Alexandria Engineering Journal 129 (2025) 1326–1339

Contents lists available at ScienceDirect

Alexandria Engineering Journal

journal homepage: www.elsevier.com/locate/aej

Original Article

YOLO-ARM: An enhanced YOLOv7 framework with adaptive attention receptive module for high-precision robotic vision object detection

Fuzhi Wang *, Changlin Song

School of Mechanical Engineering, Xihua University, Chengdu 610039, China

ARTICLE INFO

Keywords:
Convolutional neural network (CNN)
Robotic vision
Object detection
Attention mechanism
YOLOV7

ABSTRACT

This study addresses the difficulties of low detection precision, poor real-time performance, and poor model generalization in robotic vision systems under adverse circumstances through the proposition of an improved object recognition scheme based on a better convolutional neural network (CNN). To address these ends, YOLOv7-improved architecture is proposed, referred to as YOLO-ARM, which employs two new modules: the Adaptive Attention Receptive Module (ARM) and the Convolutional Block Attention Module (CBAM). ARM enhances feature extraction by adjusting the dynamic receptive field and multi-scale feature fusion, whereas CBAM improves feature maps by using channel and spatial attention procedures to improve the attention of the model towards critical features. The contributions of this paper involve the combination of ARM and CBAM in YOLOv7 to enhance the capacity of the model for handling scale changes, occlusions, and clutters. ARM module leverages group convolutions, squeeze-and-excitation blocks, and depth-wise convolutions for strengthening feature discrimination, while CBAM leverages channel and spatial attention in order to boost respective features. The proposed YOLO-ARM model outperforms other models on the MS COCO dataset, with an F1-score of 98.60 %, precision of 97.997 %, and accuracy of 99.727 %.

1. Introduction

Computer vision technology simulates human visual observation and uses computers to evaluate images. To achieve intelligent image processing, the computer must be able to understand its surroundings through images and imitate the special process of human vision. Computer vision technology is a type of artificial intelligence that simulates human perception of the surroundings. As a result, the technology incorporates numerous disciplines and technologies, such as image processing [1,2]. Object detection and recognition are serious challenges in the domain of computer vision. This entails identifying a particular item in a picture and accurately assigning it to its appropriate class. Computer vision has recently gained a lot of attention due to its varied uses. Object detection is very important in autonomous driving and safety monitoring [3]. Despite the necessity of object detection and recognition, these tasks can be difficult due to occlusion, cluttering, scale fluctuation, rotation, and changes in illumination. Different methods are used to handle these difficulties, such as genetic transforms, decision trees, geometric moment invariants, and machine learning-based approaches [4,5].

Object detection can be used to a variety of data formats, including images, video, and audio files. Computer and software technologies can locate and identify specific parts within a picture or scene. Object recognition in video functions similarly to that in photographs. The computer would be able to recognize, identify, and categorize objects in the given moving images with the use of such a tool [6,7]. Object trackers can also discern items based on distinct noises. Academic and industrial research are actively focusing on the coupling of deep learning (DL) methods and visual recognition technology. DL models can handle image pixel inputs instantly from start to end, achieve powerful semantic description abilities via layered extraction of attributes, and have made groundbreaking advancements in disciplines such as categorization and object recognition [8]. Object identification with machine learning frameworks is a set of approaches for automatically recognizing and locating objects in photographs or movies. These models are trained using labeled images, with each object of interest tagged with its equivalent class. These tagged photos are then used by the model to comprehend characteristics unique to each item class [9,10].

Prior to deep learning, object detection was primarily based on hand-designed features and typical machine learning techniques. Sliding

* Corresponding author.
E-mail addresses: paoding@mail.xhu.edu.cn (F. Wang), clsong@126.com (C. Song).

https://doi.org/10.1016/j.aej.2025.09.001
Received 16 April 2025; Received in revised form 7 August 2025; Accepted 1 September 2025
Available online 5 September 2025

window technology is commonly used with feature descriptors like SIFT (Scale Invariant Feature Transform), SURF (Speed-up Robust Features), or HOG (Histogram of Oriented Gradients) [11]. Traditional object detection approaches work by generating fake feature data from a picture based on the properties of the target, which is then used for object recognition. DL-based object recognition approaches outperform previous methods and are now widely used. Deep learning approaches are distinguished by the incorporation of learnable semantic and deep-level features that can compensate for the inadequacies of classic object detection methods [12,13].

Object detection has advanced significantly as deep learning technology has evolved, particularly in algorithmic design. These algorithms are broadly divided into two types. The first category includes region-based, two-stage detection models like Fast Region-Based CNN (Fast R-CNN), Region-Based Fully CN (R-FCN), Mask Region-Based CNN (Mask R-CNN) [14,15]. Vision transformers break images into successive blocks and use a self-attention technique to recognize objects. This class includes algorithms such as Swin transformer, detection transformer (DETR), MViT, and its descendants [16,17]. Monitoring or analyzing details on small items in DL has various hurdles, mostly with three major difficulties: modest items should lack details as a feature for CNN to create a layer; the impact of an intricate background, which generates the image to be chaotic by other items; and noise and image quality, which make small identification difficult. However, the development of new You Only Look Once (YOLO) models has overcome certain hard problems in small object detection [18,19]. YOLO combines a less expensive to compute backbone network and a variety of optimization strategies to improve object recognition efficiency, making it perfect for real-time applications. Furthermore, YOLO features a simple API interface and a model that has been trained, making it a convenient and user-friendly solution [20].

The following are the major contributions of this work:

- The paper presents an enhanced variant of YOLOv7, called YOLO-ARM, that combines two new modules: the Adaptive Attention Receptive Module (ARM) and the Convolutional Block Attention Module (CBAM).
- ARM dynamically adjusts receptive fields and combines multi-scale features with grouped convolutions, squeeze-and-excitation (SE) blocks, and depthwise convolutions.
- CBAM uses a channel and spatial attention mechanism in succession to draw attention to the important features within the feature maps. The two-stage attention mechanism assists the model in focusing better on the salient details and eliminating unwanted background noise.

The organization of the paper is the following: An overview of the pertinent information is given in 2. The strategy in question is addressed in 3. The findings are summarized in 4. 5 contains the paper's conclusion.

2. Literature review

The latest developments in object detection have utilized deep learning algorithms, specifically CNNs, to solve issues such as scale variations and occlusions. The "underwater optical detection network" (UODN) was created by Zhou et al. [21] and used the YOLOv8 architecture to address problems with the "cross stage multi-branch (CSMB)" and "big kernel spatial pyramid (LKSP)" modules. The CSMB unit works to enhance image quality by extracting more information from underwater optical images, and LKSP component works to improve the recognition of objects of variable dimensions in the network. CSMB Darknet, designed by CSMB and LKSP, can be used as a basis for underwater object detection and feature extraction. During the registration procedure, Sileo et al. [22] presented a method for using an Android manipulator to carry out a Pegin-Hole operation on a carbon fiber workpiece. The semantic picture segmentation neural network was utilized for eliminating the background, which enhanced recording at faster speeds. In the second stage, valuation of the hole location on the workpiece facade was carried out.

MC-YOLOv4 is a low-cost and efficient GPR pavement illness recognition image method that uses MobileNetV2 and CBAM attention mechanisms along with a Focal loss confidence loss function for iteration, according to Liu et al. [23]. To solve the challenge of data constraint in GPR, SAGAN-W was reinvented as an unconstrained generative adversarial neural network centered on self-attention. The mother-slave robot tracking system suggested by Lim et al. [24] tracks, predicts, and identifies targets using a single image.The suggested system uses a CNN to recognize objects and identify the target robot. Linear regression analysis was used to assess the detachment and angle amid robots, providing a cost-effective and efficient solution compared to older methods. The problem of face mask identification system verification due to the lack of genuine picture datasets was resolved by Umer et al. [25]. To overcome this limitation, a new dataset referred to as RILFD is generated using real images, which are annotated using two labels: 'with mask' and 'without mask'. The research examines different DL models such as YOLOv3 and Faster R-CNN for detecting face masks.

Specifically, Lou et al. [26] developed a concise object recognition process. This study introduced three major innovations: (1) A new downsampling approach for retaining context feature information is given. The feature fusion network is improved to efficiently combine shallow and deep information. A new network topology is presented to improve the detection accuracy in the model. YOLO-SE is a new YOLOv8-based network that Wu and Dong [27] presented, focusing these interactions in creative ways. Employing a light-weight convolution (SEConv) rather than common convolutions diminishes the parameter and quickens recognition. The paper introduced the SEF module, an upgrade of SEConv, to cater to multi-scale object detection. The SPPFE module results from incorporating a creative Efficient MultiScale Attention mechanism within the network. This extension boosts the capacity of the network to acquire features so that it can effectively process multi-scale item recognition issues. A dedicated prediction head for micro item detection is also provided, with the default recognition head substituted by a transformer prediction head. Pattern matching, computer vision in the shape of digital image processing, and artificial intelligence was employed by Yadav, et al. [28]. YOLO-based Fast CNNs were discovered to be effective in discriminating between lookalike objects, enduring continuous motion, and low image resolution.

A unique method for improving the localization accuracy of detected items was created by Balamurugan et al. [29]. In this paper, they introduce a method for iteratively refining picture region suggestions to obtain ground truth values. The Faster R-CNN (FR-CNN) appeared to be a deep CNN for object recognition. It provides the user with the perception that the network is unified and connected. To relocate erroneous region recommendations, a uniform model was created using quick predictions. This method is suitable for a variety of datasets and FR-CNN architectures, as it focuses on detecting objects. Second, we apply the joint score function to various picture attributes. The joint score function shows the hidden object's position in relation to other objects. A semi-automatic approach for unsupervised object detection in surveillance videos was created by Gomaa and Abdalrazik in 2024 [30] by combining background suppression with a modified version of the detection-based CNN YOLOv4 algorithm. To begin, this technique removes moving elements using background subtraction (BS)-based low-rank decomposition. The results of BS are then validated using a clustering approach. The revised results are employed to modify the updated YOLO v4 model for object detection and classification.

A marine biological item identification architecture based on an upgraded YOLOv5 foundation was provided by Zhang et al. [31]. Initially, the framework known as "Real-Time Models for Object Detection (RTMDet)" is explained. The essential module, Cross-Stage Partial Layer (CSPLayer), has a large convolution kernel, which allows

the detection network to collect contextual data more precisely. The YOLOv7 model was modified by Li et al. [32] to improve feature conservation and reduce loss during network processing. The SPPCSPC module was created, which incorporates feature separation and merging principles. The "Coordinate Attention for Efficient Mobile Network Design" (CA) module was added to reduce missed detections and noise effect in tiny target environments. The network was improved by including a dynamic convolutional module to tackle misdetection and leaking caused by large target size fluctuations.

The "RAST-YOLO (YOLO with Regin Attention and Swin Transformer)" method was created by Jiang and Wu in 2023 [33] to solve issues with remote sensing recognition, like wide target scale fluctuations. The Regin Attention (RA) procedure, in which Swin Transformer is employed as the backbone, is recommended to enhance object detection accuracy under complex background environments by enlarging the interaction area of the feature map and exploiting background information. The C3D module enhanced small object detection accuracy through the integration of deep and thin semantic knowledge and the optimization of remote sensing objects across numerous scales. The MCS-YOLO algorithm was presented by Cao et al. [34]. The backbone incorporates a coordinate attention component to gather cross-channel and spatial data from the feature map. A multiscale small item recognition architecture was built to enrich sensitivity in the detection of dense tiny objects. The Swin Transformer architecture was employed to pretrain the CNN to emphasize on contextual spatial features.

