Received: October 15, 2023. Revised: November 8, 2023. 428

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

A Novel YOLO-ARIA Approach for Real-Time Vehicle Detection and Classification in Urban Traffic

Aria Hendrawan[1,2]* Rahmat Gernowo[1] Oky Dwi Nurhayati[1] Christine Dewi[3]

[1]*Information System School of Postgraduate, Universitas Diponegoro, Indonesia*
[2]*Department of Information Technology and Communication, Universitas Semarang, Indonesia*
[3]*Department of Information Technology, Satya Wacana Christian University, Indonesia*
* Corresponding author's Email: ariahendrawan@students.undip.ac.id; ariahendrawan@usm.ac.id

Abstract: The number of vehicles traveling between cities has increased significantly with the acceleration of urbanization. This has resulted in several traffic-related issues, including traffic congestion and the need to collect information on the number and variety of vehicles on the road. In this study, we propose you only look once (YOLO) artificial real-time intelligent analysis (ARIA) based intelligent traffic monitoring system. YOLO is an algorithm that is capable of detecting objects in images and recordings. We improved YOLO's feature extraction capabilities to improve its vehicle detection accuracy. In addition, we proposed detection models with C3X (convolution neural network) module in the backbone of YOLO. In our experiments, the proposed system attained 99,1% accuracy with 98,3% precision and 98,4% recall on datasets obtained from the CCTV monitoring portal of the semarang city government. In addition, the proposed system has a higher average precision than other vehicle detection and classification methods. Considering the current environmental conditions, the proposed system can classify vehicles in real-time. This makes it a valuable planning and traffic-management instrument.

Keywords: Intelligent traffic monitoring system, Object detection, Object classification, You look only once (YOLO), Convolution neural network.

1. Introduction

Road traffic monitoring is an important area of study. It can help us comprehend current traffic conditions and provide traffic management agencies with actionable data. This information can enhance the quality of life for individuals [1] during holidays, for instance, traffic volume data can be used to direct vehicles away from congested areas by suggesting alternate routes. When heavy trucks routinely use a road, roadside warnings can be installed to notify vehicles and prevent accidents. In addition, the detection of a vehicle can be used to identify and monitor criminal [2] these applications all rely on data gathered by a road monitoring system. Consequently, numerous researchers have devised various methods for detecting and classifying vehicles [3].

The detection of vehicles using conventional methods can be categorized into two main groups: static-based methods[4-10]and dynamic-based methods [11, 12]The generation of vehicle prediction frames in static-based approaches involves the utilization of sliding windows or shape feature comparison. These frames are subsequently validated by considering the information stored within them. Furthermore, dynamic-based approaches employ the dynamic attributes of a mobile entity in order to recover the entity's outline from the captured image. The subsequent examples pertain to methods that are based on static principles. In their study, Mohamed et al (2015) employed Haar-like characteristics to extract vehicle geometry data, which were subsequently utilized as input for an artificial neural network in order to perform classification [13]. In their study, Wen et al (2015) employed Haar-like features to extract edge and structural vehicle features [9] these features were subsequently inputted into

AdaBoost to effectively identify and retain relevant features. The features that have undergone filtering were next subjected to classification using a support vector machine (SVM). In their respective studies, Sun et al (2002) [7] and David & Athira (2014) [5] employed Gabor filters as a means to extract vehicle attributes, which were subsequently utilized as input for a support vector machine (SVM) model. The primary objective of this approach was to ascertain the presence or absence of a vehicle within a picture. The two-step vehicle detection system developed by Wei et al (2019) serves as an illustrative instance of a dynamic-based approach [14]. Initially, the researchers employed Haar-like features and AdaBoost algorithm to detect and isolate the region of interest that contains vehicles. The region was subsequently subjected to reclassification through the utilization of the histogram of oriented gradients (HOG) technique, as outlined by Bougharriou et al., (2017) [4] in conjunction with a support vector machine (SVM). The experimental results indicated that their approach showed improved skills in detecting vehicles. In their study, Yan et al (2016) developed a vehicle identification system that employed car shadows to determine the borders of vehicles [10] this approach exemplifies the utilization of dynamic-based methods. The histogram of oriented gradients (HOG) method was subsequently employed to extract features from the boundaries. These extracted features were then utilized as input for both an AdaBoost classifier and an SVM classifier to perform validation. Nevertheless, this method possesses a limitation whereby the presence of two obstructing vehicles results in their amalgamation into a singular entity due to the interconnectedness of their shadows. This phenomenon reduces the impact of detection.

Seenouvong et al (2016) [15] introduced a vehicle recognition and counting method based on dynamic features. The researchers employed background removal to derive a discernible map from a provided image, which was subsequently utilized to segment the related foreground image. Subsequently, a range of morphological operations was employed to obtain the contour and bounding box of a mobile entity, identify mobile vehicles, and quantify the number of vehicles traversing a designated region. The conventional approaches employed for vehicle detection, which rely on either static or dynamic attributes, exhibit certain constraints. The utilization of these techniques necessitates the extraction of features by human means, a process that is both labor-intensive and prone to generating superficial features that fail to sufficiently capture the essence of vehicle modifications. Furthermore, the utilization of dynamic feature approaches can prove to be intricate and result in suboptimal outcomes when confronted with significant alterations in the background. Deep learning approaches have gradually replaced traditional methods due to their ability to autonomously acquire features from data and their increased robustness against variations in background and lighting conditions.

The previous models have demonstrated notable advantages in terms of achieving high accuracy and localization in object recognition. However, these models also exhibit certain limitations, such as the need for more intricate training procedures and a relatively slower operational speed. This is particularly evident when considering real-time object detection in a singular event [16] when it comes to categorizing object detection models, certain models such as you only look once (YOLO) developed by Redmon (2016) [17] and C.M. Liu and Juang (2021) [18], as well as the single shot MultiBox detector (SSD), demonstrate superior performance compared to other models. This is achieved by directly incorporating regression techniques into the object detection process, leading to enhanced operational speed. Nevertheless, the single shot MultiBox detector (SSD) fails to account for the interdependencies among many scales, hence resulting in constraints when it comes to detecting diminutive entities. Conversely, the you only look once (YOLO) algorithm exhibits superior aptitude in acquiring knowledge of shared attributes and exhibits enhanced operational velocity [19, 20] nevertheless, both solid state drive (SSD) and you only look once (YOLO) encounter challenges when it comes to effectively processing intricate graphical regions and exhibiting elevated identification mistakes for things that are visually identical. Moreover, the categorization of certain vehicles may not be totally precise because of minor biases that impact the confidence ratings in object detection [21]. Even with the progress in object detection techniques, a substantial challenge persists in balancing detection precision with computational efficiency.

In today's context, the significance of object detection spans various domains, from security oversight to vehicular automation. Yet, the paramount challenge has always been to elevate detection fidelity while curtailing both training and inference durations. However, the primary challenge in object detection is how to enhance detection accuracy while minimizing training and inference time. Addressing this challenge, this research introduces a novel method that leverages a unique combination of regular convolution and cross convolution, promising an innovative approach to

object detection.

The key feature of this method is the utilization of convolution three times in the neck section to extract image dataset features. This process ensures the deep extraction of characteristics from each labeled object, offering a significant advantage in detection quality. Furthermore, in the cross-convolution process, the image data is divided into two parts. While one part undergoes three convolutions, the second part is processed directly to the final section. These two parts are then combined, ensuring the integration of rich and efficient image feature information. This novel methodological approach, characterized by its unique feature extraction process, is designed to offer significant advancements in the field of object detection.

The main advantage of this approach is speed. In our experiments, the proposed method demonstrated faster detection speeds during training, recognizing and classifying vehicle objects more quickly compared to popular methods such as Yolov5x, Yolov5l, Yolov5m, Yolov5s, and Yolov5n. This speed enhancement is a testament to the efficiency of the proposed method, positioning it as a superior alternative in the realm of object detection.

Harnessing the core tenets of object detection and responding to the highlighted issues, we put forth a novel model, Yolo-ARIA was proposed, based on Yolov5 optimization, to achieve a better balance between two tasks, there are detection and classification, to obtain better performance results for vehicle detection based on vision and classification than the cutting-edge models used for the same purpose. Yolo-ARIA will implement Jia et al (2023) [22] and Liu et al (2023) [23] research that improved Yolov5 algorithm. However, these methods have limitations, such as the need for manual feature extraction and the inability to describe the changes in vehicle features effectively. Recently, techniques for deep learning have been devised to overcome these limitations. Deep learning techniques are capable of automatically extracting features from images and learning the evolution of vehicle features. As a consequence, the creation of a new model, Yolo-ARIA, is anticipated to strike a healthy equilibrium between two tasks, namely detection and classification.

2. Related work

In many fields, deep learning has been used to attain success in recent years. Convolutional neural networks (CNN) have significantly enhanced the accuracy of image recognition compared to traditional methods that require manual feature extraction. Earlier CNN models, such as LeNet [24], were utilized to identify handwritten numerals. Later models, such as AlexNet [25] enhanced the accuracy of image recognition by utilizing ReLU activation functions and dropout layers to prevent overfitting and by enhancing the model architecture. GoogLeNet [26] utilized multiple filters of varying sizes to extract features and enrich feature data. VGG-16 and VGG-19 [27] utilized multiple small convolution kernels to execute operations, demonstrating that increasing the model's depth can enhance its precision. ResNet [28] utilized residual blocks to solve the problem of gradient disappearance and the inability to converge caused by an excessively deep network. MobileNet [29] utilized deep separable convolution to extract fewer and more useful features, thereby reducing the number of redundant CNN model parameters.

Prior research has focused on enhancing CNN's ability to describe features so that they can be applied to more complex problems, such as object detection. Utilizing region-based CNN (R-CNN) [14, 30-32]models is an approach to object detection. The choice of which method to employ is dependent on the application at hand.

