https://doi.org/10.31436/iiumej.v26i1.3268

4D RADAR IMAGING AND CAMERA FUSION FOR ROAD CROSSING DETECTION AND CLASSIFICATION USING DEEP LEARNING

LIYAANA SHAHIRAH WAN ABD AZIZ[1*], FARAH NADIA MOHD ISA[1], FARIDAH ABD RAHMAN[1], ARVIND HARI NARAYANAN[2], AHMAD REZA ALGHOONEH[3], GEORGE SHAKER[4]

[1]*Department of Electrical and Computer Engineering, Kulliyyah of Engineering, International Islamic University Malaysia, 53100, Jalan Gombak, Kuala Lumpur, Malaysia*

[2]*Emrail Sdn. Bhd., 19-8 Block C1 Dataran Prima, Jalan PJU 1/41, 47301, Petaling Jaya, Selangor, Malaysia*

[3]*Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada*

[4]*Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada*

**Corresponding author: yaanashira97@gmail.com*

(Received: 19 April 2024; Accepted: 6 October 2024; Published online: 10 January 2025)

ABSTRACT: This paper presents the development of an object detection and classification system for road crossing areas, integrating 4D radar imaging and a mono-camera dataset with a deep-learning neural network. The system utilizes deep neural networks implemented via Keras and TensorFlow to detect and classify multiple targets, including pedestrians, cars, buses, and trucks. At the core of this work is Retina-4F, a multi-chip radar imaging system developed by Smart Radar System, which offers high-resolution object detection and localization capabilities. Retina-4F provides real-time 4D information on detected objects, operating in a cascading architecture with three transmitters and four receivers per chip. Two road-crossing scenes were simulated to collect data, generating a point cloud dataset labeled with target classes for neural network training and testing. Data from two main sensors—Retina-4F and a mono-camera—were pre-processed using DBSCAN and YOLOv7 for enhanced accuracy. Operating at 77 GHz, Retina-4F was tested in two road environments, generating a dataset with approximately 10,000 frames. The deep learning model demonstrated an accuracy of 84% in classifying multiple targets, including cars, pedestrians, buses, and trucks. The fusion of radar point cloud data with visual sensor data proved effective, showing strong results in distinguishing target types.

ABSTRAK: Kertas ini membentangkan pembangunan sistem pengesanan dan pengelasan objek untuk kawasan lintasan jalan raya, menggabungkan pengimejan radar 4D dan set data mono-kamera dengan rangkaian neural pembelajaran mendalam. Sistem ini menggunakan rangkaian neural mendalam yang dilaksanakan melalui *Keras* dan *TensorFlow* untuk mengesan dan mengelaskan pelbagai sasaran, termasuk pejalan kaki, kereta, bas, dan trak. Inti daripada kajian ini adalah *Retina-4F*, sistem pengimejan radar berbilang cip yang dibangunkan *oleh Smart Radar System*, yang menawarkan keupayaan pengesanan objek dan penentuan lokasi resolusi tinggi. *Retina-4F* menyediakan maklumat 4D masa nyata mengenai objek yang dikesan, beroperasi dengan tiga pemancar dan empat penerima bagi setiap cip dalam seni bina kaskad. Dua adegan lintasan jalan disimulasikan untuk mengumpul data, menghasilkan set data awan titik yang dilabel dengan kelas sasaran untuk latihan dan ujian

rangkaian neural. Data daripada dua sensor utama—*Retina-4F* dan mono-kamera—dipra-proses menggunakan *DBSCAN* dan *YOLOv7* untuk meningkatkan ketepatan. Beroperasi pada 77 GHz, *Retina-4F* diuji dalam dua persekitaran jalan yang berbeza, menghasilkan set data dengan kira-kira 10,000 bingkai. Model pembelajaran mendalam menunjukkan ketepatan sebanyak 84% dalam mengelaskan pelbagai sasaran, termasuk kereta, pejalan kaki, bas, dan trak. Penggabungan data awan titik radar dengan data sensor visual terbukti berkesan, menunjukkan hasil yang kuat dalam membezakan antara jenis sasaran.

KEYWORDS: *4D Radar Imaging, sensors fusion, deep learning, YOLOv7, Keras and TensorFlow.*

1. INTRODUCTION

The number of projects conducted in recent years involving automotive radar imaging is growing drastically. This is due to the demands in the automotive industry to move towards Adaptive driver-assistance systems (ADAS). The number of road accidents and incidents is very high as vehicles on the road have increased. Thus, this has led to a global safety issue, as these problems are commonly seen in high-income and low- and middle-income countries. According to the World Health Organization (WHO), 1.3 million people have died because of road accidents around the world, and about 20 to 50 million have suffered road accident injuries [1]. Hence, to improve the safety of road users, researchers and manufacturers have begun to support the automotive radar imaging technology [2] in autonomous vehicles, where the technology offers much higher reliability in providing high accuracy in object detection, road lane warning, blind spot detection, and an early potential crash warning system [3].

The advancement of radar for autonomous vehicles is provided by Complementary Metal-Oxide-Semiconductor (CMOS) technology, which has been integrated with automotive radar imaging and offers a much lower cost of radar on-chip and antenna on-chip mass production [4]. Moreover, CMOS technology plays a vital role in radar systems by providing low power consumption and increased data processing capacity [5]. This has enabled radar manufacturers to improve their radar chipset solutions for the automotive market [3]. Today, radar imaging systems implement the cascading of multiple chips to enhance sensor technology. Based on [6], compared to a single chip, a radar system cascaded with multiple chips allows the radar to operate with multiple transmit and receive antennas, which improves object detection.

Multiple Input Multiple Output (MIMO) radar has proven to be a reliable radar system, and the concept is mostly used in many types of radar applications. MIMO systems are used in radar imaging technology to improve angular resolution and accuracy, slow-moving target detection, and the percentage of object detection [2, 6]. For this purpose, the presence of this MIMO system led to high-resolution 4D radar imaging, resulting in a denser point cloud output [7]. The concept of 4D radar imaging is to provide high-resolution azimuth, elevation, range, and Doppler information of a target [8]. However, a single sensor system is insufficient with today's radar technology to achieve highly advanced automated driving. ADAS implementation on automated vehicles combines several sensor technologies known as Lidar, Camera, and Ultrasound [5]. Even so, radar still plays a vital role [9] in automated driving, especially with 4D radar imaging systems that enable high accuracy of distance measurement, Doppler velocity, and independence in weather conditions and lighting exposure [2].

There has been much ongoing research on radar systems using RF chips for automotive 4D radar imaging to use MIMO-based radar with frequency-modulated continuous wave (FMCW) waveform. In [10], a parking monitoring system was estimated using 77-GHz MIMO

FMCW radar to detect vacant parking spaces. The radar's performance is demonstrated by adopting radar image properties to improve the performance of radar imaging processing. Moreover, in [11], the authors demonstrated the measurement of curbstone height estimation, drain cover detection, and parking lot detection as part of safety applications in autonomous vehicles. Furthermore, as demonstrated in [12], the authors work on the performance of the MIMO FMCW automotive radar under adverse weather conditions. Using 77 GHz radar platforms, the MIMO FMCW radars are tested for snowy and foggy weather conditions.

To classify 3D radar point clouds, deep neural networks are employed as 4D radar imaging has established its credibility in the market [13-15]. Several active projects investigate the classification of human activities using Convolutional Neural Networks (CNN). Some methods for classifying human activity use CNN as the training model, which produces a point cloud image of the target when utilizing FMCW MIMO-based 4D radar image [14-16]. Recurrent Neural Network (RNN) usage is another common method for target classification [17]. In order to achieve denser point cloud features of a target, RNN is primarily used in the application of identifying moving targets [18].

Among the work done in classifying point cloud data for any application, the classification of radar point clouds faces several challenges. For example, in [14][15], radar point clouds are insufficient for classifying human activity, which requires understanding the pattern of human activity, such as sitting or standing. To overcome these obstacles, the authors determined how to merge point cloud data to produce a denser point cloud. By doing so, the distinctive signatures of each human activity can be identified and classified more easily. Training a classifier with aggregated radar point cloud data requires more computational effort than training it with a single radar data frame.

To simplify classification work, we propose a point cloud classification method using sensor fusion with a mono camera and 4D radar image. In this paper, the radar sensor used is Retina-4F, developed by Smart Radar Systems. Based on the Smart Radar system, Retina-4F is designed and developed for automotive applications [19]. With its capability of providing real-time raw data, users can work on data processing for various purposes, such as collecting datasets for deep learning and object recognition.

A study utilized a Retina-4F radar module to estimate vehicle orientation for autonomous driving [20]. Their work focuses on estimating the vehicle's orientation using three machine learning methods: Principal Component Analysis (PCA), decision tree, and Convolutional Neural Network (CNN). The radar provides the data, and the point cloud output of the vehicle is used to process the orientation estimation. Based on the authors' work, Retina-4F provides dense point cloud data, which offers a sufficient point cloud image to process the orientation angle further.

Millimeter-wave radar has the potential to provide a highly accurate location, Doppler velocity, and angular resolution of the desired target [21]. However, concerning object classification, radar point cloud sparsity could affect the recognition system due to the presence of a significant number of reflection points originating from the background. Hence, we can observe the trend of research work moving towards sensor fusion. As mentioned in [22], the authors enhance the system for detecting and tracking objects in autonomous vehicles by fusing camera data and radar 3D object detection. Therefore, combining this technology can enhance the object's visual recognition and the system's tracking capability.