An improved ASPP_BiFPN_YOLOv4 (ABYOLOv4) method for identifying human objects was suggested by Li et al. (2024) [35]. The "Atrous Spatial Pyramid Pooling (ASPP)" module replaced the standard SPP module, improving the network's receptive field and awareness of multi-scale targets. The primary Path Aggregation Network (PANet) multi-scale fusion component was substituted by a self-created "bi-layer bidirectional feature pyramid network" (Bi-FPN). A novel element was incorporated into the suggested scheme to reclaim mid and low-level information, possibly increasing the network's capacity to explicit the features of micro and medium-sized targets. Finally, Bi-FPN used depth-separable convolution to achieve an appropriate ratio of precision and parameter count. Table 1 depicts a review table summarizing the literature review section including authors, techniques, and its limitations.

Existing systems tend to have low detection precision, suboptimal real-time performance, and poor generalization in various challenging and diverse environments. Moreover, scale variations, occlusions, cluttered backgrounds, and varying illumination reduce the performance of current models even further. Although traditional techniques such as CNNs and YOLO models have improved, they remain incapable of striking a good stability among accuracy and computational speed, particularly in changing or resource-limited environments.

Table 1
Outline of the literature review.

Author(s)	Techniques	Limitations
Zhou et al. (2024)	YOLOv8 with CSMB and LKSP modules.	Focused on underwater environments; may not generalize to other domains.
Sileo et al. (2024)	Semantic image segmentation neural network for Peg-in-Hole tasks; CNN for hole position estimation.	Limited to specific robotic tasks (automotive parts); lacks broader applicability.
Liu et al. (2024)	MC-YOLOv4 with MobileNetV2, CBAM, and Focal Loss; SAGAN-W for data augmentation.	Primarily for GPR pavement disease detection; may not perform well on diverse object types.
Lim et al. (2024)	CNN for object recognition; linear regression for distance and angle estimation.	Relies on single-image input; may struggle with dynamic or occluded scenes.
Umer et al. (2023)	Custom CNN with four-stage image processing; YOLOv3 and Faster R-CNN for face mask detection.	Limited to face mask detection; performance on other objects not evaluated.
Lou et al. (2023)	Compact object detection algorithm with new downsampling strategy and feature fusion network.	Designed for specific scenarios; may lack scalability to general object detection tasks.
Wu and Dong (2023)	YOLO-SE with lightweight convolution (SEConv), SEF module, and SPPFE module.	Focused on remote sensing; may not generalize to real-time robotic vision.
Yadav et al. (2024)	YOLO, SSD, and Faster R-CNN	Comparative study without novel contributions; lacks performance improvements.
Balamurugan (2024)	Faster R-CNN with iterative refinement for object localization.	Computationally expensive; may not be suitable for real-time applications.
Gomaa and Abdalrazik (2024)	Modified YOLOv4	Semi-automatic approach requires manual tuning; limited to surveillance videos.
Zhang et al. (2023)	Improved YOLOv5 with Cross-Stage Partial Layer (CSPLayer) for marine biological object detection.	Specific to marine environments; may not perform well on terrestrial objects.
Li et al. (2023)	Improved YOLOv7 with SPPCSPC, Coordinate Attention (CA).	Focused on small object detection; may not address large-scale object challenges.
Jiang and Wu (2023)	RAST-YOLO	Complex architecture; may require high computational resources.
Cao et al. (2023)	MCS-YOLO	Designed for autonomous driving; may not generalize to other applications.
Li et al. (2024)	ABYOLOv4 with ASPP module, Bi-FPN, and depth-separable convolution for human object detection.	Specialized for human detection; may not perform well on other object classes.

3. Proposed methodology

The research methodology of this work presents an optimized object detection model, YOLO-ARM, derived from an enhanced YOLOv7 architecture to mitigate issues like low detection accuracy, poor real-time performance, and weak model generalization in robotic vision systems. The primary goals of this study are to increase detection accuracy, enhance real-time processing, and boost the model's generalization capability across varied and complex environments. To achieve these goals, the method integrates two novel components: the Adaptive ARM and the CBAM. ARM adaptively updates receptive fields and integrates multi-scale features to facilitate improved feature extraction, while CBAM applies channel and spatial attention mechanisms to improve feature maps by emphasizing important features. The model further employs advanced data preprocessing and augmentation techniques such as Gaussian filtering, minmax normalization, and data transformation like flipping, rotation, and zooming to enhance its robustness. The approach seeks to deliver a lightweight, efficient solution that is applicable in real-time scenarios like autonomous vehicles and surveillance.

3.1. Data preprocessing

The process begins by applying a Gaussian filter to the input images to reduce and eliminate noise, thereby ensuring that the input data quality is enhanced and the object detection model becomes better. The second is minmax normalization, where the pixel values of the image are standardized to a value within the range of 0–1. This normalization is important in order to have the input data in an appropriate format for the deep learning model and also in order to avoid bias on the part of the model towards certain pixel values. The sample outcome of data preprocessing is shown in the Fig. 1.

Data augmentation is used to enhance the variety of the training set by producing transformed versions of the original images. This makes

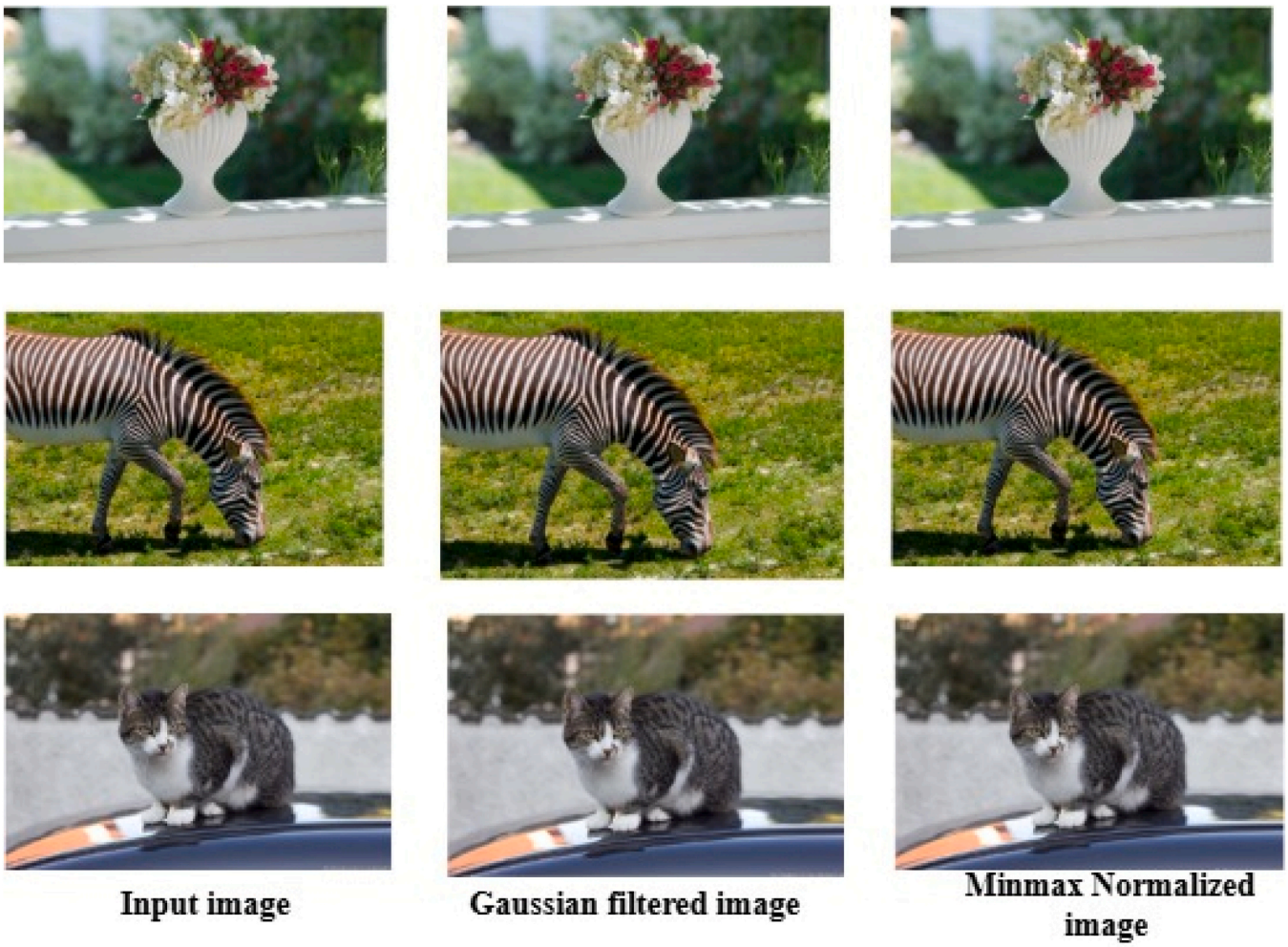

Fig. 1. Sample data pre-processing outcome.

the model generalize more to novel data and enhance its performance. Six augmentation methods: Horizontal Flipping, Vertical Flipping, Flipping both (horizontal and vertical), Rotation, Zoomed, and Color Jittered. All of these methods impose a particular kind of variation on the training data, compelling the model to acquire features that remain invariant to these transformations. The sample outcome of data pre-processing is shown in the Fig. 2.

Horizontal Flipping: This method generates a mirror reflection of the original image along the vertical axis. For object detection, this ensures that the model can distinguish objects irrespective of their left-

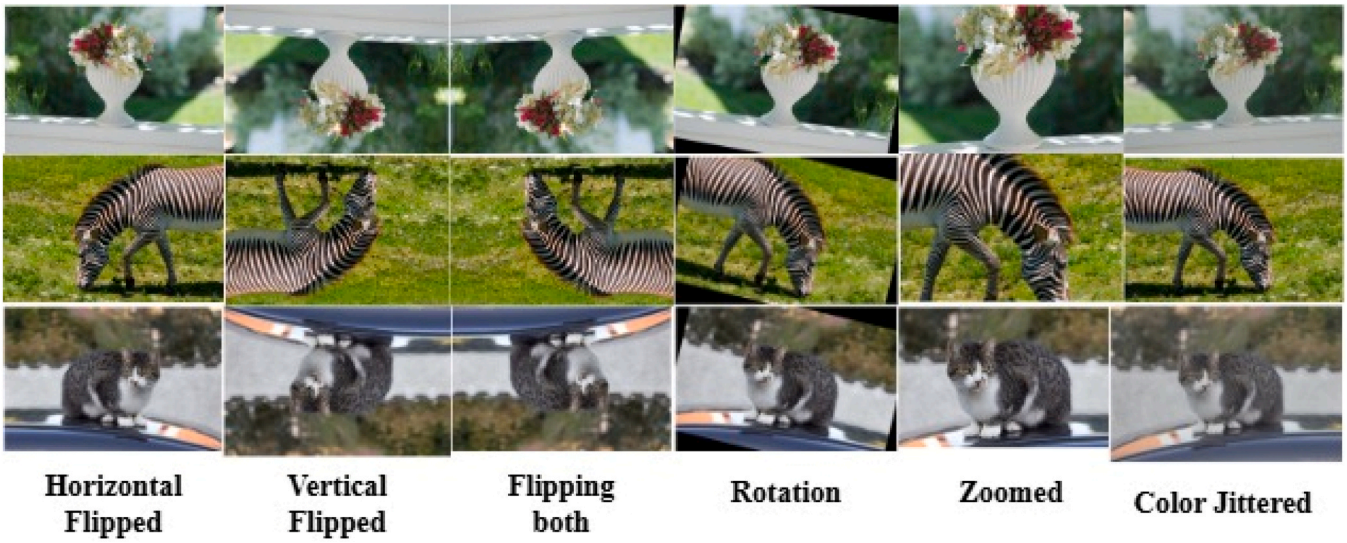

Fig. 2. Sample data pre-processing outcome.

right position. For example, a left-facing zebra should be detected as accurately as a right-facing zebra.

