Vehicle Type Recognition using an Efficient Regularization in Mask_RCNN

Noraqilah Misman
Faculty of Computing, University Malaysia
Pahang Al-Sultan Abdullah, Malaysia
aqilahms@gmail.com

Suryanti Awang
Faculty of Computing, University Malaysia
Pahang Al-Sultan Abdullah, Malaysia
suryanti@umpsa.edu.my

Mohammed Khalaf
Department of Computer Science
University of Al Maarif, Iraq
m.i.khalaf@uoa.edu.iq

Hasan Kahtan
Cardiff School of Technologies
Metropolitan University, UK
hkahtan@cardiffmet.ac.uk

Abstract: *A Vehicle Type Recognition (VTR) system faces challenges in achieving accurate classification when distinguishing between vehicle types with intra-class patterns, such as sedan cars, taxis, vans, minivans, trucks, and buses. The main challenge lies in effectively extracting and preserving discriminant features for each vehicle type to prevent misclassification. Therefore, this paper proposes an efficient regularization approach within the Mask Region-based Convolutional Neural Network (Mask_RCNN) optimizer by integrating Weighted Mean League 2 (WMean_L2) with Stochastic Gradient Distance (SGD). We introduce this model as Mask_RCNN+SGD+WMean_L2. WMean_L2 is formulated to ensure consistency in penalty regardless of model size, providing stability across architectures and simplifying hyperparameter tuning. This approach enhances the preservation of discriminant features while achieving consistent and optimal classification performance. We tested our model on the benchmark database known as Beijing Institute of Technology (BIT), evaluating its performance based on precision, recall, F-score, and accuracy. Our results demonstrate significant efficiency improvements compared to previous studies, with precision ranging from 92.31% to 100%, recall from 94.74% to 100%, and F-score from 93.51% to 100% across six vehicle classes, achieving the highest average accuracy of 97.22%.*

Keywords: *Vehicle type classification, deep learning, optimized deep learning, computational intelligence*.

Received March 5, 2025; accepted August 6, 2025
https://doi.org/10.34028/iajit/23/1/2

1. Introduction

Vehicle Type Recognition (VTR) is important for many smart transportation use cases, such as vehicle tracking, toll collection, and urban planning [5]. High accuracy in classification, however, remains a challenge, especially when dealing with intra-class variation which is a subtle variation in a same vehicle type. For example, sedans and taxis have similar looks, and van and minivan have similar forms, with conventional model recognizability having a problem in distinguishing between them. Due to that, most researchers classify the taxi and the car as a sedan car, or the truck and bus as heavy vehicles [3, 10]. The consequence is, it is not efficient when implementing the system in the real implementation [1].

A key challenge in handling intra-class issues is effectively extracting and maintaining discriminant features between almost similar vehicles but in different vehicle types. Conventional classification algorithms suffer in extracting and maintaining the discriminant features. Deep learning techniques like Convolutional Neural Networks (CNNs) and Mask Region-based Convolutional Neural Network (Mask_RCNN) architectures often struggle with overfitting and underfitting when dealing with intra-class patterns, leading to poor and unpredictable classification performance. Additionally, traditional regularization methods in deep networks may not effectively preserve the discriminant features, which can worsen classification errors. This is because penalty process in the regularization applies uniform penalty to all weights, regardless of their importance

To address this challenge, we introduce an efficient regularization method aimed to improve vehicle type classification when dealing with the intra-class patterns, supporting applications in Intelligent Transportation System (ITS) such as toll collection, traffic census, and traffic light control [11]. We propose a new regularization scheme called Weighted Mean League 2 (WMean_L2) and combine it with Stochastic Gradient Distance (SGD) in the Mask_RCNN optimizer. The WMean_L2 is formulated in the regularization based on mean squared value during the penalty process. The WMean_L2 is not influenced by the total number of weights in the model, ensuring consistent penalties, better model training, and improved the preservation discriminant features. Thus, the contributions of this paper are:

- Efficient regularization is introduced by using mean squared value named WMean_L2.

- The WMean_L2 was implemented in optimization layer with SGD as the optimizer in Mask_RCNN and named as Mask_RCNN+SGD+WMean_L2.

2. Related Works

2.1. Mask_RCNN and Regularization in Vehicle Type Recognition

Deep learning has demonstrated considerable advantages in improving accuracy when extracting discriminative features from images. A region-based deep neural network, exemplified by the Mask_RCNN, has been widely employed to extract discriminative features from intra-class patterns [9, 11, 15]. In the realm of vehicle type classification, studies utilizing Mask_RCNN have achieved commendable accuracy in identifying general vehicle classes. However, challenges arise when aiming for precise classification into specific vehicle types, leading to a decline in accuracy.

Mask_RCNN has drawbacks that can burden the backbone network's weight throughout the feature extraction process. This is because Mask_RCNN is trained using various optimization algorithms, including RMSprop, Adam, and Momentum. While this segmentation algorithm can address feature extraction challenges by generalizing and processing large datasets more efficiently [6, 17], it introduces another limitation which is model complexity. Increased complexity in the training model leads to longer training times and a decrease in accuracy due to the loss of extracted discriminative features [7]. The complexity in the Mask_RCNN framework arises from the sensitivity of hyperparameters, which affect both efficiency and practical application [18]. To mitigate this limitation and enhance the performance of the Mask_RCNN framework, regularization techniques are employed. These techniques introduce penalties to the loss function to discourage overly complex models that may overfit the training data. Such penalties take various forms but serve the common purpose of preventing overfitting and promoting the learning of more generalizable patterns.

L2 regularization has been widely used in the Mask_RCNN framework due to its advantages over other regularization techniques. Shim *et al*. [13] implemented L2 regularization in Mask_RCNN to classify vehicle types in a traffic control system. However, their work struggled with misclassifications within truck categories due to limitations in feature discrimination and a narrow focus on cars, bicycles, and trucks. Tahir *et al*. [14] developed an intelligent vehicle system that applied Mask_RCNN for real-time vehicle detection. They deployed L2 regularization to optimize the training model, but their approach also failed to achieve high accuracy when dealing with intra-class vehicle variations.

Similarly, other studies have implemented Mask_RCNN with regularization for vehicle counting and classification [8]. This approach was tested on three different video datasets and achieved precision recognition results ranging from 97.3% to 99.1%. However, it was not tested on intra-class data, limiting the ability to assess the effectiveness of their Mask_RCNN model.

L2 regularization has also been utilized in a conventional CNN architecture for vehicle detection and classification using spatio-temporal information [16]. The regularization process was implemented after feature extraction, resulting in a less complex model. However, based on their results, this model was unable to improve classification precision. This is because the CNN extracts hierarchical features at lower layers, but the model did not learn discriminative features due to improper regularization of the higher layers.

Based on these studies, it has been demonstrated that L2 regularization has been widely utilized in Mask_RCNN for recognition and classification systems. However, despite its widespread use, the default L2 regularization technique still exhibits weaknesses that can be improved. L2 regularization works by summing the squares of all weights. This sum-of-squares approach penalizes the total sum of squared weights across the entire dataset. As a result, the regularization strength is directly affected by the total number of samples in the dataset. For larger datasets, this can lead to more substantial penalties, which may influence the training dynamics and potentially result in overly conservative updates to the model parameters.

