



Boosting YOLO11: Global Attention & Hyperparameter Tuning for High-Fidelity Military Aircraft Detection

Satyo Widiyanarto¹, Ema Utami²

^{1,2}Master of Informatics, Universitas AMIKOM Yogyakarta, Yogyakarta, Indonesia

¹alexsatyo@students.amikom.ac.id, ²ema.u@amikom.ac.id

Abstract

Military aircraft detection from aerial and satellite imagery is crucial for strategic surveillance and intelligence. This study evaluated the impact of the Global Attention Mechanism (GAM) and hyperparameter optimization on the performance of the YOLO11 model for military aircraft detection. Utilizing a traditional YOLO model as a baseline, we compared precision, recall, and mean Average Precision (mAP) metrics across various configurations. These configurations included the implementation of GAM and variations in n, s variant of YOLO11, optimizers (Adam, NAdam, RAdam, Adamax, AdamW, SGD) and learning rates (0.01, 0.001, 0.0001, 0.00417). Experimental results demonstrate that the integration of GAM significantly enhances the model's detection capabilities within 300 iterations, particularly when combined with the Adamax optimizer and a learning rate of 0.001. This specific configuration achieved the highest mAP performance of 98.5%, outperforming other setups. Further confusion matrix analysis confirmed high accuracy in classifying various aircraft types, while also highlighting some challenges in distinguishing certain classes. The primary contribution of this study is the empirical demonstration of improved military aircraft detection performance by YOLO11 through the utilization of four global attention mechanism modules and effective hyperparameter tuning. These findings offer valuable insights for developing more accurate and robust object detection systems for defense and security applications.

Keywords: hyperparameters optimization; Global Attention Mechanism; object detection; military aircraft; YOLO11

How to Cite: S. Widiyanarto, and E. Utami, "Boosting YOLO11: Global Attention & Hyperparameter Tuning for High-Fidelity Military Aircraft Detection", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 9, no. 6, pp. 1306 - 1328, Dec. 2025.

Permalink/DOI: <https://doi.org/10.29207/resti.v9i6.7102>

Received: August 20, 2025

Accepted: November 16, 2025

Available Online: December 4, 2025

This is an open-access article under the CC BY 4.0 License

Published by Ikatan Ahli Informatika Indonesia

1. Introduction

Military aircraft detection from remote sensing imagery is a critical aspect in various applications such as reconnaissance, early warning, and intelligence analysis. Imagery generated through satellites or Synthetic Aperture Radar (SAR) provides strategic information highly needed for maintaining air security and sovereignty. However, the process of detecting military aircraft objects in such imagery faces significant challenges, including variations in aircraft shape and size, differing weather conditions, and complex backgrounds that interfere with accurate identification [1] - [3]. Therefore, object detection algorithms are needed that are not only fast and efficient but also capable of recognizing objects with high precision under highly diverse conditions.

Deep learning technology has become a fundamental foundation in the development of various modern object detection algorithms. The YOLO (You Only Look

Once) algorithm family stands out for its capability in real-time detection while maintaining a high level of accuracy. Recent versions such as YOLO11 show promising performance improvements, especially in the context of military aircraft detection in remote sensing imagery [1], [3]. However, the complexity of SAR imagery and highly varied backgrounds remain major obstacles that hinder comprehensive detection optimization.

Recent research endeavors to overcome these obstacles by developing various YOLO variants equipped with attention mechanisms, which are modules capable of enhancing the model's focus on important features in the imagery. For example, the YOLOv8s variant with TripletAttention integration successfully addresses detection issues on small details and overlapping areas, with performance improvements on the MAR20 dataset [1]. The DET-YOLO model, which combines feature extraction and attention mechanism modules, also

shows superior results in complex image environments [3]. Meanwhile, YOLO11, which uses Enhanced Channel Attention (ECA), can improve precision and F1 score in detecting aircraft in SAR imagery, although it still faces limitations in optimally handling image complexity [2]. Furthermore, research related to the detection of small objects such as drones and aircraft surface defects states that the Global Attention Mechanism (GAM) can significantly enhance the model's ability to focus attention on important areas and improve the detection of fine details often missed by local attention modules [4], [5]. However, the integration of global attention mechanism specifically into YOLO11 for military aircraft detection in the MAR20 dataset has not been widely explored, making it a potential research gap to be developed.