Vertical Flipping: Likewise, vertical flipping produces a mirror image across the horizontal axis. Less frequent for natural objects that possess an oriented upright position (such as the zebra or the flower pot), it can be useful for object detection that may appear upside down or in unexpected orientations.

Both Flipping: This applies both horizontal and vertical flipping, further enhancing variability in the training data. The model learns to recognize items irrespective of orientation along both axes.

Rotation: This method rotates the image by some angle. By incorporating rotated copies of the objects in the training set, the model is made stronger to variations in the orientation of the object in the image. This is especially critical in robotic vision systems where the camera angle or the pose of the object may change drastically.

Zoomed: Zooming in and out on various regions of the image exposes the model to objects at different scales. This comes directly into application with the difficulties highlighted in the research paper, including handling scales variations. With training on zoomed-in and zoomed-out images, the YOLO-ARM model improves at locating objects of different sizes in an image.

Color Jittered: This method randomly changes the color attributes of the image, including brightness, contrast, saturation, and hue. This decreases the model's compassion to lighting conditions and color casts variations, which are typical in real-world robotic vision applications and other difficult situations.

These data augmentation methods are essential in enhancing the generalization capacity of the model. By presenting the model with a larger variety of variations in the input data, the model is better equipped to withstand challenges like scale variations (tackled by zooming), occlusion (flipping and rotation partially simulate this), and cluttered backgrounds (the model learns to ignore background variations created by these augmentations).

3.2. YOLOV7- ARM architecture

YOLOv7 has a similar algorithm paradigm to its predecessor, YOLOv5. This is basically separated into four units: input, backbone, neck, and head. The Head layer proceeds to analyze the Backbone network's result image, finishing the feature map fusion method by Adaptive Attention Receptive Module (ARM), the up-sampling procedure by UPSample, and the feature map extraction of three layers of varying sizes by ELAN variation ELAN-H. Feature map is combined with two CBS components to finish feature amalgamation from Backbone network, as well as features from CBAM's dual attention approach. After REP and CBM, predict the image recognition activities and report the results.

The Backbone framework supports signature acquisition in YOLOv7. It helps extract features for target detection, resulting in feature layers. The retrieved feature layers from the Backbone play a crucial position in network building, earning them the name "effective feature layers". YOLOv7's Backbone feature extraction network uses the E-ELAN module, which has four branches for final stacking. This arrangement creates a dense residual structure from many stacks, simplifying optimization and improving accuracy through improved network depth. Effective feature layers are combined in the YOLOv7 Neck module. This module is a significant improvement to YOLOv7's layer-extraction network, specifically designed to tackle the issues of deep learning with different target sizes. Furthermore, it successfully addresses image noise.

The YOLOV7 model is modified in this architecture is the incorporation of Adaptive Attention Receptive Module (ARM) in the neck region integrating the backbone and neck and also the addition of CBAM. After the CBS, ELAN-W_1 and ARM block in the neck region, the CBAM module is added in the block to acquire the refined features.

In the proposed YOLO-ARM model, ARM is embedded in the backbone of YOLOv7, exactly after the CSPDarknet feature aggregation block. Its function is to dynamically adjust the receptive field according to the spatial context and, through it, recalibrate feature activations in terms of a mixture of grouped convolutions and dilation-based spatial encoders. Such an embedding facilitates early-stage enhancement of spatial semantics without hindering model efficiency.On the other hand, CBAM is incorporated within the neck segment, right after the PANet path aggregation block. CBAM utilizes sequential channel and spatial attention mechanisms that refine intermediate feature maps through selective boosting of informative areas while suppressing noise or irrelevant background. This attention-based refinement is done before the last detection heads, thereby ensuring the utilization of most semantically dense and spatially localized features for object classification and localization.

The augmented image data is then given to the YOLOV7 network. The model's backbone region includes four Convolutional, Batch Normalization and Sigmoid Linear Activation function forms (CBS) modules (CBS_{1-4}), four ELAN modules, and three MP modules, which extract features and provide a feature map. The CBS modules, which serve as the foundation for ELAN, MP, and ELAN-W, ARM in the network architecture which is the enhancement in this network, are constructed using the SiLU activation function, 2D batch normalization, and a typical layer of convolution with one kernel, "same" padding, and a stride of 1. The backbone employs a series of CBS modules, each comprising a convolutional layer, batch normalization, and a SiLU activation function is mathematically represented as in Eq. (1),

$$CBS(x) = SiLU(BN(W * x + b)) \tag{1}$$

where W and b are the kernel weights and bias, * represents convolution, and SiLU is the Sigmoid Linear Unit.

$$SiLU(x) = x.\sigma(x) \tag{2}$$

The backbone's first four CBS modules are connected in series, and the first four ELAN ($ELAN_1$) modules receive the output tensor from the last and fourth CBS module (CBS4). The ELAN module improves feature extraction by collecting outputs from multiple branches of convolution, enhancing gradient flow and feature diversity. Seven CBS modules plus an output concatenated unit make up the ELAN module, which is YOLOv7's primary feature extractor. In the ELAN module, the first two CBS modules, CBS_{1ELAN} and CBS_{2ELAN}, receive input tensors from the preceding layer. Four CBS modules ($CBS_{3-6ELAN}$) connected in sequence receive the output tensor from the CBS_{1ELAN} module. The output from CBS_{6ELAN}, along with the outputs of CBS_{1ELAN}, CBS_{2ELAN}, and CBS_{4ELAN}, is then concatenated and sent to CBS_{7ELAN}. The Fig. 3 illustrates the modified YOLOv7 framework integrating the ARM and CBAM.

Three CBS components plus a Max Pooling Unit make up the MP1 module. CBS7ELAN provides input to the MP and the first CBS unit (CBS1MP1). CBS_2MP_1 receives the output of the MP unit, CBS1MP1 receives the output of CBS3MP1, and the outputs of CBS_1, $3MP_1$ are merged. $MP1_1$, $MP1_2$, and $MP1_3$ are associated by the backbone in accordance with $ELAN_1$, $ELAN_2$, and $ELAN_3$, correspondingly. The neck attaches the head and backbone by concatenating feature maps that the head receives from the backbone. The head's function is to turnout the final recognitions and bounding boxes. Since it concatenates the output from all six CBS components, the ELAN module in the backbone is not the same as the ELAN-W component in the head, which only has four. Two of the four standalone CBS units feature shortcut links from ELANs 2 and 3. The network can reuse attributes from shallower areas to deeper ones thanks to shortcut connections. The two MP2 modules are structurally identical to the MP1 modules, except that the ARM module provides additional input to the combination block in $MP2_2$, while ELAN-W1 provides additional input to the concatenated block in MP21. The output tensors of ELAN-W are passed into RepConv for final detection. The Figs. 4 and 5 depicts the detailed structure of the CBS (Convolutional + Batch Normalization + SiLU), ELAN, ELAN-W, and MP (Max Pooling) modules, showcasing their interconnected layers and

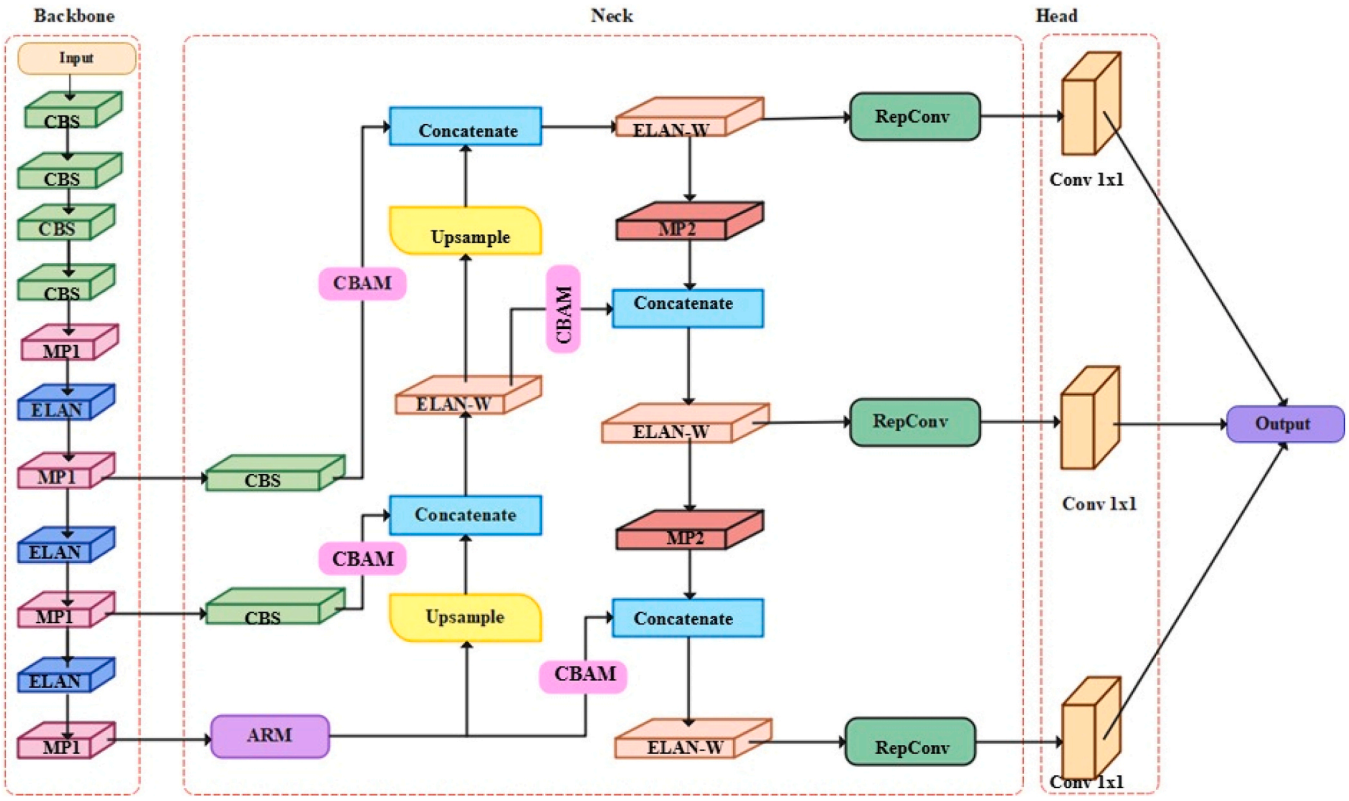

Fig. 3. The proposed YOLO-ARM model architecture.