One key limitation of L2 regularization is that it prevents the training model from effectively extracting discriminative features. This occurs because L2 regularization treats all directions in the feature space equally, shrinking all coefficients uniformly. Consequently, distinguishing discriminative features in datasets becomes more challenging. Additionally, L2 regularization does not reduce the number of features, making it less effective in high-dimensional spaces with many irrelevant features. This uniform treatment can also lead to overlapping class boundaries, further complicating the extraction of discriminative features.

Therefore, we propose an efficient regularization technique by utilizing the mean squared value to address the weaknesses of the sum-of-squares approach. The advantage of using the mean squared value is that it offers a regularization strength independent of dataset size. This method provides more consistent regularization across different batch sizes by averaging the penalty over all samples. Consequently, the mean squared approach ensures that the regularization penalty remains stable, predictable, and efficient, regardless of the dataset's scale. With this consistency, the discriminative features extracted by Mask_RCNN can be preserved, leading to improved classification performance.

3. Methodology

Our proposed regularization modification, known as WMean_L2, was implemented in the optimization layer after Regions Of Interest (ROI) align in the Mask_RCNN model, as shown in Figure 1. The Mask_RCNN model was deployed for VTR. We chose VTR to evaluate the performance of our proposed regularization modification in Mask_RCNN for addressing intra-class issues in vehicle classification.

The process consists of two main stages. Stage 1 includes three primary steps: data acquisition and pre-processing, feature extraction, and optimization. Stage 2 focuses on feature classification to obtain the final classification results. The model was deployed through training and testing phases, with these stages implemented in both phases.

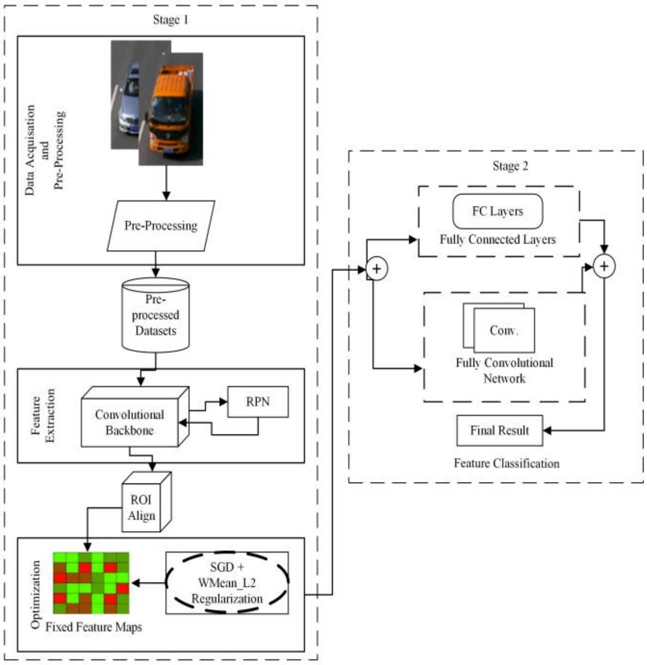
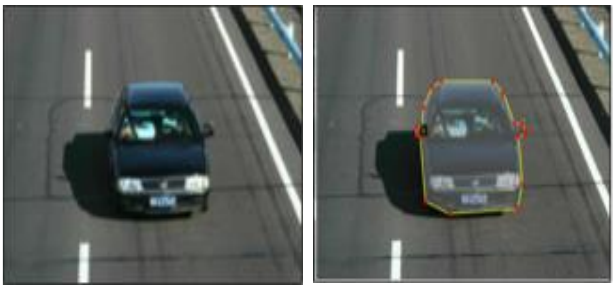


Figure 1. Mask_RCNN+SGD+WMean_L2 general framework.

3.1. Data Acquisition and Pre-Processing

The vehicle images were pre-processed in this phase. The data was acquired from benchmark database known as Beijing Institute of Technology (BIT) that contain raw vehicle images with variety of vehicle classes. Next, we implemented data annotation and data augmentation to provide information of vehicle labels and bounding box to the Mask_RCNN model during training process, and to balance the dataset.

The annotated data is stored and produced in a JavaScript Object Notation (JSON) file. While for data augmentation, we had applied three techniques that are rotate 45°, flip and warp shift. The output of this process is pre-possessed and annotated dataset that was used as input for feature extraction with image size of 1024x1024 as shown in Figure 2. The annotated dataset was used only in the training phase, whereas the testing dataset remains unannotated and is excluded from the training process. Next, features from the datasets were extracted in the convolutional backbone of the Mask_RCNN.



a) Before annotation. b) Annotated image.

Figure 2. Example of annotation image used for training phase.

3.2. Feature Extraction

In this process, we used Restnet-101 as a convolutional backbone for Mask_RCNN. The process consists of five stages, beginning with processing the raw VTR dataset image through a convolutional layer, batch normalization, ReLU activation, and max pooling to capture low-level features. In the first stage, we used 64 filters. The feature map (C1) from stage 1 is then used in the second stage to extract mid-level features, and we used 256 filters in this stage. In the third stage with 512 filters, convolutions are applied to (C2) to learn vehicle shapes and attributes, generating a new feature map (C3). The fourth stage enhances high-level feature recognition using a deeper residual block with 1024 filters to produce high-level feature maps (C4). In the fifth stage with 2048 filters, three residual blocks extract higher-level semantic information from C4, producing the final feature map (C5). The extracted feature maps (C5) serve as inputs for the Region Proposal Network (RPN), where a 3x3 convolutional layer of sliding window technique and anchor boxes are used to generate regional proposals based on the Intersection over Union (IoU) evaluation. The IoU that we used was $0.3 \geq \text{IoU} \geq 0.7$ to maintain positive anchors. This feature extraction process can be referred to Algorithm (1).

Algorithm 1: Feature extraction based on resnet-101.

```
Input: VTR images
Output: Feature maps of the extracted features
def feature_extraction (image):
//Extract features from dataset image
//Stage 1: Low-level feature extraction
C1=convolution (image)
C1=batch_normalization (C1)
C1=relu_activation (C1)
C1=max_pooling (C1)
//Stage 2: Mid-level feature extraction
  C2=residual_block (C1)
//Stage 3: Vehicle shape and attribute //learning
  C3=convolution (C2)
//Stage 4: High-level feature extraction
  C4=deeper_residual_block (C3)
//Stage 5: Higher-level semantic //information extraction
  C5=residual_block (C4)
  C5=residual_block (C5)
```

```
    C5=residual_block (C5)
return C5  # Feature map for FPN

def region_proposal_network(C5):
//Generate region proposals using RPN
    proposals = []
     for feature_map in C5:
  //Sliding window technique
    windows = sliding_window(feature_map)
 //Generate anchor boxes
    anchors = generate_anchors(windows)
 //Evaluate IoU for anchor selection
    selected_anchors = evaluate_iou(anchors)
//Store proposals
    proposals.extend(selected_anchors)
return proposals
//Example usage
    image = load_vtr_image("input_image.jpg")
  feature_maps=feature_extraction(image)
 regional_proposals=
    region_proposal_network(feature_maps)
```

Next, the regional proposals together with the feature maps undergo the ROI Align phase. The aim here is to ensure the extracted features are precisely aligned with the ROIs, which improves the accuracy of the object detection model. We used the ROI with coordinates of [10.5, 10.5, 21.5, 21.5]. The process involves extracting precise feature representations from ROIs within the feature map. This process is repeated for every ROI proposed by the RPN, allowing Mask_RCNN to make more accurate object detection and segmentation predictions. Finally, the result for each ROI is an aligned fixed-size feature map was produced.