YOLOv5 is a singular-stage object identification model that comprises four distinct components, namely input, backbone, neck, and head. In contrast to the original YOLOv4 network architecture, YOLOv5 incorporates the Focus module [33] which serves to augment the model's feature extraction capabilities. Additionally, the model incorporates the upgraded cross-stage partial network (CSPNet) as its underlying architecture, hence enhancing the effectiveness of the model. To enhance the effectiveness of the multi-scale feature fusion process, an additional layer called the path aggregation network (PANet) is incorporated, which operates in a bottom-up manner and is based on the feature pyramid structure (FPN). The primary function of the input layer is to acquire image data and convert it into a suitable format that can be processed by the neural network. The backbone is a critical element within the network architecture, as it assumes the primary responsibility of extracting features from images. The YOLOv5 model utilizes the expanded CSPNet as its underlying architecture, which is a lightweight and efficient network that can effectively extract picture information across several dimensions. The neck plays a crucial role in integrating the retrieved features from the backbone and transmitting them to the skull. The inclusion of the PANet layer within the neck component serves to enhance the efficacy of the multiscale feature fusion process, as proposed by Wang et al [32]. The primary responsibility of the head is to accurately anticipate the spatial limits and

categorical designations of the things present within the visual.

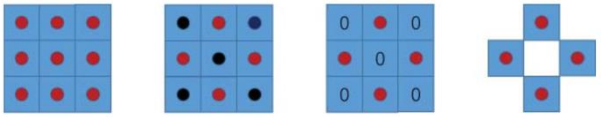
The object identification approaches possess certain limitations. Specifically, two-stage object detection methods exhibit good classification accuracy; however, the detection speed is impeded due to the large amount of network parameters. Furthermore, it is worth noting that one-stage object detection approaches have a higher real-time detection rate compared to two-stage methods, albeit at the expense of reduced accuracy. Furthermore, in order to augment the quantity of object categories, it becomes imperative to do a comprehensive reconfiguration of the complete network. This process is not only time-consuming but also diminishes the method's scalability. In summary, the existing techniques for object detection exhibit diverse tradeoffs between accuracy and performance. Two-step methods exhibit greater precision, albeit at a slower pace. One-step procedures exhibit higher speed but lower precision. When the quantity of object categories is augmented, both approaches necessitate the complete retraining of the network, resulting in a time-intensive process and a decrease in the method's scalability [3].

Recently, cross convolution neural network [34] has been used an improvement to the YOLOv5s object detection algorithm. Several changes to the algorithm, including A cross-convolution feature strengthening connection method that shortens the path of information propagation and improves the semantic information between feature pyramids. A new enhanced feature concat module that enhances the fusion of features at the same scale. As is well-known, the backbone of a CNN generates its features. These features are then continuously refined through different convolutional layers. As a result, the backbone contains the richest representation of features. This study utilizes the feature information contained in the backbone. Here is the Eq. (1) [34]:

$$F = F_{original} + F_{CCFSCA} \quad (1)$$

where F represents the features of the input, $F_{original}$ represents the features of the input of the original network, and F_{CCFSC} represents the features of the same scale between the input backbone and neck.

Cross convolutional neural network [35] is a popular approach for image classification. Cross convolution, which can be used to reduce the number of parameters in a CNN without sacrificing performance. Cross convolution is based on the principle of target calibration in the YOLO algorithm and the transformation function. It generalizes



(a) (b) (c) (d)

Figure. 1 Design process of cross convolution kernel [35]

convolution, reducing the volume of the model while still capturing high-dimensional features. Comprehensive analysis shows that at present the structure of odd x odd is widely used, which the 3x3 convolution kernel has irreplaceable role. As shown in Fig. 1 (a) is the standard 3x3 convolution kernel shape.

According to the illustration in Fig. 1 (b), the positioning of the center black dot within the new convolution kernel is deemed insignificant. To clarify, the black dot's position might be substituted with the numerical value of zero, indicating that there is no requirement for the acquisition of these particular values. In Fig. 1 (c), the authors do not effectively minimize the network parameters. Consequently, the paper proceeds to simplify the network architecture and introduces a novel convolution kernel in Fig. 1 (d). This kernel closely resembles the "+" symbol in Chinese characters and is referred to as the cross-convolution kernel. The utilization of cross-convolution kernels is a potential and novel strategy for effectively decreasing the parameter count within convolutional neural networks (CNNs). This has the potential to enhance computational efficiency and facilitate the deployment of convolutional neural networks (CNNs) on mobile devices.

There exists prior research about cross-convolutional techniques. In 2020, Arohan conducted a review of convolutional neural networks. Efforts have been made to enhance the level of efficiency and accuracy in various domains, with notable applications including object detection, digit recognition, and image recognition [36] In the study conducted by Valsesia (2020) [37], a convolutional approach was employed to assess the qualitative and quantitative outcomes of image denoising. In a study conducted by Peng Li (2019) [38] the objective was to enhance the accuracy of recognition [38] In a study conducted by Zeng Yu (2018) [39], the utilization of cross convolutional techniques was explored as a means to enhance the convergence rate of classification [39]. Verma (2018) employs the utilization of a specific method to detect the interrelationship categorization that exists between human entities [40]. However, more research is needed to evaluate the performance of the cross-convolution kernel on a wider range of tasks [35].

This research created an intelligent traffic

monitoring system that can employ a modified version of the YOLOv5 object detection model to accomplish real-time detection and enhance detection efficiency. The principal innovation of this technique are using the cross convolution module (C3X) [35]which has split into the backbone will improve the ability of classify and detection of vehicles. This paper proposes a new method for detecting vehicles that is more accurate and robust than existing methods. By automatically detecting vehicles, this method can be used to intelligent traffic monitoring systems.

The subsequent sections of this work are structured in the following manner: Section 3 principle and method improvement. The experimental findings and analysis of the proposed methodology are elaborated upon in section 4. In section 5, we provide a summary of our findings and propose potential avenues for future research.

3. Principle and method improvement

3.1 YOLOv5 for principle of detection algorithm

YOLOv5 represents the most recent iteration of the YOLO object detection system. The product is available in four distinct sizes, namely small (s), medium (m), large (l), and extra-large (x), each exhibiting variations in the quantity of parameters they possess. The YOLOv5 architecture comprises four primary components, including the input, backbone, neck, and prediction modules [41] During the input phase, YOLOv5 employs a data augmentation technique known as Mosaic. The approach presented in this study has resemblance to the CutMix technique, albeit incorporating other functionalities such as adaptive anchor box computation and adaptive picture scaling. The Mosaic technique is the process of combining disparate pictures that vary in scale and placement. This practice contributes to enhancing the diversity of the training dataset, hence increasing the model's resilience to various image categories.

The core architecture of YOLOv5 has a focus mechanism for direct processing of input photos. Subsequently, a sequence of convolutional spatial pyramid (CSP) blocks is implemented with the intention of enhancing the network's ability to fuse features. The neck of YOLOv5 incorporates a feature pyramid network (FPN) and a path aggregation network (PAN) to effectively integrate features derived from various levels of the backbone [42]. This approach enhances the model's capacity to accurately identify items across various scales.

The prediction component of YOLOv5 employs a generalized IoU (GIoU) loss function to facilitate the training process of the model for bounding box prediction. The generalized intersection over union (GIoU) loss function exhibits greater robustness compared to the conventional intersection over union (IoU) loss. Its utilization contributes to the enhancement of the model's predictive accuracy. During the post-processing phase, YOLOv5 employs a weighted non-maximum suppression (NMS) technique to eliminate redundant bounding boxes. This aids in enhancing the accuracy of the model's prognostications. In general, YOLOv5 represents a cutting-edge object identification method that demonstrates a commendable equilibrium between precision and computational efficiency. This technology demonstrates suitability for a diverse range of real-time object identification applications, including but not limited to self-driving vehicles, video surveillance systems, and augmented reality platforms [43].

3.2 Method improvement of YOLO-ARIA

YOLO-artificial real-time intelligent analysis (ARIA) represents the fifth iteration of the YOLO (you only look once) framework. The YOLO-ARIA network architecture consists of three main components: Backbone, neck, and head. YOLO-ARIA architecture shown in the Fig. 2.

YOLO-ARIA is a lightweight variant of the YOLO algorithm for detecting objects. It extracts object characteristics using convolutional and maximum pooling layers. Additionally, modify model C3X layers and modify C3 layers are utilized to extract feature information. This enables YOLOv5 to detect objects more quickly than other YOLO and SSD techniques. However, its simplified network architecture also results in less accurate detection. To enhance the detection accuracy of YOLOv5, the YOLO-ARIA variant was created. The network architecture of Yolo-ARIA is depicted in Fig. 2. Yolo-ARIA has modify of module C3-2 and additionally C3X module, to make good accuration and increase speed of detection vehicles. With the new modification of YOLO-ARIA has three outputs to predicts using three scales, then make this enables to detect vehicles with greater precision than default of YOLOv5. YOLO-ARIA is use to detect and classification vehicles. It has been trained on a database of images of vehicles and can detect vehicles in real time.

