Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.052138

ARTICLE

# MCBAN: A Small Object Detection Multi-Convolutional Block Attention Network

**Hina Bhanbhro[1,*], Yew Kwang Hooi[1], Mohammad Nordin Bin Zakaria[1], Worapan Kusakunniran[2] and Zaira Hassan Amur[1]**

[1]Computer and Information Science Department, Universiti Teknologi PETRONAS, Seri Iskandar, 31750, Malaysia

[2]Faculty of Information and Communication Technology, Mahidol University, Nakhon Pathom, 73170, Thailand

*Corresponding Author: Hina Bhanbhro. Email: hina_22007940@utp.edu.my

Received: 24 March 2024 Accepted: 23 July 2024 Published: 18 November 2024

## ABSTRACT

Object detection has made a significant leap forward in recent years. However, the detection of small objects continues to be a great difficulty for various reasons, such as they have a very small size and they are susceptible to missed detection due to background noise. Additionally, small object information is affected due to the downsampling operations. Deep learning-based detection methods have been utilized to address the challenge posed by small objects. In this work, we propose a novel method, the Multi-Convolutional Block Attention Network (MCBAN), to increase the detection accuracy of minute objects aiming to overcome the challenge of information loss during the downsampling process. The multi-convolutional attention block (MCAB); channel attention and spatial attention module (SAM) that make up MCAB, have been crafted to accomplish small object detection with higher precision. We have carried out the experiments on the Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) and Pattern Analysis, Statical Modeling and Computational Learning (PASCAL) Visual Object Classes (VOC) datasets and have followed a step-wise process to analyze the results. These experiment results demonstrate that significant gains in performance are achieved, such as 97.75% for KITTI and 88.97% for PASCAL VOC. The findings of this study assert quite unequivocally the fact that MCBAN is much more efficient in the small object detection domain as compared to other existing approaches.

## KEYWORDS

Multi-convolutional; channel attention; spatial attention; YOLO

## 1 Introduction

The detection of objects is crucial for many tasks, which include autonomous driving [1], facial recognition [2], defect detection [3], remote sensing [4], and engineering symbol classification [5]. However, the detection and classification of small objects are difficult due to the scarcity of such information and susceptibility to noise that comes from the background. Currently, there are two main types of deep learning-based object detection algorithms: two-stage methods like Region-based Convolutional Neural Network (R-CNN) [6], spatial pyramid pooling (SPP)-Net [7], Fast R-CNN [8], Faster R-CNN [9], Region-based Fully Convolutional Network (R-FCN) [3] and Mask R-CNN

[10]; and one-stage methods like you look only once (YOLO) [11], Solid-State Drive (SSD) [12], retina network (RetinaNet), and efficient network (EfficientDet) [13]. Generally, two-stage detectors include the first step which is region proposal followed by object selection to achieve highly accurate results and fast detection speed but anchored methods consider low performance in a real-time situation. While the two-stage algorithms use a fully end-to-end convolutional neural network for base, followed by a region of interest (ROI)-Pooling [14] module for region of interest localization, the latter kind of algorithms employ a single architecture that can be optimized for high speed and accurate detection.

For our study, we chose the one-stage framework YOLOv8 as the baseline model due to its robust performance against various detection tasks. However, YOLOv8, while being a general-purpose object detection network, doesn't excel in identifying small objects [15]. The object dimensions of typical objects in crowded regions of images are small, which increases the risk of losing information about small object features at the deeper layers due to YOLOv8's large down-sampling factor that may lead to information loss. Therefore, it is crucial to address the detection of small objects for better classification accuracy and overall model performance.

An effective approach involves merging low-level and high-level feature details to improve the accuracy of high-level feature information. To retain the richness of low-level details during processing, connections linking the highest and lowest levels have been established, as exemplified by the Path Aggregation Network (PANet) [16]. Additionally, down-sampling often results in the loss of important details in small objects, especially those smaller than $32 \times 32$ pixels, which can merge with the background due to their low resolution [17].

The application of attention mechanisms with detection models greatly enhances detection accuracy but often requires additional algorithm parameters. Using these attention mechanisms, convolutional neural networks (CNNs) can be fine-tuned to adjust the parameterization of input features. The network automatically identifies significant local information while disregarding irrelevant details. This integration of convolutions with attention allows CNNs to focus on key aspects and perform well across various tasks, including image classification, computer vision, and object recognition.

In the channel attention module of the Convolutional Block Attention Module (CBAM) [18], there is a problem associated with the use of the shared Multi-Layer Perceptron (MLP), which has limited its capability of capturing diverse channel-wise relations. This limitation can lead to the attention module's ability to selectively emphasize relevant channels for different parts of the input feature map [19]. Consequently, the network may not fully exploit the rich information in the feature map, potentially resulting in sub-optimal performance in tasks where capturing fine-grained channel dependencies is critical, such as object detection in cluttered scenes with small objects.

This paper introduces a novel approach called MCBAN to address the previously mentioned issues, which incorporates a novel multi-convolutional attention block (MCAB). The MCAB (Fig. 1), mechanism can adaptively re-calibrate channel feature responses, enabling the network to focus on important features while suppressing irrelevant ones, thereby significantly improving the accuracy of small object detection. We conducted extensive experiments on the KITTI and PASCAL VOC datasets to demonstrate the superior performance of MCBAN compared to mainstream algorithms such as YOLOv5, YOLOv7, and YOLOv8.

The primary contributions of this study can be outlined as follows:

1. We introduce a novel MCBAN specifically designed for detecting small objects, significantly enhancing the accuracy of small object detection.

2. A well-crafted MCAB is introduced to effectively reduce the problem of information loss for small objects during the down-sampling process.
3. The use of Channel Attention (CA) and Spatial Attention Module (SAM) improves the ability to locate small objects which increases visibility resulting in higher accuracy.
4. The MCBAN was evaluated through experiments on the KITTI and PASCAL VOC datasets.