3.3. Optimization

Optimization phase is the most important phase since our proposed WMean_L2 is implemented in this phase. Optimization phase is the most important phase since the main contribution of this paper is in this phase. In our Mask_RCNN, we deployed the optimization layer after ROI align to ensure precise spatial alignment of feature maps, capturing subtle, discriminative features accurately. The aligned feature maps from the previous phase undergo optimization process. SGD as an optimizer is deployed in the optimization layer. SGD was selected in this work due to its single-batch update rule that can minimize loss and converges efficiently to an accurate solution.

However, since SGD updates the model's weights using gradients from the entire dataset, the training process becomes longer due to frequent update steps, leading to increased model complexity and hyperparameter sensitivity. Thus, the WMean_L2 regularization was integrated with the SGD. The aim of this integration is to reduce model complexity, improve hyperparameter sensitivity issue, and preserve discriminant features of the extracted features.

WMean_L2 offers balanced and interpretable approach to weight penalization compared to standard L2. The standard L2 regularization used sum function to accumulates the squared weights and scales with the number of parameters, while WMean_L2 applies mean function to normalizing the penalty across all weights. The scale formulation in the L2 regularization ensures consistent regularization pressure regardless of model size or layer depth, which is particularly important in deep architectures. In addition, L2-based methods shrink weight rather than eliminate them, unlike L1 regularization, which induces sparsity by forcing many weights to zero. Thus, it makes L2 better than L1.

The weight penalization by utilizing mean squared value in WMean_L2 encourages smaller weights. Thus, it regularizes the model by improving the geometry of the loss surface and reducing the number of sharp local minima that could hinder convergence during optimization. Consequently, it ensures that the regularization penalty is uniformly applied across different model sizes and feature map dimensions, contributing to more stable training and improved generalization. Additionally, in deep models like Mask_RCNN, where spatial and related features are distributed across channels, preserving small but informative weights is essential for maintaining discriminative feature maps. WMean_L2 supports this by retaining subtle activations that are important for capturing intra-class variations, such as distinguishing between a car and a taxi. This helps prevent the model from over-simplifying its internal representations, leading to more robust feature learning and stable convergence during training.

The WMean_L2 regularization was formulated based on Equation (1). The equation consists of the loss function which is Sum Squared Error (SSE) plus with the penalty which is mean squared value. The mean squared value in Equation (1) was deployed based on Equation (2).

$$L = \sum_{n-1}^{N} (y_n - \hat{y}_n)^2 + \lambda \cdot \frac{1}{H \times W \times C} \sum_{i,j,k=1}^{H,W,C} \left(X_{i,j,k}\right)^2 \qquad (1)$$

Where y_n is the ground truth value for the nth sample, $\hat{y}_n$ is the predicted value for the nth sample, and N is the number of samples.

$$WMean = \frac{1}{H \times W \times C} \sum_{i,j,c}^{H,W,C} \left(X_{i,j,k}\right)^2 \qquad (2)$$

Where H and W is height and width of the feature map, C is number of channels, X is value of the feature map, and i, j, k, is row index based on height, column index-based width, channel index, respectively. While λ is regularization strength parameter to control the trade-off between fitting the data and regularization.

3.4. Feature Classification

In the feature classification phase, the optimized feature maps from the previous process undergo a few processes through fully connected layers for bounding box regression and vehicle classification. Simultaneously, these maps are passed to a fully convolutional network for object mask generation in the stage 2 as shown in Figure 1. In this phase, the flatten feature map of 1D vector is used as the input. The model computes the total loss after each prediction to measure the deviation from true labels. Backpropagation then calculates the gradients of this loss, adjusting weights and biases to minimize error. This iterative process enhances the model's ability to recognize class features over time. ReLu activation is applied to produce output neurons, which are used to classify vehicles into predefined classes like car, taxi, truck, Sport Utility Vehicle (SUV), van, or bus. Simultaneously, a Fully Convolutional Network (FCN) generates an object mask for precise pixel-wise classification, distinguishing vehicles from the background. The results from this phase were evaluated based on the standard performance measurements. The measurements are precision, recall, F-score and accuracy.

3.5. Mask_RCNN_SGD_WMean_L2

Based on the designed methodology in Figure 1, we implemented the WMean_L2 in the Mask_RCNN framework as outlined in Algorithm (2). In this paper, we focus on the feature map with one channel, that has values range from -1.0 to 1.0. These values indicate the strength and weak of detected features. The higher values represent stronger activations prominent edges, the lower values indicate weaker or absent features.

Algorithm 2: Mask_RCNN+SGD+WMean_L2.

Input: Feature Map: A tensor of shape (H, W, C) and Hyperparameter: Regularization parameter, λ
Output: Total Loss: Incorporating classification loss and WMean_L2 regularization term, and Updated Weights

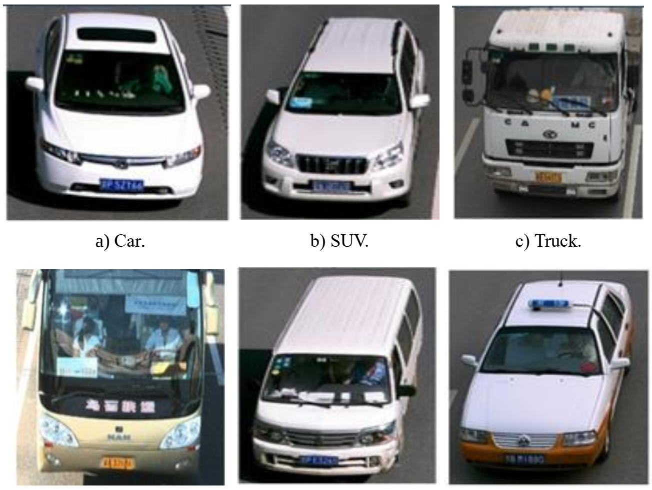
```
Initialize Mask_RCNN model
Define regularization parameter ($\lambda_{reg}$)
Initialize SGD optimizer
Define number of training iterations
while number of training iterations do
    extract feature map:
    input an image to the model to obtain the feature map
     ROI Align(Image) → Feature Map
   initialize variables:
    set feature map dimensions (H, W, C)
    set weights, biases
     flatten feature map vector, X:
         reshape (Feature Map, (H×W×C,1))→X
     compute WMean_L2 Regularization Term:
          initialize L2 Reg Term = 0
    for each channel, k in the feature map:
      calculate the squared values and accumulate:
       $L2\ Reg\ Term +\leftarrow \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{H}(X[i,j,k])^2$
      calculate the overall WMean_L2
    end for
end while
```

The process begins with calculating the sum of squares for all elements in the feature map layers, denoted as $(X_{ijk})^2$ to produce the regularization term. It involves squaring each value in the feature map, which emphasizes larger activations and minimizes smaller ones. This squaring process produce positive number for all feature map values. These values indicate the strength of detected features. The squared values are then summed across the spatial grid points and all channels, resulting in a single scalar value representing the feature map. This scalar value is then forwarded to the mean square operation, where it is divided by the total number of elements (7×7×2048) to produce the WMean_L2 regularization value in a scalar for one channel. Each feature map in other channels undergoes the same calculation process.