The algorithm presented in this study has been enhanced by employed various variants:

1. Modify module C3 into C3-2, the stages of C3-2 module as follows:

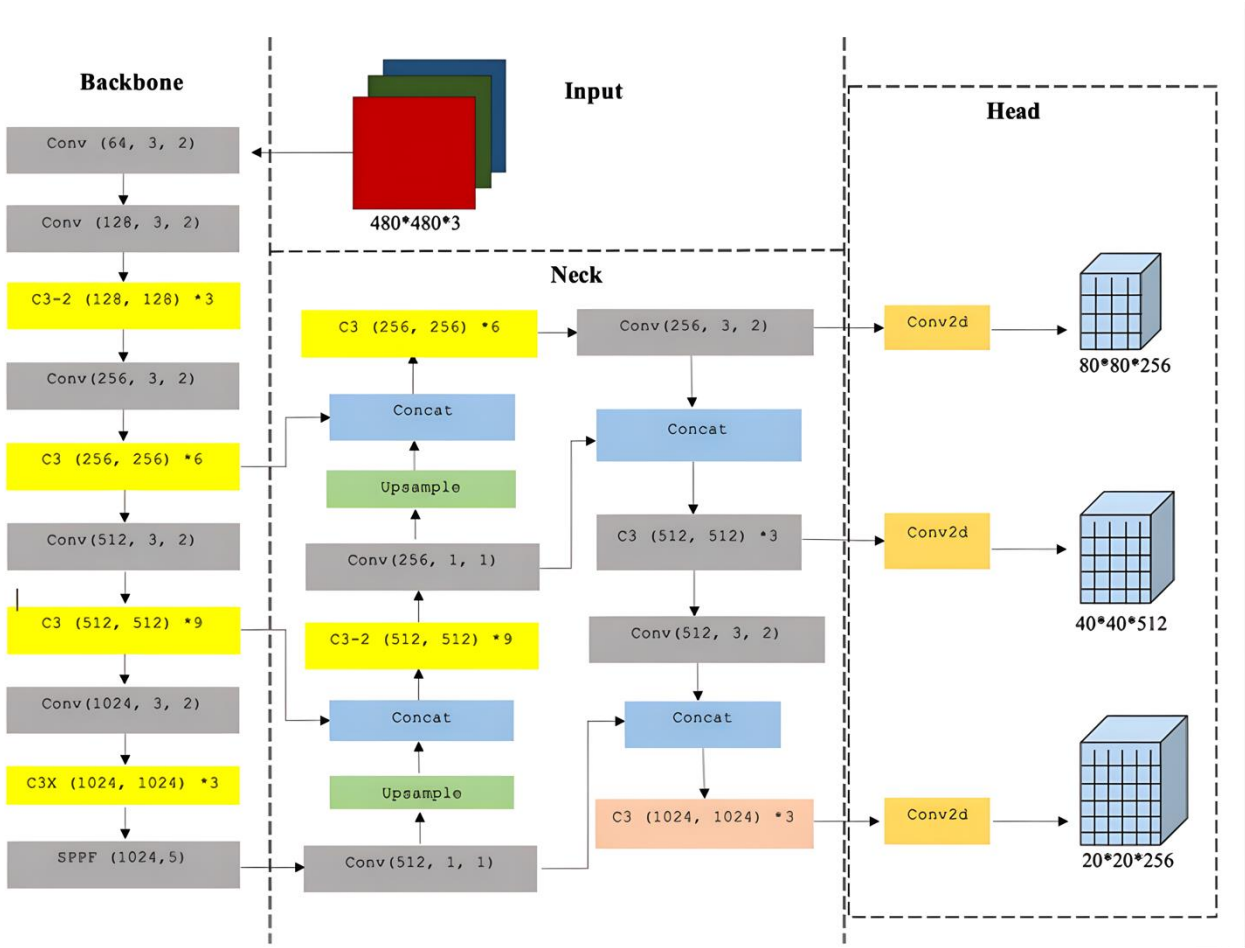


Figure. 2 Architecture Yolo-ARIA

a. Initialize the C3-2 module,
b. Calculate the hidden channels,
c. First convolution layer,
d. Second convolution layer,
e. Third convolution layer,
f. Create a sequence of Bottleneck layers without shortcut connections,
g. Forward pass through the C3 module,
h. Apply the first convolution and pass it through Bottleneck layers,
i. Apply the second convolution,
j. Concatenate the outputs of the first and second convolutions,
k. Pass the concatenated output through the third convolution,
l. return final_output.

2. Change the number of process convolutional C3 module into six times, also change the C3 module into nine times in the next steps.
3. Inserting a C3X module at the end of the backbone, the stages for C3X module as follows:
   a. Inherit from the base class C3,
   b. Calculate hidden channels,
   c. Replace the Bottleneck layers with CrossConv layers.
4. Combining C3-2 modules in the neck of YOLO-ARIA and change the number of convolutional become nine times, and
5. Substituting the number convolutional of C3 with the numbers 6 in the neck part of Yolo-Aria as depicted in Fig. 2

Both diagrams, namely C3 and C3-2, exhibit a high degree of similarity in terms of their structures. Both architectures consist of three intricate levels that are divided by two Bottleneck layers. The distinguishing factor lies in the manner in which the Bottleneck layer establishes a relationship between the input and output. In both instances, a concatenation layer (Concat) is employed to merge the outputs from two distinct pathways before to generating the ultimate output. However, inside the C3 architecture, there exists a shortcut link that facilitates the direct passage of the original input via multiple layers of bottleneck. In the case of C3-2, when the criterion shortcut=False is applied, there is an absence of shortcut connections between the levels. The primary distinction between the two can be identified as follows.

The class structure of C3X incorporates the

utilization of the cross-convolution layer. The cross-convolution layers serve as a replacement for the Bottleneck layer within the C3 module. The inputs undergo a first transit via the convolutional layers before proceeding to a sequence of cross convolution layers. Subsequently, the output generated by the cross-convolution operation is merged with the outputs originating from the preceding convoluted layer, culminating in the production of the ultimate output. The use of the cross-convolution layer in place of the bottleneck layer inside the C3X framework introduces architectural differences that have the potential to yield diverse outcomes in the context of image processing jobs. This enables the conduction of studies involving various layers to enhance the performance of the tissue in specific tasks.

3.3 Detection of vehicles

Using a single image, the YOLO object detection method can be used to identify vehicles and their locations. In genuine traffic applications, however, an input stream of image frames is provided. Different image frames detect vehicles independently, so the same vehicle may be counted multiple times, which would result in inaccurate data.

To prevent double counting, the system correlates and matches vehicles detected in various image frames using an object counting method. This is accomplished by integrating a popular and effective monitoring algorithm. Tracker is an algorithmic extension of the simple online and realtime tracking algorithm [44, 45] during real-time, CCTV-based video detection, the tracker is utilized to counteract a number of undesirable factors that can result from various camera motions. The framework's detection aspect is responsible for detecting vehicle objects that appear within a single frame. By designating a unique and distinguishable identifier for each tracking element, the tracking method follows currently monitored vehicle objects. Each tracking element is also assigned a bounding box containing the object's associated ID.

3.4 Classification of vehicles

The YOLO object detection method can be employed to accurately recognize and segment cars inside a picture. In this study, the analysis of the segmented vehicle picture is conducted through the utilization of a convolutional neural network (CNN). The primary objective of this analysis is to extract supplementary vehicle-related information, including its specific type. The process involves CNN extracting relevant picture properties, subsequently compressing them, and ultimately classifying the vehicle. The researchers developed an innovative distance metric by considering the external characteristics of the item, intending to enhance the existing methodology. The proposed approach involves the development and implementation of a deep-learning object detector that exhibits a high level of precision and recall. Subsequently, the final classification layer is eliminated. Classical architecture exhibits a substantial accumulation that gives rise to a distinctive feature vector, therefore enabling its classification. The feature vector provided serves as the "appearance descriptor" for the object under consideration, specifically a vehicle in this particular instance. After integrating this tracker into the algorithms, the three object detectors underwent training, validation, and evaluation using a diverse set of traffic surveillance photos and videos that encompassed different degrees of illumination and weather conditions. The researchers subsequently ascertained the system that could accurately assign a specific label to each image, considering its designated class.

4. Experiment and discussion

The experimental procedure shown in Fig. 3 consisted of three primary stages: dataset construction, model training, and target detection-classification vehicles. Initially, a bespoke dataset was generated through the collection and subsequent preprocessing of photographs. Subsequently, we proceeded to train a YOLO-ARIA model by making appropriate adjustments to the parameters. Ultimately, the trained model was employed to identify diminutive entities, and then, a comparative analysis was conducted with alternative methodologies.

4.1 Dataset

The utilized dataset comprises the closed-circuit television (CCTV) public site CCTV monitoring semarang city. The data was collected on September 21, 2022, at 09:48 a.m. using 1042 images. Additionally, data was collected during the nighttime on October 4, 2022, at 18:37 p.m. using 973 images. Please find the link to the dataset provided here dataset-yolo-aria. Another dataset, referred to as the VOC dataset with 12032 images, was also included in the analysis. The dataset utilized in this study comprises a total of 14,047 photographs. The VOC dataset is extensively utilized for the purpose of training and assessing deep learning models in the domains of object recognition (e.g., YOLO, Faster R-CNN, and SSD), instance segmentation (e.g., Mask

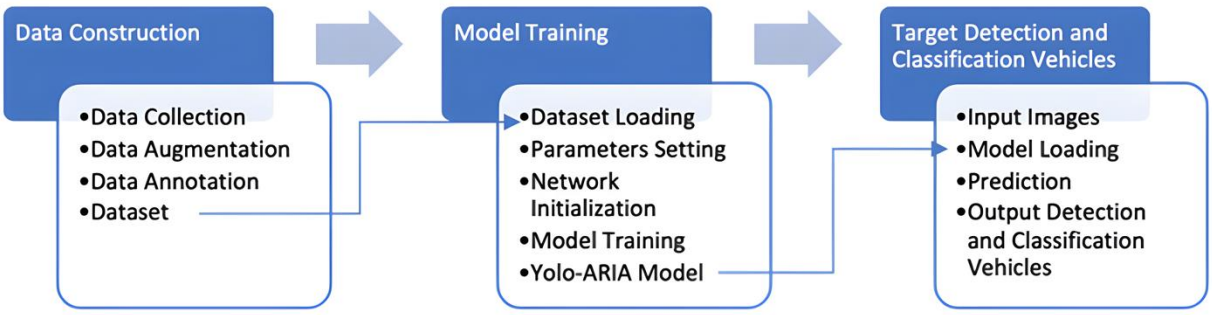


Figure. 3 Flowchart of data construction, model training and target detection-classification vehicles

R-CNN), and image classification. The dataset possesses a wide range of object categories, a substantial quantity of annotated photos, and defined assessment measures, rendering it a crucial asset for both computer vision academics and practitioners [46].