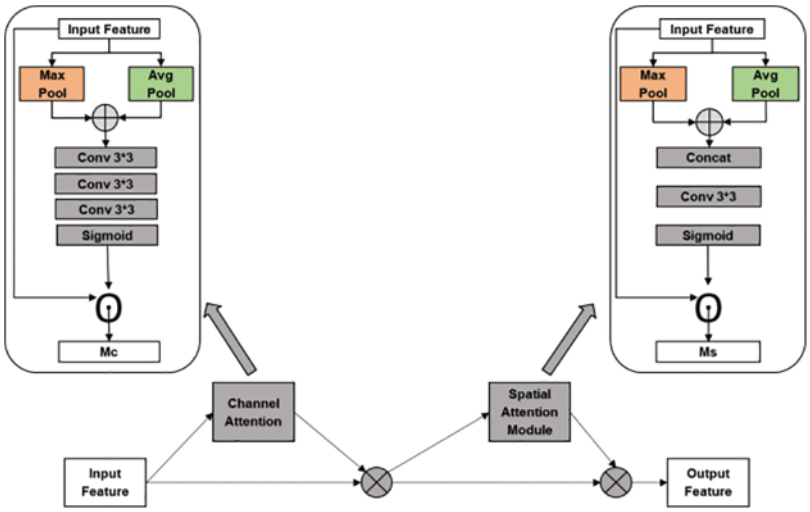


**Figure 1:** The structure of MCAB

The following sections of this paper are organized as follows: "Related work" provides for the review of literature. The novel MCBAN is presented in "Proposed method", and later on, experimental results and discussion are given in "Experiments and Discussion". Finally, the "Conclusion" summarizes the work.

## 2 Related Work

In this section, we carry out a thorough investigation of current literature related to object detection algorithms and attention mechanisms. We also focus on different techniques that will aid in addressing the problem of detecting small objects. Furthermore, we emphasize the crucial role of attention mechanisms for obtaining better accuracy and efficiency of object detection models, especially when dealing with tiny objects.

### *2.1 Methods for Detecting Small Objects*

The detection of small objects is a significant challenge and a primary focus in computer vision. The field of deep learning has seen remarkable advancements, largely driven by progress in object detection applications [20]. However, a major challenge remains: the aggregation of small objects often leads to the loss of crucial information. Many research efforts rely on large networks to boost accuracy, making the detection process increasingly resource-intensive. To address this, a layered approach using MobileNetv2 and depth-wise separable convolution has been proposed to mitigate model complexity. Additionally, the Attentional Feature Fusion Module (AFFM) [21] has been utilized to merge semantically inconsistent features, improving model accuracy for small objects. A study [22] highlighted that the precision of local details is weakened by cross-layer feedback in the

Feature Pyramid Network (FPN). This has led researchers to propose a fusion factor to regulate the information flow from deeper to shallower layers, enhancing performance in small object detection.

Wang et al. [23] concluded that if we create high-resolution images or feature maps, it would further increase the detection accuracy on small objects but this process is compute-intensive. For this purpose, they offered Single Line Electrical Drawings (SLED), which speed up the single-line symbol classification based on the novel image enhancement approach. Wang et al. in their study achieved the task of the advancement in detection accuracy on benchmark datasets, e.g., Common Objects in Context (COCO) and vision meets drone (VisDrone). Reference [23] put forward a multi-stage feature enhancement pyramid network that produces a good resolution on a small scale of the objects, and joint detection of the objects on a large scale in the remote sensing images.

In deep learning, the vanishing gradient problem occurs when gradients become very small during back-propagation, impeding the training of deep neural networks. This issue slows down or even halts learning in earlier layers. Techniques, such as rectified linear unit (ReLU) activation functions and batch normalization help mitigate this problem, enabling the effective training of deeper networks [24].

However, those approaches do help to detect small objects to some extent, but more specific research is expected to solve the problem of detecting small objects. The first issue for small objects is that they have small dimensions and content, this poses a challenge in feature recognition. Additionally, accurate positioning for smaller objects can be impaired as a result of downsampling, and this can lead to information loss. Furthermore, there is the issue of background interference which is always influenced by the variations in brightness and crowded scenes which is often another huge challenge in ensuring accurate and appropriate detection.

### *2.2 Attention Mechanism*

The attention mechanism in deep learning elaborated in [24], which resembles the visual and cognitive procedures typical for a human brain, is a part of the architecture for convolutional neural networks. It enables networks to specialize in certain segments of the input data and improves the accuracy and generalization of the model. Common attention mechanisms include sequence and excitation (SE), convolutional block attention module (CBAM), and efficient channel attention (ECA).

The work presented in [25] proposed the convolutional block attention module (CBAM), which pays attention to both the channel and the spatial dimensions of an input feature map by computing the position-wise feature input and then using the acquired map to refine the input feature map adaptively. The efficient channel attention (ECA) module proposed by [25] is also helpful in maintaining performance without reduction of the dimension as it can effectively pay attention to the information from different channels. In an effort to learn features, Shen et al. [26] put forward the Squeeze-and-Excitation (SE)-block as a Squeeze-and-Excitation framework that dynamically fashions features of the channels to improve the detection precision.

Integration of attention mechanisms certainly improves the detection ability, but the algorithms will have to incur an increase in the number of parameters involved. However, in Convolutional neural networks (CNNs), the component of attention allows for a dynamical distribution of input weights among different features with a pattern of focusing on vital elements and suppressing the others. By means of coupling [26], convolution operations with attention mechanisms, CNNs can be facilitated to discover and concentrate on the most significant features which helps get better results in such tasks as image classification and image recognition. This work introduces MCBAN, which provides additional

features to the algorithm, improving the detection performance without increasing the number of parameters.

## 3 Proposed Method

In this part, we first define our proposed method: the Multi-Convolutional Attention Network (MCBAN). Following this overview, we provide a detailed explanation of the central components of MCAB: Multi-Convolutional Attention Block (MCAB) and inducing Channel Attention (CA) spatial attention with the Spatial Attention Module (SAM).

### *3.1 Multi-Convolutional Attention Network*

The MCBAN framework is composed of three key parts: the Backbone, which is responsible for feature extraction; the Neck, which serves to enhance the feature pyramid; and the Prediction, which handles the final output, as depicted in Fig. 2. The main building block of the stem, called coarse-to-fine (C2f), replaced the second layer of convolution (C2), and the initial 6 × 6 convolution is now 3 × 3, and convolution, batch normalization, and SiLu activation functions (Constraint-Based Search (CBS)) for a block consisting of a convolution, a BatchNorm, and a sigmoid-weighted linear unit (SiLU) layer. The outputs of all bottlenecks are concatenated in C2f, while the third layer of convolution (C3) only uses the output from the preceding bottleneck.

