car camera sensor capable of predicting pedestrian orientation and the distance to the vehicle. The system identifies body features in the 2D image space and then maps them to 3D space using comprehensive statistical analysis. These statistics are continuously tracked for each pedestrian, and variations in pedestrian movement are used to alert the driver. While pedestrian datasets exhibit diverse structures, they generally contain the same core object classes. Despite the inherent challenges in pedestrian detection, the application of 2D object detection techniques for self-driving cars has seen a significant boost in effectiveness. These methods have achieved an average precision (AP) of over 90% on the widely recognized KITTI object detection benchmark [47].

While 2D techniques analyze features within the confines of a flat image, 3D techniques introduce a third dimension to the positioning and mapping of pixels, thereby uncovering more intricate spatial information within the real-world coordinate system. However, in the context of Autonomous Vehicles (AVs), a significant disparity persists between the practical application of 2D and 3D techniques [48]. Further dedicated effort is essential to bridge this gap in the utilization of 3D techniques, as a thorough understanding of 3D spatial relationships is crucial for effectively addressing various challenges in autonomous driving.

To rigorously evaluate detection performance, the most challenging detection scenarios should be analyzed using a robust and deeply integrated feature extraction backbone, rather than relying on shallower or less comprehensive feature representations. Furthermore, employing ResNet instead of VGG as the backbone architecture for models like Faster R-CNN is recommended to maximize performance. ResNet's powerful architecture enables it to learn more complex features, ultimately leading to higher detection accuracy.

## 2. Methods and materials

The methodology detailed in this section is designed for implementation in autonomous vehicles equipped with both LiDAR sensors and color cameras. Our object recognition model will leverage the MobileNet architecture for feature extraction and will be trained using the TensorFlow framework on diverse datasets, including Waymo and KITTI. To enhance the model's robustness and generalization, we will employ various data augmentation techniques such as adjustments to object positioning, rotations, horizontal and vertical flips, and color variations. This study aims to demonstrate that applying machine learning with Tensor Flow to the task of real-time object detection can achieve outstanding results. During the model training process, we will explore different architectural paradigms, specifically Faster R-CNN, YOLOv3, and MobileNet-SSD.

### 1.1.Deep Object Detection

Object detection serves as a crucial functionality, enabling the identification and localization of multiple objects within a given scene. Its performance is commonly evaluated based on classification accuracy and the precision of the generated bounding boxes. The field has witnessed the rise of deeper learning methodologies, with performance benchmarks established on widely recognized object identification datasets such as PASCAL VOC and COCO. These benchmarks are frequently used in the context of autonomous driving to assess the detection capabilities for critical elements like traffic lights, road signs, pedestrians, and vehicles.Object detection encompasses various specific domains, including face detection and pedestrian detection. At its core, an object detection algorithm relies on a feature extractor to identify the presence of objects. Contemporary deep object detection networks typically follow one of two main pipeline architectures: two-phase or one-phase detection. Our focus here is on image-based detection.

Figure 1 illustrates a step-by-step process for pedestrian detection. Leveraging the initial awareness of objects within an image, most pedestrian detection systems proceed through four sequential stages. The first stage, proposal generation, involves creating initial candidate regions or concepts from the input image. The second stage, feature extraction, focuses on outlining the characteristics of these candidate proposals generated in the first step. Various techniques, ranging from manually designed features to features learned through deep neural networks, have been employed in this stage. The third stage, classification (and filtering), is dedicated to categorizing the proposed regions as either belonging to the target class (pedestrian) or as background. Finally, the post-processing step is designed to refine the detection results by reducing redundant bounding boxes that might have been predicted for the same individual pedestrian.

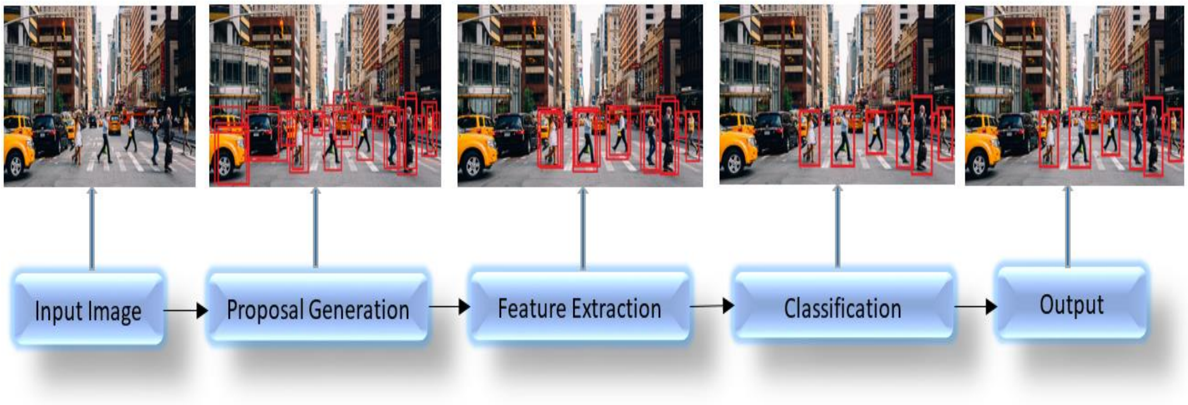


**Figure 1: Shows an example of pedestrian detection steps**

### 1.2.2D and 3D Object Detection

Numerous tasks have been addressed within the realm of 2D object detection, focusing on the analysis of information projected onto a 2D image plane [140]. However, inspecting objects in 3D space presents greater challenges compared to its 2D counterpart, primarily due to the varying distances between the object and the observing vehicle. Consequently, accurate depth information, such as that provided by LiDAR sensors, becomes highly valuable. Some research efforts [141] have explored combining data from RGB cameras and LiDAR point clouds to enhance 3D object detection. Furthermore, Liang et al. [142] proposed a multi-task learning framework to improve 3D object detection by incorporating auxiliary tasks like camera depth completion, ground plane estimation, and 2D object detection.

The increasing availability of 3D sensors and the growing number of 3D intelligence applications have propelled 3D object detection into a prominent area of research. Unlike image-based 2D detection, LiDAR point clouds offer robust spatial information that can be leveraged to accurately locate objects and delineate their shape in three dimensions. Notably, LiDAR-based 3D object detection technology generally surpasses the performance of its 2D counterparts. In this chapter, we will focus on 3D object detection. The fundamental goal of 3D object detection is to predict the 3D characteristics of objects within a driving environment based on sensory input. This process can be implicitly formulated as:

$$\mathcal{U} = E_{det}(I_{sensors}) \quad (1)$$

Where,

$\mathcal{U} = (U_1, \dots., U_N)$ is the set of N 3D objects in a background

$E_{\text{det}}$ represents the 3D object detection design

$\mathrm{I}_{sensors}$ is one or more sensory input

In most context, a 3D object is illustrated as a 3D cube-shaped that carry this object, that is as shown in Figure 2.

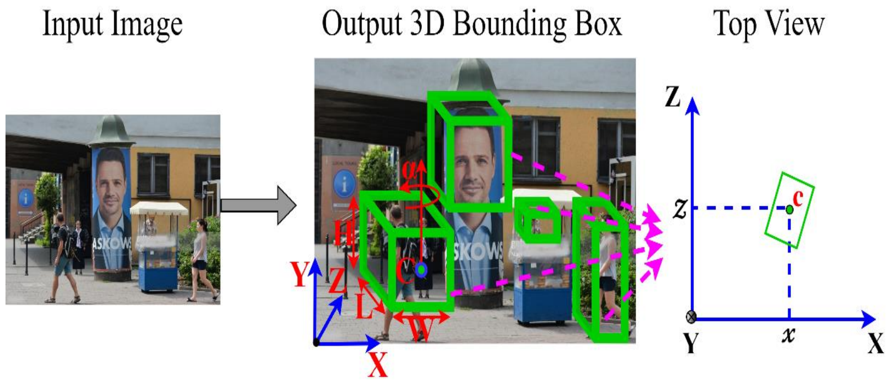


**Figure 2 Formation of 3D object detection work**

The detail of these input is presented in Table 1.

**Table 1 Description of variables for 3D bounding boxes**

| Input | Description |
|---|---|
| x, y, z | Represent the predicted location of the 3D bounding box |
| h, w, l | Represent the dimensions |
| $\boldsymbol{\theta}$ | Represent the orientation of every object |
| c (class) | Denotes the group of 3D object |

### 1.3.Framework of PDA

The primary and easy steps of Pseudo-code for pedestrian detection model is provided in Algorithm 4-1 and the structure is shown in Figure 4-4.

"The initial step involves acquiring the necessary model files corresponding to the datasets we are using, namely Waymo and KITTI. Secondly, we will initialize our model's parameters, setting default values for inputs and a threshold for subsequent processing. Following this, the third step entails loading our pre-trained model and its associated class labels, after which our model's input will be read.

For instance, if our chosen model architecture is YOLO v3, the network loading process consists of two key parts. The first part involves loading the pre-trained weights using a yolov3.weights network loader. The second part requires loading the network's configuration file using a yolov3.cfg network loader. To encapsulate this process, we will create a function named load_yolo().