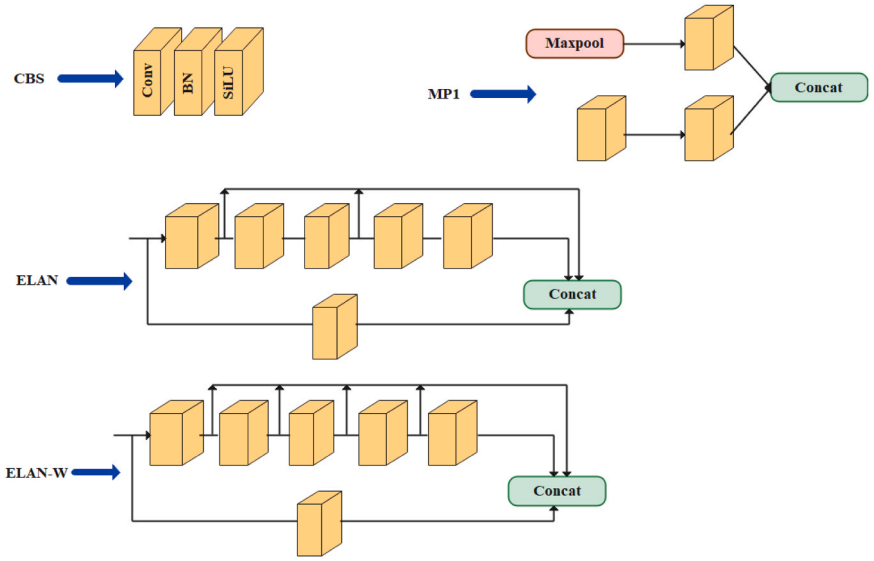

Fig. 4. The structure of CBS, ELAN, ELAN-W and MP.

feature processing pathways.

3.3. Adaptive attention receptive module

The Attention Receptive Module (ARM) is a highly evolved neural network component that is geared towards improving the feature extraction functionality of CNNs through the employment of attention mechanisms and effective receptive field adjustments. The foundation of YOLO-ARM lies in the Adaptive ARM, which fundamentally transforms feature extraction by adjusting receptive fields adaptively and incorporating multi-scale features. ARM relies on a synergy of grouped convolutions, SE blocks, and depth-wise convolutions to deepen feature discrimination. This design has special suitability to tasks involving detailed feature discrimination such as object detection, segmentation, and robotic vision. ARM refines feature maps via a sequence of operations, including 1×1 grouped convolutions for dimensionality reduction, squeeze-and-excitation (SE) blocks for channel attention, depthwise convolutions (DW-Conv) for extracting spatial features, and feature concatenation for fusing multi-scale features. The architecture 5 describes the ARM's structure, featuring grouped convolutions, SE blocks, and DW-Conv that adaptively manipulate receptive fields and enhance feature maps for better object detection.

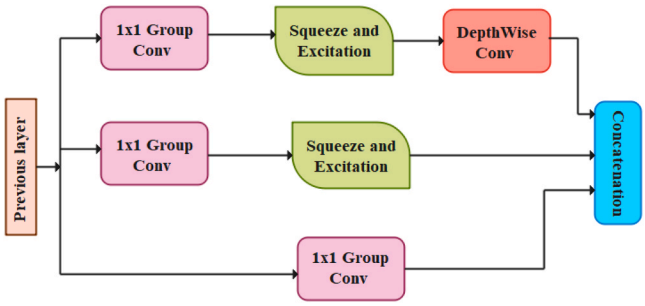

Fig. 5. The proposed architecture of Adaptive Attention Receptive Module.

The procedure begins with a 1× 1 grouped convolution, which reduces channel sizes without eliminating spatial structure, minimizing computational costs. Grouped Convolution (GConv) Lowers channel dimensions without altering spatial structure. For an input feature map $F \in R^{C\times H\times W}$, the grouped convolution divides F into G groups and performs individual convolutions on each:

$$F_{grouped} = \bigcup_{i=1}^{G} W_i * F_i \tag{3}$$

Second, the SE block dynamically remaps feature responses through global average pooling to squash out spatial details and then employ a multi-layer perceptron (MLP) and sigmoid activation to turn on important channels. This improves the model to be more sensitive to important features. The SE attention model adds a "squeeze and excitation" module to boost its expressiveness. The model learns a weight per channel in the base model and reweighs its importance in later layers to focus on important features. SE attention mechanism enhances the expressiveness and performance of the model. It is easy to train and deploy with the few parameters but might be more complex for the model with the high computational overhead of calculating the channel weights. SE block reshapes channel-wise feature responses via global average pooling and MLP

$$z_c = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W} F_c(i,j) \tag{4}$$

where z_c is the squeezed feature. Next, the processed feature map is processed by a DW-Conv, actually learning spatial features without significantly impacting the number of parameters. DW-Conv applies spatial filtering independently to each channel:

$$F_{dw} = \sum_{k=1}^{K} W_k * F_k \tag{5}$$

This process expands the receptive field, making the model better capture objects of various scales. The feature map is then processed by yet another 1× 1 grouped convolution for restoring the channel dimensions, followed by another second SE block for refining further. Finally, the module employs feature concatenation to integrate multi-scale information so that high-level semantics as well as detailed information are preserved for downstream processes. Mathematically, ARM's hierarchical processing can be defined as a sequence of transformations:

$$F_{out} = SE(GConv_{1\times1}(DWConv(SE(GConv_{1\times1}(F))))) \tag{6}$$

This structured architecture enhances feature discriminability, particularly in challenging situations such as occlusions, cluttered scenes, or finding small objects. By adding grouped convolutions for efficiency, SE blocks for channel-wise attention, and DW-Conv for spatial flexibility, ARM achieves a trade-off between performance and computational efficiency. Its light weight renders it particularly well-suited for uses like robotic vision, where speed and accuracy are both necessary. The module's ability to dynamically change the receptive fields and highlight salient features ensures robust performance in the most diverse and demanding environments. A novelty of YOLO-ARM is its strong yet lightweight nature, which has been achieved through the effective use of grouped convolutions and depth-wise operations on ARM, and the parameter-sharing MLP in CBAM. These are innovations that minimize computational costs without compromising performance, and this makes the model deployable on resource-limited environments.

3.4. CBAM

The counterpart of ARM is the CBAM, which sequentially conducts channel and spatial attention mechanisms to refine feature maps. CBAM's channel attention module (CAM) produces a channel attention map by aggregating global spatial information by using average and max pooling. The channel attention module (CAM) and spatial attention module (SAM), the two stand-alone sub-modules of the CBAM, can conserve computational power and parameters through the utilization of attention mechanisms over the channel and space, respectively. Adaptive feature refinement was achieved by multiplying the attention maps by the input feature maps after CBAM inferred each attention map separately along two independent properties. The following Eqs. (7) and (8) explains the way that CBAM processes the input feature map F:

$$F' = M_c(F) \otimes F \tag{7}$$

$$F'' = M_s(F') \otimes F' \tag{8}$$

where F' signifies the outcome of multiplying the feature map by the channel attention map, $\otimes$ indicates element multiplication, and F'' represents the refined output in its final form. The Fig. 6 shows CBAM's dual-channel and spatial attention mechanisms, where feature maps are processed in a sequential manner using spatial and channel attention sub-modules to emphasize critical features.

4. Channel attention module

To mitigate data loss while performing feature extraction, the channel attention module simplifies feature maps on a spatial scale utilizing the global maximum and global average pooling layers. The global maximum pooling layer gets the feature variance details, and the global average pooling layer acquires the complete value. Together, these two layers outperform all others. To create our channel attention map, $M_c \in \mathbb{R}^{C\times1\times1}$, the compressed F^C_{avg} and $F^C_{\max}$ descriptors are subsequently sent to a shared network. The shared network is made up of a multi-layer perceptron (MLP) and a hidden layer. Each descriptor receives the shared network, and the resulting feature vectors are subsequently aggregated via element summing. The channel attention is determined as follows in Eqs. (9) and (10),

$$M_c(F) = \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \tag{9}$$

$$M_c(F) = \sigma(W_1(W_0(F^C_{avg})) + W_1(W_0(F^C_{\max}))) \tag{10}$$

The sigmoid function is denoted by σ, $W_1 \in \mathbb{R}^{C\times C/r}$, and $W_0 \in \mathbb{R}^{C/r\times C}$. The MLP weights W_0 and W_1 are shared by the two inputs.

5. Spatial attention module

This module utilises the feature map F' output from Convolution block of DarkNet as its input feature map. To create two $H\times W\times 1$ feature maps, max pooling and]average pooling can be done based on channels. These two feature maps can then be subjected to a concat process, also known as channel splicing. It is condensed to a single channel ($H\times W\times 1$) following a 7×7 convolution. The sigmoid function then generates the spatial attention attribute. This feature is

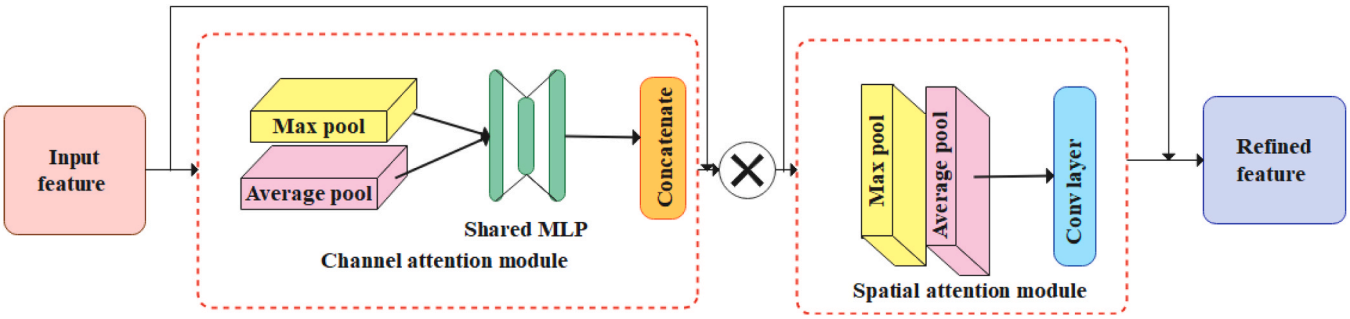

Fig. 6. The architecture of Convolutional Block Attention Module.

ultimately multiplied by the input feature of the module to produce the final generated feature. More precisely, Eq. (12) provides the computing process.

$$M_s(F) = \sigma(f^{7\times7}([AvgPool(F); MaxPool(F)])) \tag{11}$$

$$M_s(F) = \sigma(f^{7\times7}\left[F^s_{avg}; F^s_{\max}\right]) \tag{12}$$

where $f^{7\times7}$ signifies a convolution operation with a 7×7 filter size, and σ is a sigmoid function. CAM generates a channel attention map based on the inter-channel correlations of the characteristics. As every channel in a feature map is calculated as a detector of feature, each channel's focus emphases on "what" is essential given an input. Spatial attention maps are produced by SAM through the use of spatial associations between features. Spatial attention is complementary to CAM because it concentrates on the "where," as opposed to CAM. The CAM and SAM modules are used to refine the feature map F' into F''. The last feature retrieved from the network is F''. The model is trained with a compound loss function consisting of localization loss ($\mathscr{L}oss_{loc}$), confidence loss ($\mathscr{L}oss_{con}$), and classification loss ($\mathscr{L}oss_{cla}$).