4. Results and Discussion

4.1. Dataset and Experimental Settings

The first experiment is conducted using a benchmark database known as BIT vehicle dataset. This database is selected because it provides vehicle images taken using with top and frontal view of surveillance mounted camera, which is aligned with the aim of this study scope as mentioned in the introduction section. Other than that, it consists of variety of vehicle classes, for instance, bus, car (passenger car), minivan, SUV, taxi, and truck in which, other databases are not providing taxi images. However, the dataset does not provide a specific time when the images are captured. Figure 3 shows an example of vehicle images from BIT dataset.



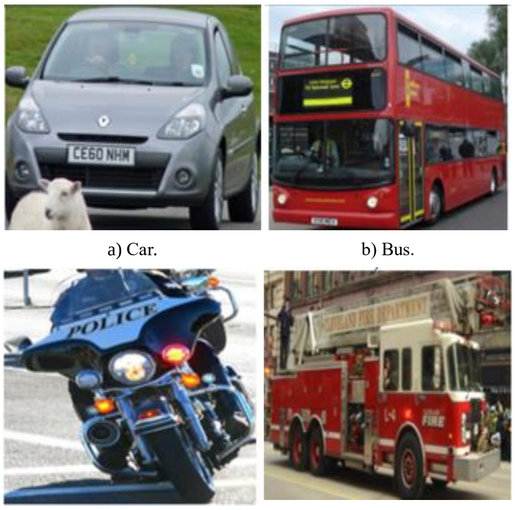
a) Car. b) SUV. c) Truck.

d) Bus. e) Van. f) Taxi.

Figure 3. Vehicle images from BIT dataset.

The total number of vehicle images is 9850 images. Car and SUV have the highest number of images with approximately 5000 and 1300 images, respectively. Taxi and minivan have the lowest number of images with not more than 600 images each. Thus, to avoid bias during the training and testing phases, 250 images were randomly selected from each class as the training dataset, and 200 images as the testing dataset. Note that, during the feature extraction process, we used images with the size of 1024x1024x3, whereby 3 was the three channels; red, green and blue.

To ensure an unbiased experiment, a second experiment was conducted using a benchmark database known as Common Objects in Context (COCO) vehicle dataset. This database is selected because it contains several vehicle classes which are car, bus, motor and truck as shown in Figure 4.



a) Car.

b) Bus.

c) Motor.

d) Truck.

Figure 4. Vehicle images from COCO dataset.

The total number of vehicle images is 9650 images. Even though, the dataset contain many images to be process, the COCO dataset lacks intra-class diversity for specific vehicle types such as taxis, SUVs, and vans, as these are all grouped under the general car class. This limitation reduces its effectiveness for detailed vehicle classification. Bus and motor class have the highest number of images with approximately 3200 and 2550 images, respectively. Car have the lowest number of images with not more than 1820 images each. Besides, this dataset has various view angle condition. Thus, to avoid bias during the training and testing phases, 100 images were randomly selected from each class as the training dataset based on frontal view that follow the study's scope, and 40 images as the testing dataset. Same as BIT dataset image settings, the COCO images will used size of 1024×1024×3 images during the feature extraction process.

For the experimental settings, the learning rate was set at 0.001, and the λ WMean_L2 was configured at 0.03. These parameter values were selected based on insights from related studies, aiming to reduce the model's loss function. A high value in the loss function could lead to an unfitted model, negatively impacting object prediction accuracy. The experiment was conducted over 300 epochs, with 1,000 steps per epoch. The model achieved a good fit at epoch 72, as indicated by the minimization of validation loss and error loss during training. This careful tuning also helped prevent overfitting throughout the training process. The following subsections present the results, demonstrating the effectiveness of the trained model when evaluated using the testing dataset.

4.2. Results for Inter-Classes Vehicle

For the first experiment, we tested the Mask_RCNN+SGD+WMean_L2 with 3 classes of vehicle types based on BIT dataset. The types were sedan, heavy vehicles and van. Car and taxi were grouped in the sedan class, bus and truck were in the heavy vehicles class, and SUV and van were in the van class. Various vehicle images were used in this experiment, including cars with sunroofs. The total testing images was 1200 images. The aim of this experiment is to observe the Mask_RCNN+SGD+WMean_L2 when dealing with inter-class classification. Thus, we classify the car and taxi as the same class, as well as the truck and bus, also the SUV and van. We present the results from this experiment in the confusion matrix and performance based on the measurements as shown in Table 1.

Table 1. Mask_RCNN+SGD+WMean+L2 performance for 3 classes based on BIT dataset.

Class		Actual			Performance measurement (%)			
		Sedan	Heavy vehicles	Van	Precision	Recall	F-score	Accuracy
Predicted	Sedan	395	0	5	98.72	99.35	99.04	98.93
	Heavy vehicles	0	397	3	99.36	99.36	99.36	
	Van	3	2	395	98.72	98.09	98.40	

The results represent the highest performance achieved when the regularization value is 0.03. Looking at Table 1, out of total 400 images for the sedan class, 395 were correctly classified, whereas 5 were incorrectly classified as van class. For the heavy vehicles class, 397 of bus and truck were correctly classified. In the van class, 395 of the SUV and van images were correctly classified, and 3 was misclassified as the sedan class and 2 as heavy vehicles. Based on that confusion matrix, the average accuracy was 98.93% and the precision for each class was more than 98%. It shows that Mask_RCNN+SGD+WMean_L2 has a low false positive rate when dealing with the inter-class classification. Other than that, the recalls for sedan class was 99.36% which is 0.01% higher than the sedan class. For the F-score, the van class was the lowest among the three classes. The high performance demonstrates that inter-class features are crucial for improving performance metrics. These features enhance the distinction between different classes, allowing the model to make more accurate predictions when they are well-separated.

Next, we tested the Mask_RCNN+SGD+WMean_L2 with 4 inter classes of vehicle types based on COCO dataset. Note that each image in the COCO dataset contains multiple objects such as vehicle, people, building, etc., different from BIT dataset that specific to a vehicle object. The types of vehicle class in COCO dataset are bus, car, motor and truck. In COCO dataset, they include car, taxi and van as the car class. While for the truck class, it consist of fire truck, lorry and pickup truck. The total testing images was 160 images. The aim of this experiment is to observe our proposed model when dealing with inter-class classification. We present the results from this experiment in the confusion matrix and performance measurements as shown in Table 2.

Table 2. Mask_RCNN+SGD+WMean+L2 performance for 4 classes based on COCO dataset.