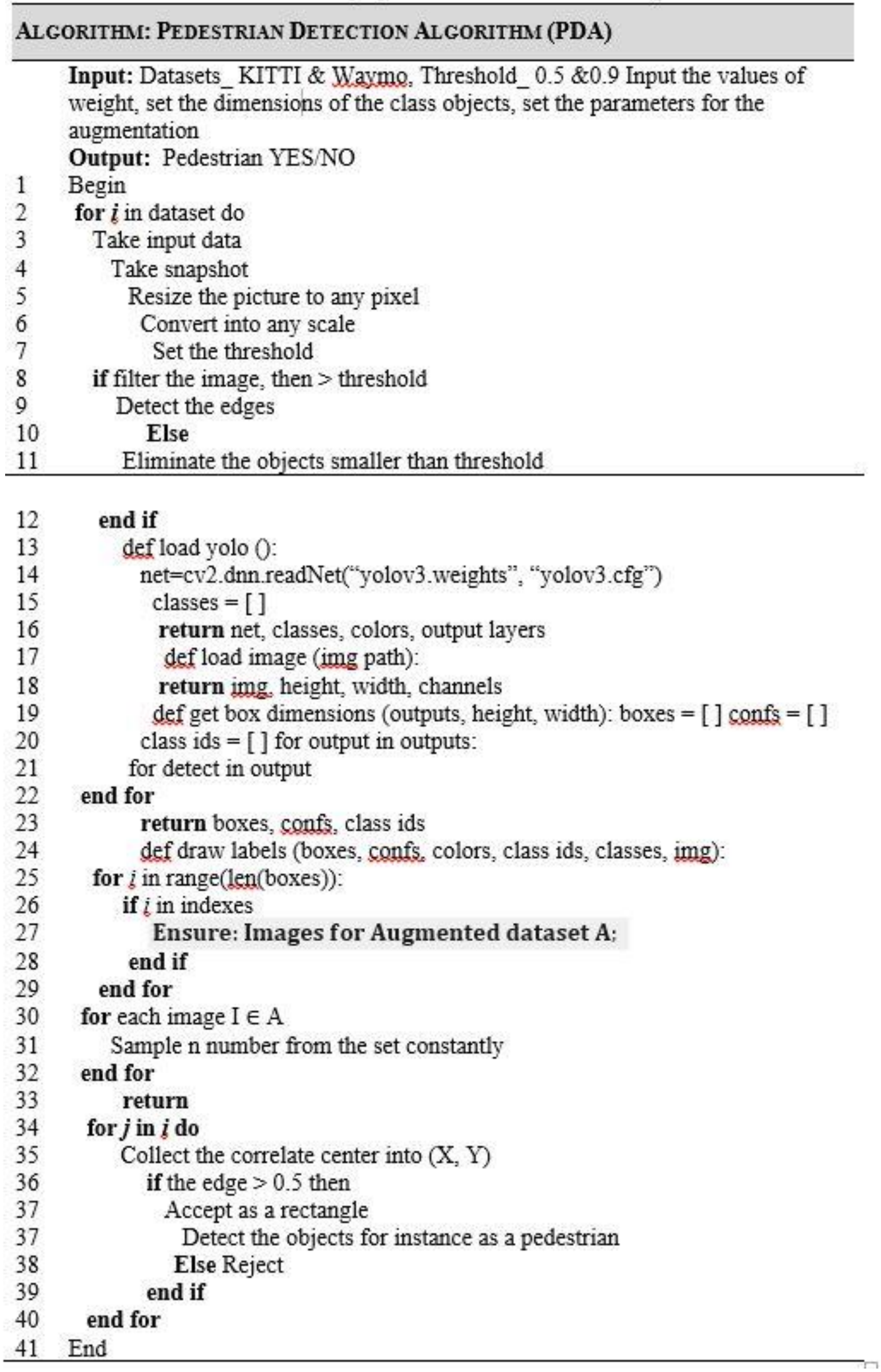
Building upon the third step, we will design a load_image() function. This function will be responsible for reading image files, resizing them to the required dimensions, and returning the processed image.

To visualize the detection results, we will use bounding boxes to predict the objects in the image. This will be achieved by calling a function named box_dimensions to obtain the coordinates of the bounding boxes and a label function to assign names to the detected objects.

To mitigate the risk of reinforcing our model with unsuitable background samples, we will apply our proposed architecture, incorporating an augmentation outlet. This outlet will reshape the image results and pre-process the output components of the bounding boxes, converting them into x and y center coordinates. This pre-processing step is crucial for further performance analysis and evaluation, ultimately leading to the desired bounding box output on the actual image.

The fundamental advantages of our proposed PDA and system model are as follows:

- A key strength of this approach lies in its ability to detect pedestrians within a defined area of interest. This focused detection enhances temporal performance, allowing for quicker delivery of results to the driver or the vehicle's onboard computer. Consequently, it eliminates the prior necessity of assessing the level of hazard for automated operation."
- The other benefit is that the false positives are significantly minimized, thus making the system robust enough to take over the irreversible security system and detect the actual pedestrian.



**ALGORITHM: PEDESTRIAN DETECTION ALGORITHM (PDA)**

**Input:** Datasets_ KITTI & Waymo, Threshold_ 0.5 &0.9 Input the values of weight, set the dimensions of the class objects, set the parameters for the augmentation

**Output:** Pedestrian YES/NO

```
Begin
 for i in dataset do
    Take input data
       Take snapshot
          Resize the picture to any pixel
            Convert into any scale
              Set the threshold
    if filter the image, then > threshold
        Detect the edges
             Else
           Eliminate the objects smaller than threshold
     end if
        def load yolo ():
           net=cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")
             classes = [ ]
              return net, classes, colors, output layers
               def load image (img path):
              return img, height, width, channels
             def get box dimensions (outputs, height, width): boxes = [ ] confs = [ ]
           class ids = [ ] for output in outputs:
         for detect in output
   end for
           return boxes, confs, class ids
           def draw labels (boxes, confs, colors, class ids, classes, img):
    for i in range(len(boxes)):
        if i in indexes
            Ensure: Images for Augmented dataset A:
         end if
     end for
   for each image I ∈ A
       Sample n number from the set constantly
   end for
        return
   for j in i do
        Collect the correlate center into (X, Y)
            if the edge > 0.5 then
               Accept as a rectangle
                  Detect the objects for instance as a pedestrian
                 Else Reject
            end if
    end for
End
```

"Pedestrian detection algorithms generally adhere to the structural framework illustrated in Figure 3. The initial step involves the sensor system capturing data, typically in the form of images. Following this, a region proposal technique is employed as the second step. Region of Interest (ROI) proposals, a common visual approach used with cameras and stereo vision, serve as the foundational element for subsequent system tracking. Edges, lines, and shapes are identified and then fed into classifiers to determine the category of a potential target (e.g., whether it is a person or not). These ROIs represent candidate areas within the image where pedestrians might be present.

To generate these ROIs, various techniques are utilized, including the sliding window approach, Locally Decorrelated Channel Features (LDCF), and selective search. Subsequently, in the third step, features are extracted from these identified ROIs. For object detection, algorithms employing either manually engineered features or Deep Learning (DL)-based object detection techniques are used for this feature extraction and subsequent classification.

Hand-crafted feature extraction techniques are applied to models that rely on low-level image features to manually define ROIs. However, these handcrafted approaches can be limiting and may not be sufficiently robust, as complex features are challenging to design manually. In contrast, DL methods empower the network to automatically learn and specify relevant features. This often leads to superior levels of feature extraction. Finally, in the last step, these extracted features are fed into a classifier to make the final prediction about the presence and location of pedestrians."

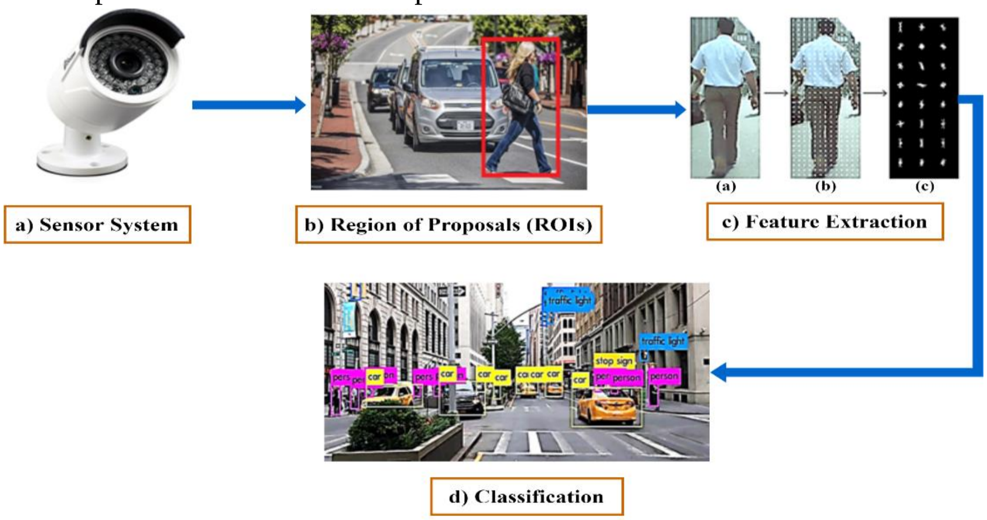


**Figure 1 Primary detection system framework of pedestrian**

The output feature produced by the subtraction step is provided as input to the classifier in an attempt to determine whether pedestrians or other obstacles in terms of binary labels are present inside the recommended region. However, since the advent of DL, more numbers of CNN-based sequence methods are being adopted for classification as opposed to using conventional classifiers, e.g., AdaBoost and SVM.