The photos are partitioned into training, validation, and testing sets at a ratio of 7:2:1 in a random manner. In order to address the issue of sample imbalance and enhance the realism of the dataset, three image are employed on each group with different techniques of Yolo algorithm. These techniques include Yolov5x, Yolov5l, Yolov5m, Yolov5s, Yolov5n and Yolo-ARIA. The dataset has been annotated in the YOLO format using the LabelImg software. In order to enhance the assessment of the detection efficacy of objects under conditions of visibility of day and night, the dataset has been annotated with four distinct groups, namely car, box car, motorcycle, and truck for dataset of closed circuit television (CCTV) public site CCTV monitoring semarang city. And for the VOC dataset have 20 groups, namely aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, sofa, train, and tv monitor The annotations for the closed circuit television (CCTV) public site CCTV monitoring semarang city are stored in roboflow using format Yolov5 Pytorch, and VOC dataset using VOC.yaml in link VOC dataset.

The bounding box mark tool (Chen et al., 2023) was utilized to generate a bounding box for each sign. The labeling technique is conducted individually for every class. A single image has the potential to be assigned several marks. In the detection phase, a singular class detector model was employed, with each class label being linked to an individual training model. The return values of the bounding box labeling tool consist of object coordinates in the form of x1, x2, y1, y2. The coordinates of the items differ from the input value of Yolo. In contrast, the Yolo input value encompasses the coordinates of the center point, width, and height (x, y, w, h). Consequently, the system is required to adapt the bounding box coordinates within the Yolo input format. The alteration technique is centered on the utilization Eqs. (2-7) [47].

$$dw = 1/W \tag{2}$$

$$\mathrm{x} = \frac{(x_1+x_2)}{2} \times dw \tag{3}$$

$$dh = 1/H \tag{4}$$

$$\mathrm{y} = \frac{(y_1+y_2)}{2} \times dh \tag{5}$$

$$w = (x_2 - x_1) \times dw \tag{6}$$

$$h = (y_2 - y_1) \times dh \tag{7}$$

The dimensions of an image can be described as follows: H represents the image height, dh represents the absolute image height, W represents the image width, and dw represents the absolute image width. This implies that the permissible values for the dimensions of the picture, represented as dw and dh, can vary from 0.0 to 1.0 in floating-point format.

4.2 Network training

The experimental development platform for network training utilizes Google Colaboration Pro+ with the spesification graphics processing unit (GPU) utilized in system is the NVIDIA A100-SXM4-40GB, RAM 40.5GB, the current version of Python is

Table 1. Hyperparameter of the model

Hyperparameter	Description	Value
lr0	Initial learning rate	0.01
lrf	Learning rate at the end of training	0.01
momentum	Momentum term used in the optimizer	0.937
weight_decay	Weight decay term used in the optimizer	0.0005
warmup_epochs	Number of epochs during which the learning rate is gradually increased	3
warmup_momentum	Momentum term used during warmup	0.8
warmup_bias_lr	Learning rate used for the bias terms during warmup	0.1
box	Scale factor for the bounding boxes	0.05
cls	Scale factor for the classification scores	0.5
cls_pw	Weight of the classification loss	1
obj	Scale factor for the objectness scores	1
obj_pw	Weight of the objectness loss	1
iou_t	IoU threshold for the non-maximum suppression	0.2
anchor_t	Anchor threshold for the non-maximum suppression	4
fl_gamma	Gamma parameter for the focal loss	0
hsv_h	Hue jitter range	0.015
hsv_s	Saturation jitter range	0.7
hsv_v	Value jitter range	0.4
degrees	Rotation range	0
translate	Translation range	0.1
scale	Scaling range	0.5
shear	Shear range	0
perspective	Perspective range	0
flipud	Probability of flipping the image vertically	0
fliplr	Probability of flipping the image horizontally	0.5
mosaic	Probability of applying mosaic augmentation	1
mixup	Probability of applying mixup augmentation	0
copy_paste	Probability of applying copy-paste augmentation	0

3.10.12, The experimental framework employed in this study is PyTorch version 2.0.1 which is a deep learning platform. The current version of CUDA is 11.8 and script python YOLOv5 v7.0-215-ga6659d0.

Table 1 presents an outline of the essential hyperparameters required for network training. Prior to commencing the training process, purposeful adjustments are made to these parameters in order to optimize the effectiveness of the model and mitigate the potential for overfitting. As an example, the batch size has been set to 16, the learning rate has been fixed at 0.01, the number of iterations has been constrained to 100, and the Adam optimizer has been employed. Every individual entry within Table 1 fulfills a distinct and specific purpose. For example, the initial learning rate (lr0) determines the rate at which the model learns. A higher learning rate (lr0) can accelerate the convergence process, but it may also increase the risk of overfitting. In contrast, the final learning rate (lrf) has a significant role in shaping the learning trajectory during the later stages of training, where a lower lrf can effectively prevent overfitting. The momentum factor is a crucial determinant of the update frequency of the learning rate throughout each epoch. A higher momentum value leads to less frequent updates, whereas a lower momentum value results in more frequent updates.

It is crucial to thoroughly understand and carefully adjust these hyperparameters as they significantly influence the dataset's ability to detect and classify. During the training period, hyperparameters were manually selected to optimize the model. Our selections have been grounded in thorough examination, recognizing the potential for even minor adjustments to significantly influence the results. The systematic approach employed in this process ensures that our model undergoes thorough training, so positioning it to achieve optimal performance.

Prior to network training, the hyperparameter is configured to optimize the model's performance and mitigate the risk of overfitting. The number of iterations was set to 100, and the Adam optimizer was employed. According to the data presented in Tables 2 and 4, it is evident that the value of the loss function exhibits a significant decrease during the initial 0 to 40 iterations, followed by a gradual decline in the subsequent 40 to 80 iterations. Following the

Table 2. The mAP comparison of different methods through different epochs

Model	Epoch=0	Epoch=20	Epoch=40	Epoch=50	Epoch=60	Epoch=80	Epoch=100
YOLO-ARIA	0.004298	0.91161	0.97412	0.99121	0.99308	0.98855	0.99161
Yolov5x	0.0019022	0.87665	0.98353	0.98925	0.99155	0.98981	0.9914
Yolov5l	0.0029256	0.9363	0.9815	0.98525	0.98839	0.98855	0.98923
Yolov5m	0.0040827	0.81562	0.98504	0.98894	0.99225	0.99174	0.99045
Yolov5s	0.0039138	0.90867	0.98278	0.98902	0.99047	0.99189	0.99239
Yolov5n	0.0035396	0.83096	0.90727	0.96576	0.97812	0.97998	0.98575

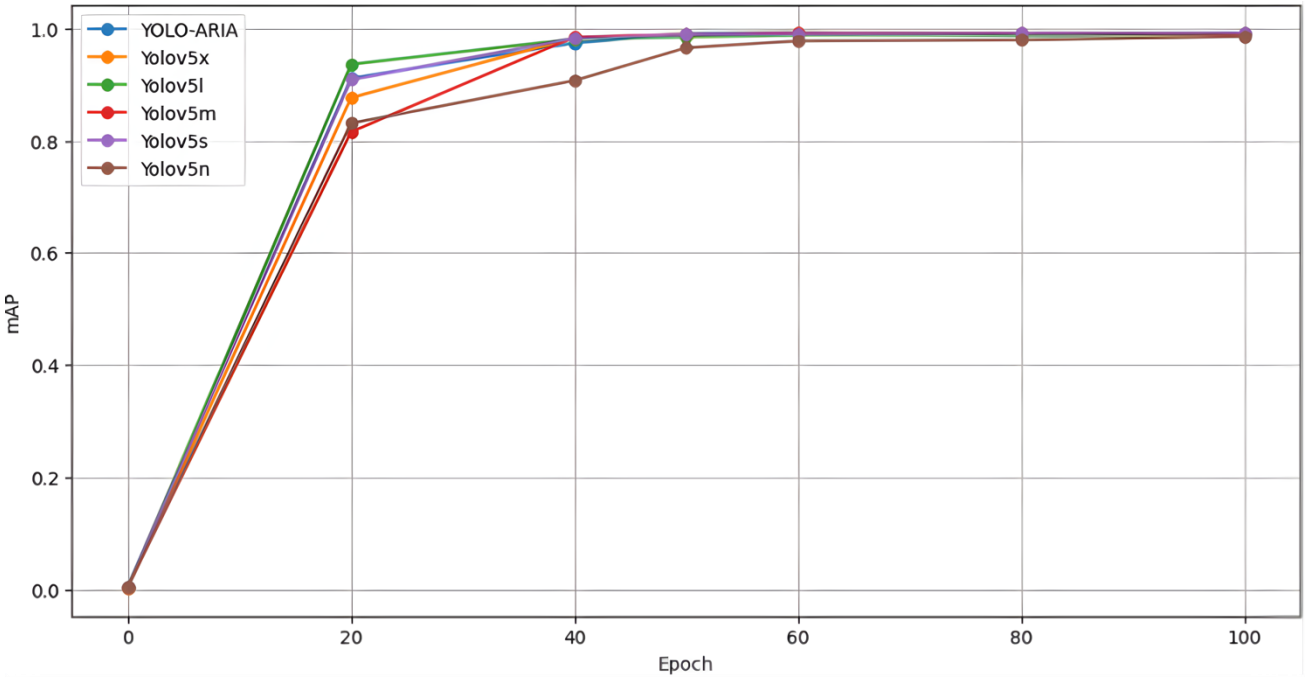


Figure. 4 The variations mAP accross different models and epochs

conclusion of the epoch 80, the depreciation value demonstrates a tendency towards stability, ultimately leading to the model's optimal state.