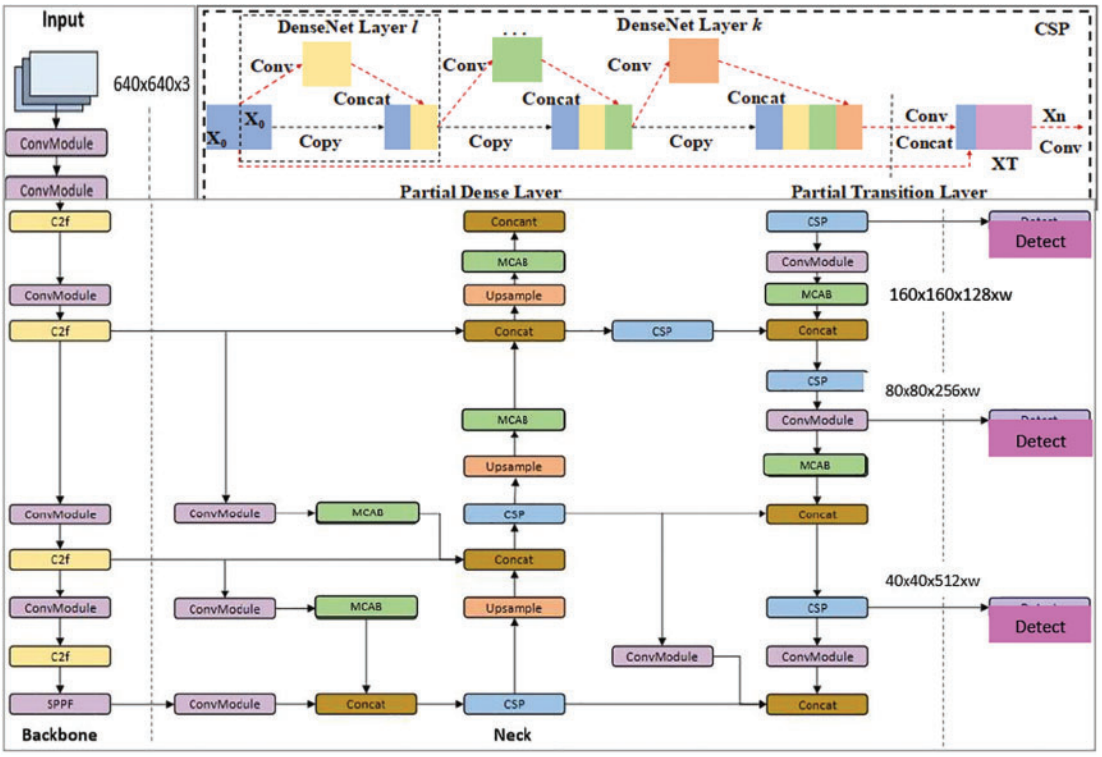


**Figure 2:** An outline of MCBAN. This model is built upon the YOLOv8 architecture and introduces two new attention modules. It also retains several existing modules from YOLOv8, including Conv, C2f, SPPF, Concat, Upsample, and Detect

The kernel size of the first convolution was changed from 2 × 2 to 3 × 3, but the bottleneck structure remains consistent with YOLOv8. This change suggests a return to the Residual Network (ResNet) block established in 2015. Features are concatenated directly in the neck without needing to match channel dimensions, reducing parameter count and overall tensor size. However, YOLOv8 differs from anchor-based models by being anchor-free. Instead of predicting an object's offset from a predefined anchor box, it directly predicts the object's center. This approach simplifies non-maximum suppression (NMS), a complex post-processing step that filters out redundant detection, by reducing the number of bounding box predictions.

In this study, we utilize the KITTI and Pascal VOC datasets for object detection as a case study. Our methodology starts with an input image of size 640 × 640, which undergoes a series of operations through the Backbone network and Neck, followed by further feature integration and channel adjustment in the Prediction phase. The YOLOv8 loss function is constructed by merging the bounding box, classification, and confidence loss components. Eq. (1) is applied to determine the position of the bounding box:

$$\cup_{x}^{y}\ x, y = IoU_{P}^{G} \tag{1}$$

In Eq. (1), the variables $x$ and $y$ denote the coordinates of the $y$th bounding box within the $x$th grid cell. The probability associated with this bounding box is denoted by ∪. If the $y$th bounding box contains an object, $P_{x,y}$ is set to 1; otherwise, it is set to 0. The Intersection over Union ($IoU$) between the predicted class $P$ and the actual ground truth $G$ is measured by $IoU_{groundtruth}$, where a higher $IoU$ signifies more precise bounding box predictions. The complete YOLOv8 [27] loss function is described by Eq. (2):

$$loss_{YOLOv8} = loss_{boundingbox} + loss_{classification} + loss_{confidence} \tag{2}$$

where $loss_{boundingbox}$ computes the accuracy of predicted bounding boxes as a difference of ground truth and predicted values using IoU. Whereas, the $loss_{classification}$ component calculates the cross-entropy loss to predict class labels. Lastly, the $loss_{confidence}$ measures the confidence scores of predicted bounding boxes to ensure that the models learn to predict accurate bounding boxes. Minimizing the total loss function during training helps enhance the overall performance of the YOLOv8 object detection model.

### *3.2 Multi-Convolutional Attention Block*

We place the MCAB module behind the Conv or Upsample module in the feature fusion process to make the model only focus dynamically on input areas after feature extraction. MCAB aims to enhance the sharp outputs of input feature maps, increasing the ability to learn from the details at a better rate. The MCBAN model is an extension of the YOLOv8 architecture, integrating two novel attention modules, including the MCAB while preserving key components from YOLOv8. These components consist of the Convolutional (Conv) layers, the C2f module, the Spatial Pyramid Pooling with Feature Fusion (SPPF) module, the Concatenation (Concat) operation, the Upsample module, and the Detection (Detect) module. By leveraging the YOLOv8 framework and incorporating attention mechanisms like the MCAB, the MCBAN model aims to enhance its object detection performance, emphasizing the importance of specific channels in the feature maps for more accurate and efficient object localization and recognition.