2. Methods

This research gap arises because many previous studies have relied more on local attention modules such as ECA or TripletAttention and have not comprehensively examined the effectiveness of global attention mechanisms in improving generalization capabilities and detection accuracy, especially in highly complex and low-resolution image conditions [1], [2], [5]. Therefore, this study aims to fill this gap by improving YOLO11 through the integration of the global attention mechanism, which is expected to enhance the detection capability of military aircraft in the MAR20 dataset in remote sensing imagery. In this experiment, several research stages will be carried out as shown in Figure 1, including dataset collection, data pre-processing, augmentation and data splitting, and model training using YOLO11 modified by adding a Global Attention Mechanism module.

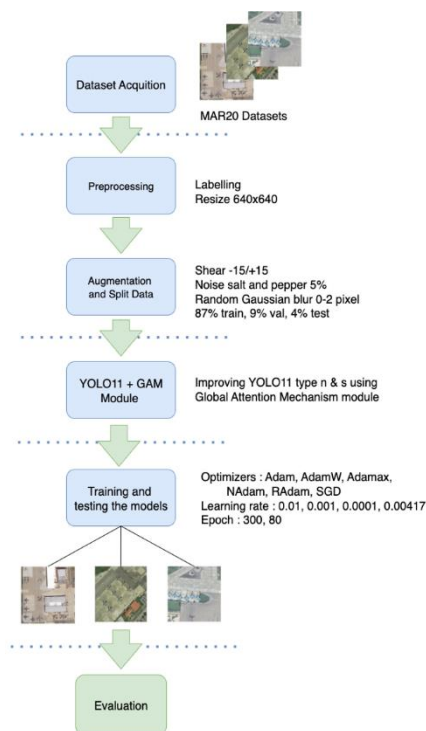


Figure 1. The stages of research

Training will be conducted with various parameters such as different optimizers and learning rates with 80 epochs and maximum of 300 epochs. The obtained results will be evaluated by measuring the mAP, Recall, and Precision values to then measure their improvement from the original model. The research approach used is experimental, where the researchers control conditions to observe the impact of applying various parameter tuning variations on the YOLO11 architecture.

2.1 Experimental Setup

Table 1 is some information about hardware, software, and parameter settings.

Table 1. Information of hardware, software and parameter settings

CPU	AMD Ryzen 5 5600
Memory	DDR 32 GB
GPU	RTX 3050 8 GB
Operational System	Ubuntu 24.04.1 LTS
Framework	Python 3.12.3 torch 2.5.1+cu124 CUDA
Model	YOLO11n, YOLO11s
Optimizers	Adam, Adamax, AdamW, NAdam, RAdam, SGD
Learning Rate	0.01, 0.001, 0.0001, 0.00417
Training Iterations	300, 80
Batch size	16
Image size	512
Momentum	0.9
Weight decay	0.0005
IoU-threshold	0.7

2.2 Dataset

The dataset used in this research is MAR20 (Military Aircraft Recognition) [6]. This dataset allows research to test military aircraft detection capabilities in various complex and diverse backgrounds. Figure 2 shows an example of the MAR20 dataset.



Figure 2. Sampel dataset MAR20

Consisting of 3,842 images, MAR20 contains 20 types of military aircraft with a total of 22,341 instances, where each instance is equipped with two types of annotations. Annotations are available in both horizontal and oriented bounding box representations.

This dataset was collected from 60 military bases in various countries using Google Earth imagery, thus covering diverse weather conditions, seasons, lighting, and various obstacles that add to the difficulty level in the aircraft identification process.

2.3 Preprocessing

In this research, a preprocessing stage was carried out on the dataset to ensure the quality and effectiveness of the data used. The pre-processing process included resizing images to 640x640 pixels and adjusting class names according to the actual military aircraft type. For example, in Table 2, class A1 is labeled SU-35, referring to an advanced Russian fighter jet. The dataset covers various aircraft types, ranging from fighter jets like F-16 and F-22, heavy transport aircraft like C-130 and C-17, to reconnaissance and aerial refueling aircraft, such as E-3 and KC-135. This class renaming aims to associate each class with a specific aircraft type or model, thus facilitating the identification and classification process based on the function and specific capabilities of each aircraft type.