4.3 Evaluation indicators

The researchers used the category with the highest model output value (top-1) as the classification result when evaluating the model's output results. Then, accuracy was used as the evaluation metric. The Eq. (8) for calculating precision is as follows [3]:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (8)$$

True positive, false positive, true negative, and false negative are abbreviated as TP, FP, TN, and FN, respectively. In addition, the mean average precision (mAP) Eq. (9), precision Eq. (10), and recall Eq. (11) were utilized to assess the efficacy of various object detection models. The following are the formulas for calculating this Eqs. (8-11) indicators [48]:

$$mAP = \frac{\sum_{k=1}^{k=n} AP_k}{n} \quad (9)$$

$$Precission = \frac{TP}{TP + FP} \quad (10)$$

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

Yolo loss function based on [47] shown Eq. (12).

$$\text{Yolo Loss Function} = \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y - \hat{y}_i)^2] + \boldsymbol{\lambda_{coord}} \sum_{\boldsymbol{i=0}}^{\boldsymbol{s^2}} \sum_{\boldsymbol{j=0}}^{\boldsymbol{B}} \mathbb{I}_{\boldsymbol{ij}}^{\boldsymbol{obj}} \left[\left(\sqrt{\boldsymbol{w_i}} - \sqrt{\widehat{\boldsymbol{w}}_{\boldsymbol{i}}}\right)^2 + \left(\sqrt{\boldsymbol{h_i}} - \sqrt{\widehat{\boldsymbol{h}}_{\boldsymbol{i}}}\right)^2 \right] + \sum_{\boldsymbol{i=0}}^{\boldsymbol{s^2}} \sum_{\boldsymbol{j=0}}^{\boldsymbol{B}} \mathbb{I}_{\boldsymbol{ij}}^{\boldsymbol{obj}} \left(\boldsymbol{C_i} - \widehat{\boldsymbol{C}}_{\boldsymbol{i}}\right)^2 + \lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{noobj} \left(C_i - \hat{C}_i\right)^2 + \sum_{i=0}^{s^2} \mathbb{I}_i^{obj} \sum_{c \epsilon classes} (p_i(c) - \hat{p}_i(c))^2 \quad (12)$$

The term l_{ij}^{obj} is used to indicate the presence of an object in cell i, whereas j^{th} bounding box

Table 3. Experiment training result

Model	Class	Images	Instances	Precission	Recall	mAP	Epoch	Times	Layers	GFLOPs
Yolov5x	all	100	1042	0.979	0.984	0.99	100	1h 13s	322	203.8
Yolov5l	all	100	1042	0.985	0.984	0.992	100	33m 34s	267	107.7
Yolov5m	all	100	1042	0.975	0.981	0.993	100	21m 45s	212	47.9
Yolov5s	all	100	1042	0.984	0.979	0.993	100	13 m 47s	157	15.8
Yolo-ARIA	all	100	1042	0.99	0.98	0.993	100	12 m 28s	172	16.6
Yolov5n	all	100	1042	0.981	0.967	0.992	100	12 m 11s	157	4.1

Table 4. Experiment testing result

Model	Class	Images	Instances	Precission	Recall	mAP
Yolov5x	all	100	1042	0.985	0.798	0.82
Yolov5l	all	100	1042	0.975	0.727	0.745
Yolov5m	all	100	1042	0.981	0.981	0.992
Yolov5s	all	100	1042	0.973	0.702	0.723
Yolo-ARIA	all	100	1042	0.983	0.984	0.991
Yolov5n	all	100	1042	0.883	0.672	0.685

Table 5. Experiment training result of poor visibility conditions

Model	Class	Images	Instances	Precission	Recall	mAP	Epoch	Times	Layers	GFLOPs
Yolov5x	all	200	973	0.864	0.982	0.955	100	16 m 37s	322	203.8
Yolov5l	all	200	973	0.853	0.984	0.966	100	12 m 13s	267	107.7
Yolov5m	all	200	973	0.809	0.948	0.974	100	9 m 40s	212	47.9
Yolov5s	all	200	973	0.856	0.655	0.972	100	7 m 59s	157	15.8
Yolo-ARIA	all	200	973	0.882	0.655	0.949	100	8 m 17s	172	16.6
Yolov5n	all	200	973	0.965	0.633	0.938	100	7 m 38s	157	4.1

Table 6. Experiment testing result of poor visibility conditions

Model	Class	Images	Instances	Precission	Recall	mAP
Yolov5x	all	200	973	0.864	0.982	0.955
Yolov5l	all	200	973	0.853	0.984	0.966
Yolov5m	all	200	973	0.81	0.948	0.974
Yolov5s	all	200	973	0.856	0.655	0.972
Yolo-ARIA	all	200	973	0.882	0.654	0.948
Yolov5n	all	200	973	0.965	0.633	0.946

predictor in cell indicates the responsibility of predicting the bounding box. Subsequently, $\left(\hat{x}, \hat{y}, \hat{w}, \hat{h}, \hat{c}, and \hat{p}\right)$ are the anticipated bounding box's center coordinates, width, height, confidence, and category probability serve as their representations. The symbols lacking a cusp are indeed authentic designations. In addition, our research establishes the value of λ_{coord} as 0.5, signifying that the impact of width and height mistakes on the calculation is reduced. Subsequently, the parameter λ_{coord} = 0.5 is incorporated in order to mitigate the impact of several grids without objects on the loss function.

4.4 Comparative experiments

To enhance the validation of the detection capabilities of the YOLO-ARIA algorithm described in this study, a comparative analysis was conducted with several one-stage object detection methods, including Yolov5x, Yolov5l, Yolov5m, YOLOv5s, and Yolov5n [49].

In the training phase, Fig. 4 illustrates the comparison of the mean average precision (mAP) curve, between our technique and other methods over a span of about 100 epochs. According to the data presented in Table 2, it can be observed that the mean average precision (mAP) exhibits a significant rise during the initial 80 epochs, followed by a tendency to stabilize once the epoch count reaches 100.

Throughout the training and testing phase, our technique continuously outperforms other methods in terms of detection accuracy, as evidenced by the superior curve of the YOLO-ARIA network. In addition, the enhanced model demonstrates a more gradual and consistent trajectory, indicating enhanced stability in experiment testing result shown in Table 4. The achieved training phase mean Average Precision (mAP) of our proposed technique

Table 7. Experiment training result of VOC dataset

Model	Class	Images	Instances	Precission	Recall	mAP	Epoch	Times	Layers	GFLOPs
Yolov5x	all	4952	12032	0.81	0.782	0.843	100	6h 55min 56s	322	204.2
Yolov5l	all	4952	12032	0.8	0.771	0.829	100	4h 51min 57s	267	108
Yolov5m	all	4952	12032	0.772	0.756	0.804	100	3h 45min 1s	212	48.1
Yolov5s	all	4952	12032	0.729	0.701	0.746	100	3h 49s	157	15.9
Yolo-ARIA	all	4952	12032	0.739	0.697	0.754	100	3h 4min 15s	172	16.8
Yolov5n	all	4952	12032	0.659	0.622	0.654	100	2h 49min 28s	157	4.2

Table 8. Experiment testing result of VOC dataset

Model	Class	Images	Instances	Precission	Recall	mAP
Yolov5x	all	200	973	0.864	0.982	0.955
Yolov5l	all	200	973	0.853	0.984	0.966
Yolov5m	all	200	973	0.81	0.948	0.974
Yolov5s	all	200	973	0.856	0.655	0.972
Yolo-ARIA	all	200	973	0.882	0.654	0.948
Yolov5n	all	200	973	0.965	0.633	0.946