### *3.2.1 Channel Attention (CA)*

We propose the use of a Channel Attention (CA) module to introduce multi-convolutional operations that enhance feature extraction from small objects while simultaneously reducing the number of parameters and increasing detection accuracy, as shown in Fig. 3. The module operates as follows:

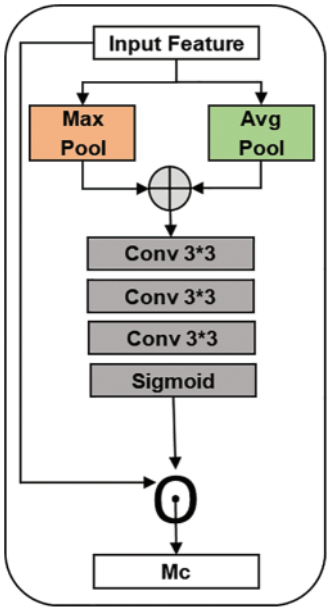


**Figure 3:** The structure of CA

The process begins with a 3 × 3 convolution to decrease the channel dimension and reduce the parameter count of the feature map. The module is crafted to improve the extraction of features from small objects while minimizing parameters and enhancing detection accuracy. This block includes two branches. The first branch starts with a 3 × 3 convolution, followed by max pooling with a stride of 2 and a kernel size of 3, focusing on capturing edge information. The second branch begins by halving the feature map size using a 3 × 3 convolution with a stride of 2. Both branches subsequently apply a 3 × 3 convolution to extract features from small objects, fostering cross-channel interaction and information integration.

The process starts in the CA with the application of the adaptive average pooling and max pooling on the input feature map leading to the channel-wise representation. Such incision is of great significance because it allows for a better illustration of the global context of the main functions of channels. Following that, a convolutional operation is included in order for the network to learn channel-wise interactions, in which it could also focus on the pure channels and figure out the relevant spatial patterns that will be extracted. The output of the cross-operation submits to an activation function of the sigmoid to receive the attention weights that show how important it is in each channel. The mentioned attention weights are then relied upon assisting element-wise multiplication, which results in highlighting the important relations and leaving behind the less important ones. Thus, such a channel-wise attention mechanism makes the network adaptive at its feature level. It enables the network to reshape its feature response, and eventually, it leads to better performance in the areas of

object detection and image classification. The overall function is given in Eq. (3):

$$Mc(F) = \sigma(W * (Favg + Fmax)) \tag{3}$$

where $Mc(F)$ is the input feature map, and $Favg$ denotes adaptive average pooling and $Fmax$ is max pooling. Whereas, $W$ represents weight matrices, and $\sigma$ is a non-linear activation function.

The computation of adaptive $Favg$ is represented by Eq. (4), The adaptive average pooling operation computes a $3 \times 3$ output for each channel of the input feature map. This can be achieved using a convolutional layer with a kernel size equal to the height and width of the input feature map, and a stride equal to the height and width of the input feature map.

### *3.2.2 Spatial Attention Module*

In images with small objects, the importance of information varies across different parts of the image. For instance, edge position information for small objects is generally more crucial than information from other areas. Therefore, the Spatial Attention Module (SAM) plays a critical role in enhancing such vital information. This paper proposes a combined attention module, where a SAM module is utilized after a Channel Attention (CA) to generate a two-dimensional SAM map. Unlike channel-wise attention, SAM complements and extends the functionality of CA by focusing more on content information in spatial positions. By assigning weights to each spatial position, SAM identifies the most important spatial position information, enhancing the features of that particular area while suppressing noise features. Following channel-wise attention, a weight-shared SAM block is employed to refine spatial information. The structure of the SAM module is illustrated in Fig. 4.

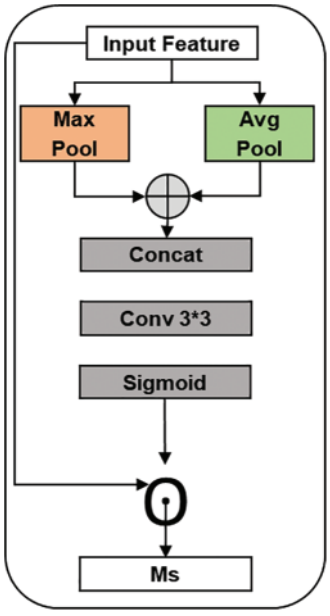


**Figure 4:** Architecture outline of Spatial Attention Module (SAM)

In the SAM module, max-pooling and average pooling operations are applied to the channel axis of the feature map. These operations help aggregate channel information, improving the retention and extraction of texture features. The process is depicted as follows:

$$G1 = [maxPool(F) \cdot avgPool(F)]\pi r^2 \tag{4}$$

where $F$ represents the input feature map, and $[\cdot]$ indicates a concatenation process. After three $3 \times 3$ convolution operations, the receptive field of the feature map is extended. A two-dimensional SAM map is created using a sigmoid function, which extracts local details from the feature map. This local information is then integrated with global data obtained from the MCAB module. The computation is expressed as follows:

$$AG2 = \sigma f\, 3 \times 3\, ([Favg;\ Fmax]) \tag{5}$$

Next, the weight of the final Spatial Attention Module (SAM) is applied to the original feature map, as calculated in Eq. (6).

$$G3\,(F) = W \times F \tag{6}$$

where $W$ denotes the weight acquired by the SAM module. This allows the network model to extract diverse features by considering both local and global information.

## 4 Experiments and Discussion

In this section, we compared MCBAN with state-of-the-art approaches through experiments conducted on the PASCAL VOC and KITTI datasets. We also carried out ablation studies to evaluate the impact of the techniques implemented in MCBAN.

### *4.1 Implementation Details*

The experimental setup comprises a 13th Gen Intel Core i5-13500 processor, an Intel UHD (Ultra High Definition) Graphics 770 GPU (Graphics Processing Unit), 16 GB of (Random Access Memory) RAM, and Windows 10 as the operating system. The experiments are conducted using PyCharm as the integrated development environment, Python as the programming language, and PyTorch as the deep learning framework. For network optimization, the Adam optimizer is employed, with input image dimensions of (640, 640). The initial learning rate is set at 0.001, with a batch size of 4, and the network is trained over 100 epochs.

### *4.2 Datasets*

The KITTI dataset is gathered using a vehicle outfitted with a dashboard camera and additional sensors to facilitate testing and benchmarking for autonomous driving [28]. It comprises 7481 training images annotated with seven distinct classes: cars, vans, trams, trucks, pedestrians, people sitting, and cyclists.