In obtaining a dataset for developing a neural network point cloud classifier, only a few datasets are available in radar point cloud target classification. Hence, a point cloud dataset

with target labeling is needed before developing a neural network classifier. We proposed a sensor fusion that combined radar point cloud data and image data to create a custom dataset. Many studies adopt this method in combining sensor fusion for higher efficiency in target detection and classification [23-25], especially for fully autonomous applications.

Our paper aims to develop target detection and classification using the deep learning method by performing measurements using a 77 GHz automotive radar image system for object detection by varying targets commonly seen on road crossings, like pedestrians, cars, buses, and trucks. With the aid of the 4D radar image and mono camera, we obtained the target's data information and used the Python environment and YOLOv7 for data processing and manipulation. Based on the information obtained, like target point cloud data and image data, it is used for data fusion to associate the sensor data for object classification. Based on this data fusion, we obtained a dataset of targets with their point cloud features and class labels.

The rest of the paper is organized as follows: Section 2 describes the materials and methodology used in this work, along with a brief workflow discussion. Section 3 discussed the processing involved after obtaining data from the sensors and the medium used for processing the data. Section 4 focuses mainly on sensor data fusion, preparing datasets for neural network training, and discussing the development of neural network models. In Section V, we discussed the performance of the neural network classifier and the analysis of the results obtained. In the last section, we share the conclusion of this work and future improvement areas.

2. MATERIALS AND METHOD

2.1. Retina-4F

For this work, we employed Retina-4F FMCW-MIMO millimeter-wave imaging radar from the Smart Radar System. This radar operates in medium-range mode, with a maximum detectable distance of 100 m for cars and 40 m for pedestrians. Table 1 lists the radar specifications.

Table 1. Retina-4F Specifications

Parameter	Value
Number of chips cascading	TI AWR2243 x 4
Number of Tx/Rx	12 Tx / 16 Rx
Frequency	77 GHz
Bandwidth	3.8 GHz
Range resolution	0.5 m
Azimuth angle resolution	2.0°
Azimuth field of view	100° ± 50°
Elevation angle resolution	2.0°
Elevation field of view	24° ± 12°
Point cloud output	Max 6, 144 per frame

Four Texas Instruments (TI) AWR2243 chips have been integrated into this radar system, enabling RETINA-4F to differentiate several objects without experiencing a significant loss in detection and localization accuracy. The radar uses four cascade chips, each of which has patch

https://doi.org/10.31436/iiumej.v26i1.3268

antennas with four transmitters (Tx) and three receivers (Rx), for a total of 12 transmitter antennas, 16 receiver antennas, and 192 virtual channels.

2.2. Experimental Setup and Workflow

Pedestrians, cars, buses, and trucks are the items of interest for classification in the experiment that involves gathering data at road crossings since they are the most frequent types of objects at these locations. The experimental setup for data collection at a road crossing is shown in Figure 1, with the distance between the crossing and the radar set to be roughly 50m. All measurements were made outside at two different road crossings. Due to the radar's azimuth field of view, which is 100° ± 50°, and its location approximately 50 meters from the crossing, it is possible to observe and measure the motions of various objects, including pedestrians, before and after they cross the road. Many more actions can be collected by having a much wider view than when the radar image is focused exclusively on the crossing. For example, a car waiting for pedestrians to cross will have a very low, approaching zero, Doppler velocity value. This information is useful for training the classifier to recognize that cars have Doppler features with different ranges of Doppler velocity.

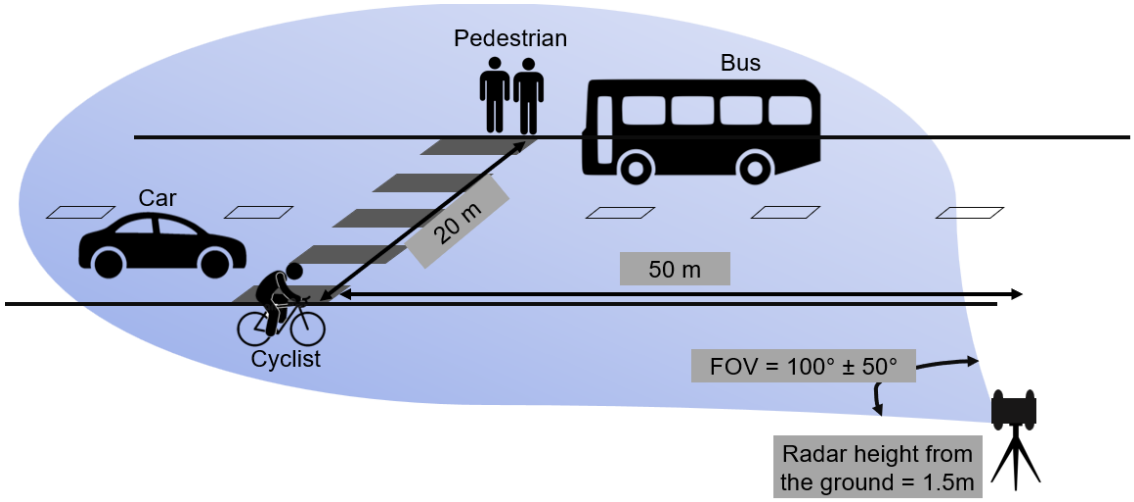

Figure 1. Experimental setup data collection.

Based on Figure 2, 4D Radar Imaging and a monocular camera are used for the measurement. Due to the unavailability of datasets currently available for radar point clouds at road crossing areas, gathering data to train a deep neural network for classification is necessary. This data must include the objects' features as well as their labels. The camera and radar are combined only for deep neural network training. The robotic operating system (ROS) environment in which Retina-4F operates makes data collection easier since it allows for simultaneous control over radar and camera nodes. As a result, the timestamp can be used to synchronize the radar point cloud data and camera images.

Referring to Figure 2, the mounting arrangement between the radar and camera is made for the simple data fusion process. In this case study, the data obtained from the camera is passed through the You Only Look Once version 7 (YOLOv7) object detection algorithm to produce bounding box predictions of objects present in the image. Meanwhile, radar data from Retina-4F are being preprocessed using density-based spatial clustering of applications with noise (DBSCAN) for point cloud clustering.

https://doi.org/10.31436/iiumej.v26i1.3268

Figure 2. Camera and radar sensors setup.

Based on the bounding box and cluster data obtained from the preprocessing, the next step is to associate both sensors' data using the data fusion method. The sensor data fusion provides a complete dataset of point clouds with class labels. This dataset is utilized to train a deep learning model for object classification and is divided into two subsets: the training set and the test set. The entire experimental workflow is illustrated in Figure 3.

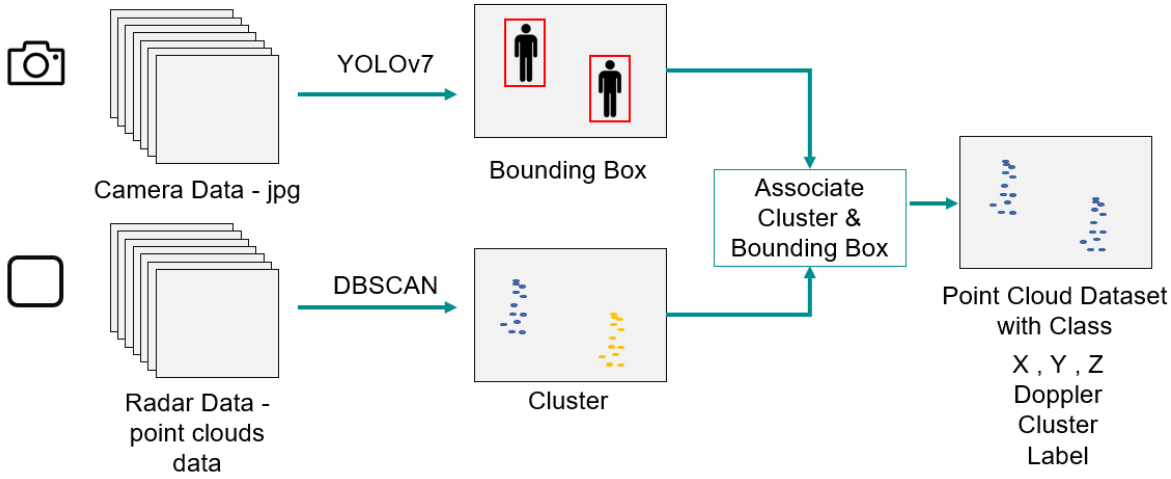

Figure 3. Experimental workflow for point cloud dataset preparation.

3. MEASURED DATA PROCESSING

Both sets of data require different processing approaches to prepare the point cloud data and camera data gathered from the measurement. Referring to the experimental workflow block

https://doi.org/10.31436/iiumej.v26i1.3268

diagram, the data fusion commences after processing the sensor data. The point cloud data from the radar sensor undergoes clustering through DBSCAN. Image data captured by a mono camera is processed using YOLOv7 for object classification and recognition.

3.1. DBSCAN

Retina-4F point cloud data output has the following information: timestamp, Cartesian coordinates (X, Y, and Z), Doppler, and Power Intensity. Lidar point cloud clustering is usually used for segmentation to distinguish different segments of the point cloud features, for instance, to segment the car point cloud between the tires and body of the car. However, in this project, the clustering process focuses on grouping the point cloud solely based on the type of objects. This means that the point cloud of each object belongs to different clusters. The point cloud segmentation is unnecessary for the fusion process with image data. Moreover, the sparsity of the radar point cloud is still too large compared to the lidar point cloud, which makes it very complex for segmentation.