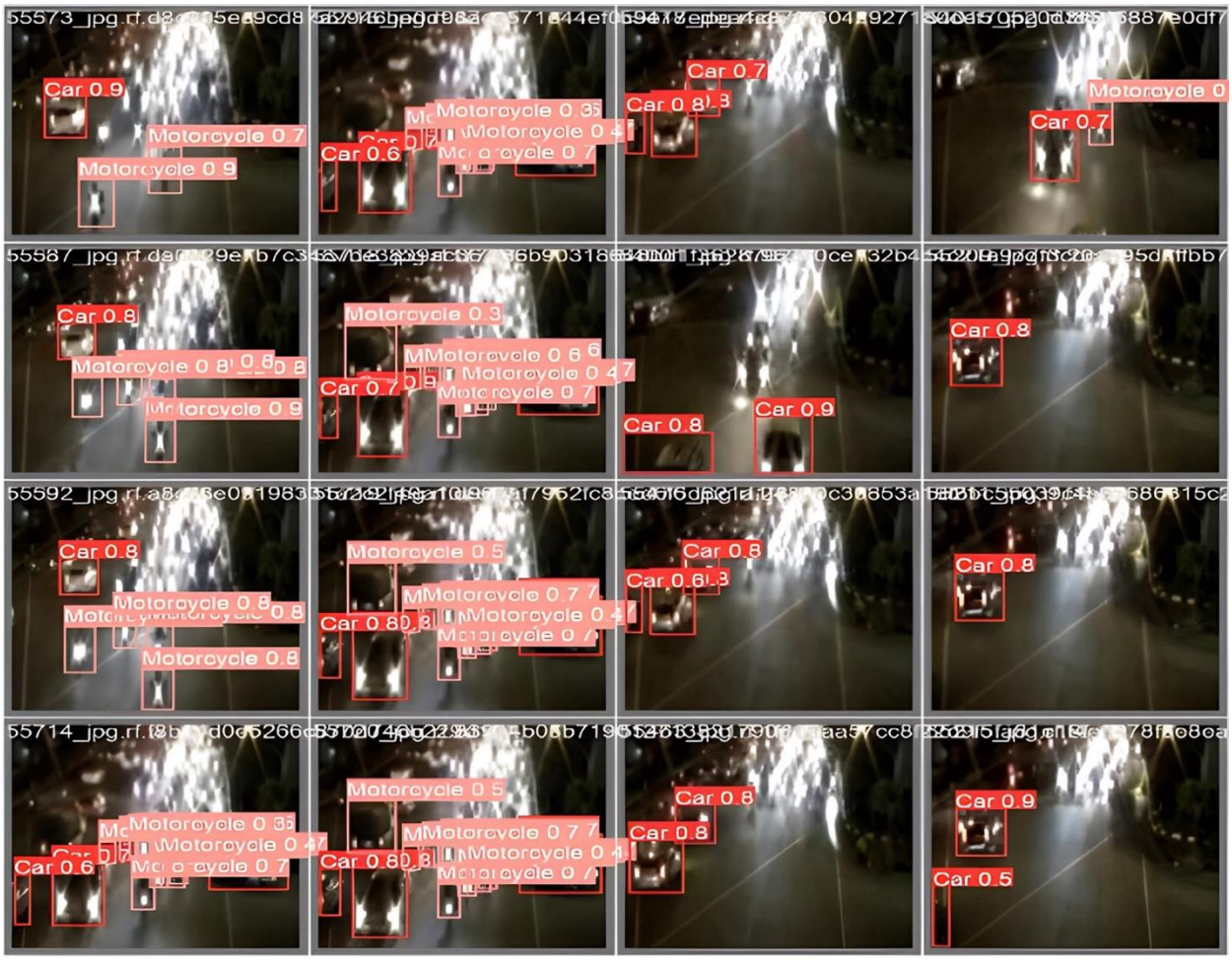


Figure. 5 Detection vehicles YOLO-ARIA in low light

is 0.993, equal with the mAP of 0.99 obtained by the original YOLOv5x model. The achieved testing phase mean Average Precision (mAP) of our proposed technique is 0.991, surpassing the mAP of 0.82 obtained by the original YOLOv5x model. This improvement amounts to an estimated rise of 1%, providing evidence for the enhanced performance of the YOLO-ARIA.

To assess the detection capabilities of the YOLO-ARIA algorithm described in this research, a comparative experiment was undertaken using a custom dataset comprising diverse scenarios

characterized by poor visibility conditions and using VOC dataset. The performance of YOLO-ARIA was evaluated in comparison to YOLOv5x, YOLOv5l, YOLOv5m, YOLOv5s, and YOLOv5n. In the identical experimental setting, the speed times that exhibit the most favorable training outcome are preserved as the speed times in other models.

The evaluation metrics employed for comparative trials encompass precision (P), recall (R), mean average precision (mAP), number of layers, and GFLOPs values. The optimization of our approach in terms of detection performance is demonstrated in Tables 5 and 6. In comparison to Yolov5x, the mean average precision (mAP) of detection exhibits a notable same value but YOLO-ARIA is superior in detection speed and a small number of training layers. Specifically, the speed detection has demonstrated a noteworthy improvement of 50%. Additionally, the giga floating point operations per second (GFLOPs) have shown a substantial decrease of 50% than Yolov5x. GFLOPS is often used to measure the performance of a graphical processing unit (GPU) and a central processing unit (CPU) in the context of graphic processing, image processing or model training YOLO. In the interim, there has been a noteworthy reduction of 50% in the number of layers than Yolov5x. By having a smaller GFLOPS value compared to YOLOv5x, as well as a small number of layers as well, YOLO-ARIA still has the same mean average precision value of 0.85 with Yolov5x. Of the smaller layer number and smaller value of GflOPS, this is the advantage of its detection speed that increases by 50% than Yolov 5x.

In comparison to VOC dataset the initial YOLO-ARIA model, there is an observed rise of 50% in speed detection training times and reduction 50% in using layers network training. Furthermore, the mean

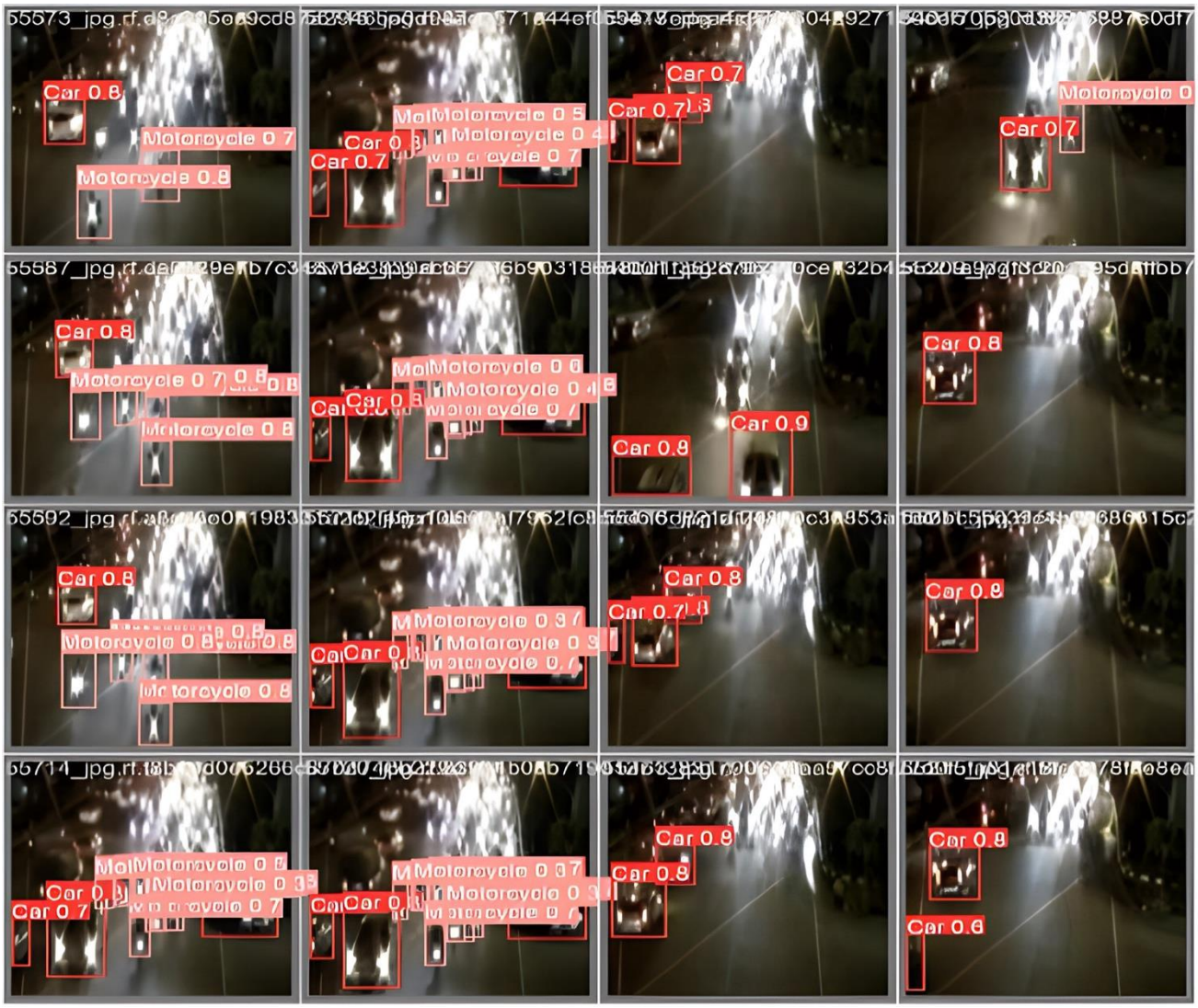


Figure. 6 Detection vehicles Yolov5x in low light

Figure. 7 Detection vehicles Yolov5x in high light

Figure. 8 Detection vehicles YOLO-ARIA in high light

average precision (mAP) has shown equal with the Yolov5x, estimation in 0.8. Additionally, In the interim, there has been a reduction of 8% in the quantity of GFLOPs, leading to a significant decrease of 50% in the size of the layers network. However, except for a minor decline in recall value, all other evaluation metrics demonstrate progress.

Based on an evaluation of the intricacy of each model and the obtained detection outcomes, a general inference can be drawn that YOLO-ARIA exhibits