### 1.4. Dataset and Pre-processing

For training and testing 3D CNNs Recommendation, KITTI and Waymo dataset is selected. This dataset provides the given data of multiple sensors. Gives 16 beam LiDAR readings, 360° image generated by 6-color camera, highly accurate GPS position and inertia information. The coordinate system of multiple sensors is recorded, so it's an easy task to transform the data to another sensor. All these aforementioned devices are mounted on an electric vehicle. So, this dataset contains a lot of features of different driving conditions such as busy roads, pedestrians and common car parks and underground parking. It also depicts different thunder-light and weather conditions. This approach is making use of LiDAR and front color camera. We utilized YOLOv3 eliminates them automatically since the dataset does not provide the object labels. This is standard in machine learning approach known as pseudo-labeling.

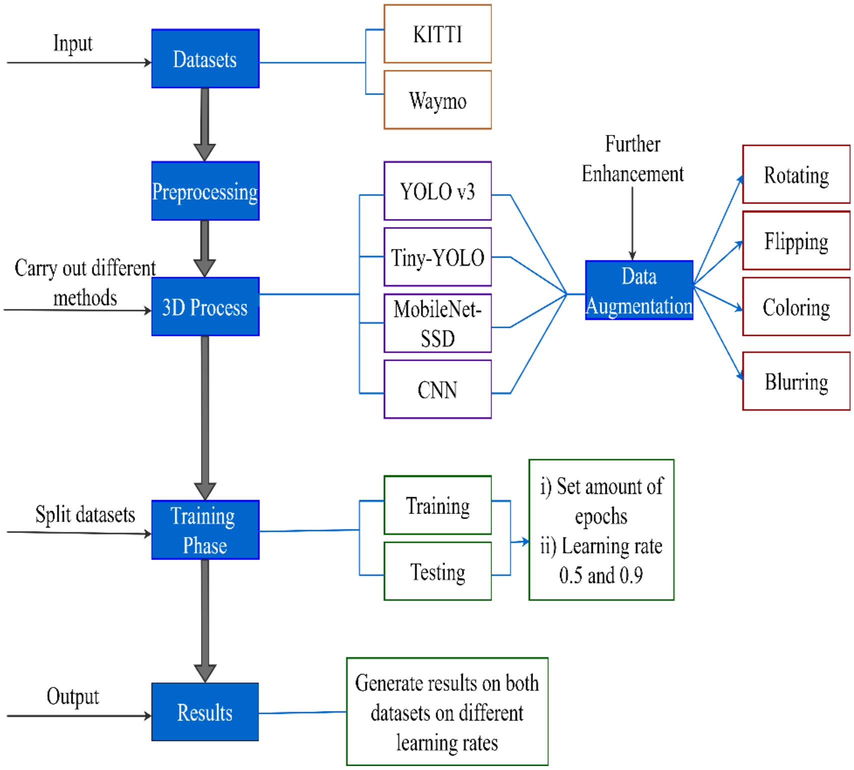


**Figure** Error! No text of specified style in document. **Basic steps for procedure exploration**

Actually, we are executing the strategy described above. We simply analyze the odd-numbered voxelization techniques of the ranks 1 through 19 of KITTI and Waymo dataset. Consequently, we have instances from pedestrian training once again, we generate 5 randomly to fill up the non-pedestrian class of the fake AOI in the input image. Though within the image the random sizes and random positioning crossroads are not needed for pedestrian sensing. later on all the AOIs are handled like a pedestrian AOI. finally the dataset has over 24,000 pairs of sample images and corresponding point clouds. Data set is balanced so both classes incur equal number of samples. For the training and testing the dataset is divided into 70% and 30%. It is important to note that this event was considered. The action range is 10m because of the 16 beam LIDAR sensor. Do not give us so much 3D information beyond this threshold, too few dots above 10 meters. In this way, all samples lie above the 10m threshold which is rejected.

### 1.5. 3D Process

Although our method work is getting closer to pedestrian detectors, we have three options and we select YOLO v3 since the first one is the most accurate method to locate an object of original model. It offers the capability to detect multiple objects for pedestrians. A single neural network cooperates with the provided YOLO v3 to the entire image. The network divides images among regions. It estimates every person area bounding boxes along with the probabilities. The bounding boxes are weighted by estimation possibility. Over the classification system, this model has so many benefits. Seeing the whole picture in the test, therefore time estimates its forecasts from a global perspective. The image also uses the same mesh for the prediction and unlike other systems requires thousands of reviews picture. This is one of the top-performing architectures regarding the object detection task. Then we also explored the Tiny-YOLO. There is a lot of learning apparently acquired in YOLO v3 because it has the attribute of architecture as there are three regional regression networks in three different regions at the depth level. Tiny-YOLO does the same YOLO v3 scheme until the first region sends it back to the network. By this, it is quite faster compared to the vanilla YOLO, but accuracy somehow is weaker too. Finally, we also tested MobileNet-SDD. This model is a good idea of SSD technique. SSD method for object location image with single deep neural network. Asymptotically the adjustment of the bounding box of the output region towards set of defaults. Cells of varying aspect ratio and scale based on the feature map while estimating space, the network comes up with a score. The existence of each object class in every default box alters the boxes to fit better. Besides the object's shape, the network also supports predictions. From several feature maps with varying resolutions naturally capture objects of any size.

The SSD method is less complex than methods requiring object advice since it completely eliminates the proposal generation and the subsequent pixel or feature is the sampling stage and condense the all-numbers data processing within a single lattice.

This method can be adapted to various convolutional networks applications, that's why we select the mobile network as the backbone. Mobile networks are a collection of optimal models known as mobility and embedded image applications. On the basis of mobile network in a silky architecture using depth separators gentle bending for constructing deep neural networks. We introduce two simple global hyper parameters which is trade delays and accuracy effectively. These hyperparameters allow the model creators to make the correct decision regarding the restricted app-sized model issue So, through the combination of SSD and MobileNet we have MobileNet-SSD, both are correct and swift. Look at these object detectors because they are providing great accuracy or decent trading uptime and accuracy. Additionally, any object detector could be suitable to this framework.

Finally, the proposed pipeline will take as input the potential object, investigation suggested pedestrians and ultimately tell whether they are pedestrians or no pedestrians. Actually, this section takes the volume of the created outcome voxel lattice volume within each 3D points. The 3DCNN scheme is another point net-based architecture. It's an ethical 3D convolutional neural network which possesses a network size and is able to classify these among the various 10 classes. The original form of the purpose architecture is to operate with artificial data sets or tools such as a connection device that provides high-compactness point clouds. In order to connect your network to the existing issue We have done a number of changes to the original PointNet. On further, the number of input voxel has been reduced by 30 × 30 × 30 filter size to 25 × 25 × 25 voxels for surpass introduce 3D data even though they are very sparse. Similarly, the depth of the system has also been decreased so that cartographical features are not lose due to falling essence of convolutions. The most recent alteration was the following on cell residents of voxel lattice volume size. The point net original displays each cell density of process point cloud. However, we altered it for a dual covered network one. In our demo, all cells are programmed with points in 3D space matching for entry point cloud. Thus, the central suction process is the result of 25 × 25 × 25 voxels, 1 in solitary confinement means that the concerned place has provided a score and 0 means there are no points in the concerned places. Finally, the endmost fully connected was also substituted as per our problem. That is, the output neuron count Increasing from 10 to 2 which convey to the existence of the sample and thus approved the pedestrian or no pedestrian. Ultimately the 3D-CNN gives the following configuration:Get 25x25x25 voxels lattice as an input layer

- 3D Convolutional Layer attributing 100 filtrate of dimensions 3x3x3
- Get 50% likelihood as a dropout layer.
- 3D Convolutional Layer attributing 100 filtrate of dimensions 2x2x2
- Get 50% likelihood as a dropout layer.
- Resulting 2 output neurons by fully connected layers.

### 1.6. Process of Data Augmentation

In the majority of the computer vision tasks, data augmentation play a leading role. For pedestrian detection, we introduce to employ a search space of data augmentation with different hyper-parameters. A new data augmentation strategy can well enable training of data changes by efficiently transferring people from other data sets to victim scenes. Apart from the shortage of algorithms, inadequate training sample coverage is another key reason for poor detection performance, particularly for deep learning. Practically speaking, it is impossible to capture enough pedestrian samples. Thus, numerous researchers make use of data enhancement techniques in order to boost data coverage fully leveraging available training data. Style Delegate and Pedestrian embedding are two-phase data augmentation program. Pedestrian Embedding: Take out the pedestrians from other datasets and arbitrarily embed them into the victim view.

- Style Delegate: Flip, rotate, color and blur the person patches from placed images, transfer their style with proposed model, and embed them back. In our work we used the style delegate augmentation which comprises of rotation, flipping, coloring and blurring.