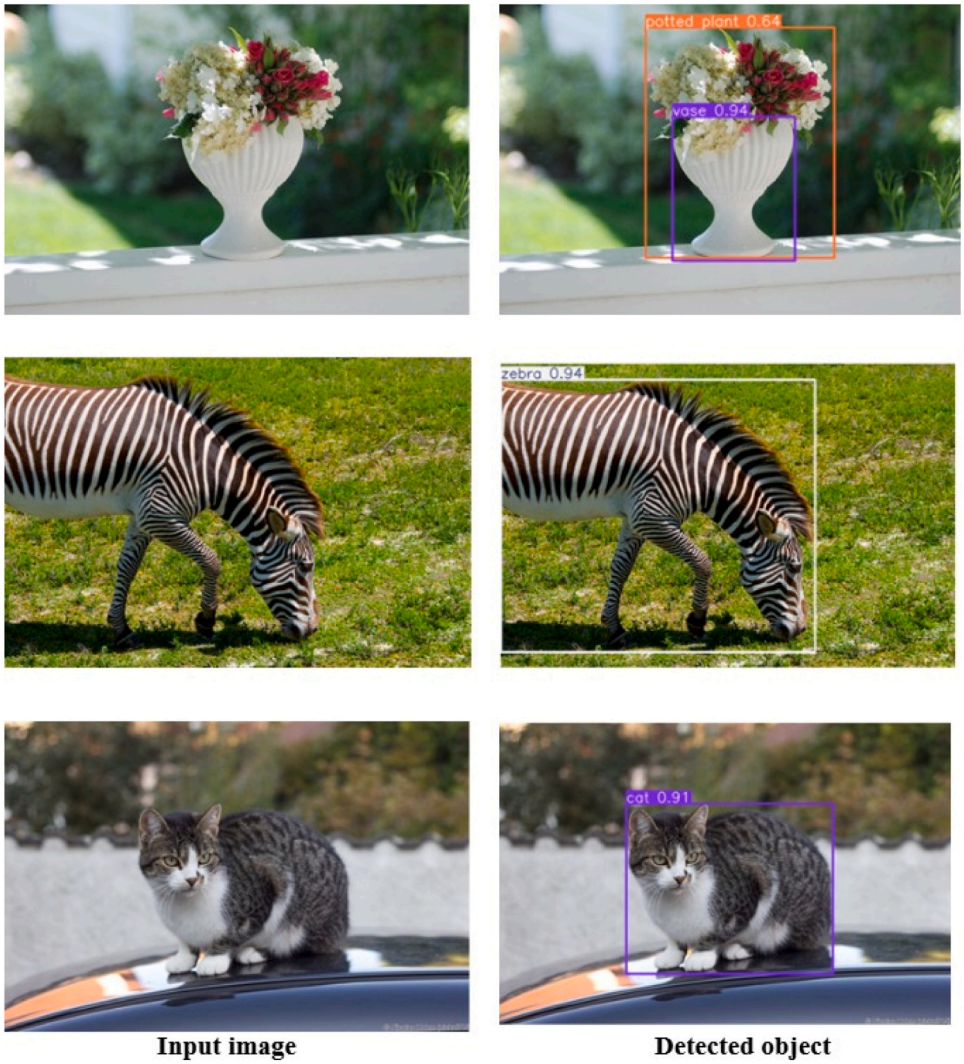

Fig. 7. Sample of outcome of input and object detection.

$$\mathscr{L}oss = \lambda_1 \mathscr{L}oss_{loc} + \lambda_2 \mathscr{L}oss_{con} + \lambda_3 \mathscr{L}oss_{cla} \quad (13)$$

where λi are balance weights. Localization loss is done with IoU (Intersection over Union) for bounding box regression

$$\mathscr{L}oss_{loc} = 1 - IoU(B_{pred}, \quad B_{gt}) \quad (14)$$

and the confidence loss utilizes focal loss to address class imbalance

$$\mathscr{L}oss_{con} = -\alpha_t (1 - p_t)^{\gamma} \log(p_t) \quad (15)$$

Where p_t is the estimated probability, α_t weights classes proportionally, and γ emphasizes hard examples. The sample outcome of object detection is shown in the Fig. 7.

6. Result and discussion

Python is used to implement the proposed model. Performance criteria such as precision, accuracy, specificity, sensitivity, F-measure, negative predictive value (NPV), false positive ratio (FPR), and false negative ratio (FNR) are assessed for the proposed model. The proposed approach is assessed using the following performance metrics:

Accuracy: Accuracy is defined as the proportion of correctly identified samples compared to all samples.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (16)$$

Precision: The precision measure stands for the ratio of properly expected positive observations to all anticipated positive observations. The model's capacity to identify negative samples as positive is assessed.

$$Precision = \frac{TP}{TP + FP} \quad (17)$$

Sensitivity: Sensitivity is the ratio of actual positive samples that were correctly anticipated to be positive.

$$Sensitivity = \frac{TP}{TP + TN} \quad (18)$$

Specificity: Specificity is the ratio of actual negative instances that were precisely estimated to be negative. It assesses the model's capability to identify each negative sample.

$$Specificity = \frac{TN}{TN + FP} \quad (19)$$

F-Measure: The recall and precision harmonic mean, which equates the two measures into a single score. It displays the model's exactness in capturing both positive and negative inputs.

$$F - Measure = \frac{2 * precision * recall}{Precision + recall} \quad (20)$$

FPR: The percentage of real negative instances that were mistakenly identified as positive is displayed by the FPR. It evaluates the model's propensity to classify negative samples as positive.

$$FPR = \frac{FP}{FP + TN} \quad (21)$$

FNR: FNR is the percentage of true positive instances that were mistakenly classified as negative. It evaluates the model's propensity to label positive data as negative.

$$FNR = \frac{FN}{TP + FN} \quad (22)$$

MCC: True and false positive and negative data are combined into a single number by the MCC, which runs on a scale from −1 to +1.

$$MCC = \frac{((TP * TN) - (FP * FN))}{\sqrt{((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}} \quad (23)$$

NPV: Out of all the samples that were anticipated to be negative during the analysis, NPV determines the ratio of true negative samples that were properly recognized as such.

$$NPV = \frac{TN}{TN + FN} \quad (24)$$

The numbers of mistakenly negatively categorized samples are symbolized by FP (False Positives), mistakenly categorized positively classified instances by FN (False Negatives), properly categorized positively classified samples by TP (True Positives), and properly categorized negatively classified instances by TN (True Negatives).

6.1. Dataset description

The proposed YOLO-ARM model was verified on the MS COCO (Microsoft Common Objects in Context) dataset, a widely used object detection benchmark found on Kaggle (https://www.kaggle.com/datasets/hariwh0/ms-coco-dataset). The dataset has 328,000 images categorized into 91 object labels and includes object detection, segmentation, and captioning annotations. For detection tasks alone, it contains 2.5 million labeled instances with varied object scales, intricate scenes, and difficult occlusions. The validation set of the dataset (2017 version) was employed for evaluation, comprising 5000 images with 36,781 bounding box annotations over 80 categories. This diverse dataset is especially useful for model robustness testing because it contains: (1) scale variation (32 % of objects take up <1 % of image space); (2) crowding (mean 7.7 instances per image); and (3) real-world complexity (indoor/outdoor scenes, changing lighting/occlusions).

6.2. Overall performance analysis

The YOLO-ARM model proposed was comprehensively tested on the MS COCO dataset, a benchmarking dataset that is well known for its variability in object sizes, occlusions, and intricate scenes. The performance of the model was compared with state-of-the-art object detection models such as CNNs, R-CNN, YOLOv5, and FPN using conventional metrics like accuracy, precision, F1-score, specificity, sensitivity, and error rates. The findings prove the superiority of YOLO-ARM in detecting high accuracy while ensuring real-time efficiency. The Fig. 8 illustrates a comparison of the accuracy of the proposed YOLO-ARM model with a number of popular object detection frameworks, namely CNN, R-CNN, YOLOv5, and FPN.

The proposed model has an impressive accuracy of 99.727 %, far

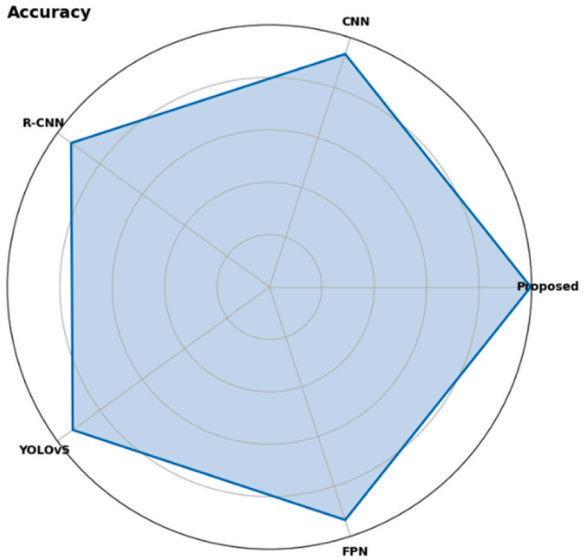

Fig. 8. Accuracy comparison of proposed and existing model.

exceeding all others. On the other hand, classical CNNs of 93.51 %, R-CNN of 93.53 %, and FPN of 93.52 % show similar but relatively lower performance with negligible differences among them. Most notably, 92.83 % YOLOv5 trails slightly behind these three because it makes a compromise between speed and accuracy in its standard settings. The accuracy of the proposed model at 99.727 % reflects the efficiency of its new elements, including the ARM and light feature recalibration, in improving feature discrimination without compromising computational power. The 4.2–6.9 percentage point advance over standard CNNs and R-CNN-based techniques demonstrates the shortcomings of traditional methods in addressing intricate situations such as occlusions, scale change, or crowded backgrounds. Even when compared to YOLOv5, the proposed architecture reflects a 6.9 percentage point improvement, confirming its optimizations in attention mechanisms. The provided Fig. 9 illustrates the precision metrics exceptional capability of the proposed YOLO-ARM model to reduce false positives over other top object detection architectures.

The proposed model attains a precision of 97.997 % and outperforms conventional CNNs of 91.70 % by 6.3 percentage points significantly, illustrating how its ARM and dynamic multi-scale fusion minimize misclassifications. Though R-CNN of 93.77 %, YOLOv5 of 94.08 %, and FPN of 93.73 % are as precise as each other mostly because they are region-based or multi-scale models, they are still behind the proposed model by 3.9–4.3 percentage points. Most importantly, YOLOv5's marginally higher accuracy (94.08 %) than R-CNN and FPN indicates its compromise between performance and efficiency, but the precision of the proposed model at 97.997 % indicates a pivotal improvement. For example, the 6.3-point difference from CNNs highlights how conventional convolutional methods are challenged by complicated scenes where contextual reasoning must take place.

The F1-score outcomes reflect the better trade-off between precision and recall of the proposed YOLO-ARM model over other models. With a remarkable F1-score of 98.60 %, the proposed model clearly outperforms all others significantly, reflecting its high ability to tackle both false positives and negatives. The baseline CNNs reach 94.35 %, while R-CNN of 94.62 % and FPN of 95.29 % reflect incremental improvements owing to their more advanced architectures. Interestingly, YOLOv5 at 93.51 %, which is indicative of its intrinsic speed versus detection quality trade-off. The proposed model's 4.25–5.09 percentage point advantage in F1-score over other methods highlights the effectiveness of its novel components.