Table 2. Modify Class Name

Old Class	New Class
A1	SU-35
A2	C-130
A3	C-17
A4	C-5
A5	F-16
A6	TU-160
A7	E-3
A8	B-52
A9	P-3C
A10	B-1B
A11	E-8
A12	TU-22
A13	F-15
A14	KC-135
A15	F-22
A16	F-18
A17	TU-95
A18	KC-10
A19	SU-34
A20	SU-24

2.4 Data Augmentation and Split

After the pre-processing stage, the data was then augmented using shear methods of -15/+15 (horizontal/vertical), salt and pepper noise of 5%, and Random Gaussian blur with a range of 0-2 pixels, which collectively totaled 9,222 images. Subsequently, the data was divided into a training set of 87% (7,070 images), validation of 9% (768 images), and testing of 4% (384 images).

The size of the dataset significantly influences the proportion of data allocated for training, validation, and testing. For large datasets, the proportion for validation and testing can be smaller because the absolute number of samples remains sufficient for accurate evaluation. Conversely, for smaller datasets, the proportion for validation and testing tends to be larger to ensure more reliable and less biased evaluation results [7][8].

With a total of 9,222 images, the 87% allocation for training is sufficient for effective training. The 9% validation proportion provides an adequate size for the tuning process without significantly reducing the data for training. Meanwhile, 4% for testing is sufficient to evaluate the model's performance in real-world conditions.

2.5 YOLO11 Architecture

The YOLO11 framework, officially released by Ultralytics as a direct successor to YOLOv8, advances real-time object detection through key architectural innovations. It integrates a C3k2 (Cross Stage Partial with three convolutions and two 3x3 kernels) block, which replaces the C2f block found in previous versions for more efficient feature extraction, a SPPF (Spatial Pyramid Pooling Fast), and a C2PSA (Cross Stage Partial with Spatial Attention) as shown in Figure 3. These innovations aim to strengthen the model's capabilities in feature extraction and improve computational efficiency [9].

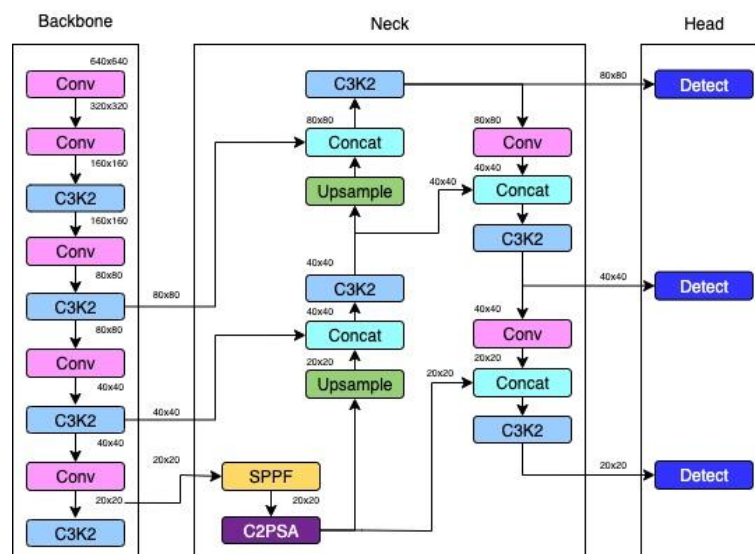


Figure 3. YOLO11 Architecture

The experimental methodology employed in this study involved a comparative analysis of two specific YOLO11 variants, namely YOLO11n and YOLO11s, to evaluate their performance trade-offs between speed and accuracy. These variants were benchmarked against other models as depicted in Figure 4, which illustrates the relationship between latency and COCO mAP50-95 scores. YOLO11n was characterized by its lower latency, operating at approximately 1.5ms, though achieving a COCO mAP50-95 score of about 39%. Conversely, YOLO11s, while exhibiting a marginally higher latency of approximately 2.5ms, demonstrated a notable improvement in accuracy with a COCO mAP50-95 score of around 47%. This comparative assessment allowed for the identification of optimal use cases for each variant: YOLO11n for applications prioritizing extreme speed, and YOLO11s for real-time scenarios where a balance of enhanced accuracy and efficient processing is critical.