The KITTI dataset, beyond its diverse range of object classes, includes a variety of data modalities such as stereo images, 3D point clouds, global positioning system (GPS) coordinates, and inertial measurement unit (IMU) data. This multimodal aspect allows for the development and testing of algorithms that can leverage multiple sources of information for improved object detection and scene understanding. Furthermore, KITTI's benchmarks cover several tasks including object detection, tracking, and road/lane estimation, making it a versatile dataset for comprehensive evaluation of autonomous driving systems. The complexity of KITTI's real-world scenarios, such as varying traffic densities and dynamic environments, poses significant challenges that help to push the boundaries of current detection models.

The PASCAL VOC dataset, created by Everingham et al. in 2010 [29], is a publicly accessible dataset designed for object detection tasks. It contains 20 object categories with variations in scales

and poses. For this study, we utilize the training sets from PASCAL VOC 2007 and PASCAL VOC 2012, totaling 16,551 images, for training purposes. The model's performance is evaluated using the PASCAL VOC 2007 test set, which consists of 4952 images.

The PASCAL VOC dataset, in addition to its 20 object categories, provides extensive annotations that include object class labels, bounding boxes, and detailed segmentations. This rich annotation allows researchers to explore various aspects of object detection, such as localization, classification, and instance segmentation. PASCAL VOC also features a variety of visual contexts, from cluttered backgrounds to objects in diverse poses and occlusions, which test a model's ability to generalize across different conditions. Moreover, the PASCAL VOC challenges have historically driven significant advancements in the field by providing a competitive platform for comparing state-of-the-art methods, thus fostering innovation and continuous improvement in object detection techniques.

#### *4.2.1 Dataset Description and Image Selection Criteria*

The datasets were analyzed to isolate the subsets of small objects. These subsets of images from the datasets are used to assess our network's ability to recognize small objects. The criteria and methodology for selection are as follows:

*Subset Image Selection Criteria*

We isolated a subset of 5700 images by selecting those that contain small objects, defined as objects with bounding box areas below a certain threshold (e.g., less than 5% of the total image area). This threshold was determined based on the distribution of object sizes in the dataset. We used a stratified sampling approach to ensure that the subset is representative of the overall distribution of small objects across different categories and positions within the images.

This provides a comprehensive description of the main datasets employed, including the training set of 4560 images (80% of total images), test sets of 570 images (10% of total images), and validate sets of 570 images (10% of total images). These sets consist instances of small annotated objects, ensuring a balanced representation of various object categories and sizes.

The bounding boxes provide visualization, which gives a clear understanding of the characteristics of the objects, such as spatial distribution, which is crucial for evaluating the model's performance.

### *4.3 Evaluation Metrics*

This paper assesses model performance using multiple metrics, including frames per second (*FPS*), average precision (*AP*), mean average precision (*mAP*), *F1–score*, precision (*P*), and recall (*R*). The *FPS* quantifies the number of frames processed each second. Thu the *AP* represents the area under the precision-recall (*P*-*R*) curve, which plots recall on the *x*-axis and precision on the *y*-axis. Precision and recall are determined using Eqs. (7) and (8).

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

where *TP* denotes the count of correctly detected positive samples, *FP* indicates the count of incorrectly detected positive samples, and *FN* represents the count of falsely detected negative samples, which corresponds to missed detections.

$$A = \int_0^1 P(R)dR \tag{9}$$

where *AP* denotes the area under the precision-recall (*P*-*R*) curve and can be computed using Eq. (10). While the mean average precision (*mAP*) is the average of various *AP* values and can be computed using Eq. (10).

$$mAP = AP1 + AP2 + \cdots + APn/n \tag{10}$$

where *n* represents the total number of object categories.

$$F1 - score = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = \frac{2 \text{ Precision . Recall}}{\text{Precision } + \text{ Recall}} \tag{11}$$

where *F1–score* is the harmonic mean of precision (*P*) and recall (*R*), and it can be computed using Eq. (11).

## 5 Results and Discussion

In this part, we present the results of various experiments that were carried out for this study. A detailed evaluation of results of different models on KITTI and Pascal VOC datasets. Furthermore, the comparison of these results is also discussed in this section.

### *5.1 Evaluation of Results on KITTI Dataset*

To demonstrate the effectiveness of MCBAN, we compared its performance with several mainstream algorithms on the KITTI dataset, including YOLOv5, YOLOv7, and YOLOv8. The comparison results are shown in Table 1, with the results of our model highlighted in bold.

**Table 1:** The results of experiments with different models on the KITTI dataset

| Model | R/% | P/% | F1 | mAP/% | mAP@0.5:0.05:0.95 | Size/MB | Params/M | FPS |
|---|---|---|---|---|---|---|---|---|
| YOLOv5 | 89.7 | 94.7 | 97% | 90.90 | 80.10 | 179.5 | 46 | 14 |
| YOLOv7 | 91.3 | 95.5 | 93% | 91.67 | 87.90 | 17.6 | 47.09 | 49 |
| YOLOv8 | 92.8 | 89.0 | 89% | 92.89 | 88.99 | 180.53 | 65.35 | 28 |
| **MCBAN** | **94.9** | **90.91** | **91%** | **97.75** | **89.89** | **176.96** | **64** | **28** |

Table 1 shows that MCBAN achieves an *mAP* of 97.75% while maintaining a detection speed of 28 *FPS*. Compared to YOLOv8, MCBAN achieves a higher *mAP* by 4.86%, with improvements in recall (*R*), precision (*P*), and *F1–score* by 2.1%, 1.91%, and 0.02%, respectively. This improvement is achieved while reducing the model size by 3.57 MB, demonstrating the effectiveness of the MCBAN algorithm.

This study evaluates the detection accuracy of MCBAN against several leading models for each class in the KITTI dataset, as detailed in Table 2. The highest Average Precision (AP) for an individual object category is highlighted in bold among the four models considered. The results

indicate that MCBAN surpasses mainstream algorithms in most categories, demonstrating superior accuracy. Compared to YOLOv8, which shows the same detection accuracy for three categories and improved accuracy for four categories, MCBAN achieves an increase in detection accuracy of 2.56% for buses and 1.75% for cyclists. Additionally, MCBAN reduces the number of parameters by 1.35%, maintaining the same detection speed while enhancing accuracy. The KITTI dataset includes eight categories: bus, boat, van, airplane, person sitting, truck, cyclist, and car.