Class		Actual				Performance measurement (%)			
		Bus	Car	Motor	Truck	Precision	Recall	F-score	Accuracy
Predicted	Bus	40	0	0	0	100	85.11	91.95	89.38
	Car	6	25	0	9	62.5	96.15	75.76	
	Motor	0	0	40	0	100	100	100	
	Truck	1	1	0	38	95	80.85	87.36	

The results in Table 2 represent the highest performance achieved when the regularization value is 0.03. Looking at Table 2, out of total 40 images for the bus and motor class, both obtained 40 images were correctly classified. For the car class, 25 images were correctly classified, whereas 9 were incorrectly classified as truck class, and 6 as bus class. For the truck vehicles class, 38 images of truck were correctly classified whereas 1 were incorrectly classified as bus class, and 1 as car class. Based on that confusion matrix, the average accuracy was 89.38% and the precision for each class was more than 62%. For motor class, it achieved 100% in precision, recall, and F-score. This result shows that motor classes have distinctive features that differentiate them from other classes. However, in terms of overall performance the model showed a high false positive rate when classifying inter-classes especially in the car class.

The model achieved a lower precision of 62.5% in the car class, because the car images were wrongly predicted as truck and bus. The truck class achieved recall with 80.85% due to truck images are often misclassified as cars. Similarly, the bus class had a recall of 85.11%, due to bus images are misclassified as trucks. These results show that the model has difficulty distinguishing vehicles with overlapping similar appearance. To improve performance, the model needs feature learning to capture more discriminative features for each class. Although the overall accuracy was 89.38%, the differences in precision, recall, and F-score between classes show that it’s still a challenge to classify similar appearance vehicles accurately.

From both performances between the BIT and COCO datasets using the proposed model, the BIT dataset shows high accuracy and consistent results. This suggests that the BIT dataset is suitable for vehicle type recognition to be implemented in applications related to ITS as mentioned in the introduction section. This is because the BIT dataset provides top and frontal views of vehicle images that are captured from mounted surveillance cameras. In addition, BIT dataset focuses on vehicle type domain, well-balanced class distribution, and clearer visual distinctions between vehicle type classes. These characteristics enable the proposed model to learn more discriminative features that are crucial in determining intra-class patterns compared to COCO. In contrast, COCO’s dataset contains various and overlapping objects that does not align with this scope of study. This caused the model to make it difficult to classify the vehicle types and increases the chance of confusion between similar classes.

4.3. Results for Intra-Classes Vehicle

For the second experiment, we tested the proposed model with 6 classes; car, taxi, van, bus, truck and SUV from BIT dataset. The aim in this experiment is to observe the performance of the Mask_RCNN+SGD+WMean_L2 when dealing with the intra-class classification. Similar to the previous experiment, we used various vehicle images including cars with sunroofs. Thus, we can see if the car with sunroofs are able to be classified as the car class or will be misclassified as the taxi class. Table 3 shows the results of the proposed model based on the confusion matrix, precision, recall, F-score and accuracy.

Table 3. Mask_RCNN+SGD+WMean+L2 performance for 6 classes.

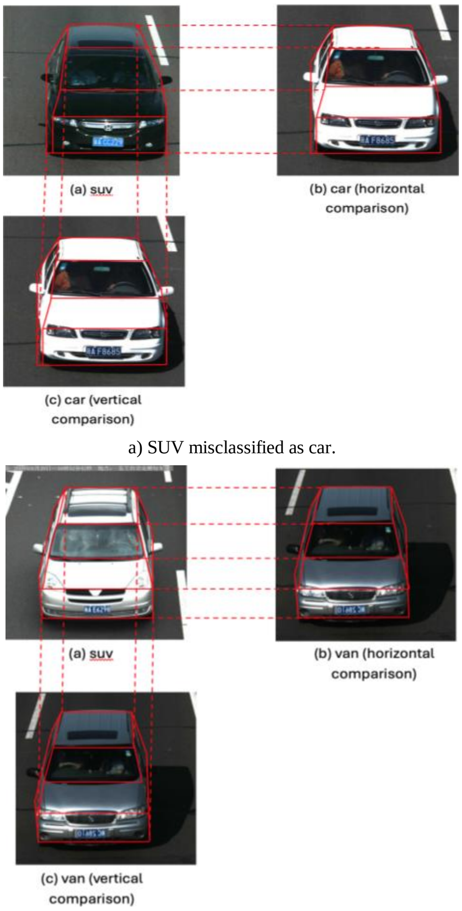
Class		Actual						Performance measurement (%)			
		Car	Truck	SUV	Van	Bus	Taxi	Precision	Recall	F-score	Accuracy
Predicted	Car	197	0	0	0	0	3	98.72	97.47	98.09	97.22
	Truck	0	197	0	3	0	0	98.72	97.47	98.09	
	SUV	3	3	187	7	0	0	93.59	94.81	94.19	
	Van	2	2	11	185	0	0	92.31	94.74	93.51	
	Bus	0	0	0	0	200	0	100	100	100	
	Taxi	0	0	0	0	0	200	100	98.73	99.36	

In Table 3, out of 200 total images for the car class, 197 images were correctly classified, whereas 3 was incorrectly classified as the taxi class. For the truck class, 197 of the truck images were correctly classified. In the SUV class, 187 of the SUV images were correctly classified, and 7 were misclassified as the van class, while 3 was misclassified as the car and the truck, respectively. 185 of van images were correctly classified, and 11 misclassified as the SUV class. Looking at the bus class, all of 200 images were correctly classified. The most interesting part is 200 of the taxi images were correctly classified as the taxi although in the taxi images are almost similar like sedan car with the sunroof. Based on that confusion matrix, the average accuracy of the proposed technique was 97.22% and the precision for each class is more than 92.31%. The results shows a promising performance since we classify the class into 6 classes although the results were slightly decrease compared to the 3 classes.

Thus, we can see that the performance of Mask_RCNN+SGD+WMean_L2 was comparable in both inter-class and intra-class features. It shows that the proposed model able to preserve the discriminant intra-class features. The discriminant intra-class features reduce the overlap between different classes by clearly defining boundaries within each class. This reduction in overlap decreases misclassification, thereby increasing the model's overall accuracy. By minimizing intra-class variability, these features ensure the model makes fewer false positive predictions, leading to higher precision. Additionally, they help capture more true positives by reducing false negatives, which improves recall. Since the F-score is the harmonic mean of precision and recall, enhancements in both metrics due to discriminant intra-class features naturally result in a higher F-score, providing a balanced measure of performance.

Although the proposed model reduces the number of false positive predictions, misclassification in intra-class still occurred, specifically for SUVs with 13, and vans with 15 were incorrectly classified. This occurs due to a combination of visual indistinctness in images and the effect of regularization on feature learning as shown in Figure 5-a) and (b). These figures illustrate the example of cases where SUVs are misclassified as either cars or vans. From both horizontal and top-down perspectives, SUV image was misclassified due to identical regions with cars and vans, which are similar rooflines, window shapes, and body proportions. These similarities can be misclassified, especially when distinctive SUV traits like higher ground clearance, larger wheel arches, and a bulkier rear bumper are either not visible or not emphasized in the input image.