#### 1.6.1. Training Phase

For 10,000 epochs training of 3D CNN model was performed. Best loss-to-loss accuracy ratio was achieved on epochs 557, and therefore this model was selected for experimentation process. Then the described method indicates the overfitting issues. Finally, the correction and training loss were 0.95 and 0.1297, and the correction and testing loss were 0.92 and 0.2381.

#### 1.6.2. Structure of CNN

The field of artificial intelligence has experienced significant advancements recently, largely propelled by the deep learning revolution. Within computer vision, deep neural networks, particularly deep Convolutional Neural Networks (CNNs), have become exceptionally powerful learning tools. They excel at extracting meaningful representations, ranging from low-level image processing tasks to high-level perception tasks like object detection and scene understanding. A key advantage of CNNs lies in their ability to leverage contextual information and pre-learned knowledge acquired from vast amounts of training data. The application of deep learning methodologies to pedestrian tracking has marked a significant leap in improving performance, with some approaches capable of handling variations in lighting and complex scenarios involving multiple pedestrians [143].

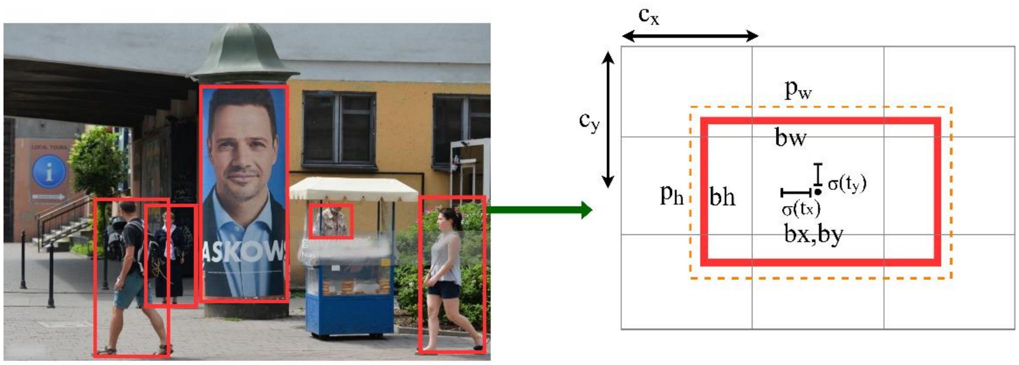
In this , we explored robust neural network architectures such as Faster R-CNN and SSD-MobileNet for pedestrian detection. Pedestrian classification using deep CNNs often relies on the TensorFlow framework. While there's typically a trade-off between computational cost and network depth, the concept of the Inception architecture offers a technique to increase network depth and width without a proportional increase in computational demands. To enhance the efficiency of resources, Inception modules are designed to process information locally and are stacked according to the local regions within the image, as depicted in Figure 4-5. To further improve the performance of the Inception module, the Inception V2 model was developed by increasing the number of convolutional layers and reducing the size of the convolution filters.

The Faster R-CNN system comprises two main components: a Region Proposal Network (RPN) and a Convolutional Neural Network for classification and bounding box regression. By integrating these two parts, Faster R-CNN functions as a unified network. Within Faster R-CNN, multi-scaled feature maps are utilized to locate different body parts across the convolutional layers. Conversely, other region-based network types are also employed for body part localization. The feature maps are provided to the classifier through a Region of Interest (RoI) pooling operation.

A significant advantage of using CNNs compared to other classification methods or traditional neural networks is their effective utilization of image templates and spatial relationships within the data. For instance, in Recurrent Neural Networks (RNNs), the output's dependence on all preceding values can lead to suboptimal results for image data. Additionally, in iterative architectures like Long Short-Term Memory networks (LSTMs), the output tends to become increasingly biased towards the most recent context as the input sequence grows very long. Considering that our model is intended for a Raspberry Pi-based vehicle with limited computing power, a carefully constructed convolutional neural network represents the most suitable architecture. As such, the design incorporates only approximately 20,000 parameters to ensure efficient processing.

### 1.6.3. Darknet of YOLOv3

As a backbone, this variant uses the Darknet53 classifier and a multi-scale indicator. YOLOv3, according to the feature pyramid network (FPN), uses the approach of feature map fusion in order to achieve multi-scale prediction. Feature extraction is facilitated through the help of Darknet-53. 53 convolution layers need to be trained for ImageNet and a large number of convolution kernels with sizes 1 × 1 and 3 × 3 are utilized. The bounding feature is down sampled since convolutional layers are two-step and in three different dimensions it does the detection. In parallel, for verifying the normalization of the input in intense layers through batch normalization of convolutional layers is shown. Darknet-53 offers higher accuracy than Darknet-19. Along with this, to prevent over-fitting Leaky RELU is utilized. With more rooms for convolutional layers, this update raises the low-measure feature that improves pedestrian detection and other challenges and enhances its speed. YOLOv3 enhances the anchor box idea of YOLOv2 by using dimension clustering in the prediction of the bounding box as illustrated in Figure 6.



(a) Bounding Boxes (b) Explaining Bounding Boxes

**Figure 6 (a) Shows the bounding boxes (b) Image explaining the bounding boxes**

The calculation of a single bounding box needs four values, that is, coordinates bx and by, width bw, and height bh. Let the coordinate of the top left corner of the network element in the feature map be (cx, cy), and the anchor box width and height be pw and ph. Then the position and size of the predicted bounding box can be represented as follows:

$$b_x = \sigma(t_x) + c_x \quad (2)$$
$$b_y = \sigma(t_y) + c_y \quad (3)$$
$$b_w = p_w e t^w \quad (4)$$
$$b_h = p_h e t^h \quad (5)$$

Where the explanation of all variables that are utilized in the above equations from (4-2) to (4-5) are given in Table 4-2. The prediction utilizes logistic regression for the object score. The score must be one if it coincides with the ground object. The threshold must be 0.5 and 0.9. While training, it sums up all the squared error loss [145].

In mathematical decision hypothesis and correction, a cost function or loss function (also referred to as an error function) is a function that describes some event-related "cost" onto a real number into more variables which solve the problem and attempt to minimize the damage function. The loss function for pedestrian detection based on deep learning includes loss function for bounding box regression. In this paper, the loss function which is utilized for pedestrian detection is the sum of three different losses which are: (i) localization/regression bounding loss, (ii) confidence loss, and (iii) classification loss, and can be described as follows:

$$L = \lambda_3 L_{reg} + \lambda_1 L_{conf} + \lambda_2 L_{cla} \qquad (6)$$

$$L_{reg} = \lambda_{coord} \sum_{i=0}^{S\times S} \sum_{j=0}^{M} \mathbb{1}_{ij}^{obj} (2 - w_i \times h_i)[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \qquad (7)$$

$$+\lambda_{coord} \sum_{i=0}^{S\times S} \sum_{j=0}^{M} \mathbb{1}_{ij}^{obj} (2 - w_i \times h_i)[(w_i - \widehat{w}_i)^2 + \left(h_i - \hat{h}_i\right)^2] \qquad (8)$$

$$L_{conf} = -\lambda_{obj} \sum_{\substack{i=0\\i=0}}^{S\times S} \sum_{j}^{M} \mathbb{1}_{ij}^{obj} [\hat{C}_i \log(C_i) + \left(1 - \hat{C}_i\right) \log(1 - C_i)] \qquad (9)$$

$$+\lambda_{noobj} \sum_{i=0}^{S\times S} \sum_{j}^{M} \mathbb{1}_{ij}^{noobj} [\hat{C}_i \log(C_i) + \left(1 - \hat{C}_i\right) \log(1 - C_i)] \qquad (10)$$

$$L_{cla} = -\sum_{\substack{i=0\\i=0}}^{S\times S} \sum_{j=0}^{M} \mathbb{1}_{ij}^{obj} \sum_{c \in classes} [\hat{p}_i(c) \log(p_i(c)) + \left(1 - \hat{p}_i(c)\right) \log(1 - p_i(c))] \quad (11)$$

### 1.7. MobileNet-SSD

The MobileNet is a good performing model for mobile or embedded device vision applications that have constrained resources. MobileNet is successfully applied in the SSD meta-architecture and leverages the deep separation agreement. The MobileNet convolution block has depth direction 3×3 convolution and applies a filter to each input channel. This is followed by 1 × 1 point-by-point convolution to produce a linear combination achievement of the prior layer. Batch normalization and ReLU activation layers are also incorporated after each convolutional layer. In MobileNetV2 [146], the Convolution block is the inverse of the rest block of ResNet. Extensions are applied prior to bottlenecks and shortcuts are utilized directly between bottlenecks in MobileNetV2. This reverse method has been more memory efficient. Employ two basic universal hyperparameters that strike a balance between delay and accuracy. These hyper-parameters enable the model builder to make the right choice constraint-based application size model. This is achieved since SSD and MobileNet, we have MobileNet-SSD, both which are fast and accurate.

### 1.8. Data Augmentation

Data augmentation is a crucial technique used to increase the size and diversity of datasets available for training machine learning models. Deep Learning (DL) methodologies often necessitate substantial amounts of training data to make accurate and robust predictions, which isn't always readily available. Therefore, we artificially expand the existing data to develop a more generalized model. Common data augmentation methods, such as cropping, padding (filling), and translation (panning), are frequently employed when training large neural networks. These techniques generate various modified versions of the original dataset, effectively increasing its size and enhancing the reliability of the machine learning model.