**Table 2:** The average precision (AP) percentage for each category within the KITTI dataset

| Class | YOLOv5 | YOLOv7 | YOLOv8 | MCBAN |
|---|---|---|---|---|
| Bus | 89.98 | 95.10 | 95.34 | **97.90** |
| Boat | 100 | 98.98 | 100 | **100** |
| Van | 97.90 | 100 | 100 | **100** |
| Airplane | 78.20 | 53.98 | 65.89 | **74.77** |
| Person_sitting | 90.87 | 88.89 | 99.0 | **99.0** |
| Truck | 90.10 | 86.78 | 96.10 | 90.78 |
| Cyclist | 89.90 | 90.89 | 94.35 | **96.10** |
| Car | 41.80 | 67.90 | 61.90 | **71.34** |

Fig. 4 visually represents the outcomes from Table 2, illustrating that MCBAN achieves the top position in most categories, highlighting its outstanding detection performance.

Fig. 5 displays the distribution of detection accuracy and speed among various models on the KITTI dataset. The results demonstrate that the MCBAN algorithm achieves significantly improved detection performance on the KITTI dataset.

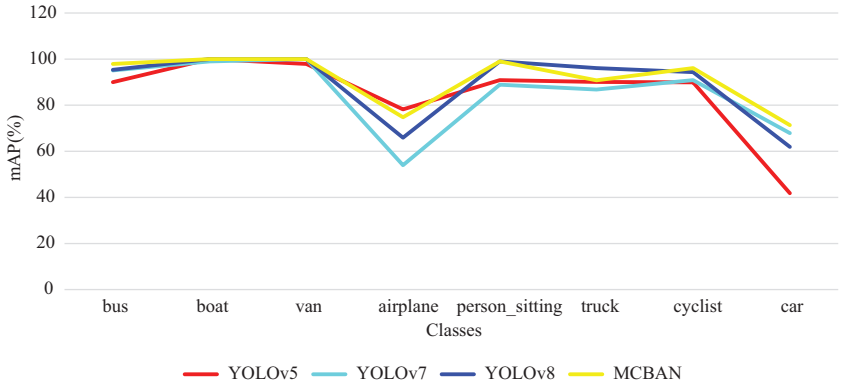


**Figure 5:** The comparison of AP of 8 classes on the KITTI dataset

The results shown in Fig. 6 confirm that MCBAN outperforms other algorithms, consistently achieving the highest scores in most classes. This underscores MCBAN's exceptional ability to detect small objects.

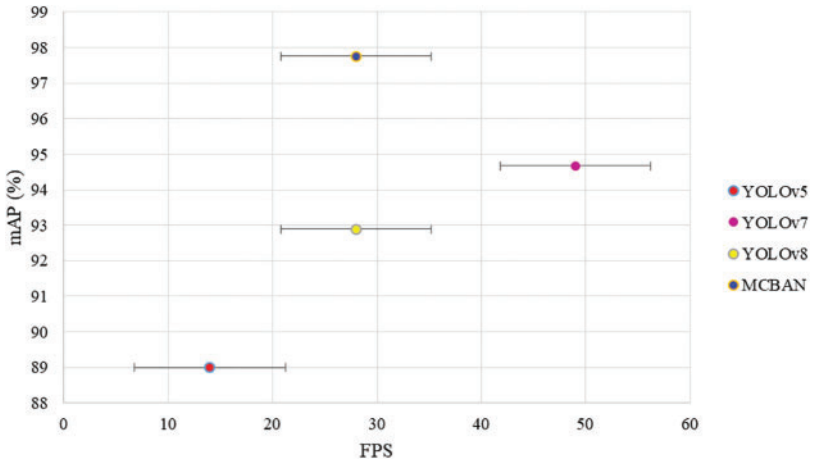


**Figure 6:** The variation in accuracy and speed across different methods on the KITTI dataset

### *5.2 Evaluation of Results on PASCAL VOC Dataset*

Table 3 compares the detection performance of the proposed MCBAN on the PASCAL VOC dataset against YOLOv5, YOLOv7, and YOLOv8. The results for our model are highlighted in bold.

**Table 3:** The results of experiments conducted with various models on the PASCAL VOC dataset

| Model | R/% | P/% | F1 | mAP/% | mAP@0.5:0.05:0.95 | Size/MB | Params/M | FPS |
|---|---|---|---|---|---|---|---|---|
| YOLOv5 | 89.7 | 94.7 | 97% | 87.90 | 78.10 | 179.5 | 46 | 14 |
| YOLOv7 | 91.3 | 95.5 | 93% | 84.67 | 81.90 | 17.6 | 47.09 | 49 |
| YOLOv8 | 91.0 | 90.96 | 88% | 87.0 | 86.99 | 180.53 | 65.35 | 28 |
| **MCBAN** | **91.88** | **92.0** | 90% | **88.97** | **81.89** | **176.96** | **64** | **28** |

Table 3 shows that MCBAN achieves a *mAP* of 88.97% while maintaining a detection speed of 28 *FPS*. Compared to YOLOv8, MCBAN reduces the model size by 4.05 MB and the number of parameters by 1.35%. Furthermore, MCBAN enhances Recall (*R*), Precision (*P*), *F1–score*, and mean Average Precision (*mAP*) by 0.88%, 1.04%, 0.2%, and 1.97%, respectively. These findings show that the proposed algorithm significantly improves detection accuracy for small objects while still meeting real-time detection requirements.

To assess the detection capabilities of MCBAN, this research evaluates its accuracy against mainstream algorithms across each category using the PASCAL VOC dataset, as presented in Table 4. The results for MCBAN are emphasized in bold. MCBAN outperforms mainstream algorithms in accuracy for most categories containing smaller objects and achieves optimal detection results for 10 object classes. Compared to YOLOv8, MCBAN enhances feature capture, leading to higher accuracy in 5 classes. Specifically, for small objects like cat, sofa, and chair, MCBAN improves detection accuracy by 0.95%, 1.22%, and 1.3%, respectively. Although MCBAN shows lower accuracy in three categories compared to YOLOv8, it maintains consistent detection speed, reduces parameter count, and improves mean Average Precision (mAP), leading to superior performance across the majority of classes.