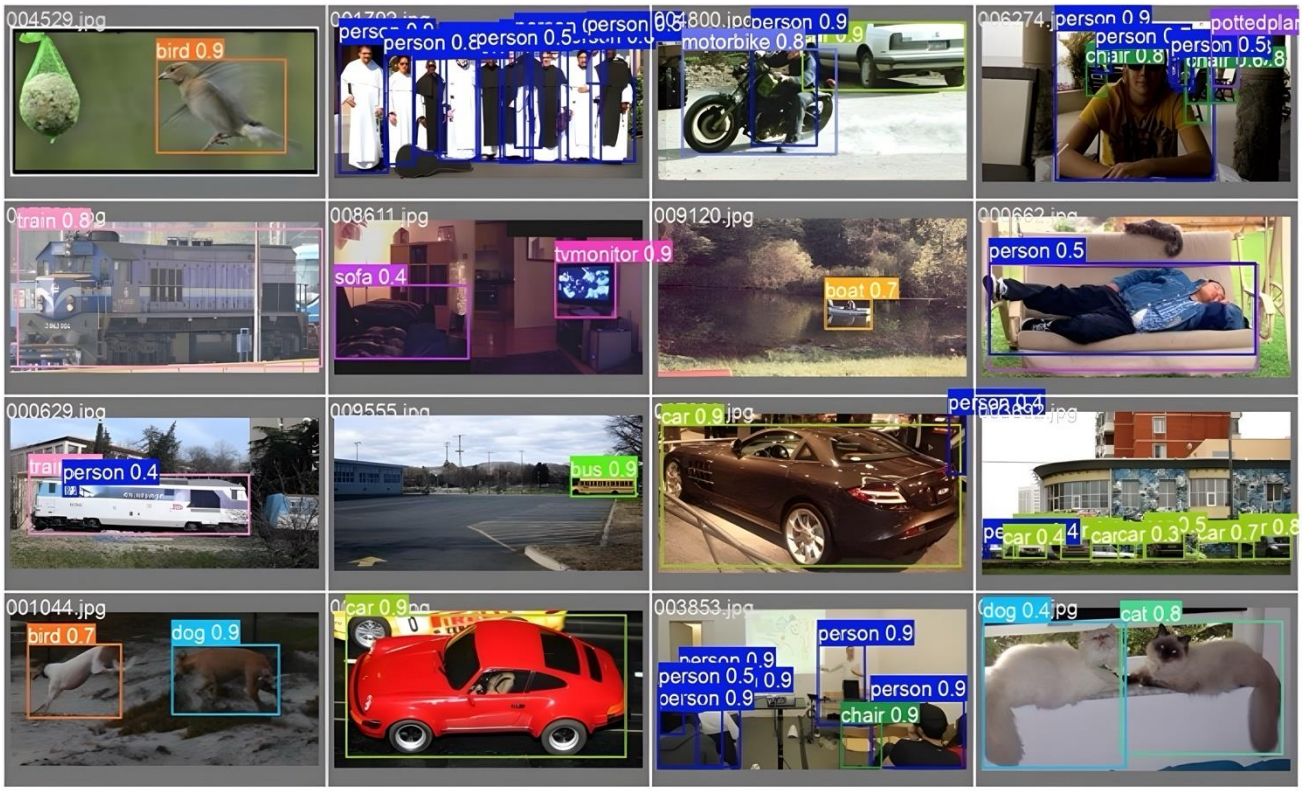


Figure. 9 Detection vehicles Yolov5x in VOC datasets

Figure. 10 Detection vehicles YOLO-ARIA in VOC datasets

superior performance in comparison to the several one-stage object detection methods, including Yolov5x, Yolov5l, Yolov5m, YOLOv5s, and Yolov5n.

4.5 Ablation experiments and result discussions

To assess the impact of each suggested module on network optimization, this part utilizes ablation experiments for the purpose of verification.

The respective functionalities of each module are presented in Tables 5 and 6. When comparing with YOLOv5x, it is observed that employing model Yolov5x with C3 module in the backbone part, equal values with the YOLO-ARIA using the C3X, which is implementation of cross convolution neural network in the part of backbone YOLO, that is 0.95 of mean average precision. Although, YOLO-ARIA has decrease in the recall values than Yolov5x but has increase in the speed of detection for timing.

To evaluate the visual efficacy of our technique, we conducted a comparison examination of the detection capabilities between the original YOLOv5 model and our improved method, YOLO-ARIA. A selection of photos randomly chosen from the dataset for the specific purpose of evaluating detection performance. The Yolo-ARIA algorithm exhibits a heightened level of certainty, approximately 1%, in effectively identifying various objects such as autos, motorbikes, and trucks. This is illustrated in Fig. 5 additionally, Fig. 6 provides an account of the outcomes derived from the detection and classification procedures performed on nighttime datasets characterized by diminished levels of illumination. Nevertheless, YOLO-ARIA exhibits a minor limitation in accurately recognizing objects of trucks, leading to lower recall results in comparison to YOLOv5x. The outcomes obtained from applying YOLO-ARIA to well-illuminated conditions, as depicted in Figs. 7 and 8, exhibit identical detection and classification outcomes to Yolov5x. Additionally, the researchers present the results achieved utilizing the VOC datasets, which are illustrated in Figs. 9 and 10. Utilizing the aforementioned dataset, The YOLOv5x model demonstrates a modest advantage in the detection of tiny objects.

The YOLO-ARIA method, as proposed, demonstrates a high level of efficacy in efficiently tackling the inherent difficulty of detecting diminutive targets under conditions of limited sight. The implementation of this approach leads in a notable decrease in instances of missed detections and a reduction in errors, hence enhancing the accuracy and dependability of the target detection outcomes.

5. Conclusion

This study aims to tackle the obstacles associated with achieving a more optimal equilibrium between two fundamental objectives, namely detection and classification. The objective is to enhance the performance outcomes in the domain of vehicle detection. To this end, we introduce YOLO-ARIA, a novel approach that seeks to enhance the efficiency and accuracy of detection by leveraging advanced techniques. The primary contributions of this research article encompass: The incorporation of C3X at the input side enables enhanced adaptability to target features and improved performance in detection. The integration of the C3-2 module into the backbone and neck of the YOLO-ARIA model enhances the model's capacity to efficiently recognize targets in situations with limited visibility. The network utilizes a convolutional network for the purpose of feature extraction, as opposed to employing fully linked layers. Additionally, it incorporates the detection and classification of automobiles.

A dataset was generated comprising a range of situations with low visibility conditions, which was utilized for both training the model and conducting performance testing. The experimental findings indicate that our proposed strategy yields substantial enhancements when compared to the original YOLOv5. The mean average precision (mAP) has experienced a notable improvement of 8.3%, while the speed detection has exhibited a significant enhancement of 50%. Furthermore, there has been a 50% reduction in the number of layers in the network, leading to a significant 50% fall in the number of GFLOPs. In contrast to previous models that have been developed in recent years, the YOLO-ARIA model exhibits some advantages in terms of mean average precision (mAP), speed of detection, number of layers, and GFLOPs value.

Nevertheless, it is important to acknowledge that the strategy provided in this article does possess certain limitations in our future study. The presence of overlapping between small light sources and the detection targets may result in a reduction in the accuracy of the algorithm's detection capabilities. Additional research and development are required in order to devise methodologies that can accurately distinguish between minuscule sources of light and genuine detection targets.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, Aria Hendrawan and Oky Dwi Nurhayati; methodology, Aria Hendrawan and Christine Dewi; software, Aria Hendrawan; validation, Aria Hendrawan, Rahmat Gernowo, and Oky Dwi Nurhayati; formal analysis, Aria Hendrawan; investigation, Rahmat Gernowo, Oky Dwi Nurhayati; resources, Aria Hendrawan; data curation, Aria Hendrawan; writing—original draft preparation, Aria Hendrawan; writing—review and editing, Aria Hendrawan, Oky Dwi Nurhayati, and Christine Dewi; visualization, Aria Hendrawan; supervision, Rahmat Gernowo, and Oky Dwi Nurhayati; project administration, Aria Hendrawan; funding acquisition, Aria Hendrawan.

Acknowledgments

This work was supported of Universitas Semarang, and Universitas Diponegoro, Semarang, Indonesia.

References

[1] N. Nigam, D. P. Singh, and J. Choudhary, "A Review of Different Components of the Intelligent Traffic Management System (ITMS)", *Symmetry*, Vol. 15, No. 3, 2023, doi: 10.3390/sym15030583.

[2] V. Desai, S. Degadwala, and D. Vyas, "Multi-Categories Vehicle Detection For Urban Traffic Management", In: *Proc. of 2023 Second International Conference on Electronics and Renewable Systems (ICEARS)*, pp. 1486–1490, 2023, doi: 10.1109/ICEARS56392.2023.10085376.

[3] C. J. Lin and J. Y. Jhang, "Intelligent Traffic-Monitoring System Based on YOLO and Convolutional Fuzzy Neural Networks", *IEEE Access*, Vol. 10, pp. 14120–14133, 2022, doi: 10.1109/ACCESS.2022.3147866.

[4] S. Bougharriou, F. Hamdaoui, and A. Mtibaa, "Linear SVM classifier based HOG car detection", In: *Proc. of 2017 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, pp. 241–245, 2017, doi: 10.1109/STA.2017.8314922.

[5] H. David and T. A. Athira, "Improving the Performance of Vehicle Detection and Verification by Log Gabor Filter Optimization", In: *Proc. of 2014 Fourth International Conference on Advances in Computing and Communications*, pp. 50–55, 2014, doi: 10.1109/ICACC.2014.18.

[6] M. I. B. Ahmed, R. Zaghdoud, M. S. Ahmed, R. Sendi, S. Alsharif, J. Alabdulkarim, B. A. A. Saad, R. Alsabt, A. Rahman, and G. Krishnasamy, "A Real-Time Computer Vision Based Approach to Detection and Classification of Traffic Incidents", *Big Data and Cognitive Computing*, Vol. 7, No. 1, 2023, doi: 10.3390/bdcc7010022.

[7] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection using Gabor filters and support vector machines", In: *Proc. of 2002 14th International Conference on Digital Signal Processing Proceedings. DSP 2002 (Cat. No.02TH8628)*, Vol. 2, pp. 1019–1022, 2002.

[8] Y. Wei, Q. Tian, J. Guo, W. Huang, and J. Cao, "Multi-vehicle detection algorithm through combining Harr and HOG features", *Math Comput Simul*, Vol. 155, pp. 130–145, 2019, doi: 10.1016/j.matcom.2017.12.011.

[9] X. Wen, L. Shao, W. Fang, and Y. Xue, "Efficient Feature Selection and Classification for Vehicle Detection", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 25, No. 3, pp. 508–517, 2015, doi: 10.1109/TCSVT.2014.2358031.

[10] G. Yan, M. Yu, Y. Yu, and L. Fan, "Real-time vehicle detection using histograms of oriented gradients and AdaBoost classification", *Optik (Stuttg)*, Vol. 127, No. 19, pp. 7941–7951, 2016, doi: 10.1016/j.ijleo.2016.05.092.

[11] N. Baghdadi and S. Aboutabit, "A Comparative Study of Vehicle Detection Methods", In: *Proc. of Advanced Intelligent Systems for Sustainable Development (AI2SD'2018)*, pp. 916–927, 2019.