Experimental results have shown that DL models trained with augmented images generally outperform those trained without augmentation in terms of training loss (penalties for incorrect predictions), precision, recall, and overall verification accuracy. Data augmentation can also help reduce overfitting and improve the generalization ability of our model. By introducing variations in the training data, we enhance the heterogeneity of our training set, making the model more resilient to unseen data. In the specific domain of pedestrian detection, bounding box labels are typically acquired naturally to identify pedestrians in images. Consequently, the patterns learned during pedestrian detector training can be seamlessly integrated with the original training data. Our proposed data augmentation scheme, illustrated in Figure 7, can be effectively applied in several scenarios:

First, pedestrians from richly annotated datasets can be transferred and superimposed onto datasets with limited or no labels. This helps bridge the domain gap between different datasets and allows the model to learn from a wider variety of backgrounds and contexts.

Second, even when training and testing are performed within the same domain, we can introduce pedestrians from other annotated datasets to create a virtually unlimited number of augmented samples. This significantly improves the robustness and overall capabilities of the pedestrian recognition system.

Lastly, our approach can specifically generate variations featuring small and indoor pedestrians. This targeted augmentation is designed to enhance the performance of pedestrian detectors in these particularly challenging scenarios, where pedestrians might be less prominent or appear under different lighting and scale conditions."

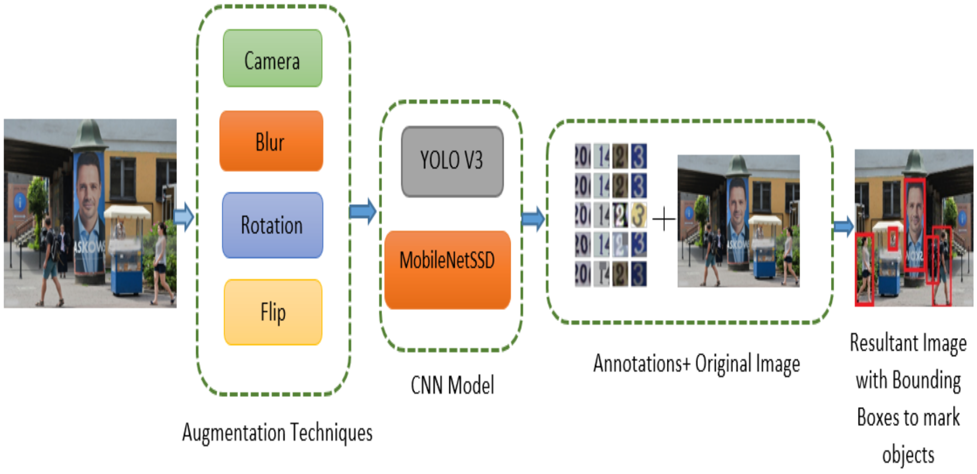

Camera
Blur
Rotation
Flip
Augmentation Techniques
YOLO V3
MobileNetSSD
CNN Model
Annotations+ Original Image
Resultant Image with Bounding Boxes to mark objects

**Figure 2 .Augmentation technique Framework for Object Detection**

## 2. Implementation and Results

The experiment is about the validity of 3D CNN in rejecting true false positives. In the first step, YOLO v3, Tiny-YOLO and MobileNet-SSD is used as a pedestrian detector. Post the first step 3D CNN is used to classify found objects as pedestrian or not. The algorithm is executed on the Waymo and KITTI dataset and store prediction results. Besides, 110 samples Random selected and individuals labeled by our system the agent asserts that the prediction is correct. To evaluate the performance of our proposed PDA models, the evaluation metric is typically utilized to identify the real-time pedestrian in time for example AP (average precision), precision and recall. The ratio of correct predictions is equal to the accuracy value reported in this section.Discussion Hardware and Software Settings

In this work, the hardware and software settings that are used during the implementation of our proposed PDA design named YOLO-3D CNN are presented in Table 2 and Table 3.

| TABLE 2 HARDWARE SETTINGS | |
|---|---|
| **PROCESSOR** | Intel® core™ i5-7200U CPU@ 3.1 GHz |
| **OS** | Free DOS (support up to windows 10) |
| **MEMORY** | 16GiB |
| **GRAPHICS** | NVIDIA GeForce 920MX, 2GB |
| **OS TYPE** | 64-bit |
| **STORAGE** | 500 HDD |

| TABLE 3 SOFTWARE SETTINGS | |
|---|---|
| **WINDOWS** | 10 pro |
| **PYTHON** | 3.6 |
| **BASE SYSTEM** | Darknet |
| **CUDA** | 9.0 |
| **CUDNN** | 7.0.5 |

### 2.1.Effecting Factors for Pedestrian Detection

For actual word instances, diverse factors affect pedestrians' performance such as weather conditions as day/night hours, fogy and raining conditions, clothes color of body components like face, head, height, etc. and pedestrians' speed essentially comprise size group, age group, sex, etc. Weather conditions create a predominant criterion for occluded, low resolution, and multispectral pedestrians particularly night times. In addition, pedestrian speed and body components must be considered for true/false pedestrian.

### 2.1.1. KITTI and Waymo Benchmarks

In our research, we utilized two prominent datasets, Waymo (referenced as source 3) and KITTI (source 4), to train and evaluate our innovative PDA model for 3D Convolutional Neural Networks. These datasets are rich in 3D object information, gathered from diverse sensors integrated into the data collection platforms described in sources 3 and 4.

The Waymo dataset offers comprehensive data, including 16-beam LiDAR scans, a full 360-degree view captured by 6 high-resolution color cameras (boasting 12.5 million 3D bounding box annotations), and highly precise GPS and inertial measurement unit data. Notably, the coordinate systems of these various sensors are carefully calibrated, simplifying the process of transforming data between different sensor perspectives. The data acquisition vehicles are equipped with a full suite of sensors, capturing a wide array of real-world driving scenarios encountered throughout the year, such as congested urban roads, areas with pedestrian activity, and shared parking spaces like garages. The dataset also includes instances of diverse weather conditions, even capturing events like thunderstorms.

It's important to note that traditional approaches often leverage the strengths of either LiDAR or front-facing color cameras individually. However, recent advancements have led to the creation of rich, multi-modal datasets that combine RGB imagery with 3D LiDAR point clouds. This fundamental combination of multi-modal data is now considered crucial for achieving significant progress in the field of advanced 3D object detection.

The KITTI dataset, a valuable asset for training and testing 3D object detection systems, provides synchronized stereo image pairs, dense LiDAR point clouds, and brief textual descriptions from a front-facing camera. This dataset includes roughly 80,000 meticulously labeled 3D bounding boxes distributed across approximately 15,000 individual frames.

Initially, when implementing our proposed PDA model, we set the training duration to 10,000 steps. We observed that increasing the number of training steps generally led to improved performance. For established models like Faster R-CNN and MobileNet-SSD, we considered around 70,000 training steps. This decision was based on the observation that beyond this point, the reduction in the models' loss value tended to become less significant. In contrast, our initial assessment at 10,000 epochs indicated that the model loss was still actively decreasing.

Our experiments revealed that as the training progressed, the final loss value for the Faster R-CNN model settled at approximately 1.23. MobileNet-SSD, while starting with a higher initial loss of 1.42, ultimately achieved a notably lower loss of 0.24 after training. A comprehensive overview of the total loss for all models throughout the training process is visually presented in Figure 8.

To optimize the training process and prevent overfitting, we began with a default learning rate of 0.5. Subsequently, we adjusted this rate upwards to 0.9 based on the observed training results. Furthermore, we set the batch size to 12 for our experiments.

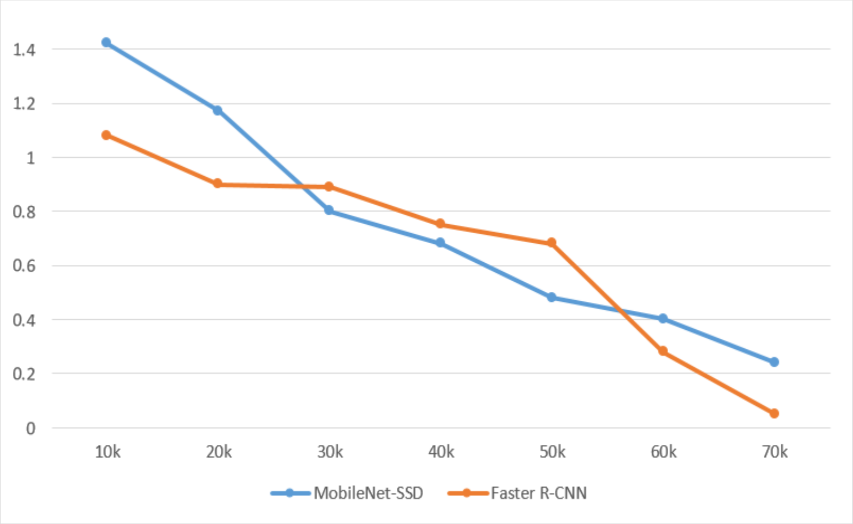


**Figure 8 Shows the epochs of SSD-MobileNet and Faster R-CNN. The blue and orange lines constitute the sum of the losses for Faster R-CNN and SSD-MobileNet**

At training time confirmed the design file of our proposed PDA model without applying any data augmentation. Then added four types of data augmentation to the design file. Data augmentation technique before and after results are shown in Table 4. Color based enhancement performs best among all other best data enhanced fuzzy enhancement technologies, AP increased by 0.25%. The before and after augmentation variance performance is not so remarkable., the highest AP increase is 0.75%.