One commonly known method for clustering non-linear data is DBSCAN [26]. The algorithm used in DBSCAN finds the nearest neighbor around each point of the point cloud and defines the cluster by connecting the points that are within the range. The region can be defined by adjusting DBSCAN parameters known as Epsilon and minimum points. Epsilon defines the radius around a point, and the algorithm will use that radius to search for neighboring points. Any points within the radius region will be part of the same cluster. Minimum points refer to the minimum number of points within a core point's radius to form a cluster. This is where the algorithm defines noise's presence in the point cloud. By identifying a core point and having a minimum number of points within the radius to form a cluster, it will be considered noise if the minimum number of points is not satisfied.

In this experiment, the optimal values for epsilon and minimum point chosen were 1.5 and 5, respectively. Next, the point cloud data can be processed for clustering after defining the epsilon and minimum point values. For clustering, four features of the point cloud data are passed to DBSCAN: the Cartesian coordinates of the point (x, y, and z) and Doppler velocity.

DBSCAN's point cloud clustering produces a cluster of points; hence, preparing the dataset for neural network classification algorithm training and testing is insufficient. To aid the neural network in recognizing and predicting the type of class a point cloud belongs to, a further step is required to provide a class label for the point cloud data. Therefore, YOLOv7 is incorporated into the workflow to create a dataset of point clouds with a class label.

3.2. YOLO-V7

The mono-camera's image data must be processed with an object detection algorithm to get the bounding box information for labeling the point cloud to its class. In this experiment, we use YOLOv7, a real-time object detection known for its high-speed processing and accuracy [27]. The YOLOv7 model is pre-trained using the Microsoft COCO dataset [28] and has around 90 classes of objects. Hence, objects that are of interest, such as cars, pedestrians, trucks, and buses, can be easily acquired from this model. By performing the object detection task on the image collected by the mono-camera, it gives information on bounding boxes, type of class, and class probability for all the objects recognized in the image. Most users used YOLOv7 after its release due to its high accuracy and efficiency in processing up to 160 frames per second of real-time data.

https://doi.org/10.31436/iiumej.v26i1.3268

The neural network in YOLOv7 uses the forward propagation method compared to another neural network, which normally uses both forward and backward propagation to perform predictions. The algorithm will first find and locate objects in the frame based on the input, either image or video. This process is usually called object detection. Correspondingly, the algorithm will generate bounding box predictions as well as the labels and confidence thresholds of the objects. This process is called object recognition. However, these two processes happen together as YOLOv7 is based on a one-stage architecture. As seen in Figure 4, the YOLO network architecture consists of 24 convolutional layers and 2 fully linked layers.

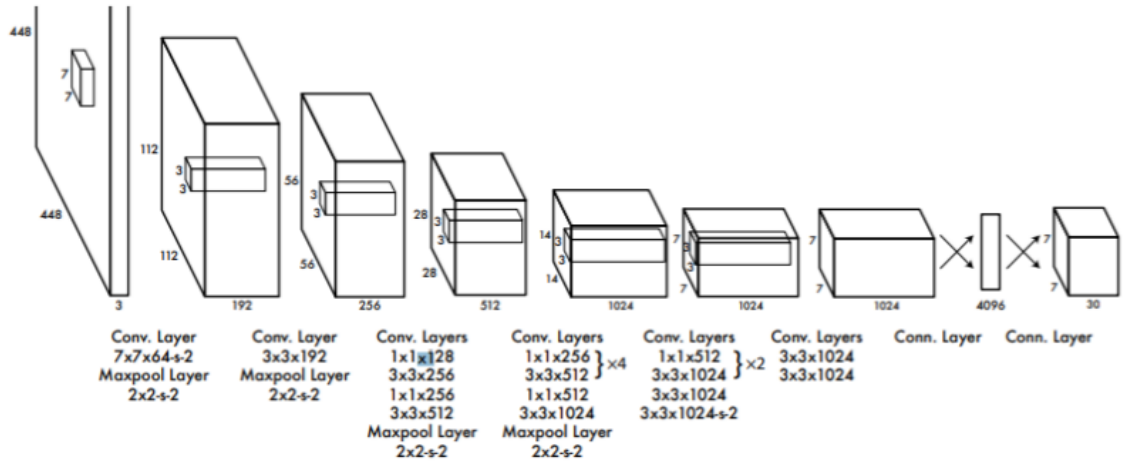

Figure 4. YOLO network architecture comprising Convolutional Neural Network (CNN) [29].

Based on the input, YOLO divides the image into grid cells, and a grid cell is in charge of detecting an object if its center falls within that grid cell. YOLO then predicts the bounding box and confidence scores of the box that encloses the object detected in the grid cell. The network defines the bounding boxes given by the center coordinate of the bounding box (x, y), the width (w), and the height (h). The confidence score is defined by the probability of an object with respect to the class label that is present inside the bounding box. To get the prediction, the YOLO network predicts the class label in each grid cell as the YOLO model is already pre-trained with many types of objects that allow the neural network to be able to classify objects in the input image into its specific labels.

Due to the prediction done by the YOLO network for each grid cell, multiple bounding boxes are generated. The network needs to assign just one bounding box predictor to handle each object during training. Based on whose prediction has the highest current intersection over union (IOU) with the ground truth, YOLO designates one predictor for predicting an object. Using the IOU method allows the network to improve its accuracy in detecting objects with different proportions. Another approach that YOLO uses to tackle the multiple bounding box generation is by implementing a post-processing step known as Non-Maximum Suppression (NMS). NMS is used mainly to eliminate redundant and overlapping bounding boxes by suppressing the bounding that has the largest IOU with the current highest probability bounding box. This guarantees that each object is only detected once, and one bounding box is obtained. As a result, YOLO produces information about the object's presence in an image, including details about the object's bounding box location and size, class label, and confidence score.

https://doi.org/10.31436/iiumej.v26i1.3268

4. PROJECTION AND CLASSIFICATION USING DEEP NEURAL NETWORK

We propose using point cloud projection onto an image to classify the point cloud data after clustering. This part of the work involves the fusion of the camera and radar data. The first step of this work is to calibrate the camera and radar before taking measurements at the road crossings. Calibration is crucial as we want accurate position and orientation of both sensors with respect to the measurement scene. From here, the intrinsic and extrinsic parameters of the sensors can be acquired to help in the projection process; this calibration method is usually known as Extrinsic Calibration. The significance of the calibration lies mainly in the misalignment between the camera data and radar data. A more detailed explanation of the projection work will be further discussed below.

4.1. Point Cloud Projection

Figure 5 shows the block diagram of the steps taken to project point cloud data onto the image. The point cloud data must be transformed from radar Cartesian coordinates (XR, YR, ZR) into a camera coordinate system (XC, YC, ZC) and then to image coordinates. The coordinate system between the radar coordinate and camera coordinate system is significantly different, as presented in Figure 6. The X-axis of the camera and the radar coordinate system remain the same; however, it is observed that the Y-axis of the radar coordinate system is facing the same axis as the z-axis in the camera coordinate system. To transform the Z-axis of the radar coordinate system into a camera coordinate, it is facing downwards with respect to the camera's principal axis.

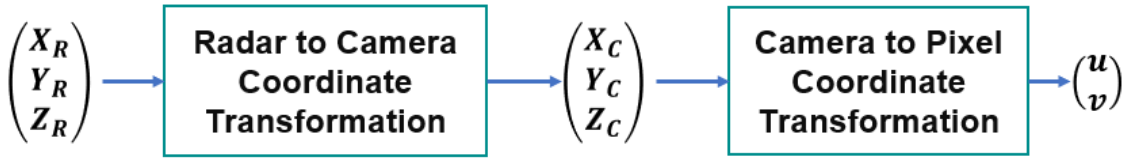

Figure 5. Radar to pixel coordinate system transformation.

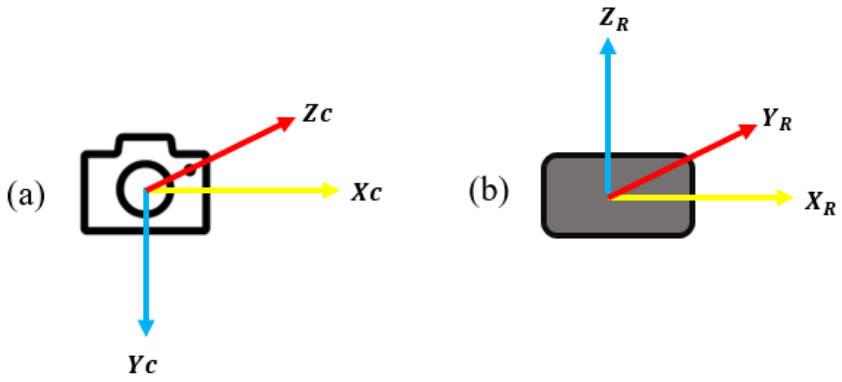

Figure 6. Difference between camera and radar coordinate system. (a) Camera coordinate system. (b) Radar coordinate system.