To enhance model stability and generalization, the WMean_L2 regularization method applies a uniform penalty to all weights by averaging their squared values. This helps reduce overfitting, but it can also limit the model's ability to learn subtle features that are specific to each class. Therefore, the model tends to focus more on common, shared features, which increases the chance of misclassifying certain inputs. This issue is further worsened when the training data lacks variety in viewing angles images of certain vehicle types, for example SUV in this case. Due to that, the model tends to misclassify SUVs for dominant classes, such as cars or vans. The annotated diagram supports this finding by showing how overlapping and suppressed features from different angles make it harder for the model to tell vehicle types apart under these conditions.


(a) suv
(b) car (horizontal comparison)
(c) car (vertical comparison)

a) SUV misclassified as car.

(a) suv
(b) van (horizontal comparison)
(c) van (vertical comparison)

b) SUV misclassified as van class.

Figure 5. Comparison of misclassification SUV, as car and van class.

We also observed the Mask_RCNN+SGD performance by using different regularization techniques which are dropout, L1 regularization, and the default L2 regularization. The aim of this observation is to see how the proposed WMean_L2 able to enhance the classification performance compared to other regularizations when dealing with intra-class. Table 4 shows the results comparison based on the precision, recall, F-score and accuracy. Figure 6 depicts a bar chart to visualize the performance of the techniques for each vehicle class.

Based on the results in Table 4, the proposed technique (Mask_RCNN+SGD+WMean_L2) consistently outperformed other techniques across most vehicle classes, achieving the highest precision, recall, and F-score. It showed significant improvements for the car and truck classes, with precision and recall both around 98.72% and 97.47%. For the SUV and van classes, the improvements were more modest. All techniques performed well in classifying buses, but the proposed technique still outperformed others. For the taxi class, it showed significant gains in precision and F-score, though recall was not as high.

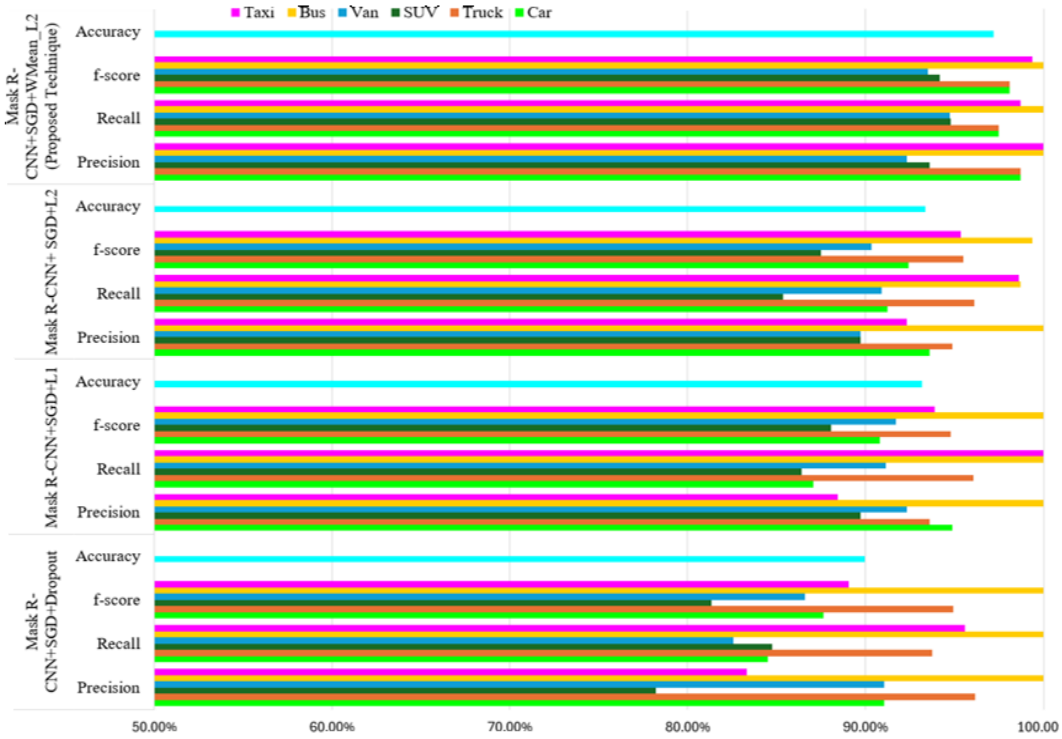


Figure 6. Performance comparison based on vehicle classes and different regularization techniques.

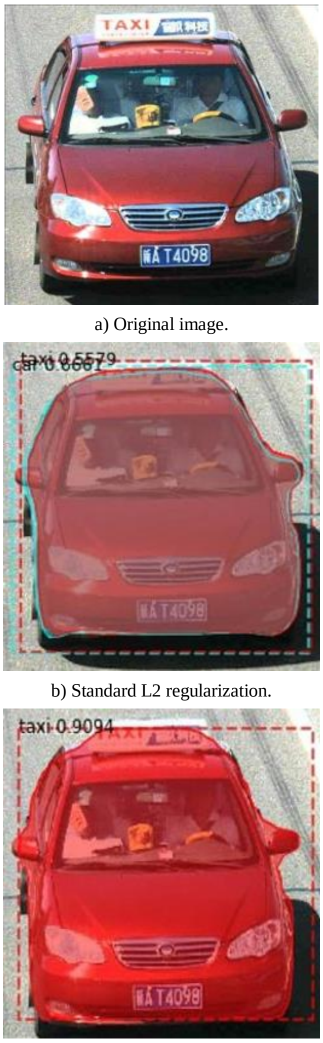
Table 4. Mask_RCNN+SGD performance based on different regularization methods.

Technique	Class	Performance measurement (%)			
		Precision	Recall	F-score	Accuracy
Mask_RCNN+SGD+Dropout	Car	91.03	84.52	87.65	89.96
	Truck	96.15	93.75	94.94	
	SUV	78.21	84.72	81.33	
	Van	91.03	82.56	86.59	
	Bus	100.00	100.00	100.00	
	Taxi	83.33	95.59	89.04	
Mask_RCNN+SGD+L1	Car	94.87	87.06	90.80	93.16
	Truck	93.59	96.05	94.81	
	SUV	89.74	86.42	88.05	
	Van	92.31	91.14	91.72	
	Bus	100.00	100.00	100.00	
	Taxi	88.46	100.00	93.88	
Mask_RCNN+SGD+L2	Car	93.59	91.25	92.41	93.38
	Truck	94.87	96.10	95.48	
	SUV	89.74	85.37	87.50	
	Van	89.74	90.91	90.32	
	Bus	100.00	98.73	99.36	
	Taxi	92.31	98.63	95.36	
Mask_RCNN+SGD+WMean_L2 (Proposed technique)	Car	98.72	97.47	98.09	97.22
	Truck	98.72	97.47	98.09	
	SUV	93.59	94.81	94.19	
	Van	92.31	94.74	93.51	
	Bus	100.00	100.00	100.00	
	Taxi	100.00	98.73	99.36	

Figure 7 justifies how the proposed model performance is obtained. The evaluation of vehicle using baseline standard L2 regularization and the proposed WMean_L2 regularization reveals significant differences in classification performance and the way each technique influences weight penalization and feature discrimination. In Figure 7-b), the model with standard L2 regularization misclassified the vehicle as a car with a confidence score of 0.6661 compared to the taxi class with 0.5579. In contrast, Figure 7-c) shows that the proposed model correctly classified the vehicle as a taxi with a confidence score of 0.9094. This score shows improved discriminative capability in the proposed model.