The Fig. 10 illustrates specificity of the proposed YOLO-ARM model's superior capacity to accurately identify negative instances, with a highly impressive specificity of 99.02 %. This is a wide 4.9–5.2 percentage point higher than rivaling architectures, such as CNNs of 93.87 %, R-CNN of 94.70 %, YOLOv5 of 93.83 %, and FPN of 94.62 %. Such high-performance points to the strength of the model in reducing false positives, especially in dense, cluttered settings where background noise could otherwise result in spurious detections. The specificity gains are especially notable when compared to YOLOv5, where the 5.2 percentage point difference underscores the limitations of conventional single-stage detectors in background suppression.

The Fig. 10 illustrates the sensitivity metrics that indicate the superior capability of the proposed YOLO-ARM model to identify true positive instances with an impressive sensitivity of 98.31 %. This is a 3.5–4.5 percentage point gain over the rival models, such as CNNs of 94.76 %, R-CNN of 94.81 %, YOLOv5 of 93.83 %, and FPN of 95.36 %. These findings indicate the model's remarkable ability to reduce false negatives, especially for difficult cases such as small, occluded, or partially occluded objects. The 4.48 percentage point margin above YOLOv5 is especially striking, showing the degree to which legacy single-stage detectors compromise on sensitivity in the pursuit of speed. The 3.5 percentage point improvement over FPN, a multi-scale architecture by design, is indicative of the dynamic feature fusion approach's dominance.

The Fig. 11 depicts the MCC values clearly demonstrate the better overall detection performance of the suggested YOLO-ARM model with an impressive MCC value of 98.48 %. This is an improvement of 2.7–4.6 percentage points better than alternative architectures, such as CNNs at 95.36 %, R-CNN at 95.78 %, YOLOv5 at 93.83 %, and FPN at 94.93 %. These results show that the model achieves impressive balanced performance over all categories of classification (true/false positives/negatives) in highly adverse where typical class imbalances or adverse conditions for detection abound. The 4.65 percentage point gain over YOLOv5 is especially striking, illustrating how classical single-stage detectors falter with well-balanced performance in spite of their speed benefits. The 2.7 percentage point margin over R-CNN, otherwise a high-accuracy two-stage detector, further exemplifies the strength of the proposed model to best the inherent limitations in region-based methods. The 3.5 percentage point gains over all comparison models indicate the architectural advancements create universal improvements independent of baseline methodology.

The Fig. 11 depicts the NPV metrics that emphasize the superior performance of the suggested YOLO-ARM model in identifying negative instances accurately with an outstanding NPV of 99.33 %. The proposed YOLO-ARM model performs significantly better than CNNs, whose NPV

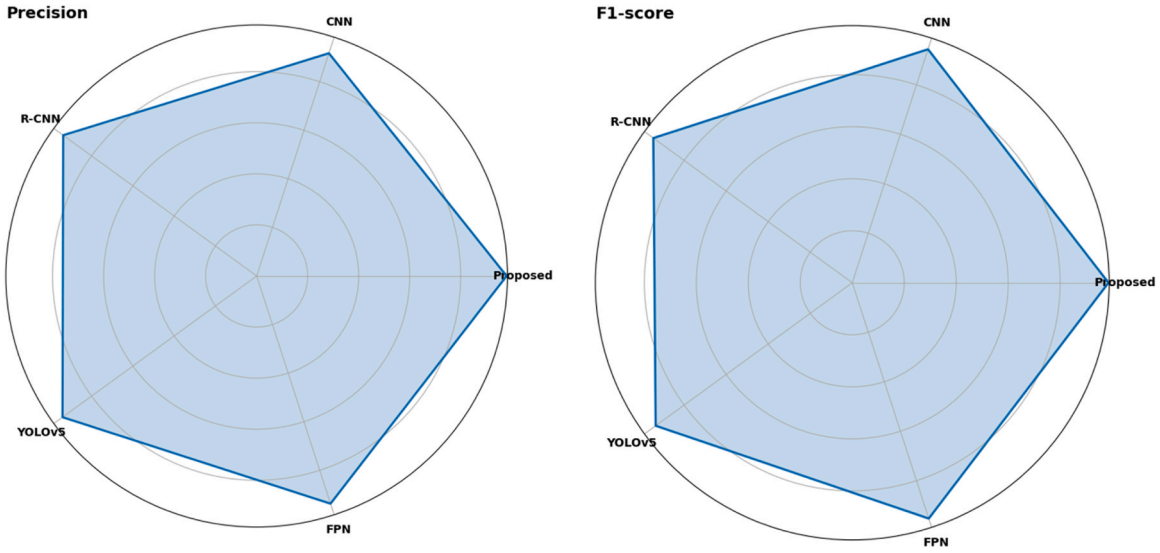

Fig. 9. Precision and F1-score comparison of proposed and existing model.

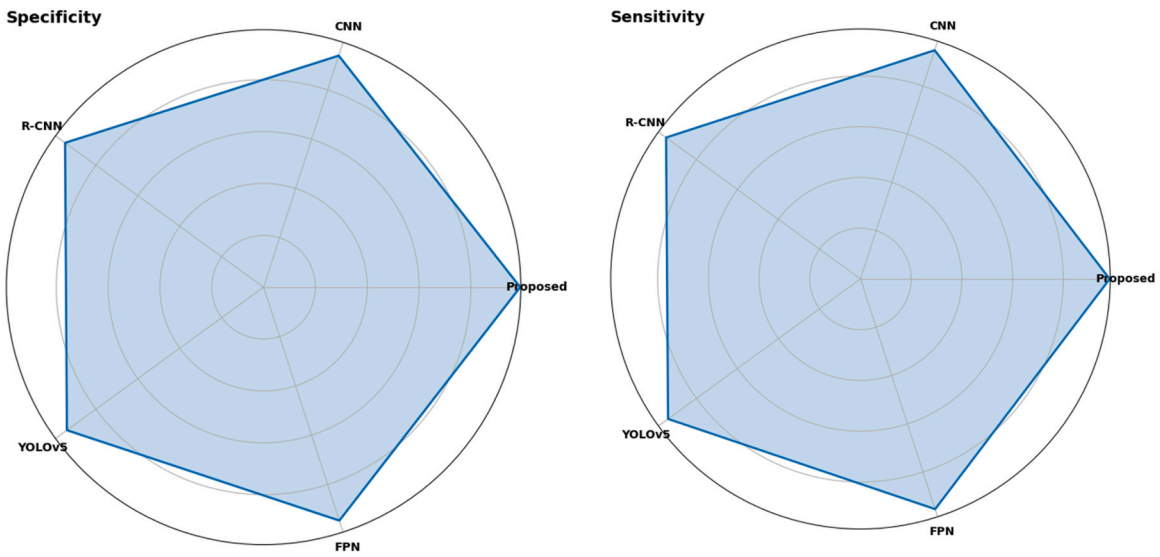

Fig. 10. Specificity and sensitivity comparison of proposed and existing model.

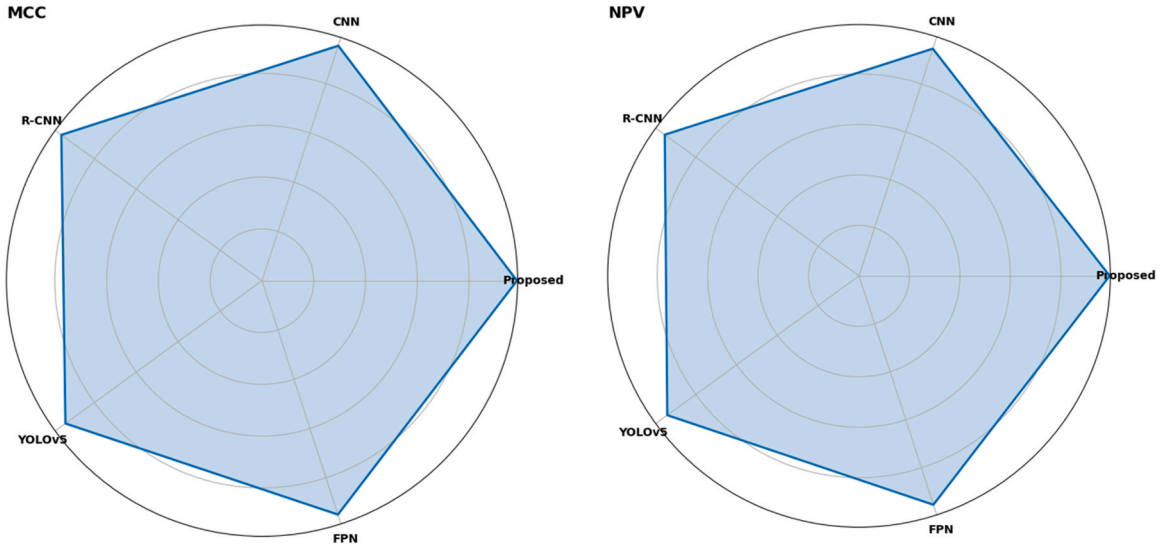

Fig. 11. MCC and NPV comparison of proposed and existing model.

is 94.66 %, with a remarkable lead of 4.68 percentage points. This reflects that the YOLO-ARM model performs much better than conventional CNN structures in identifying negative cases correctly. In comparison with R-CNN, with NPV of 95.18 %, the proposed model improves by 4.15 percentage points. YOLOv5 achieves an NPV of 93.94 %, the lowest of the existing models. The NPV of the proposed model is 5.40 percentage points greater than that of YOLOv5, indicating a significant advantage in identifying correctly the absence of objects. This emphasizes that although YOLOv5 is more focused on speed, it is behind the proposed model in identifying correctly negative instances. FPN has an NPV of 95.37 %. The model proposed performs better than FPN by 3.97 percentage points, suggesting that its architecture is better suited to reducing false negatives.

The Fig. 12 represents the FPR results that show the efficiency of the proposed YOLO-ARM model in reducing false alarms, registering an extremely low FPR of 0.00851. This shows that the model is very precise in identifying negative examples correctly and hardly misclassifies them as positive. The YOLO-ARM model significantly outperforms CNNs with an FPR of 0.059 by a huge margin of around 5.06 percentage points. This means that the YOLO-ARM model makes much fewer false positive detections than standard CNNs. In comparison with R-CNN, with an FPR of 0.063, the proposed model is improved by around 5.45 percentage points. YOLOv5 has the largest FPR of 0.069 among the models being compared. The FPR of the proposed model is around 6.09 percentage points less than that of YOLOv5, indicating a significant advantage in reducing false alarms. FPN's FPR is 0.066. The proposed model performs better than FPN by around 5.77 percentage points, meaning that it performs better at avoiding false positive errors.