**Table 4 Augmentation Results of SDD-MobileNet OD on datasets**

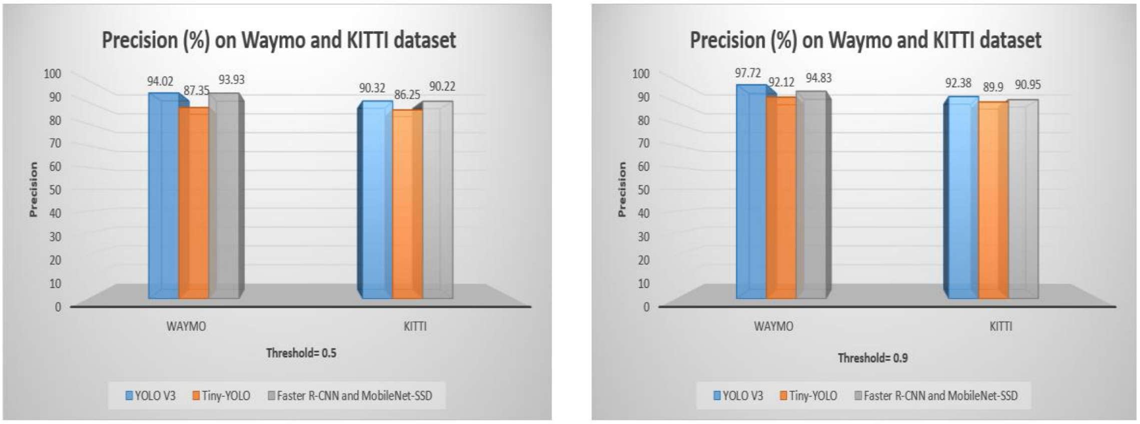
| Model | | Average Precision (Ap%)<br>Augmentation | | Pedestrian |
|---|---|---|---|---|
| SSD-MobileNet | | None | | 1.04 |
| SSD-MobileNet | | Rotating | | 0.30 |
| SSD-MobileNet | | Flipping | | 1.27 |
| SSD-MobileNet | | Color | | 1.56 |
| SSD-MobileNet | | Blurring | | 1.27 |

To set up and test our novel PDA model, we initially employed the YOLO v3 object detection framework in conjunction with the extensive Waymo dataset.

Our analysis of the proposed PDA model's performance on the Waymo dataset revealed impressive precision and recall rates. Specifically, with a learning rate of 0.5, we achieved a precision of 94% and a recall of 92%, as illustrated in Figures 5-2(a) and 5-3(a). Further optimization by increasing the learning rate to 0.9 led to even higher accuracy, reaching 97.72% for precision and 97.63% for recall, as depicted in Figures 5-2(b) and 5-3(b).

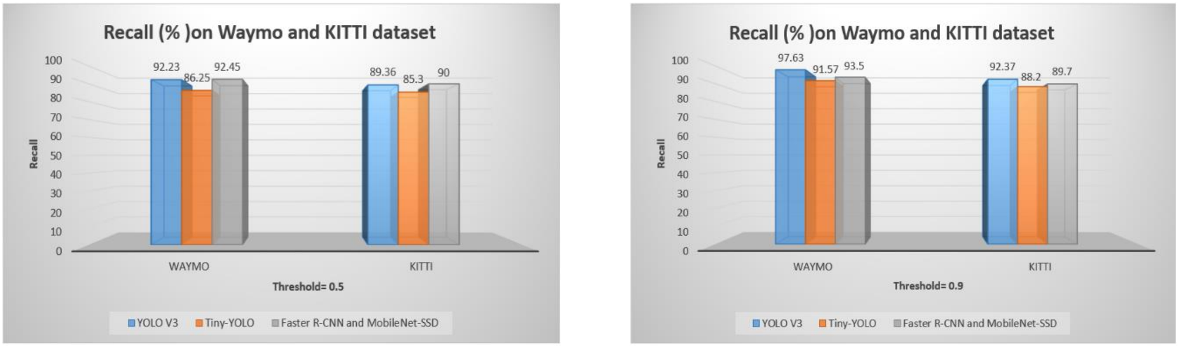
Subsequently, we experimented with Tiny-YOLO as an alternative object detector within our model configuration. With a learning rate of 0.5, the precision and recall accuracy dropped to 87% and 86%, respectively. This indicates that Tiny-YOLO appears to be less robust compared to YOLOv3 in this context, as shown in Figures 9(a) and 10(a). Increasing the learning rate for Tiny-YOLO to 0.9 resulted in improved accuracy, achieving 92% precision and 91% recall, as demonstrated in Figures 9(b) and 10(b).

Finally, we integrated MobileNet-SSD and Faster R-CNN as object detectors within our PDA model. Utilizing a higher learning rate, ranging from 0.5 up to 0.9, these configurations achieved precision and recall accuracy rates of up to 94.83% and 93.5%, respectively, as detailed in Figures 9 and 10.The primary goals in evaluating these different model configurations were to achieve exceptional accuracy, maintain efficient processing times, and effectively identify various components of objects, particularly beyond the initial convoluteional layers.



***(a)*** *Precision output on 0.5* ***(b)*** *Precision output on 0.9*

**Figure 9 The precision result of the YOLO-3D CNN on novel object detectors using two different datasets**



**(a)** Recall output on 0.5 threshold.

**(b)** Recall output on 0.5 threshold.

Similarly, when we evaluated our proposed model on the KITTI dataset, using YOLO v3 as the object detector yielded precision and recall accuracy rates of 92.38% and 92.37%, respectively, with an increased learning rate. In contrast, when Tiny-YOLO was employed, the accuracy rates decreased to 89% for precision and 88% for recall. Lastly, the accuracy rates achieved with MobileNet-SSD and Faster R-CNN were slightly lower, around 90% for precision and 89% for recall, as illustrated in Figures 5-2(b) and 5-3(b). Consequently, our findings indicate that YOLO v3 provided the most favorable object detection performance within our proposed model, particularly when utilizing the Waymo dataset, outperforming Tiny-YOLO. This superior performance on the Waymo dataset, compared to KITTI, can likely be attributed to the fact that Waymo is a more recently released dataset offering a substantial amount of sensor data from various modalities, including comprehensive LiDAR data and high-resolution camera imagery. It encompasses a wider variety of real-world scenarios, with Waymo data originating from diverse environments such as peak traffic scenes in Phoenix, covering both daytime and nighttime navigation. The performance of all the tested object detectors across both datasets is visually summarized in Figures 5-2 and 5-3.

Figure 11 presents a performance graph of our proposed PDA model using YOLO v3 at learning rates of 0.5 and 0.9. Notably, our proposed PDA model with YOLO v3 achieved the highest accuracy, in terms of both precision and recall, at the 0.9 learning rate. This specific configuration demonstrated improved performance compared to prior work and effectively minimized errors in classifying true positives and false positives. A key factor contributing to this enhanced performance is YOLO v3's ability to process the entire image during training, enabling the detector model to perform real-time predictions more accurately.

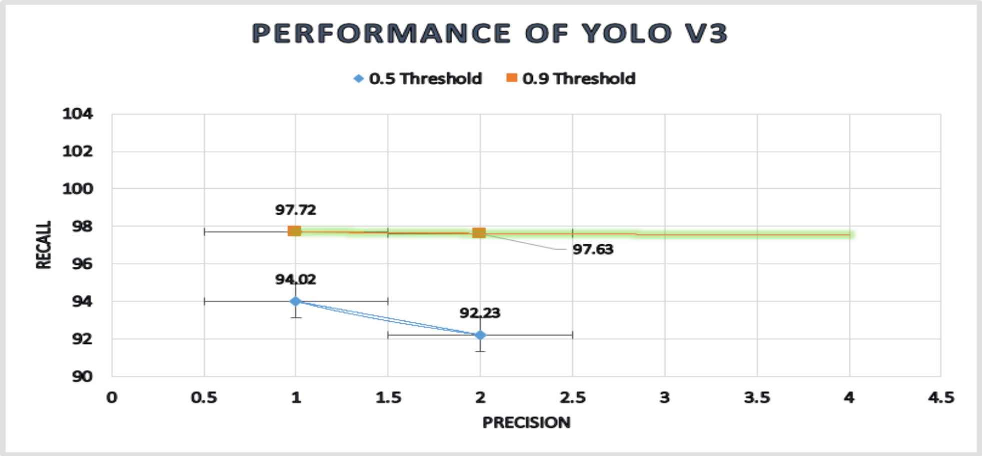


**Figure 11 Performance graph of our proposed model YOLOv3 on different learning rate**

To assess the effectiveness of our proposed system, we conducted evaluations on a dedicated pedestrian test set. This set was created by selecting 20% of the pedestrian images from the various datasets we utilized. We focused on measuring the Average Precision (AP) and the speed of detection, which are standard metrics for evaluating a model's performance on unseen data. Furthermore, Table 5-6 provides a comparative analysis of our proposed network's performance against existing algorithms on several well-established datasets, including the Pascal VOC dataset. Our network was also tested on the Caltech Pedestrian dataset, INRIA, and COCO datasets, where it demonstrated encouraging results. As clearly shown in Table 5-6, the YOLO v3 network achieved a notable detection rate of 82.4% in precision and 88.5% in recall. This significantly surpasses the detection rate of Tiny-YOLO by +12.11% in precision and +17.27% in recall, highlighting the superior performance of YOLO v3 in our evaluations.