Figures 8-a) and (b) extend the analysis presented in Figure 7 by comparing the baseline and the proposed model to observe their influence on the decision boundaries formed by the classification model. This analysis is based on feature weight behavior. Data distribution in that figure represent features for a car (pink) and a taxi (blue). In Figure 8-a), the effect of standard L2 regularization to the decision boundaries are aligned with the zero axis. This is due to the equal penalization across all weights, which lead to more generalized boundaries and not based on the actual data pattern. Although this approach can help reduce overfitting, it may also limit the model's ability to distinguish subtle differences within intra-class features.

In contrast, Figure 8-b) illustrates the impact of WMean_L2 regularization on the model's decision boundaries. The decision boundaries are aligned with the data distribution which is different from the standard L2 regularization. It indicates that the WMean_L2 allows the model to adapt more flexible to the actual data pattern. This flexibility of decision boundaries helps the model to focus on relevant discriminative features while ignoring less relevant ones. As a result, the WMean_L2 can improve the model's ability to separate intra-class categories. For example is to distinguishing between car and taxi classes, which share almost similar features.

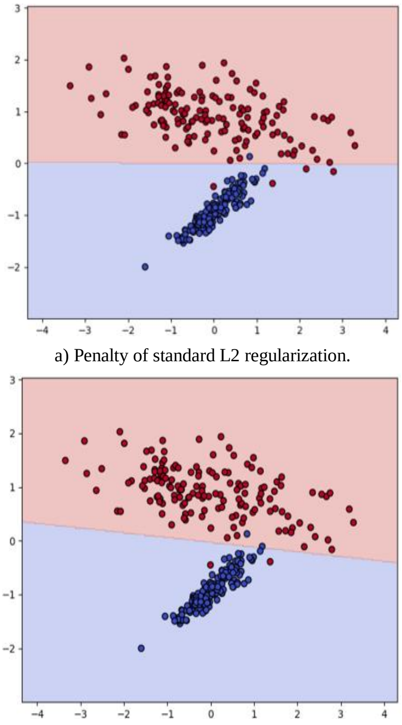


a) Original image.

b) Standard L2 regularization.

c) WMean_L2 regularization.

Figure 7. Vehicle type classification based on baseline and proposed techniques.



a) Penalty of standard L2 regularization.

b) Penalty of WMean_L2 regularization.

Figure 8. The effect of regularization techniques to the decision boundary.

4.4. Comparison Results with other State-of-Art Techniques

We compared the results obtained using the proposed approach (Mask_RCNN+SGD+WMean_L2) across six vehicle categories and compared them to existing VTR methods that emphasized intra-class classification through deep learning techniques. The techniques are Three-Channels of CNN known as TC-SF-CNNLS [12], and semi-supervised CNN [2]. Table 5 shows the comparison of the results among those techniques based on accuracy, precision, recall, and F-score. Based on the table, the proposed technique achieves the highest accuracy, performing better across all vehicle classes compared to other techniques.

Table 5. Comparison results.

Technique/Class		Performance measurement (%)			
		Precision	Recall	F-score	Accuracy
Zhang *et al.* [18]	**Car**	91.92	87.50	89.66	89.20
	Truck	89.11	88.24	88.67	
	SUV	84.00	87.50	85.71	
	Van	83.00	83.84	83.42	
	Bus	98.00	97.03	97.51	
	Taxi	-	-	-	
Zhang *et al.* [17]	**Car**	88.37	95.0	91.56	90.41
	Truck	92.78	90.00	91.37	
	SUV	87.62	85.00	86.29	
	Van	84.24	85.50	84.86	
	Bus	91.18	93.00	92.08	
	Taxi	98.95	93.99	96.41	
Proposed technique	**Car**	98.72	97.47	98.09	97.22
	Truck	98.72	97.47	98.09	
	SUV	93.59	94.81	94.19	
	Van	92.31	94.74	93.51	
	Bus	100.00	100.00	100.00	
	Taxi	100.00	98.73	99.36	

In terms of average accuracy, the proposed technique achieves an accuracy of 97.22%, surpassing the accuracies obtained by [2, 12], which are 89.20% and 90.41%, respectively. For the taxi class, only our proposed technique and the method presented by Satyanarayana *et al.* [12] performed classification. The proposed technique was outperformed in all performance metrics, achieving a precision of 100%, a recall of 98.73%, and an F-score of 99.36%. For other classes, the proposed technique also outperformed existing methods across all performance metrics. This demonstrates that the proposed technique enhances precision in intra-class challenges, as the classes often share highly similar appearances.

While the results show that our model performs well overall, a closer look reveals some limitations. Specifically, the model showed smaller improvements for the SUV and van classes, with F-scores of 94.19% and 93.51%. These two vehicle types often have very similar region, which makes it difficult for the model to clearly separate them. As a result, some misclassifications still occur. This suggests that although WMean_L2 improves classification between similar vehicle types, it still faces challenges when the visual differences are subtle.

These challenges are not only theoretical but have practical consequences. In real-world applications such as automated toll collection, traffic monitoring, or autonomous driving, the inability to correctly differentiate between classes, for example, a van and an SUV, could impact decision-making systems that rely on accurate vehicle classification for pricing, enforcement, or path planning.

5. Conclusions

VTR is one of the systems facing challenges related to intra-class patterns. Mask_RCNN is one of the deep learning techniques widely used in VTR due to its ability to extract region-based features. An optimization layer in Mask_RCNN is implemented to minimize the loss function by adjusting weights and biases, thereby reducing classification errors and ensuring efficient model convergence. L2 regularization is particularly popular in optimization due to its stability and ability to keep weights small and evenly distributed, which helps capture detailed patterns and balance model complexity for better generalization to new data.

The standard L2 regularization, which relies on summing squared values, has limitations. It is affected by weight scale, making it less effective at discouraging correlated weights within features of the same class. This shortcoming can reduce the accuracy of VTR, where differentiating between visually similar categories, like taxis and cars, is essential. To overcome these challenges, we introduce a modified L2 regularization approach called WMean_L2. Instead of sum-squared values, it utilizes the mean-squared value, ensuring scale independence, better model comparability, and greater stability during architectural changes. These advantages contribute to more consistent optimization outcomes.

We integrated WMean_L2 into Mask_RCNN, using SGD as the optimizer, creating Mask_RCNN+SGD+WMean_L2. This model was tested in VTR to enhance intra-class classification accuracy. To assess its performance, we used the vehicle dataset from the BIT. Results demonstrated notable improvements across multiple evaluation metrics, confirming that this modification in L2 regularization strengthens classification efficiency, particularly in distinguishing closely related categories.