The Fig. 12 illustrates the FNR values of the proposed YOLO-ARM model's excellent ability to minimize missed detection, having a low FNR of 0.0077. The proposed YOLO-ARM model performs better than CNNs with an FNR of 0.0488 by about 4.12 percentage points. This proves that the YOLO-ARM model has a better ability to minimize the frequency of missed detection than conventional CNNs. Relative to R-CNN, with an FNR of 0.0485, the model presented here demonstrates an improvement of about 4.09 percentage points. YOLOv5 has the largest FNR of 0.067 among the models being compared. The FNR of the proposed model is about 5.94 percentage points lower than that of YOLOv5, indicating a huge benefit in minimizing missed detections. This shows that although YOLOv5 focuses on speed, it has a greater tendency to

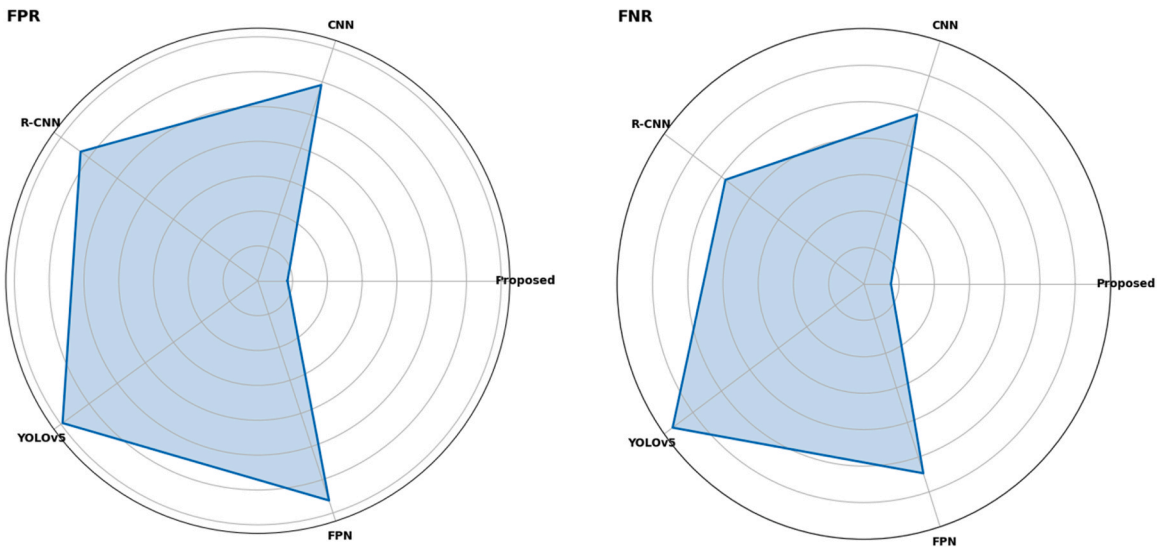

Fig. 12. FPR and FNR comparison of proposed and existing model.

miss detecting objects than the proposed model. FPN has an FNR of 0.0546. The new model is superior to FPN by around 4.70 percentage points and proves its high performance in reducing the cases when objects are not detected. The Fig. 13 displays the Receiver Operating Characteristic (ROC) curves for the suggested model, CNN, R-CNN, YOLOv5, and FPN, with a dashed line showing a random guess.

The suggested model has the highest AUC of 0.99727, which is nearly as close to the ideal value of 1. This reflects a very good capacity to differentiate between positive and negative instances. For comparison, CNN yields an AUC value of 0.93511, R-CNN yields an AUC value of 0.93533, YOLOv5 yields an AUC value of 0.92831, and FPN yields an AUC value of 0.93518. These are significantly lower than the proposed model, signifying that it has a better classification performance at different threshold settings. Fig. 14 indicates the training and validation performance of a model for 10 epochs. The graph on the left indicates the training and validation accuracy, and the one on the right indicates the training and validation loss.

The training accuracy (blue line) begins around 0.27 during the initial epoch and gradually rises throughout the training process. By the 10th epoch, it has risen to around 0.97. The validation accuracy (orange line) also rises, beginning around 0.35. It fluctuates to some degree between epochs 3 and 7, reaching a peak at around 0.89 near epoch 7,

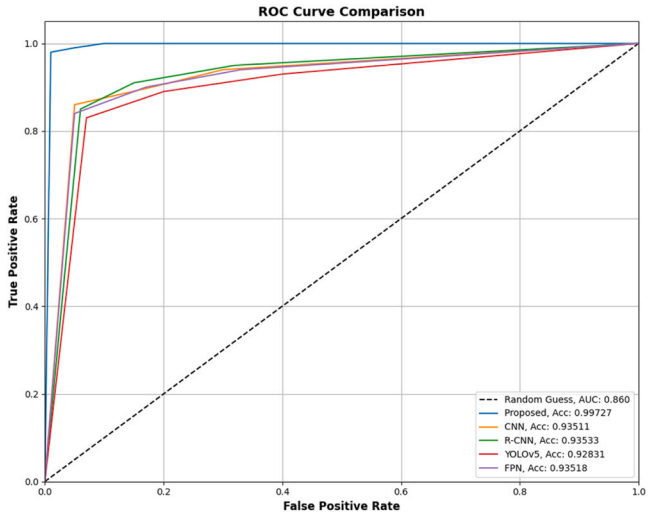

Fig. 13. ROC curve analysis of proposed and existing model.

and then slightly rises to around 0.96 by the 10th epoch. Training loss (blue) begins at the maximum value around 0.075 in the beginning epoch and dips dramatically during early epochs to tend towards an appreciable value closer to zero around the 10th epoch. Validation loss (orange line) also descends, initiating with a value close to around 0.06. It drops quickly in the initial epochs, to a low point at epoch 7 at about 0.004, and then increases slightly to about 0.006 by epoch 10. The validation loss beginning to rise slightly while the training loss still falls is another sign that the model may be beginning to overfit the training data towards the end of training. The proposed YOLO-ARM model shows better performance in both inference speed and computational cost compared to other models and is analysed in Table 2.

With an inference speed of 31 FPS, it performs better than CNNs (14 FPS), R-CNN (6 FPS), YOLOv5 (27 FPS), and FPN (18 FPS), making the quickest among the models tested. Besides, YOLO-ARM also has low CPU usage of 49 %, which is far less compared to R-CNN (76 %) and FPN (62 %), and similar to CNNs (48 %) and YOLOv5 (52 %). This blend of high speed and effective resource consumption serves to underscore YOLO-ARM's real-time capabilities, including autonomous driving and surveillance applications, where performance and computational efficiency must be balanced. The fact that the model can deliver 31 FPS while maintaining CPU usage at under 50 % is further testament to its optimization and lightweight nature, making it an applicable choice for rollout in resource-limited contexts. The ablation experiments verify the incremental gains from each module in the new YOLO-ARM framework and is analysed and compared in Table 3.

Baseline YOLOv7 has 94.87 % accuracy, 92.12 % precision, and an F1-score of 93.42 % with an inference rate of 30 FPS. When the Adaptive Attention Receptive Module (ARM) is combined into YOLOv7, the performance is greatly enhanced, with accuracy upgraded to 97.13 %, precision up to 94.91 %, and the F1-score up to 96.01 %, while the inference speed reduces slightly to 29 FPS. In the same way, adding only the CBAM to YOLOv7 achieves 96.48 % accuracy, 94.2 % precision, and an F1-score of 95.29 % with a slight decrease in speed to 28 FPS. But the highest performance metrics are obtained from the proposed YOLO-ARM model by incorporating both ARM and CBAM: an accuracy of 99.73 %, precision of 97.99 %, and F1-score of 98.6 %, and along with that, a real-time inference speed of 31 FPS Table 4.

The comparison of the proposed YOLO-ARM model with the existing models on the MS COCO and PASCAL VOC datasets proves its high performance. For the MS COCO dataset, YOLO-ARM has 99.73 % accuracy, 97.99 % precision, and an F1-score of 98.6 %, far better than CNNs (93.51 %, 91.7 %, 94.35 %), R-CNN (93.53 %, 93.77 %,

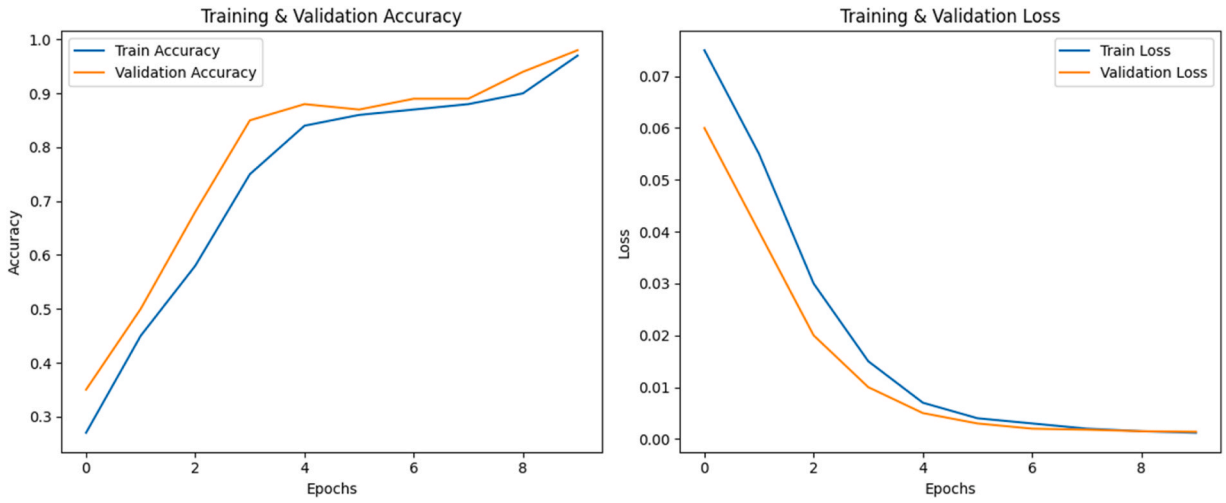

Fig. 14. Training and validation comparison of the proposed model.

Table 2
Comparison of inference time and CPU Utilization.

Model	Inference Time (FPS)	CPU Utilization (%)
CNNs (Baseline)	14 FPS	48 %
R-CNN	6 FPS	76 %
YOLOv5	27 FPS	52 %
FPN	18 FPS	62 %
Proposed YOLO-ARM	**31 FPS**	**49 %**

Table 3
Comparison of ablation studies analysis.

Model Variant	Accuracy (%)	Precision (%)	F1-Score (%)	FPS
YOLOv7 (Baseline)	94.87	92.12	93.42	30
YOLOv7 + ARM	97.13	94.91	96.01	29
YOLOv7 + CBAM	96.48	94.2	95.29	28
YOLO-ARM (Proposed)	**99.73**	**97.99**	**98.6**	**31**

Table 4
Comparison and analysis of dataset evaluation.

Model	Dataset	Accuracy (%)	Precision (%)	F1-Score (%)
CNNs	MS COCO	93.51	91.7	94.35
	PASCAL VOC	91.87	90.42	92.31
R-CNN	MS COCO	93.53	93.77	94.62
	PASCAL VOC	92.13	91.6	93.04
YOLOv5	MS COCO	92.83	94.08	93.51
	PASCAL VOC	91.36	92.79	91.82
FPN	MS COCO	93.52	93.73	95.29
	PASCAL VOC	92.2	92.45	94.01
YOLO-ARM	MS COCO	99.73	97.99	98.6
	PASCAL VOC	98.91	96.58	97.84

94.62 %), YOLOv5 (92.83 %, 94.08 %, 93.51 %), and FPN (93.52 %, 93.73 %, 95.29 %).Also, on the PASCAL VOC benchmark, YOLO-ARM continued high performance with 98.91 % accuracy, 96.58 % precision, and 97.84 % F1-score, performing better than CNNs (91.87 %, 90.42 %, 92.31 %), R-CNN (92.13 %, 91.6 %, 93.04 %), YOLOv5 (91.36 %, 92.79 %, 91.82 %), and FPN (92.2 %, 92.45 %, 94.01 %). These findings reveal the strong robustness and generalization ability of YOLO-ARM on various datasets, positioning it as a state-of-the-art model for high-accuracy object detection tasks. The model's across-the-board superiority in both datasets confirms the effectiveness of its Adaptive ARM and CBAM to handle scale changes, occlusion, and cluttered backgrounds.