[12] N. Seenouvong, U. Watchareeruetai, C. Nuthong, K. Khongsomboon, and N. Ohnishi, "Vehicle detection and classification system based on virtual detection zone", In: *Proc. of 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pp. 1–5, 2016, doi: 10.1109/JCSSE.2016.7748886.

[13] A. Mohamed, A. Issam, B. Mohamed, and B. Abdellatif, "Real-time Detection of Vehicles Using the Haar-like Features and Artificial Neuron Networks☆", *Procedia Comput Sci*, Vol. 73, pp. 24–31, 2015.

[14] W. Zhang, Y. Zheng, Q. Gao, and Z. Mi, "Part-Aware Region Proposal for Vehicle Detection in High Occlusion Environment", *IEEE Access*, Vol. 7, pp. 100383–100393, 2019, doi: 10.1109/ACCESS.2019.2929432.

[15] N. Seenouvong, U. Watchareeruetai, C. Nuthong, K. Khongsomboon, and N. Ohnishi, "A computer vision based vehicle detection and

counting system", In: *Proc. of 2016 8th International Conference on Knowledge and Smart Technology (KST)*, pp. 224–227, 2016, doi: 10.1109/KST.2016.7440510.

[16] M. Haris and A. Glowacz, "Road Object Detection: A Comparative Study of Deep Learning-Based Algorithms", *Electronics (Basel)*, Vol. 10, No. 16, 2021, doi: 10.3390/electronics10161932.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", In: *Proc. of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[18] C. M. Liu and J. C. Juang, "Estimation of Lane-Level Traffic Flow Using a Deep Learning Technique", *Applied Sciences*, Vol. 11, No. 12, 2021, doi: 10.3390/app11125619.

[19] K. Wanzeng, H. Jichen, J. Mingyang, Y. Jinliang, C. Weihua, H. Hua, and Z. Haigang, "YOLOv3-DPFIN: A Dual-Path Feature Fusion Neural Network for Robust Real-Time Sonar Target Detection", *IEEE Sens J*, Vol. 20, No. 7, pp. 3745–3756, 2020, doi: 10.1109/JSEN.2019.2960796.

[20] R. Ojala, J. Vepsäläinen, and K. Tammi, "Motion detection and classification: ultra-fast road user detection", *J Big Data*, Vol. 9, No. 1, 2022, doi: 10.1186/s40537-022-00581-8.

[21] J. Z. Jingyi, H. Shengnan, D. Chenxu, Z. Haiyang, Z. Li, J. Zhanlin, and G. Ivan, "Improved Vision-Based Vehicle Detection and Classification by Optimized YOLOv4", *IEEE Access*, Vol. 10, pp. 8590–8603, 2022, doi: 10.1109/ACCESS.2022.3143365.

[22] X. Jia, Y. Tong, H. Qiao, M. Li, J. Tong, and B. Liang, "Fast and accurate object detector for autonomous driving based on improved YOLOv5", *Sci Rep*, Vol. 13, No. 1, 2023, doi: 10.1038/s41598-023-36868-w.

[23] H. Liu, X. Duan, H. Lou, J. Gu, H. Chen, and L. Bi, "Improved GBS-YOLOv5 algorithm based on YOLOv5 applied to UAV intelligent traffic", *Sci Rep*, Vol. 13, No. 1, 2023, doi: 10.1038/s41598-023-36781-2.

[24] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", In: *Proc of the IEEE*, Vol. 86, No. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", *Advances in Neural Information Processing Systems*, F. Pereira, C. J. Burges, L. Bottou, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[26] S. Christian, W. Liu, Y. Jia, S. Pierre, R. Scott, A. Dragomir, E. Dumitru, V. Vincent, and R. Andrew, "Going deeper with convolutions", In: *Proc. of 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA, pp. 1–9, 2015, doi: 10.1109/CVPR.2015.7298594.

[27] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", In: *Proc. of 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA*, 2015, 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", In: *Proc. of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[29] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", *ArXiv*, Vol. abs/1704.04861, 2017.

[30] C. Hsu, C. L. Huang, and C. H. Chuang, "Vehicle detection using simplified fast R-CNN", In: *Proc. of 2018 International Workshop on Advanced Image Technology (IWAIT)*, pp. 1–3, 2018, doi: 10.1109/IWAIT.2018.8369767.

[31] K. Shi, H. Bao, and N. Ma, "Forward Vehicle Detection Based on Incremental Learning and Fast R-CNN", In: *Proc. of 2017 13th International Conference on Computational Intelligence and Security (CIS)*, pp. 73–76, 2017, doi: 10.1109/CIS.2017.00024.

[32] L. Wang, Y. Lu, H. Wang, Y. Zheng, H. Ye, and X. Xue, "Evolving boxes for fast vehicle detection", In: *Proc. of 2017 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1135–1140, 2017, doi: 10.1109/ICME.2017.8019461.

[33] F. Jubayer, J. A. Soeb, A. N. Mojumder, M. K. Paul, P. Barua, S. Kayshar, S. S. Akter, M. Rahman, and A. Islam, "Detection of mold on the food surface using YOLOv5", *Curr Res Food Sci*, Vol. 4, pp. 724–728, 2021, doi: 10.1016/j.crfs.2021.10.003.

[34] X. Zhu, J. Liu, X. Zhou, S. Qian, and J. Yu,

"Enhanced feature Fusion structure of YOLO v5 for detecting small defects on metal surfaces", *International Journal of Machine Learning and Cybernetics*, 2023, doi: 10.1007/s13042-022-01744-y.

[35] G. C. Yang and Z. W. Dong, "Cross Convolutional Neural Networks", In: *Proc. of 2020 IEEE 6th International Conference on Control Science and Systems Engineering (ICCSSE)*, Beijing, China, 2020.

[36] A. Ajit, K. Acharya, and A. Samanta, "A Review of Convolutional Neural Networks", In: *Proc. of 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, pp. 1–5, 2020, doi: 10.1109/ic-ETITE47903.2020.049.

[37] D. Valsesia, G. Fracastoro, and E. Magli, "Deep Graph-Convolutional Image Denoising", *IEEE Transactions on Image Processing*, Vol. 29, pp. 8226–8237, 2020, doi: 10.1109/TIP.2020.3013166.

[38] P. Li, P. Jiang, S. Zeng, and R. Fan,"Parallel Concatenated Network with Cross-layer Connections for Image Recognition", In: *Proc. of Service-Oriented Computing – ICSOC 2018 Workshops*, pp. 80–88, 2019.

[39] Z. Yu, T. Li, G. Luo, H. Fujita, N. Yu, and Y. Pan, "Convolutional networks with cross-layer neurons for image recognition", *Inf Sci (N Y)*, Vol. 433–434, pp. 241–254, 2018, doi: 10.1016/j.ins.2017.12.045.

[40] A. Verma, T. Meenpal, and B. Acharya, "A Deep Convolutional Neural Network for Interrelationship Identification between Humans from Images", In: *Proc. of 2018 Conference on Information and Communication Technology (CICT)*, pp. 1–6, 2018, doi: 10.1109/INFOCOMTECH.2018.8722391.

[41] H. Zhang, M. Tian, G. Shao, J. Cheng, and J. Liu, "Target Detection of Forward-Looking Sonar Image Based on Improved YOLOv5", *IEEE Access*, Vol. 10, pp. 18023–18034, 2022, doi: 10.1109/ACCESS.2022.3150339.

[42] Y. Fang, Y. Ma, X. Zhang, and Y. Wang, "Enhanced YOLOv5 algorithm for helmet wearing detection via combining bi-directional feature pyramid, attention mechanism and transfer learning", *Multimed Tools Appl*, Vol. 82, No. 18, pp. 28617–28641, 2023, doi: 10.1007/s11042-023-14395-0.

[43] S. Guo, L. Li, T. Guo, Y. Cao, and Y. Li, "Research on Mask-Wearing Detection Algorithm Based on Improved YOLOv5", *Sensors*, Vol. 22, No. 13, Jul. 2022, doi: 10.3390/s22134933.

[44] M. I. B. Ahmed, R. Zaghdoud, M. S. Ahmed, R. Sendi, S. Alsharif, J. Alabdulkarim, B. A. A. Saad, R. Alsabt, A. Rahman, and G. Krishnasamy, "A Real-Time Computer Vision Based Approach to Detection and Classification of Traffic Incidents", *Big Data and Cognitive Computing*, Vol. 7, No. 1, 2023, doi: 10.3390/bdcc7010022.

[45] Z. Xu, B. Wei, and J. Zhang, "Reproduction of spatial–temporal distribution of traffic loads on freeway bridges via fusion of camera video and ETC data", *Structures*, Vol. 53, pp. 1476–1488, 2023, doi: 10.1016/j.istruc.2023.05.023.

[46] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge", *Int J Comput Vis*, Vol. 88, No. 2, pp. 303–338, 2010, doi: 10.1007/s11263-009-0275-4.

[47] R. C. Chen, C. Dewi, Y. C. Zhuang, and J. K. Chen, "Contrast Limited Adaptive Histogram Equalization for Recognizing Road Marking at Night Based on Yolo Models", *IEEE Access*, Vol. 11, pp. 92926–92942, 2023, doi: 10.1109/ACCESS.2023.3309410.

[48] C. J. Lin and J. Y. Jhang, "Intelligent Traffic-Monitoring System Based on YOLO and Convolutional Fuzzy Neural Networks", *IEEE Access*, Vol. 10, pp. 14120–14133, 2022, doi: 10.1109/ACCESS.2022.3147866.

[49] G. Jocher, "YOLOv5 by Ultralytics (Version 7.0) [Computer software]", 2020, Accessed: Nov. 03, 2023. [Online]. Available: https://doi.org/10.5281/zenodo.3908559