From here on, the intrinsic matrix (K), rotation matrix (R), and translation matrix (t) are the main parameters to construct the camera projection matrix [30]. Assuming that a scatterer point exists in an image coordinate given by (X_C, Y_C, Z_C), we want to project the point into a 2D

https://doi.org/10.31436/iiumej.v26i1.3268

image plane. Hence, there is a need to transform from image coordinate to pixel known as $[u, v]$. The transformation matrix can be obtained from the camera-radar calibration and represented as

$$[u, v, w]^T = K[R|t][x, y, z, 1]^T \quad (1)$$

where K is the intrinsic parameter of the camera obtained by camera-radar calibration represented as

$$K = \left[f_x, 0, c_x, 0, f_y, c_y, 0,0,1\right] \quad (2)$$

where f_x and f_y are the camera's focal length along the x-axis and y-axis. This focal length determines the camera's zoom level and horizontal field of view. Moreover, cx and cy represent the principal points of the x-axis and y-axis, where the optical axis and image plane intersect. This point is presented in pixels. Note that the Rotation and Translation, $[R|t]$ matrix is concatenated, where it is known as an extrinsic parameter obtained after calibration. The Rotation matrix is represented as:

$$\begin{gathered} R = R_x R_y R_z \\ R_x = [1,0,0,0, \cos\theta\ , -\sin\theta\ , 0, \sin\theta\ , \cos\theta] \\ R_y = [\cos\theta\ , 0, \sin\theta\ , 0,1,0, -\sin\theta\ , 0, \cos\theta] \\ R_z = [\cos\theta\ , -\sin\theta\ , 0, \sin\theta\ , \cos\theta\ , 0,0,0,1] \end{gathered} \quad (3)$$

and the translation matrix is represented as:

$$t = \left[t_x, t_y, t_z\right]^T \quad (4)$$

Point cloud projection onto an image can be made after calibrating the camera-radar sensor using the Extrinsic Calibration method and converting the 3D point cloud coordinate into the 2D image coordinate. Figure 9 (left) – Figure 16 (left), which shows the 3D point cloud projected onto a 2D image, demonstrates this. From here on, the point cloud cluster that is contained inside the bounding box may be retrieved using the bounding box information that we obtained using the YOLOv7 technique. The center coordinate of the bounding box shows the location of the bounding box in 2D image coordinates, along with the width and height.

Through simple data processing using Python, after transforming the 3D point cloud onto the image, any point cloud cluster within the bounding box can be extracted. At this stage, the point cloud cluster will be labeled according to the type of class of target in the image included within the bounding box. After a few processing steps, this produces a custom dataset of radar point clouds with class labels, which can be used for neural network classification training and testing.

The custom dataset comprises radar data containing point cloud clusters, target labels, Doppler velocity, and power intensity. The entire dataset is subsequently divided into two subsets, which are the training set and the testing set. This division allocates 80% of the dataset for training purposes and leaves the remaining 20% for testing. The distribution of data points for each class for both subsets is presented in Table 2 and shown in Figure 7. It can be observed that the pedestrian class makes up around 28% of the dataset. In comparison, cars and big vehicles (such as buses and trucks) constitute approximately 25% and 13% of the dataset, respectively. The clutter class comprises 34% of the dataset.

https://doi.org/10.31436/iiumej.v26i1.3268

Table 2. Number of Data Points for Each Classes

	Training	Testing	Total
Person	203 341	51 108	254 449
Car	183 126	45 632	228 758
Big Vehicle	93 085	23 248	116 333

Figure 7. Illustration of the custom dataset distribution.

The imbalance in class frequencies observed in the point cloud dataset, which includes people, cars, buses, and trucks, can be linked to two primary sources. To begin with, it is common for the distribution of these classes in real-world scenarios to display a natural imbalance. In road crossing settings, there is often a greater majority of pedestrians and cars than buses and trucks. As a result, the imbalanced representation of the custom dataset may effectively reflect the real-world condition. Furthermore, it is important to acknowledge that data-gathering processes have the potential to induce bias, which might subsequently result in class imbalance. This potential bias may arise due to several factors related to the data-gathering process, including the selection of the timing of data collection. However, the overall custom dataset is satisfactory enough to train and test deep neural networks.

4.2. Deep Learning Model Using TensorFlow and Keras

In this work, we develop a deep learning model using an open-source machine learning library known as Keras. To be more specific, Keras is a neural network library that is built on top of TensorFlow [31]. Keras uses the Python programming language, which makes it a more popular choice for an application programming interface (API) than any other neural network library, such as PyTorch. The integration between Keras and TensorFlow makes this open-source machine-learning library more advantageous. TensorFlow works as the backend, providing low-level computational complexity on either CPUs or GPUs to train neural networks, while Keras is the frontend, which offers a high-level and user-friendly API.

The network architecture of Keras as a deep neural network model is called a Feedforward Neural Network (FNN) [32], as presented in Figure 8. The network is made up of several layers

https://doi.org/10.31436/iiumej.v26i1.3268

consisting of neurons in each layer and fully connected to each neuron in the following layers. In other words, the neurons in the first layer are connected to adjacent neurons in the following layers, but they are not connected to the neurons within the same layer. This explains why this network architecture is known as a feedforward neural network since it goes from the input layer to the output layer in a single direction.

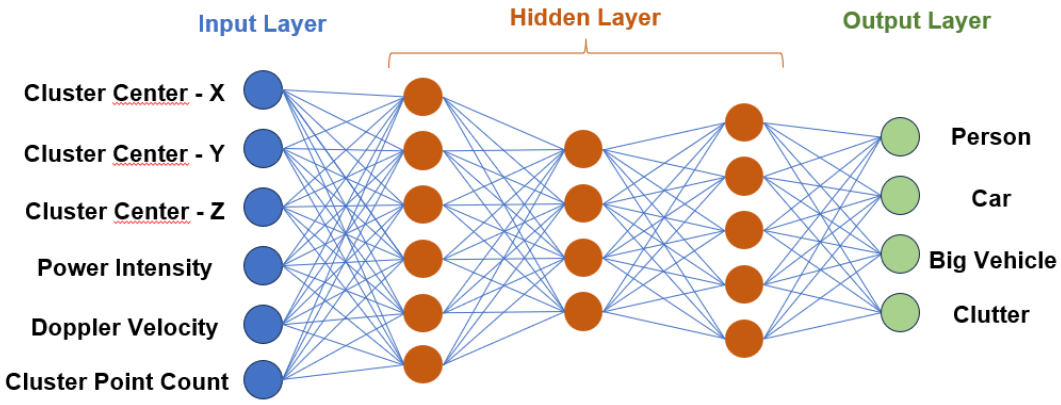

Figure 8. Feedforward neural network architecture.

The feedforward neural network has three main layers: the input, hidden, and output layers. As the name suggests, the input layer will receive the input data the user passes. In this work, we pass point cloud features to the input layer, which is the point cloud's cluster center coordinates: X, Y, and Z, Doppler velocity, power intensity, and cluster point count. The presence of neurons in the input layer depends on the dimension of the input. For instance, in this work, the dimension size is 6; hence, there are 6 neurons in the input layer. The hidden layer refers to the layers between the input and output layers, which perform computational tasks and nonlinear transformations of the input data.

The neural network's final layer, the output layer, focuses mostly on providing the network's final prediction. The neurons in this layer are determined by the user's desired prediction for the network's final result. Binary classification tasks and multi-class classification are the two categories into which it is typically split. In the output layer of a binary classification task, there is often simply one neuron that provides the likelihood that the input belongs to one of the two classes. More than one neuron is often included in the output layer of a multi-class classification task to determine which specific class the input data would belong to from more than two classes [33].

To handle the numerous kinds of classes, we want the neural network to learn and predict in this study; we employ multi-class classification. As a result, our model's final output includes four classes: person, car, big vehicles, and clutter. Since there are four classes that need to be distinguished, the final layer of the neural network will consist of four nodes for each class, as was previously stated.

The fundamental component of the neural network is the activation function within the hidden layers. The deep neural network designed utilizes ReLU (Rectified Linear Unit) activation functions in this study. The Rectified Linear Unit (ReLU) is a commonly employed activation function recognized for its straightforwardness and efficiency in deep learning. The network integrates non-linearity by enabling positive values to remain undisturbed while assigning a value of zero to negative values [34]. This characteristic assists in reducing the issue of vanishing gradients and helps the development of the neural network during the

https://doi.org/10.31436/iiumej.v26i1.3268

training process. In short, this is how feedforward neural networks work: the user passes on the data features to the layers, and each neuron in each layer of the neural network completes its task by weighing the inputs, applying activation functions, and sending the output that has been transformed to the following layer.

5. RESULTS AND ANALYSIS

This study collects and processes images and radar point clouds for target classification. We trained the model with the dataset obtained after the data fusion processing. Two road-crossing scenes are used for data collection, and the dataset is randomly divided into a training and testing set. The combination of both scenes gives a dataset consisting of approximately 10,000 frames.

In our Python data processing system, each target has been labeled using distinct colors representing different classes. In this context, people are associated with the color blue on the plot, cars are represented by the color red, orange denotes buses, and trucks are represented by the color purple on the plot. In addition, aside from the point cloud associated with the targets, all other point clouds of the background and uninterested targets are now assumed to be cluttered and colored grey.

Figures 9 to 16 display the outcomes from the data fusion, including 3D point cloud projection onto a 2D image and plotting point cloud data with labeling of several targets. Figure 9 (left) to Figure 16 (left) display targets, bounding boxes, and point clouds. Based on the projection process of the point cloud onto the 2D image, the point cloud that is inside the bounding box is extracted using simple Python processing and labeled as the target according to the label of the bounding box. We now have a point cloud dataset with points relevant to the target being labeled appropriately due to the processing. This can be observed by the point cloud labeled as several targets in Figure 9 (right) to Figure 16 (right). The point cloud related to the targets is accurately mapped when plotted along the X-Y axis.