The proposed YOLO-ARM model exhibits higher accuracy and real-time inference performance on benchmark datasets; however, it is worth mentioning a number of possible limitations that could impact its wider use. First, the incorporation of attention mechanisms—i.e., the ARM and CBAM necessarily complicates the model's training complexity and convergence time, particularly when utilizing large-scale datasets. While the inference stays lightweight, training consumes extra computational efforts and more epochs compared to the original YOLOv7 architecture. Second, although YOLO-ARM is very competitive on datasets in the size range of MS COCO and PASCAL VOC, scaling up to huge dimensions as in Open Images or domain-specific datasets with millions of images could be further optimized in terms of memory handling and distributed training strategies.

The findings conclusively prove that YOLO-ARM establishes a new benchmark for object detection, achieving state-of-the-art accuracy of 99.727 % and precision of 97.997 % with real-time efficiency. Its novel employment of ARM and CBAM resolves fundamental issues in robotic vision, including scale changes, occlusion, and complex backgrounds. The low error rates of the model FPR of 0.0085 and FNR of 0.0077 and its balanced performance MCC of 98.48 % make it an attractive solution for autonomous systems, surveillance, and industrial automation. Domain adaptation and hardware optimization will be the areas of future work to extend its applicability. The YOLO-ARM model establishes a new state of the art in object recognition, overcoming key limitations of current frameworks with novel attention mechanisms and adaptive fusion of features. Its high accuracy, low error rates, and real-time efficiency make it a revolutionary solution for robotic vision systems. Domain-specific optimizations (e.g., underwater or medical imaging) could be investigated in future work to further extend its applicability.

7. Conclusion

The paper introduced an enhanced YOLOv7-driven object detection system, improved with an Adaptive ARM and CBAM, which attains outstanding performance results. The contributions of this work are the incorporation of ARM for adaptive feature refinement and CBAM for dual-channel and spatial attention, greatly enhancing feature discrimination in complicated situations. The proposed model surpasses current models, such as CNNs, R-CNN, YOLOv5, and FPN, achieving an accuracy of 99.727 %, precision of 97.997 %, and an F1-score of 98.60 %, showing its better ability to balance detection accuracy and efficiency. Key observations identify the proposed model's outstanding

performance in lowering FPR of 0.00851 and FNR of 0.0077 and its ability to perform well under small and highly concentrated objects. The model's specificity of 99.02 % and sensitivity of 98.31 % also support its performance in different surroundings. The implications of this work are significant for real-time and high-precision object recognition applications like independent driving, surveillance, and robotics vision. The light model design and the model's flexibility in being applicable make it appropriate to deploy in low-resource environments. Future research may consider combining transformer-based models with other architectures to further improve multi-scale feature extraction and generalization to diverse datasets. To further develop high-precision robotic vision object detection, a few directions can be pursued. One, combining transformer-based architectures with the current YOLO-ARM framework could improve multi-scale feature extraction and contextual awareness, especially for difficult scenes. Two, domain-specific adaptations, for example, special models for medical imaging or robotic vision application (e.g., autonomous navigation, robotic arms) can be explored to tackle special issues such as low contrast or turbidity.

CRediT authorship contribution statement

Fuzhi Wang: Validation, Supervision, Software, Project administration, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Changlin Song:** Writing – review & editing, Writing – original draft, Validation, Software, Methodology.

Consent to publish

Not applicable.

Ethics approval

Not applicable

Funding

This work was supported by the Chunhui project of the Ministry of Education of China (NO. 12202528) and the Key Project of Xihua University (NO. Z1120223)

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] Che, C., Zheng, H., Huang, Z., Jiang, W. and Liu, B., 2024. Intelligent robotic control system based on computer vision technology. arXiv preprint arXiv: 2404.01116.
[2] Y.L. Chen, Y.R. Cai, M.Y. Cheng, Vision-based robotic object grasping—a deep reinforcement learning approach, Machines 11 (2) (2023) 275.
[3] Y. Li, C. Yin, Y. Lei, J. Zhang, Y. Yan, RDD-YOLO: road damage detection algorithm based on improved you only look once version 8, Appl. Sci. 14 (8) (2024) 3360.
[4] M.W. Ahmed, A. Jalal, Robust object recognition with genetic algorithm and composite saliency map (February). 2024 5th International Conference on Advancements in Computational Sciences (ICACS), IEEE, 2024, pp. 1–7 (February).
[5] Naseer, A., Al Mudawi, N., Abdelhaq, M., Alonazi, M., Alazaib, A., Algarni, A. and Jalal, A., 2024. CNN-based object detection via segmentation capabilities in Outdoor natural scenes. Ieee Access.
[6] U. Sirisha, S.P. Praveen, P.N. Srinivasu, P. Barsocchi, A.K. Bhoi, Statistical analysis of design aspects of various YOLO-based deep learning models for object detection, Int. J. Comput. Intell. Syst. 16 (1) (2023) 126.
[7] X. Wang, N. He, C. Hong, Q. Wang, M. Chen, Improved YOLOX-X based UAV aerial photography object detection algorithm, Image Vis. Comput. 135 (2023) 104697.
[8] Liu, B., Yu, L., Che, C., Lin, Q., Hu, H. and Zhao, X., 2023. Integration and performance analysis of artificial intelligence and computer vision based on deep learning algorithms. arXiv preprint arXiv:2312.12872.
[9] A.A. Rafique, M. Gochoo, A. Jalal, K. Kim, Maximum entropy scaled super pixels segmentation for multi-object detection and scene recognition via deep belief network, Multimed. Tool Appl. 82 (9) (2023) 13401–13430.
[10] Y. Li, C. Ma, L. Li, R. Wang, Z. Liu, Z. Sun, Lightweight tunnel obstacle detection based on improved YOLOv5, Sensors 24 (2) (2024) 395.
[11] J. Yang, Y. Gapar, Improved object detection algorithm based on multi-scale and variability convolutional neural networks, IECE Trans. Emerg. Top. Artif. Intell. 1 (1) (2024) 31–43.
[12] A. Li, S. Sun, Z. Zhang, M. Feng, C. Wu, W. Li, A multi-scale traffic object detection algorithm for road scenes based on improved YOLOv5, Electronics 12 (4) (2023) 878.
[13] H. Nie, H. Pang, M. Ma, R. Zheng, A lightweight remote sensing small target image detection algorithm based on improved YOLOv8, Sensors 24 (9) (2024) 2952.
[14] H.C. Dan, P. Yan, J. Tan, Y. Zhou, B. Lu, Multiple distresses detection for asphalt pavement using improved you only look once algorithm based on convolutional neural network, Int. J. Pavement Eng. 25 (1) (2024) 2308169.
[15] P. Liu, W. Qian, Y. Wang, YWnet: a convolutional block attention-based fusion deep learning method for complex underwater small target detection, Ecol. Inform. 79 (2024) 102401.
[16] S. Zhou, H. Zhou, Detection based on semantics and a detail infusion feature pyramid network and a coordinate adaptive spatial feature fusion mechanism remote sensing small object detector, Remote Sens. 16 (13) (2024) 2416.
[17] A. Naseer, A. Jalal, Integrating semantic segmentation and object detection for multi-object labeling in aerial images, ICACS (2024).
[18] L.D. Quach, K.N. Quoc, A.N. Quynh, H.T. Ngoc, Evaluating the effectiveness of YOLO models in different sized object detection and feature-based classification of small objects, J. Adv. Inf. Technol. 14 (5) (2023) 907–917.
[19] S. Zhou, H. Zhou, L. Qian, A multi-scale small object detection algorithm SMA-YOLO for UAV remote sensing images, Sci. Rep. 15 (1) (2025) 9255.
[20] X. Geng, Y. Su, X. Cao, H. Li, L. Liu, YOLOFM: an improved fire and smoke object detection algorithm based on YOLOv5n, Sci. Rep. 14 (1) (2024) 4543.
[21] H. Zhou, M. Kong, H. Yuan, Y. Pan, X. Wang, R. Chen, W. Lu, R. Wang, Q. Yang, Real-time underwater object detection technology for complex underwater environments based on deep learning, Ecol. Inform. 82 (2024) 102680.
[22] M. Sileo, N. Capece, M. Gruosso, M. Nigro, D.D. Bloisi, F. Pierri, U. Erra, Vision-enhanced Peg-in-Hole for automotive body parts using semantic image segmentation and object detection, Eng. Appl. Artif. Intell. 128 (2024) 107486.
[23] M. Sileo, N. Capece, M. Gruosso, M. Nigro, D.D. Bloisi, F. Pierri, U. Erra, Vision-enhanced Peg-in-Hole for automotive body parts using semantic image segmentation and object detection, Eng. Appl. Artif. Intell. 128 (2024) 107486.
[24] D. Lim, J. Kim, H. Kim, Efficient robot tracking system using single-image-based object detection and position estimation, ICT Express 10 (1) (2024) 125–131.
[25] M. Umer, S. Sadiq, R.M. Alhebshi, S. Alsubai, A. Al Hejaili, A.A. Eshmawi, M. Nappi, I. Ashraf, Face mask detection using deep convolutional neural network and multi-stage image processing, Image Vis. Comput. 133 (2023) 104657.
[26] H. Lou, X. Duan, J. Guo, H. Liu, J. Gu, L. Bi, H. Chen, DC-YOLOv8: Small-size object detection algorithm based on camera sensor, Electronics 12 (10) (2023) 2323.
[27] T. Wu, Y. Dong, YOLO-SE: improved YOLOv8 for remote sensing object detection and recognition, Appl. Sci. 13 (24) (2023) 12977.
[28] S.P. Yadav, M. Jindal, P. Rani, V.H.C. de Albuquerque, C. dos Santos Nascimento, M. Kumar, An improved deep learning-based optimal object detection system from images, Multimed. Tool Appl. 83 (10) (2024) 30045–30072.
[29] G. Balamurugan, Faster region based convolution neural network with context iterative refinement for object detection, Meas. Sens. 31 (2024) 101025.
[30] A. Gomaa, A. Abdalrazik, Novel deep learning domain adaptation approach for object detection using semi-self building dataset and modified yolov4, World Electr. Veh. J. 15 (6) (2024) 255.
[31] J. Zhang, J. Zhang, K. Zhou, Y. Zhang, H. Chen, X. Yan, An improved YOLOv5-based underwater object-detection framework, Sensors 23 (7) (2023) 3693.
[32] K. Li, Y. Wang, Z. Hu, Improved YOLOv7 for small object detection algorithm based on attention and dynamic convolution, Appl. Sci. 13 (16) (2023) 9316.
[33] X. Jiang, Y. Wu, Remote sensing object detection based on convolution and swin transformer, IEEE Access 11 (2023) 38643–38656.
[34] Y. Cao, C. Li, Y. Peng, H. Ru, MCS-YOLO: a multiscale object detection method for autonomous driving road environment recognition, IEEE Access 11 (2023) 22342–22354.
[35] R. Li, X. Zeng, S. Yang, Q. Li, A. Yan, D. Li, ABYOLOv4: improved YOLOv4 human object detection based on enhanced multi-scale feature fusion, EURASIP J. Adv. Signal Process. 2024 (1) (2024) 6.