**Table 4:** The Average Precision (AP) percentage for each category within the PASCAL VOC dataset

| Class | YOLOv5 | YOLOv7 | YOLOv8 | MCBAN |
|---|---|---|---|---|
| Cat | 87.90 | 85.10 | 89.40 | **90.35** |
| Bird | 80.79 | 78.98 | 100 | **100** |
| Horse | 98.0 | 80.90 | 100 | **100** |
| Bicycle | 78.20 | 30.90 | 90.71 | **90.10** |
| Person | 39.90 | 89.90 | 99.0 | **99.0** |
| Dog | 80.98 | 70.76 | 96.10 | 90.78 |
| Sheep | 70.10 | 41.50 | 100 | **100** |
| Sofa | 89.89 | 88.86 | 98.78 | **100** |
| Chair | 79.89 | 56.60 | 94.70 | **96.0** |
| Cow | 65.87 | 78.89 | 71.35 | **81.55** |

Fig. 7 presents the distribution results on the PASCAL VOC dataset. The results reveal that MCBAN exceeds YOLOv5 and YOLOv7 in speed and outperforms YOLOv5, YOLOv7, and YOLOv8 in mean Average Precision (*mAP*). This indicates that our model has enhanced the detection of small objects in the dataset, leading to an overall improvement in performance. In conclusion, the proposed MCBAN shows impressive detection accuracy while maintaining a consistent detection speed.

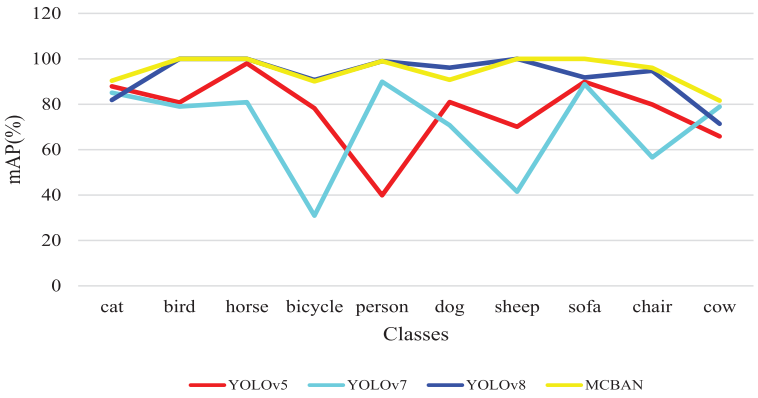


**Figure 7:** The comparison of Average Precision (AP) across 10 classes on the PASCAL VOC dataset

## 6 Ablation Experiments

To validate the effectiveness of each proposed strategy in this study, we conducted ablation experiments on the baseline model using the KITTI and PASCAL VOC datasets, with the experimental results displayed in Table 5. In Table 5, MCAB, Channel Attention (CA), and Spatial Attention Module (SAM) refer to attention module-based detection layers that emphasize small features and extract information from input feature maps.

Table 5 presents experimental results demonstrating that each enhancement strategy improved detection performance to varying extents when integrated into the baseline model. The table details the

detection outcomes of various model configurations on the KITTI and PASCAL VOC datasets, showcasing the performance gains achieved through different strategies. The baseline model, YOLOv8, shows robust results on both datasets, with KITTI achieving higher recall and mAP compared to PASCAL VOC. The Channel Attention (CA) and Spatial Attention Module (SAM) strategies indicate improvements in some metrics but vary between datasets, with CA showing better precision and mAP on KITTI, while SAM provides balanced improvements across both datasets. The MCBAN model demonstrates significant enhancements, particularly on KITTI, achieving the highest recall (94.9% and 91.88%), precision (90.91% and 92.88%), F1–score (91% and 90%), and mAP (97.75% and 88.97%) among all models, reflecting the effectiveness of attention module-based detection layers. Notably, the CA and SAM models reduce the model size and parameters, indicating a trade-off between complexity and performance. The consistent frames per second (FPS) across all models suggest that these improvements do not come at the cost of processing speed. Overall, the table underscores how tailored improvements can significantly enhance model performance on specific datasets.

**Table 5:** The detection results after the introduction of different improved strategies (the bold data indicates the best results in the table)

| Model | Dataset | R/% | P/% | F1 | mAP/% | Size/MB | Params/M | FPS |
|---|---|---|---|---|---|---|---|---|
| YOLOv8 | KITTI | 92.8 | 89.0 | 89% | 92.89 | 180.53 | 65.35 | 28 |
| YOLOv8 | PASCAL VOC | 91.0 | 90.96 | 88% | 87.0 | 180.53 | 65.35 | 28 |
| CA | KITTI | 90.3 | 91.1 | 87% | 88.67 | 22.6 | 57.09 | 28 |
| CA | PASCAL VOC | 90.7 | 89.8 | 86% | 80.76 | 24 | 50 | 28 |
| SAM | KITTI | 89.0 | 85.03 | 81% | 86.0 | 118.3 | 55.35 | 28 |
| SAM | PASCAL VOC | 90.0 | 86.96 | 80% | 84.0 | 122 | 58.3 | 28 |
| **MCBAN** | KITTI | **94.9** | **90.91** | 91% | **97.75** | **176.96** | **64** | **28** |
| **MCBAN** | Pascal VOC | **91.88** | **92.0** | **90%** | **88.97** | **176** | **64** | **28** |

## 7 Conclusion

In conclusion, we have introduced MCBAN, a novel approach that addresses the challenge of information loss for small objects during down-sampling. By combining Channel Attention (CA) with a multi-convolutional attention mechanism and Spatial Attention Module (SAM), MCBAN effectively reduces interference from irrelevant information, leading to improved accuracy in regressing and localizing small objects. Our evaluation of the KITTI and PASCAL VOC datasets demonstrates the superiority of MCBAN over other state-of-the-art algorithms in small object detection, achieving a mean Average Precision (mAP) of 97.75% on KITTI and 88.97% on PASCAL VOC. MCBAN achieves these advancements while maintaining detection speed, showcasing its potential for real-world applications. Additionally, the modular design of MCBAN allows for easy integration into existing object detection frameworks, making it a practical choice for researchers and developers. However, despite the promising performance, the effectiveness of MCBAN might be limited when applied to datasets with different characteristics compared to KITTI or PASCAL VOC, thus compromising the generalizability of the model. Additionally, the computational complexity of the proposed model might pose challenges for real-time applications on resource-constrained devices.

**Acknowledgement:** The authors appreciate the support and cooperation of the Computer and Information Science Department.