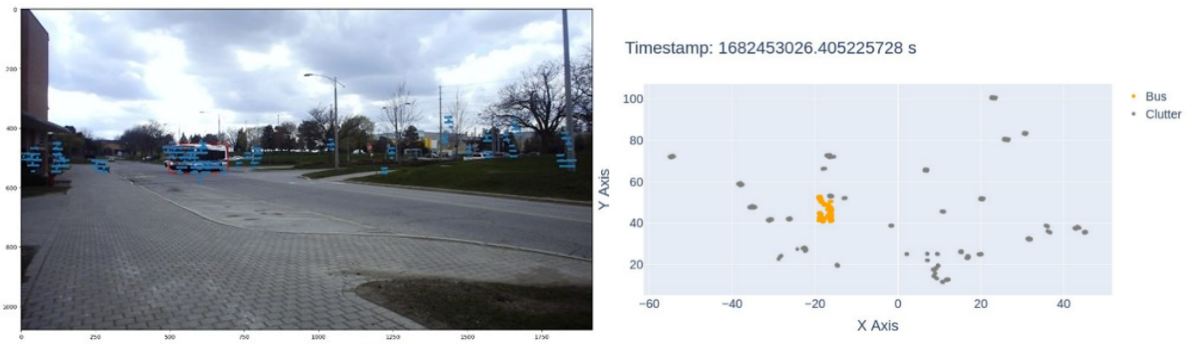

Figure 9. Bus in scene 1. (left) 3D Point cloud projection onto the 2D image. (right) X-Y axis scatter plot for point cloud dataset with target class.

https://doi.org/10.31436/iiumej.v26i1.3268

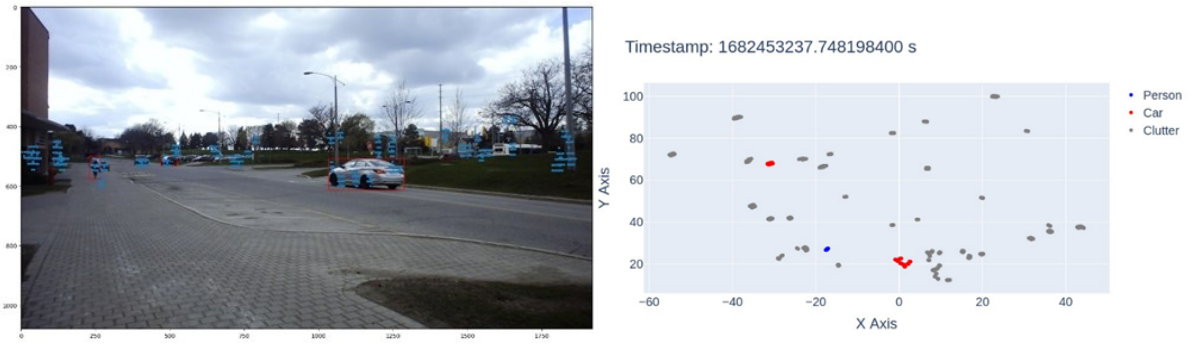

Figure 10. Car in scene 1. (left) 3D Point cloud projection onto the 2D image. (right) X-Y axis scatter plot for point cloud dataset with target class.

Figure 11. Pedestrian in scene 1. (left) 3D Point cloud projection onto the 2D image. (right) X-Y axis scatter plot for point cloud dataset with target class.

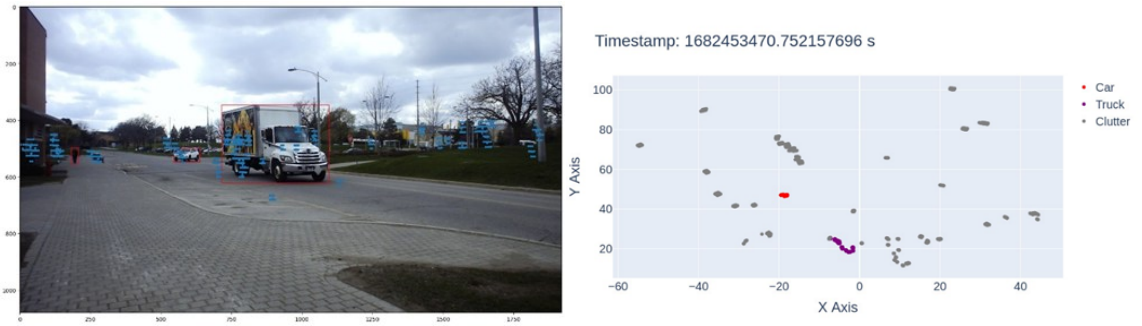

Figure 12. Truck in scene 1. (left) 3D Point cloud projection onto the 2D image. (right) X-Y axis scatter plot for point cloud dataset with target class.

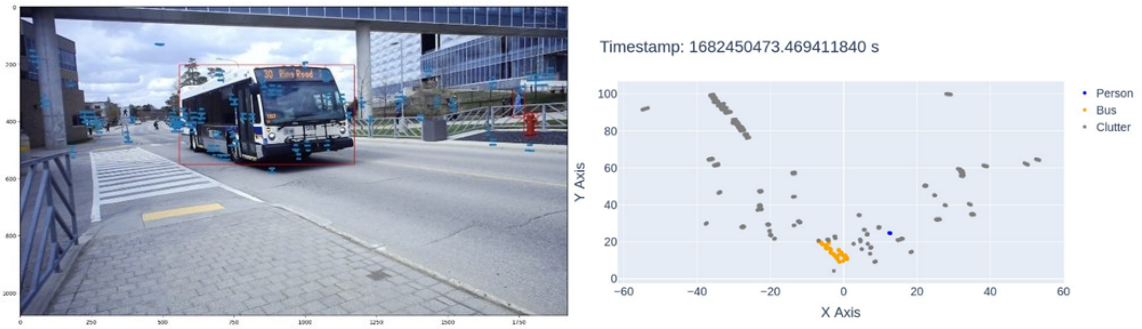

Figure 13. Bus in scene 2. (left) 3D Point cloud projection onto the 2D image. (right) X-Y axis scatter plot for point cloud dataset with target class.

https://doi.org/10.31436/iiumej.v26i1.3268

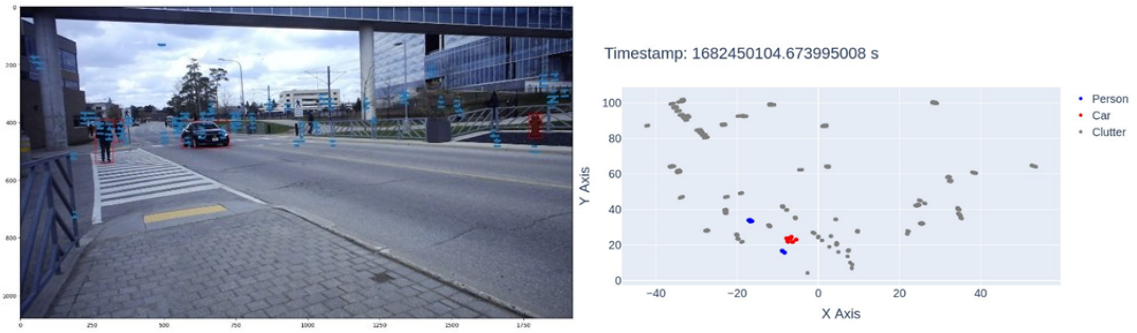

Figure 14. Car in scene 2. (left) 3D Point cloud projection onto the 2D image. (right) X-Y axis scatter plot for point cloud dataset with target class.

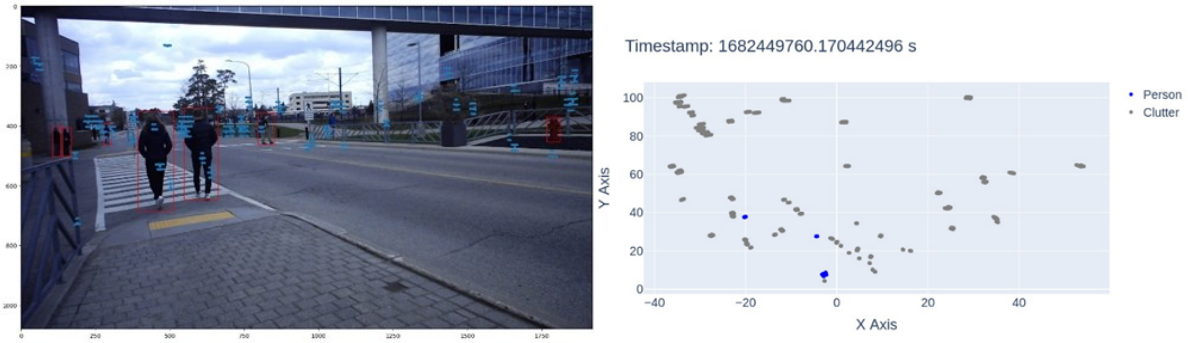

Figure 15. Pedestrian in scene 2. (left) 3D Point cloud projection onto 2D image. (right) X-Y axis scatter plot for point cloud dataset with target class.