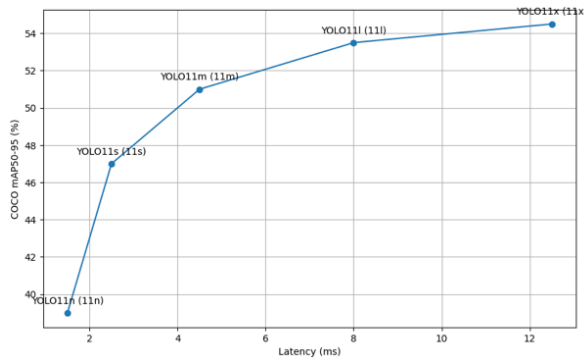


Figure 4. Benchmarking of variant YOLO11

2.6 Global Attention Mechanism Module

The Global Attention Mechanism (GAM) was introduced by Liu, Shao, and Hoffmann in the article titled "Global Attention Mechanism: Retain Information to Enhance Channel-Spatial Interactions" [10]. The architecture of GAM is shown in Figure 5.

GAM is designed to enhance cross-dimensional interaction in neural networks by retaining information from both Channel Attention and Spatial Attention aspects. GAM utilizes a channel attention sub-module that leverages 3D permutations and a multilayer perceptron (MLP) to strengthen spatial attention. In the spatial attention sub-module, two convolutional layers are used to aggregate spatial information without using pooling, which can reduce important information.

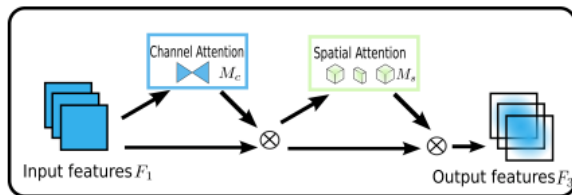


Figure 5. Global Attention Mechanism (GAM) Module

$$F1 \in RC \times H \times W \quad (1)$$

$$F2 = Mc(F1) \odot F1 \quad (2)$$

$$F3 = Ms(F2) \odot F2 \quad (3)$$

The equations for the Global Attention Mechanism (GAM) can be explained as follows. Given the input feature map in Equation 1, where C is the number of channels, H is the height, and W is the width, the GAM process consists of two main steps, namely Channel In Equation 2, Channel Attention, denoted as Mc is the channel attention map generated by processing $F1$ using 3D permutation and a multilayer perceptron (MLP), and $\odot$ represents element-wise multiplication.

In Equation 3, Spatial Attention, denoted as Ms , is the spatial attention map generated by processing $F2$ using two convolutional layers to aggregate spatial information. Thus, the final output $F3$ is the result of applying Channel Attention and Spatial Attention sequentially, which enhances the cross-dimensional interaction between Channel Attention and Spatial Attention.

2.7 YOLO11 with GAM Module Architecture

The implementation of the Global Attention Mechanism (GAM) module in the YOLO architecture is a significant innovation aimed at improving object detection accuracy, particularly for small targets that are often difficult to recognize in images captured by Satellites or UAVs. In the experiment [11], GAM was integrated into the YOLOv8 backbone module to capture crucial feature information across various dimensions, thereby reducing the difficulty in detecting small objects affected by factors such as blur due to rapid movement and long shooting distances.

This study introduces a novel modified YOLO11 architecture, detailed in Figure 6, which integrates four Global Attention Mechanism (GAM) modules within its Neck for enhanced military aircraft detection. The network comprises a Backbone for progressive feature downsampling from 640x640 inputs, a Neck for robust multi-scale feature aggregation, and a Head for object detection.

The four GAM modules are strategically placed within the Neck: the first two after initial concatenation and C3K2 blocks, processing 80x80 and 40x40 features respectively. The third GAM module refines 20x20 feature maps after processing, and a fourth is integrated into the deepest part of the Neck, operating on 20x20 features within a path including SPPF and C2PSA modules. This strategic placement ensures that global attention enhances feature representation and flow across various scales, from fine-grained to abstract. This integration of GAMs within the Neck aims to improve the model's ability to handle scale variations and occlusions, thereby boosting overall detection performance in the Head.