Looking ahead, optimizing the model's hyperparameters will be a key focus. In this study, we manually adjusted these settings. Moving forward, developing a configurable deep learning model will be essential to achieving the best possible performance.

Acknowledgment

We would like to show our gratitude to University Malaysia Pahang Al-Sultan Abdullah and University of Al Maarif for supporting this work as part of the grant number UIC241504.

References

[1] Awang S., Azmi N., and Ghani N., "Road Enforcement Monitoring System based on Vehicle Type Recognition Using Sparse Filtering Convolutional Neural Network with Layer Skipping Strategy," *in Proceedings of the IEEE 6th International Conference on Industrial Engineering and Applications*, Tokyo, pp. 475-479, 2019. https://doi.org/10.1109/IEA.2019.8715122

[2] Awang S., Azmi N., and Rahman A., "Vehicle Type Classification using an Enhanced Sparse-Filtered Convolutional Neural Network with Layer-Skipping Strategy," *IEEE Access*, vol. 8, pp. 14265-14277, 2020. https://doi.org/10.1109/ACCESS.2019.2963486

[3] Chiang C., Jaber M., Chai K., and Loo J., "Distributed Acoustic Sensor Systems for Vehicle Detection and Classification," *IEEE Access*, vol. 11, pp. 31293-31303, 2023. https://doi.org/10.1109/ACCESS.2023.3260780

[4] Dong Z., Wu Y., Pei M., and Jia Y., "Vehicle Type Classification Using a Semisupervised Convolutional Neural Network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, pp. 2247-2256, 2015. https://doi.org/10.1109/TITS.2015.2402438

[5] Elassy M., Al-Hattab M., Takruri M., and Badawi S., "Intelligent Transportation Systems for Sustainable Smart Cities," *Transportation Engineering*, vol. 16, pp. 100252, 2024. https://doi.org/10.1016/j.treng.2024.100252

[6] Lin K., Zhao H., Lv J., Zhan J., and et al., "Face Detection and Segmentation with Generalized Intersection over Union based on Mask_RCNN," *in Proceedings of the 10th International Conference Advances in Brain Inspired Cognitive Systems*, Guangzhou, pp. 106-116, 2020. https://doi.org/10.1007/978-3-030-39431-8_11

[7] Minkesh A., Worranitta K., and Taizo M., "Human Extraction and Scene Transition Utilizing Mask_RCNN," *arXiv Preprint*, pp. 1-6, 2025. https://arxiv.org/abs/1907.08884v2

[8] Ojha A., Sahu S., and Dewangan D., "Vehicle Detection Through Instance Segmentation Using Mask_RCNN for Intelligent Vehicle System," *in Proceedings of the 5th International Conference on Intelligent Computing and Control Systems*, Madurai, pp. 954-959, 2021. https://doi.org/10.1109/ICICCS51141.2021.9432374

[9] Ojha G., Poudel D., Khanal J., and Pokhrel N., "Design and Analysis of Computer Vision Techniques for Object Detection and Recognition in ADAS," *Journal of Innovations in Engineering Education*, vol. 5, pp. 47-58, 2022. https://doi.org/10.3126/jiee.v5i1.43682

[10] Qian Z., Zhao C., Zhang B., Lin S., and et al., "Classification of Vehicle Types using Fused Deep Convolutional Neural Networks," *Journal of Intelligent and Fuzzy Systems*, vol. 42, pp. 5125-5137, 2022. https://doi.org/10.3233/JIFS-211505

[11] Ramakrishnan D. and Radhakrishnan K., "Applying Deep Convolutional Neural Network Algorithm in the Cloud Autonomous Vehicles Traffic Model," *The International Arab Journal of Information Technology*, vol. 19, no. 2, pp. 186-194, 2022. DOI: 10.34028/iajit/19/2/5

[12] Satyanarayana G., Deshmukh P., and Das S., "Vehicle Detection and Classification with Spatio-Temporal Information Obtained from CNN," *Displays*, vol. 75, pp. 102294, 2022. https://doi.org/10.1016/j.displa.2022.102294

[13] Shim K., Yoon S., Ko K., and Kim C., "Multi-Target Multi-Camera Vehicle Tracking for City-Scale Traffic Management," *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Nashville, pp. 4193-4200, 2021. https://doi.org/10.1109/CVPRW53098.2021.00473

[14] Tahir H., Khan M., and Tariq M., "Performance Analysis and Comparison of Faster R-CNN, Mask_RCNN and ResNet50 for the Detection and Counting of Vehicles," *in Proceedings of the International Conference on Computing, Communication, and Intelligent Systems*, Greater Noida, pp. 587-594, 2021. https://doi.org/10.1109/ICCCIS51004.2021.9397079

[15] Tahir N., Bature U., Baba M., Abubakar K., and Yarima S., "Image Recognition Based Autonomous Driving: A Deep Learning Approach," *International Journal of Engineering and Manufacturing*, vol. 10, pp. 11-19, 2020. https://doi.org/10.5815/ijem.2020.06.02

[16] Zhang B. and Zhang J., "A Traffic Surveillance System for Obtaining Comprehensive Information of the Passing Vehicles Based on Instance Segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, pp. 7040-7055, 2020. https://doi.org/10.1109/TITS.2020.3001154

[17] Zhang L., Wang P., Li H., Li Z., and et al., "A Robust Attentional Framework for License Plate Recognition in the Wild," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, pp. 6967-6976, 2020. https://doi.org/10.1109/TITS.2020.3000072

[18] Zhang Y., Huang Y., Yu S., and Wang L., "Cross-View Gait Recognition by Discriminative Feature Learning," *IEEE Transactions on Image Processing*, vol. 29, pp. 1001-1015, 2019. https://doi.org/10.1109/TIP.2019.2926208

Noraqilah Misman received the Master's degree in Computer Science from Universiti Teknologi Malaysia, Johor Bahru, Malaysia, in 2015. She is currently pursuing the Ph.D. degree in Computer Science with Universiti Malaysia Pahang Al-Sultan Abdullah, Malaysia. Her research interest includes Image Recognition Using Machine Learning.

Suryanti Awang is an Associate Professor at the Faculty of Computing, Universiti Malaysia Pahang Al-Sultan Abdullah, Malaysia. She holds a PhD in Electrical Engineering from the Universiti Teknologi Malaysia. Her expert areas are in Artificial Intelligence, Machine Learning to solve various problems related to Pattern Recognition. She has graduated 5 PhD students and supervising 3 ongoing PhD students.

Mohammed Khalaf is an Assistant Professor in the Department of Computer Science at the University of Al Maarif. He holds a PhD in Computer Science from Liverpool John Moors University, UK. His research interests include Artificial Intelligence, Healthcare Bioinformatics, Machine Learning, and Data Science.

Hasan Kahtan is a Senior Lecturer in Software Engineering at the Department of Applied Computing and Engineering, Cardiff School of Technologies, Cardiff Metropolitan University. Hasan is a committed lecturer with over ten years University of Malaya, Universiti Malaysia Pahang, National University of Malaysia, and Universiti Teknologi MARA. He has a strong interest in academic research and publications especially in Mobile Cloud Computing, and Machine Learning.