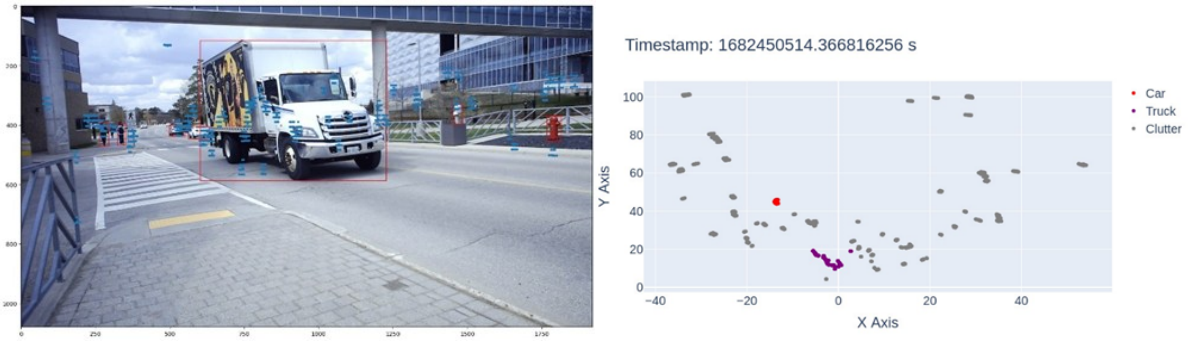

Figure 16. Truck in scene 2. (left) 3D Point cloud projection onto the 2D image. (right) X-Y axis scatter plot for point cloud dataset with target class.

This work aims to classify 4D point cloud data using a deep learning method to monitor road crossings. Following data fusion and the subsequent generation of a customized dataset, the next step involves training and testing the deep neural network. The model performed training with different numbers of layers for 10 epochs using a training dataset consisting of over 10,000 frames of point cloud data. Using 10 epochs is due to the computational resource hardware limitation. 1 epoch has taken an average of 307 seconds, making training deep neural networks with a large dataset require too much time. Subsequently, predictions were made on the radar point cloud using the training set.

https://doi.org/10.31436/iiumej.v26i1.3268

In this work, we construct 4 neural network layers to classify the object into respective classes. Figure 17 displays the confusion matrix, which provides an overview of the model's predictions for classifying point cloud clusters as either a person, car, or big vehicle. Based on an analysis of the confusion matrix, it can be observed that the model successfully predicts point cloud clusters of pedestrians as persons with an accuracy of 98.09%. However, there is still a notable misclassification rate of approximately 1.91 % in which the pedestrian cluster is erroneously identified as other objects. Next, the 4 layers model illustrated high accuracy, predicting the point cloud cluster of cars as cars with a precision of 94.79%. Despite the misclassification of person and car class within the corresponding class, its impact does not greatly affect the model's performance.

In addition, in the classification of large vehicles, which consists of bus and truck class types, a significant amount of misclassification is still observed. The model for the large vehicle class type demonstrates a predictive accuracy of 78.41 %. Nevertheless, a significant proportion of instances involve the incorrect classification of big vehicles as cars, up to 21.08 %.

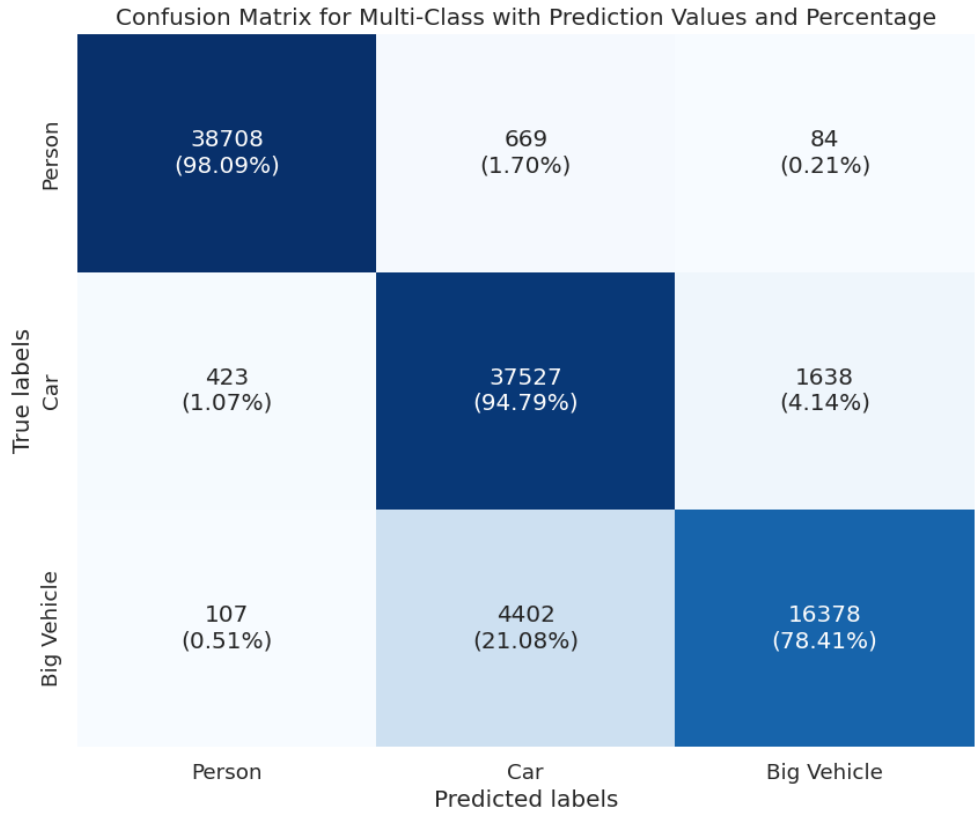

Figure 17. Confusion Matrix of 4 Layers Neural Network Model

Evaluating a deep learning model is facilitated using a confusion matrix, which is crucial for gaining valuable insights into the model's predictive capabilities. The confusion matrix provides essential information for assessing the success of a model. It allows us to calculate important metrics such as precision, accuracy, and F1 score, which comprehensively evaluate the model's performance. Additionally, in the context of multi-class classifications, it is

https://doi.org/10.31436/iiumej.v26i1.3268

important to consider the average metric that provides a comprehensive understanding of the performance exhibited by a neural network model designed to classify multiple classes.

Firstly, as a quantitative measure, it evaluates the model's ability to identify true positives among all positive predictions accurately. The attribute of high precision indicates that the model has a low rate of misclassifying negative instances as positive ones. This characteristic is significant in situations where the occurrence of false positives is undesirable. Next, recall is a metric that measures the model's ability to correctly detect all instances that truly belong to a specific category. High recall is crucial when recording many true positive instances, even at the cost of accepting some false positive instances. Lastly, the F1-score is a measurement that effectively balances precision and recall. The metric offers a comprehensive evaluation of a model's effectiveness, which is particularly advantageous in scenarios with imbalanced datasets or where there is a requirement to achieve a balance between precision and recall. F1-score is computed by taking the harmonic mean of precision and recall, providing a unified measure that reflects the balance between these two crucial elements of model efficacy.

Furthermore, based on the model evaluation shown in Table 3, the micro-average, macro-average, and weighted-average F1 scores are present. These findings imply a considerable level of proficiency exhibited by the model in its entirety. The micro-average score measures the model's effectiveness in producing accurate predictions for all categories without exhibiting bias towards any specific class. In this context, the micro-average represents the accuracy of the model's performance. The macro-average F1 score is calculated by determining the F1 score for each class and then taking the average. This metric provides evidence that the model demonstrates consistent performance across different classes. The calculation of the weighted average score addresses the issue of class imbalance by assigning a higher weight to classes with a larger number of occurrences.

Table 3. Model Evaluation Report (4 layers)

	Precision	**Recall**	**F1-Score**
Person	0.99	0.76	0.86
Car	0.88	0.82	0.85
Big Vehicle	0.90	0.70	0.79
Micro avg	0.93	0.77	0.84
Macro avg	0.92	0.76	0.83
Weighted avg	0.93	0.77	0.84

The 4-layer model, which consists of layers with 64, 32, 16, and 4 neurons, respectively, shows a significant improvement with a micro-average F1-score of 0.84. The additional layer in this model helps enhance its representational capacity, allowing it to capture more intricate features and relationships in the data. This higher score suggests that the model's increased depth enables better learning and generalization. The balance achieved in this model between complexity and performance indicates that it is well-suited for the given task, effectively leveraging the additional layer to improve predictive accuracy without introducing substantial overfitting.

The 4-layer model's superior performance with a micro-average F1-score of 0.84 suggests it strikes the right balance between model complexity and performance. The additional layer allows it to learn more sophisticated features without overcomplicating the training process or leading to overfitting. The 3-layer model, while simpler, performs reasonably well with a

micro-average F1-score of 0.81, indicating that it is effective but might benefit from a slight increase in complexity to improve performance further. On the other hand, the 5-layer model's performance, with a micro-average F1-score of 0.80, suggests that its added complexity does not translate into better performance, possibly due to overfitting and optimization difficulties.

Comparing our study with related work from [35], it becomes evident that our approach has several notable differences and improvements. The work aims to evaluate the performance of radar-based object categorization in real-world traffic monitoring, demonstrating the efficacy of Gaussian Mixture Models (GMM) in processing radar point cloud data despite the difficulties caused by noise and clutter.