The YOLO v3 network demonstrates a stronger capability and greater suitability for real-time pedestrian detection due to its effective balance between processing speed and detection accuracy when compared to Tiny-YOLO. Notably, within our proposed network, YOLO v3 achieved impressive accuracy rates of 97.72% for precision and 97.63% for recall on the Waymo dataset using a 0.9 threshold. In contrast, Faster R-CNN and MobileNet-SSD yielded accuracy rates of 94.83% for precision and 93.5% for recall, which, as shown in Table 5-6 representing our proposed PDA model's results, is more consistent than the performance of Tiny-YOLO.

Table 5-6 further presents a comparison between the results obtained by our proposed PDA model, utilizing YOLOv3, Tiny YOLO, Faster R-CNN, and MobileNet-SSD, and the findings of prior research. As evident from the table, our proposed PDA model delivers more precise results compared to existing studies and effectively minimizes errors in distinguishing between true and false detections.

**Table 5: Comparison of our proposed PDA model's test results (YOLOv3, Tiny YOLO, Faster R-CNN, and MobileNet-SSD) with existing work**

| Our Proposed PDA Model | | | Previous Work | | |
|---|---|---|---|---|---|
| **Detection Network** | Average Precision (Ap%) | Average Recall (AR%) | Detection Network | Average Precision (Ap%) | Average Recall (AR%) |
| **Tiny YOLO** | 92.12 | 91.57 | Tiny YOLO [147] | 70.30 | 71.23 |
| **YOLO v3** | 97.72 | 97.63 | YOLO v3 [148] | 82.4 | 88.5 |
| **Faster R-CNN** | 94.83 | 93.5 | Faster R-CNN [149] | 88.8 | 84.3 |
| **MobileNet-SSD** | 94.83 | 93.5 | MobileNet-SSD [149] | 73.5 | 56.9 |

## 3. Conclusion & Future Direction

Accurate pedestrian detection stands as a cornerstone for the safety of self-driving vehicles, playing a vital role in preventing potential accidents. However, achieving reliable pedestrian detection is a formidable challenge due to several key factors: (i) the frequent obstruction or partial visibility of pedestrians (occlusion), (ii) the limitations of image quality, including low resolution and the use of multi-spectral data, and (iii) the inherent difficulty in accurately distinguishing between actual pedestrians and other objects, thus avoiding false positives.

In recent years, Deep Learning (DL) technologies have emerged as a powerful tool, demonstrating significant promise in tackling these persistent challenges associated with pedestrian detection in the realm of autonomous driving. This paper offers a concise introduction to DL techniques as a contemporary approach aimed at resolving the aforementioned issues.

The central contribution of this thesis is the proposal of a novel Pedestrian Detection Algorithm (PDA) specifically engineered to enhance the accuracy of identifying genuine pedestrians while minimizing the occurrence of false detections. Our PDA incorporates a YOLO-3D CNN model designed to improve the differentiation between true pedestrians and non-pedestrian entities. The architecture of our PDA strategically leverages the strengths of several key components. Initially, it employs YOLOv3 to analyze the entire input image, enabling the training of a detector model capable of making predictions in real-time. Secondly, it integrates MobileNet-SSD as a feature extractor, selected for its favorable combination of high accuracy and efficient processing speed. Lastly, the PDA utilizes the Faster R-CNN technique to precisely pinpoint the locations of various parts within detected objects, building upon the feature representations learned by the convolutional layers. To further enhance the robustness and generalizability of our PDA model, we also implemented data augmentation techniques. These methods effectively expand the diversity of our training dataset by generating modified versions of existing samples, thereby maximizing the information gained from the available training data.

### 3.1. Future Direction

In the future, we will take into consideration the following work.

- "Obstruction" or "blockage" is also one of the primary pitfalls of pedestrian detection. In current literature, it is discovered that majority of the papers employed deep learning methods and traditional algorithms on tiny and limited blockage. The DL approach achieves the best performance among traditional algorithms. However, the accuracy would drop rapidly if there is dense and prolonged blockage in complicated prospect. Thus, the future is to manage long-term and heavy occlusion through DL techniques using support from generalized techniques and several datasets in an attempt to enhance the accuracy of the occlusion.
- For pedestrian detection, better results have been obtained with the help of DL techniques. But even today, the existing algorithms are still facing the issue of detecting small, moderate, and occluded objects. In the future, one can think/address the above issues.
- Also, more than adequate is yet to be achieved to examine the detection production improving in unfavorable conditions of weather and illumination. Addressing this concern in the future can be made possible by training the two models simultaneously using both day-time as well as night-time models in one paradigm to further the capabilities of generalization.
- Further, it is also required to study more approaches that are compounded into the detection algorithms to increase the accuracy boost. In the future, few effective techniques could be compounded together to enhance pedestrian detection system accuracy.
- Algorithms based on fuzzy logic can be blended with DL algorithms for refining the pedestrian detection process.
- A challenging future project could be to explore/ integrate 3D measurements with 2D information to enhance detections and classifications.
- Multi-class methodologies must be integrated, not merely to account for alternative pedestrian models, but also to inspect for alternative targets (e.g., cars) and render the system more robust.

- A DL algorithm-based pedestrian detection has solved most of the issues in pedestrian detection, but these are extremely slow, and interpretability is extremely low. Hence, the major issue in pedestrian detection is speed and accuracy. Future research can be focused on improving the computation speed and detection accuracy.

## References

Ahamed, N.N. and R. Vignesh. 2022. Smart agriculture and food industry with blockchain and artificial intelligence. J. Comput. Sci. 18:1–17.

Ahmad, S., S.H. Awan, Y. Khan, N. Safwan, S.S. Qurashi and M.Z. Hashim. 2021. A combo smart model of blockchain with the Internet of Things (IoT) for the transformation of agriculture sector. Wirel. Pers. Commun. 121:2233–2249.

Ahmed, R.A., E.E. Hemdan, W. El-Shafai, Z.A. Ahmed, E.M. El-Rabaie and F.E. Abd El-Samie. 2022. Climate-smart agriculture using intelligent techniques, blockchain and Internet of Things: Concepts, challenges, and opportunities. Trans. Emerg. Tele. Tech. 33:4607-4620.

Ahsan, T., F. Zeeshan khan, Z. Iqbal, M. Ahmed, R. Alroobaea, A.M. Baqasah, I. Ali and M.A. Raza. 2022. IoT devices, user authentication, and data management in a secure, validated manner through the blockchain system. Wirel. Commun. Mob. Comput. 2022:1–13.

M. Waqas, Z. Khan, S. U. Ahmed and A. Raza, "MIL-Mixer: A Robust Bag Encoding Strategy for Multiple Instance Learning (MIL) using MLP-Mixer," 2023 18th International Conference on Emerging Technologies (ICET), Peshawar, Pakistan, 2023, pp. 22-26. DOI: 10.1109/ICET59753.2023.10374927.

Alam, S. 2023. Security Concerns in Smart Agriculture and Blockchain-Based Solution. 2022 OPJU. Inter. Techn. Conf. Emer. Tech. IEEE. pp.1–6.

Aldhyani, T.H.H. and H. Alkahtani. 2023. Cyber security for detecting distributed denial of service attacks in agriculture 4.0: Deep learning model. Int. J. Math. 11:233.

Awan, S., S. Ahmed, F. Ullah, A. Nawaz, A. Khan, M.I. Uddin, A. Alharbi, W. Alosaimi and H. Alyami. 2021a. IoT with BlockChain: A Futuristic Approach in Agriculture and Food Supply Chain. Wirel. Commun. Mob. Comput. 2021:5580179.

Khan, Zia, Saif Ur Rehman Khan, Omair Bilal, Asif Raza, and Ghazanfar Ali. "Optimizing Cervical Lesion Detection Using Deep Learning with Particle Swarm Optimization." In 2025 6th International Conference on Advancements in Computational Sciences (ICACS), pp. 1-7. IEEE, 2025. DOI: 10.1109/ICACS64902.2025.10937863

Shahzad, Inzamam, Asif Raza, and Muhammad Waqas. "Medical Image Retrieval using Hybrid Features and Advanced Computational Intelligence Techniques." Spectrum of engineering sciences 3, no. 1 (2025): 22-65

Raza, A., Salahuddin, & Inzamam Shahzad. (2024). Residual Learning Model-Based Classification of COVID-19 Using Chest Radiographs. Spectrum of Engineering Sciences, 2(3), 367–396.

Bhat, S.A., N.-F. Huang, I.B. Sofi and M. Sultan. 2021. Agriculture-food supply chain management based on blockchain and IoT: a narrative on enterprise blockchain interoperability. Agri. 12:40-52.