**Funding Statement:** This research was funded by Yayasan UTP FRG (YUTP-FRG), grant number 015LC0-280 and Computer and Information Science Department of Universiti Teknologi PETRONAS.

**Author Contributions:** Methodology, Hina Bhanbhro, Yew Kwang Hooi and Worapan Kusakunniran; Software, Mohammad Nordin Bin Zakaria and Hina Bhanbhro; Supervision, Yew Kwang Hooi, Mohammad Nordin Bin Zakaria and Worapan Kusakunniran; Writing—original draft, Hina Bhanbhro; Writing—review and editing, Hina Bhanbhro and Zaira Hassan Amur. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** Data will be shared upon the request of readers.

**Ethics Approval:** This study did not include human or animal subjects. Therefore, ethical approval is not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest to report regarding the present study.

## References

[1] J. Jing, D. Zhuo, H. Zhang, Y. Liang, and M. Zheng, "Fabric defect detection using the improved YOLOv3 model," *J. Eng. Fibers Fabrics*, vol. 15, 2020. doi: 10.1177/1558925020908268.

[2] Z. Yu, H. Huang, W. Chen, Y. Su, Y. Liu and X. Wang, "YOLO-FaceV2: A scale and occlusion aware face detector," 2019, *arXiv:2208.0*.

[3] L. Du, R. Zhang, and X. Wang, "Overview of two-stage object detection algorithms," in *5th Int. Conf. Intell. Comput. Signal Process. (ICSP)*, Suzhou, China, 20–22 Mar., 2020, vol. 1544, no. 1, 2020.

[4] D. Xu and Y. Wu, "MRFF-YOLO: A multi-receptive fields fusion network for remote sensing target detection," *Remote Sens.*, vol. 12, no. 19, 2020, Art. no. 3118. doi: 10.3390/rs12193118.

[5] H. Bhanbhro, Y. K. Hooi, and Z. Hassan, "Modern approaches towards object detection of complex engineering drawings," in *IEEE Int. Conf. Digital Transform. Intell. (ICDI)*, Kuching, Sarawak, Malaysia, Dec. 2022, pp. 1–6.

[6] H. Qiu *et al.*, "Hierarchical context features embedding for object detection," *IEEE Trans. Multimed.*, vol. 22, no. 12, pp. 3039–3050, 2020. doi: 10.1109/TMM.2020.2971175.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Analysis Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, 2015. doi: 10.1109/TPAMI.2015.2389824.

[8] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 1440–1448.

[9] S. Ren *et al.*, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Analysis Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2016. doi: 10.1109/TPAMI.2016.2577031.

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Visi.*, Venice, Italy, Oct. 2017, pp. 2961–2969.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788.

[12] D. Biswas *et al.*, "An automatic traffic density estimation using Single Shot Detection (SSD) and MobileNet-SSD," *Phys Chem. Earth A/B/C*, vol. 110, pp. 176–184, 2019. doi: 10.1016/j.pce.2018.12.001.

[13] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, Jun. 2020, pp. 10781–10790.
[14] H. Bhanbhro, Y. Kwang Hooi, W. Kusakunniran, and Z. H. Amur, "A symbol recognition system for single-line diagrams developed using a deep-learning approach," *Appl. Sci.*, vol. 13, no. 15, 2023, Art. no. 8816. doi: 10.3390/app13158816.
[15] P. Shen, X. Li, N. Yang, and A. Chen, "Lightweight YOLOv8 PCB defect detection algorithm based on triple attention," *Microelectron. Comput.*, vol. 41, no. 4, pp. 20–30, 2024.
[16] F. N. M. Zamri, T. S. Gunawan, S. H. Yusoff, A. A. Alzahrani, A. Bramantoro and M. Kartiwi, "Enhanced small drone detection using optimized YOLOv8 with attention mechanisms," *IEEE Access*, vol. 12, pp. 90629–90643, 2024.
[17] B. Yan, J. Li, Z. Yang, X. Zhang, and X. Hao, "AIE-YOLO: Auxiliary information enhanced YOLO for small object detection," *Sensors*, vol. 22, no. 21, 2022, Art. no. 8221. doi: 10.3390/s22218221.
[18] S. Woo, J. Park, J. -Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," in *Proc. European Conf. Comput. Vis.*, Munich, Germany, Sep. 2018, pp. 3–19.
[19] L. Zhao *et al.*, "YOLOv8-QR: An improved YOLOv8 model via attention mechanism for object detection of QR code defects," *Comput. Electri. Eng.*, vol. 118, 2024, Art. no. 109376. doi: 10.1016/j.compeleceng.2024.109376.
[20] M. Zhang, S. Xu, W. Song, Q. He, and Q. Wei, "Lightweight underwater object detection based on yolo v4 and multi-scale attentional feature fusion," *Remote Sens.*, vol. 13, no. 22, 2021, Art. no. 4706. doi: 10.3390/rs13224706.
[21] H. Bhanbhro, Y. K. Hooi, Z. Hassan, and N. Sohu, "Modern deep learning approaches for symbol detection in complex engineering drawings," in *IEEE Int. Conf. Digital Transform. Intell. (ICDI)*, Kuching, Dec. 2022, pp. 121–126.
[22] Y. Zhao, L. Zhao, Z. Liu, D. Hu, G. Kuang and L. Liu, "Attentional feature refinement and alignment network for aircraft detection in SAR imagery," 2022, *arXiv:2201.07124*.
[23] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, Jun. 2020, pp. 11534–11542.
[24] Y. Chen *et al.*, "Image super-resolution reconstruction based on feature map attention mechanism," *Appl. Intell.*, vol. 51, pp. 4367–4380, 2021. doi: 10.1007/s10489-020-02116-1.
[25] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 7132–7141.
[26] L. Shen, B. Lang, and Z. Song, "DS-YOLOv8-Based object detection method for remote sensing images," *IEEE Access*, vol. 11, pp. 125122–125137, 2023. doi: 10.1109/ACCESS.2023.3330844.
[27] A. Dumitriu, F. Tatui, F. Miron, R. T. Ionescu, and R. Timofte, "Rip current segmentation: A novel benchmark and YOLOv8 baseline results," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Vancouver, BC, Canada, Jun. 2023, pp. 1261–1271.
[28] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, Oct. 2017, pp. 2980–2988.
[29] M. Everingham, L. Van Gool, C. K. 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.