The benchmark study and this research contrast notably regarding the radar devices utilized. The benchmark study used the Texas Instrument AWR1843BOOST radar chipboard, but this study employed the TI AWR2243. This differentiation suggests that each research may gain advantages from varied capabilities and performance criteria inherent to their corresponding radar evaluation boards. In terms of measurement resolutions, the benchmark study achieved a range resolution of 9 cm, a Doppler resolution of 0.8 m/s, an azimuth angle resolution of 15°, and an elevation angle resolution of 28°. Conversely, this study proposed using radar with a coarser range resolution of 50 cm but a significantly finer Doppler resolution of 0.06 m/s, an azimuth angle resolution of 2°, and an elevation angle resolution of 4.7°. In addition, the types of objects considered in each study also vary. The benchmark study focused on pedestrians and cars, while this study expanded the scope to include buses and trucks. This study's broader range of transportation modes indicates a more complex and challenging classification task, which could lead to more robust and versatile detection capabilities.

In terms of data collection, the benchmark study collected 8035 frames of training data over 13 minutes and 1222 frames of testing data over 2 minutes. However, in this study, the data is collected over 10,000 frames with a total measurement time of 2.5 hours, indicating a more extensive data collection process. This larger dataset could contribute to more robust training and testing, potentially leading to more reliable classification outcomes. Data processing tools and methods also varied between the studies. The benchmark study utilized Scikit-learn APIs to fit GMM for classification, an unsupervised learning method. In contrast, in this work, DBSCAN is used for clustering radar point cloud data and YOLOv7 for image processing and labeling, combining sensor fusion for data association. This approach will likely lead to more accurate and reliable classifications, combining unsupervised clustering with supervised labeling. Moreover, the classification methods employed were different. The benchmark study used GMM, which, as an unsupervised learning method, does not necessarily align predicted labels with ground truth. In this work, we applied supervised learning using KERAS and TensorFlow for deep neural network models. Given the nature of supervised learning, this approach likely results in more accurate classifications.

The confusion matrix from the benchmark study in Figure 20 presents the classification accuracy for three categories: clutter, car, and pedestrian. The true positive rates for clutter, car, and pedestrian are 89%, 61%, and 93%, respectively. The matrix reveals several key points. Firstly, the benchmark study accurately detects clutter, with 89% of true clutter instances correctly classified. This high detection rate indicates that the model distinguishes irrelevant data points from actual objects. Secondly, the car detection accuracy is moderate at 61%. This lower accuracy suggests that the model has difficulty distinguishing cars from pedestrians, as evidenced by 34% of cars being misclassified as pedestrians. Lastly, pedestrian detection shows a high accuracy of 93%, indicating that the model performs well in identifying pedestrians within the radar point cloud data. Overall, the benchmark study's results indicate

https://doi.org/10.31436/iiumej.v26i1.3268

that while the model is proficient in identifying pedestrians and clutter, it struggles with accurately classifying cars, often confusing them with pedestrians.

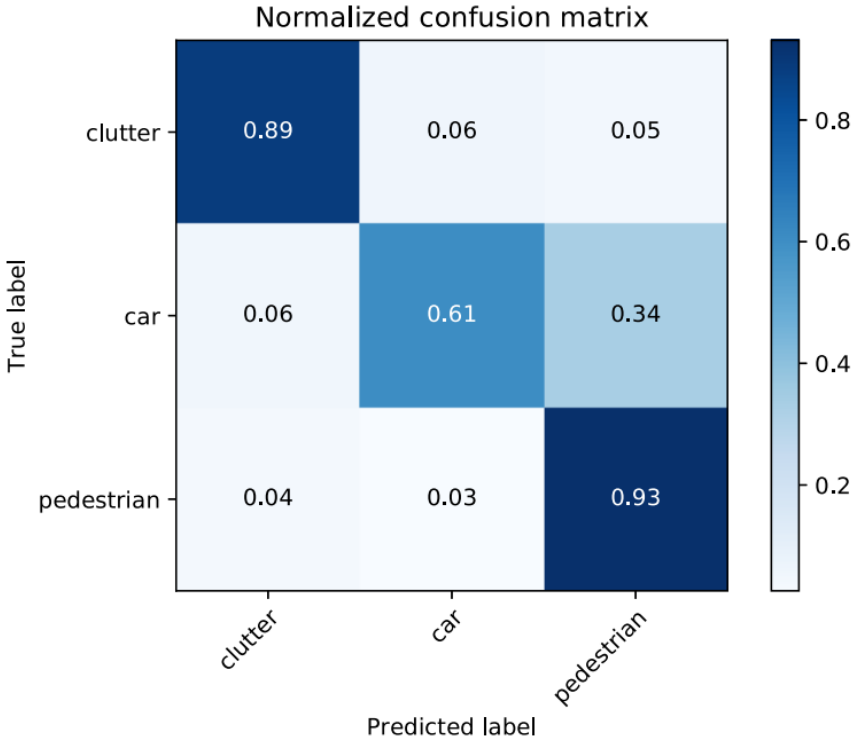

Figure 20. Confusion Matrix [35]

In contrast, this study's confusion matrix includes three categories: person, car, and big vehicle. The true positive rates for a person, car, and big vehicle are 98.09%, 94.79%, and 78.41%, respectively. Several critical points emerge from this matrix. Firstly, the person detection rate is exceptionally high at 98.09%, demonstrating that the model is highly effective at identifying individuals within the radar data. Secondly, the car detection accuracy is also high at 94.79%, significantly improving over the benchmark study. This improvement suggests that using YOLOv7 for image processing, DBSCAN for clustering, and sensor fusion has enhanced the model's capability to classify cars accurately. Lastly, detecting big vehicles (buses and trucks) shows a lower accuracy of 78.41%. This lower accuracy indicates that while the model performs well, there is still room for improvement in distinguishing between large vehicles and other categories.

In terms of performance and results, the confusion matrix in this study shows marked improvements in the detection of cars and persons compared to the benchmark study. Specifically, the car detection rate of 94.79% is significantly higher than the benchmark study's 61%, highlighting the effectiveness of this work model's approach. The person detection accuracy in this study (98.09%) also surpasses the pedestrian detection accuracy in the benchmark study (93%), indicating superior performance in this category. Although there is an additional challenge of detecting big vehicles in this work, the accuracy rate of 78.41% suggests that while effective, there is still potential for further refinement in distinguishing large vehicles from other categories.

In the benchmark study, the precision, recall, and F1-score for detecting persons were 0.85, 0.93, and 0.89, respectively, as shown in Table 4. This indicates a high recall, suggesting that the benchmark model was effective at identifying most instances of persons, but the precision

https://doi.org/10.31436/iiumej.v26i1.3268

was slightly lower, meaning there were some false positives. For car detection, the precision was 0.88, the recall was 0.61, and the F1-score was 0.72, showing a significant drop in recall compared to precision. This suggests that the benchmark model struggled more with correctly identifying cars, possibly confusing them with other categories. Clutter detection had a precision of 0.71, recall of 0.89, and an F1-score of 0.79, indicating a higher tendency to identify clutter correctly but with a relatively lower precision, reflecting a notable number of false positives.

In comparison, this work demonstrated improved performance metrics across all categories. The precision, recall, and F1-score for person detection were 0.99, 0.76, and 0.86, respectively, as shown in Table 3. While the recall is lower than the benchmark study, the precision is significantly higher, suggesting the model produces fewer false positives when detecting persons. For car detection, the model achieved a precision of 0.88, recall of 0.82, and F1-score of 0.85, showing a balanced and robust performance. This represents a substantial improvement in recall over the benchmark study, indicating the model's enhanced capability to identify cars correctly. For big vehicle detection, the precision was 0.90, the recall was 0.70, and the F1-score was 0.79. Although this category was not included in the benchmark study, the metrics indicate a solid performance in identifying large vehicles, albeit with a lower recall than precision.

Table 4. Model Evaluation Report [35]

	Recall	**F1-Score**
Person	0.93	0.89
Car	0.61	0.72
Clutter	0.89	0.79

This study also reports micro, macro, and weighted averages for the classification performance. The micro average precision, recall, and F1-score are 0.93, 0.77, and 0.84, respectively. These metrics provide an overall performance evaluation considering all instances across categories, showing a strong general performance with high precision and good recall. The macro averages, which consider the performance across categories equally, were 0.92, 0.76, and 0.83 for precision, recall, and F1-score, respectively. These averages reflect the model's balanced performance across different categories. The weighted averages, which take into account the proportion of each category, were 0.93, 0.77, and 0.84, respectively, indicating that the model maintains high performance even when the category distribution is considered.

In summary, the comparative analysis shows that this study's classification model demonstrates superior precision across all categories compared to the benchmark study. This indicates that the model in this work produces fewer false positives, particularly for person and car detection. The recall for car detection in this work is significantly improved, highlighting the model's enhanced capability to identify cars correctly. While the recall for person detection is lower than the benchmark, the higher precision compensates for this by reducing false positives. Including big vehicle detection in this study adds a layer of complexity, with good performance metrics indicating the model's capability to handle more diverse categories. The overall averages further underscore the robustness and effectiveness of this work classification model, showing high precision and balanced performance across all categories.

6. CONCLUSION

With the rigorous technical improvement of the presented approach on radar-powered intelligent infrastructure, diverse application domains, from autonomous navigation to microclimate monitoring, can benefit from enhanced environmental awareness.