Bhatia, J., K. Italiya, K. Jadeja, M. Kumhar, U. Chauhan, S. Tanwar, M. Bhavsar, R. Sharma, D.L. Manea and M. Verdes. 2022. An overview of fog data analytics for IoT applications. Sensors 23:199-209.

Raza, A., Soomro, M. H., Shahzad, I., & Batool, S. (2024). Abstractive Text Summarization for Urdu Language. Journal of Computing & Biomedical Informatics, 7(02).

Bhushan, B., C. Sahoo, P. Sinha and A. Khamparia. 2021. Unification of Blockchain and Internet of Things (BIoT): requirements, working model, challenges and future directions. Wir. Net. 27:55–90.

M. Wajid, M. K. Abid, A. Asif Raza, M. Haroon, and A. Q. Mudasar, “Flood Prediction System Using IOT & Artificial Neural Network”, VFAST trans. softw. eng., vol. 12, no. 1, pp. 210–224, Mar. 2024. DOI: 10.21015/vtse.v12i1.1603

Khan, Z., Hossain, M. Z., Mayumu, N., Yasmin, F., & Aziz, Y. (2024, November). Boosting the Prediction of Brain Tumor Using Two Stage BiGait Architecture. In 2024 International Conference on Digital Image Computing: Techniques and Applications (DICTA) (pp. 411-418). IEEE.

Khan, S. U. R., Raza, A., Shahzad, I., & Ali, G. (2024). Enhancing concrete and pavement crack prediction through hierarchical feature integration with VGG16 and triple classifier ensemble. In 2024 Horizons of Information Technology and Engineering (HITE)(pp. 1-6). IEEE https://doi. org/10.1109/HITE63532.

Khan, S.U.R., Zhao, M. & Li, Y. Detection of MRI brain tumor using residual skip block based modified MobileNet model. Cluster Comput 28, 248 (2025). https://doi.org/10.1007/s10586-024-04940-3

Khan, U. S., & Khan, S. U. R. (2024). Boost diagnostic performance in retinal disease classification utilizing deep ensemble classifiers based on OCT. Multimedia Tools and Applications, 1-21.

Asif, S., Khan, S. U. R., Amjad, K., & Awais, M. (2024). SKINC-NET: an efficient Lightweight Deep Learning Model for Multiclass skin lesion classification in dermoscopic images. Multimedia Tools and Applications, 1-27.

Asif, S., Awais, M., & Khan, S. U. R. (2023). IR-CNN: Inception residual network for detecting kidney abnormalities from CT images. Network Modeling Analysis in Health Informatics and Bioinformatics, 12(1), 35.

Khan, M. A., Khan, S. U. R., Haider, S. Z. Q., Khan, S. A., & Bilal, O. (2024). Evolving knowledge representation learning with the dynamic asymmetric embedding model. Evolving Systems, 1-16.

Raza, A., & Meeran, M. T. (2019). Routine of encryption in cognitive radio network. Mehran University Research Journal of Engineering & Technology, 38(3), 609-618.

Al-Khasawneh, M. A., Raza, A., Khan, S. U. R., & Khan, Z. (2024). Stock Market Trend Prediction Using Deep Learning Approach. Computational Economics, 1-32.

Khan, U. S., Ishfaque, M., Khan, S. U. R., Xu, F., Chen, L., & Lei, Y. (2024). Comparative analysis of twelve transfer learning models for the prediction and crack detection in concrete dams, based on borehole images. Frontiers of Structural and Civil Engineering, 1-17.

Khan, S. U. R., & Asif, S. (2024). Oral cancer detection using feature-level fusion and novel self-attention mechanisms. Biomedical Signal Processing and Control, 95, 106437.

Farooq, M. U., Khan, S. U. R., & Beg, M. O. (2019, November). Melta: A method level energy estimation technique for android development. In 2019 International Conference on Innovative Computing (ICIC) (pp. 1-10). IEEE.

Raza, A.; Meeran, M.T.; Bilhaj, U. Enhancing Breast Cancer Detection through Thermal Imaging and Customized 2D CNN Classifiers. VFAST Trans. Softw. Eng. 2023, 11, 80–92.

Dai, Q., Ishfaque, M., Khan, S. U. R., Luo, Y. L., Lei, Y., Zhang, B., & Zhou, W. (2024). Image classification for sub-surface crack identification in concrete dam based on borehole CCTV images using deep dense hybrid model. Stochastic Environmental Research and Risk Assessment, 1-18.

Khan, S.U.R.; Asif, S.; Bilal, O.; Ali, S. Deep hybrid model for Mpox disease diagnosis from skin lesion images. Int. J. Imaging Syst. Technol. 2024, 34, e23044.

Khan, S.U.R.; Zhao, M.; Asif, S.; Chen, X.; Zhu, Y. GLNET: Global–local CNN's-based informed model for detection of breast cancer categories from histopathological slides. J. Supercomput. 2023, 80, 7316–7348.

Hekmat, Arash, Zuping Zhang, Saif Ur Rehman Khan, Ifza Shad, and Omair Bilal. "An attention-fused architecture for brain tumor diagnosis." Biomedical Signal Processing and Control 101 (2025): 107221.

Khan, S.U.R.; Zhao, M.; Asif, S.; Chen, X. Hybrid-NET: A fusion of DenseNet169 and advanced machine learning classifiers for enhanced brain tumor diagnosis. Int. J. Imaging Syst. Technol. 2024, 34, e22975.

Khan, S.U.R.; Raza, A.;Waqas, M.; Zia, M.A.R. Efficient and Accurate Image Classification Via Spatial Pyramid Matching and SURF Sparse Coding. Lahore Garrison Univ. Res. J. Comput. Sci. Inf. Technol. 2023, 7, 10–23.

Farooq, M.U.; Beg, M.O. Bigdata analysis of stack overflow for energy consumption of android framework. In Proceedings of the 2019 International Conference on Innovative Computing (ICIC), Lahore, Pakistan, 1–2 November 2019; pp. 1–9.

HUSSAIN, S., Raza, A., MEERAN, M. T., IJAZ, H. M., & JAMALI, S. (2020). Domain Ontology Based Similarity and Analysis in Higher Education. IEEEP New Horizons Journal, 102(1), 11-16.

Shahzad, I., Khan, S. U. R., Waseem, A., Abideen, Z. U., & Liu, J. (2024). Enhancing ASD classification through hybrid attention-based learning of facial features. Signal, Image and Video Processing, 1-14.

Mahmood, F., Abbas, K., Raza, A., Khan,M.A., & Khan, P.W. (2019 ). Three Dimensional Agricultural Land Modeling using Unmanned Aerial System (UAS). International Journal of Advanced Computer Science and Applications (IJACSA) [p-ISSN : 2158-107X, e-ISSN : 2156-5570], 10(1).

Meeran, M. T., Raza, A., & Din, M. (2018). Advancement in GSM Network to Access Cloud Services. Pakistan Journal of Engineering, Technology & Science [ISSN: 2224-2333], 7(1).

Khan, S. R., Raza, A., Shahzad, I., & Ijaz, H. M. (2024). Deep transfer CNNs models performance evaluation using unbalanced histopathological breast cancer dataset. Lahore Garrison University Research Journal of Computer Science and Information Technology, 8(1).

Bilal, Omair, Asif Raza, and Ghazanfar Ali. "A Contemporary Secure Microservices Discovery Architecture with Service Tags for Smart City Infrastructures." VFAST Transactions on Software Engineering 12, no. 1 (2024): 79-92.

Bilal, O., Asif, S., Zhao, M., Khan, S. U. R., & Li, Y. (2025). An amalgamation of deep neural networks optimized with Salp swarm algorithm for cervical cancer detection. Computers and Electrical Engineering, 123, 110106.

Khan, S. U. R., Asif, S., Zhao, M., Zou, W., Li, Y., & Li, X. (2025). Optimized deep learning model for comprehensive medical image analysis across multiple modalities. Neurocomputing, 619, 129182.

Khan, S. U. R., Asif, S., Zhao, M., Zou, W., & Li, Y. (2025). Optimize brain tumor multiclass classification with manta ray foraging and improved residual block techniques. Multimedia Systems, 31(1), 1-27.

Khan, S. U. R., Asim, M. N., Vollmer, S., & Dengel, A. (2025). AI-Driven Diabetic Retinopathy Diagnosis Enhancement through Image Processing and Salp Swarm Algorithm-Optimized Ensemble Network. arXiv preprint arXiv:2503.14209.

Khan, Z., Khan, S. U. R., Bilal, O., Raza, A., & Ali, G. (2025, February). Optimizing Cervical Lesion Detection Using Deep Learning with Particle Swarm Optimization. In 2025 6th International Conference on Advancements in Computational Sciences (ICACS) (pp. 1-7). IEEE.

Khan, S.U.R., Raza, A., Shahzad, I., Khan, S. (2025). Subcellular Structures Classification in Fluorescence Microscopic Images. In: Arif, M., Jaffar, A., Geman, O. (eds) Computing and Emerging Technologies. ICCET 2023. Communications in Computer and Information Science, vol 2056. Springer, Cham. https://doi.org/10.1007/978-3-031-77620-5_20