This work demonstrates a target detection and classification approach by fusing 4D radar imaging with deep neural networks. This work is highly motivated to solve challenges relating to point cloud classification using a deep neural network as part of a monitoring system. With the presence of high-resolution radar imaging in the automotive market, given its advantages, there is a trend in research work toward developing a radar-based monitoring system with object classification capability. However, researchers face the challenges of object classification using radar imaging due to the sparsity characteristic of the point. This work proposes a method to solve this challenge through sensor data fusion. We integrate radar imaging and cameras to create a customized dataset of point clouds and use it to train a deep neural network. In this dataset, all the spatial relationships and information of the radar point cloud data are preserved. Hence, the deep neural network employed in this work could learn the complex relationships of the point cloud data and make predictions.

The results show 84% classification accuracy on three common roadway objects, indicating feasibility but also room for improvement. First, the custom dataset can be expanded to improve generalizability. Additional data capture in varied weather conditions, times of day, and geographic locales will reduce overfitting and aid robustness. Second, augmenting the radar point cloud with multi-perspective views from a lidar sensor could provide complementary geometric cues to resolve ambiguities. Third, modifying the deep network architecture itself may boost accuracy. Attention mechanisms could help focus feature extraction on more discriminative spatial regions of the point cloud. Also, transformer networks have shown value in processing unordered point sets and may outperform convolutional architectures. Lastly, implementing the deep network on dedicated hardware like GPUs or FPGAs would enable real-time inference critical for practical systems. The framework developed here establishes a baseline—further optimization of the sensor fusion, data representations, and neural network design provides an exciting path toward deployable 4D radar perception systems.

REFERENCES

[1] “Road traffic injuries,” World Health Organization, 20-Jun-2022. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries. [Accessed: 25-Aug-2022].
[2] C. Waldschmidt, J. Hasch and W. Menzel, "Automotive Radar — From First Efforts to Future Systems," in IEEE Journal of Microwaves, vol. 1, no. 1, pp. 135-148, Jan. 2021.
[3] X. Gao, G. Xing, S. Roy and H. Liu, "Experiments with mmWave Automotive Radar Test-bed," 2019 53rd Asilomar Conference on Signals, Systems, and Computers, 2019, pp. 1-6.
[4] I. Bilik, O. Longman, S. Villeval and J. Tabrikian, "The Rise of Radar for Autonomous Vehicles: Signal Processing Solutions and Future Research Directions," in IEEE Signal Processing Magazine, vol. 36, no. 5, pp. 20-31, Sept. 2019.
[5] E. Ragonese, G. Papotto, C. Nocera, A. Cavarra and G. Palmisano, "CMOS Automotive Radar Sensors: mm-Wave Circuit Design Challenges," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 69, no. 6, pp. 2610-2616, June 2022.
[6] I. Bilik et al., "Automotive multi-mode cascaded radar data processing embedded system," 2018 IEEE Radar Conference (RadarConf18), 2018, pp. 0372-0376.

[7] Y. Cheng, J. Su, H. Chen and Y. Liu, "A New Automotive Radar 4D Point Clouds Detector by Using Deep Learning," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 8398-8402.
[8] S. Sun and Y. D. Zhang, "4D Automotive Radar Sensing for Autonomous Vehicles: A Sparsity-Oriented Approach," in IEEE Journal of Selected Topics in Signal Processing, vol. 15, no. 4, pp. 879-891, June 2021.
[9] M. Stolz, M. Wolf, F. Meinl, M. Kunert and W. Menzel, "A New Antenna Array and Signal Processing Concept for an Automotive 4D Radar," 2018 15th European Radar Conference (EuRAD), 2018.
[10] J. Martínez García, D. Zoeke and M. Vossiek, "MIMO-FMCW Radar-Based Parking Monitoring Application With a Modified Convolutional Neural Network With Spatial Priors," in IEEE Access, vol. 6, pp. 41391-41398, 2018.
[11] G. Li et al., "Novel 4D 79 GHz Radar Concept for Object Detection and Active Safety Applications," 2019 12th German Microwave Conference (GeMiC), 2019, pp. 87-90.
[12] X. Gao, S. Roy, G. Xing and S. Jin, "Perception Through 2D-MIMO FMCW Automotive Radar Under Adverse Weather," 2021 IEEE International Conference on Autonomous Systems (ICAS), 2021, pp. 1-5.
[13] J. Wu, Z. Zhu and H. Wang, "Human Detection and Action Classification Based on Millimeter Wave Radar Point Cloud Imaging Technology," 2021 Signal Processing Symposium (SPSympo), LODZ, Poland, 2021, pp. 294-299.
[14] Y. Kim, I. Alnujaim and D. Oh, "Human Activity Classification Based on Point Clouds Measured by Millimeter Wave MIMO Radar With Deep Recurrent Neural Networks," in IEEE Sensors Journal, vol. 21, no. 12, pp. 13522-13529, 15 June15, 2021.
[15] I. Alujaim, I. Park and Y. Kim, "Human Motion Detection Using Planar Array FMCW Radar Through 3D Point Clouds," 2020 14th European Conference on Antennas and Propagation (EuCAP), Copenhagen, Denmark, 2020, pp. 1-3.
[16] Z. Yu et al., "A Radar-Based Human Activity Recognition Using a Novel 3-D Point Cloud Classifier," in IEEE Sensors Journal, vol. 22, no. 19, pp. 18218-18227, 1 Oct.1, 2022.
[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proc IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2014, pp. 580–587.
[18] X. Cai, M. Giallorenzo and K. Sarabandi, "Machine Learning-Based Target Classification for MMW Radar in Autonomous Driving," in IEEE Transactions on Intelligent Vehicles, vol. 6, no. 4, pp. 678-689, Dec. 2021.
[19] “Smart Radar System,” srs.ai. https://www.smartradarsystem.com/kr/index.html (accessed Mar. 02, 2021).
[20] S. Lim, J. Jung, B. -h. Lee, J. Choi and S. -C. Kim, "Radar Sensor Based Estimation of Vehicle Orientation for Autonomous Driving," in IEEE Sensors Journal, 2022.
[21] X. Gao, G. Xing, S. Roy and H. Liu, "Experiments with mmWave Automotive Radar Test-bed," 2019 53rd Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 2019, pp. 1-6.
[22] Nabati, R., & Qi, H. (2021). Centerfusion: Center-based radar and camera fusion for 3d object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 1527-1536).
[23] M. Abdalwohab, W. Zhang, A. M. S. Abdelgader and I. Abdelazeem, "Deep learning based camera and radar fusion for object detection and classification," 2021 IEEE 4th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE), Shenyang, China, 2021, pp. 322-326.
[24] A. Sengupta, L. Cheng and S. Cao, "Robust Multiobject Tracking Using Mmwave Radar-Camera Sensor Fusion," in IEEE Sensors Letters, vol. 6, no. 10, pp. 1-4, Oct. 2022.
[25] J S. Guo, P. Wang, J. Ding and H. Liu, "Deep Model Based Road User Classification Using mm-Wave Radar," 2021 CIE International Conference on Radar (Radar), Haikou, Hainan, China, 2021, pp. 2843-2846.

[26] F. Jin, A. Sengupta, S. Cao and Y. -J. Wu, "MmWave Radar Point Cloud Segmentation using GMM in Multimodal Traffic Monitoring," 2020 IEEE International Radar Conference (RADAR), Washington, DC, USA, 2020, pp. 732-737.
[27] A. P. Rangari, A. R. Chouthmol, C. Kadadas, P. Pal and S. Kumar Singh, "Deep Learning based smart traffic light system using Image Processing with YOLO v7," 2022 4th International Conference on Circuits, Control, Communication and Computing (I4C), Bangalore, India, 2022, pp. 129-132.
[28] C. Wang, A. Bochkovskiy and H. Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," in 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 2023 pp. 7464-7475.
[29] Y. A. Khan, S. Imaduddin, A. Ahmad and Y. Rafat, "Image-based Foreign Object Detection using YOLO v7 Algorithm for Electric Vehicle Wireless Charging Applications," 2023 5th International Conference on Power, Control & Embedded Systems (ICPCES), Allahabad, India, 2023, pp. 1-6.
[30] J. Oh, K. -S. Kim, M. Park and S. Kim, "A Comparative Study on Camera-Radar Calibration Methods," 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, 2018, pp. 1057-1062.
[31] S. D. Boncolmo, E. V. Calaquian and M. V. C. Caya, "Gender Identification Using Keras Model Through Detection of Face," 2021 IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), Manila, Philippines, 2021, pp. 1-6.
[32] M. Si, T. J. Tarnoczi, B. M. Wiens and K. Du, "Development of Predictive Emissions Monitoring System Using Open Source Machine Learning Library – Keras: A Case Study on a Cogeneration Unit," in IEEE Access, vol. 7, pp. 113463-113475.
[33] S. -H. Chen, C. -S. Hung, J. -Y. Wang, C. -H. Chen and K. -C. Hsu, "The Implementation of Hybrid Electric Vehicle Battery Fault and Abnormal Early Warning System Using Keras Neural Network Technology," 2021 9th International Conference on Orange Technology (ICOT), Tainan, Taiwan, 2021, pp. 1-6.
[34] Wang, Y., Li, Y., Song, Y., & Rong, X. (2020). The influence of the activation function in a convolution neural network model of facial expression recognition. *Applied Sciences*, *10*(5), 1897.as
[35] Jin, F., Sengupta, A., Cao, S., & Wu, Y. J. (2020, April). Mmwave radar point cloud segmentation using gmm in multimodal traffic monitoring. In 2020 IEEE International Radar Conference (RADAR) (pp. 732-737). IEEE.