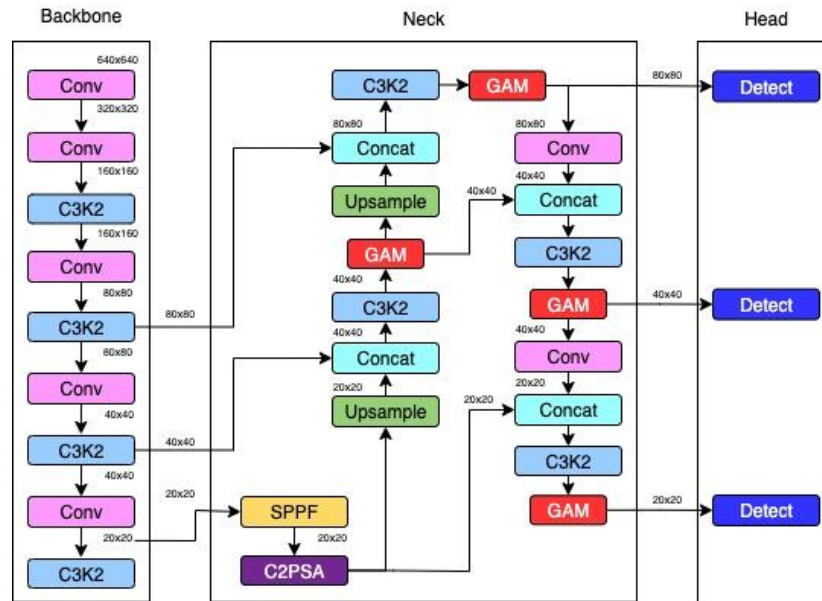


Figure 6. YOLO11 with GAM Module Architecture

2.8 Training Scheme

Table 3 outlines Training Scheme I, which employs the YOLO11s model augmented with the Global Attention Mechanism (GAM). The training commenced with the configuration of optimizer parameters, including options such as Adam[12], Adamax [13], AdamW, NAdam, RAdam [14], and SGD [15]. After selecting an optimizer, the next step is to determine the learning rate (lr), with choices among 0.01, 0.001, or 0.0001. Next, a parameter of 300 epochs is set as the maximum number of training iterations for the network. Once all parameters are configured, the process proceeds to the Training step, where the model undergoes the learning process according to the previously determined parameters. After training is complete, a Results Recording step is performed to record or save the outcomes of the training process. Subsequently, it is evaluated whether all parameters have been trained well and correctly. If not, the flow returns to the initial parameter setting stage for adjustment or other configurations so that training can be rerun. If all parameters have been trained, the process ends.

Table 4 details Training Scheme II, which was designed to conduct a comparative analysis with prior research by [3] using identical hyperparameters. This scheme involved training twelve distinct models, categorized into two primary groups. The first group, comprising Models 19 through 24, utilized the baseline YOLO11n architecture. The second group, Models 25 through 30, incorporated the Global Attention Mechanism (GAM) into the YOLO11n architecture (denoted as YOLO11n+GAM) to investigate its impact. Across both groups, a comprehensive set of six different optimizers AdamW, RAdam, NAdam, Adamax, SGD, and Adam were systematically applied to each model. Crucially, all training runs under Scheme II shared consistent global training parameters: an initial learning

rate of 0.00417 and a fixed training duration of 80 epochs. This meticulous design ensures a direct and fair comparison of the performance characteristics of various optimizers, both with and without the GAM integration, under conditions mirroring those of the referenced study.

Table 3. Training Scheme I

Model	Learning Rate (lr)	Optimizer
Model 1	0.01	AdamW
Model 2	0.01	Adam
Model 3	0.01	RAdam
Model 4	0.01	NAdam
Model 5	0.01	Adamax
Model 6	0.01	SGD
Model 7	0.001	AdamW
Model 8	0.001	Adam
Model 9	0.001	RAdam
Model 10	0.001	NAdam
Model 11	0.001	Adamax
Model 12	0.001	SGD
Model 13	0.0001	AdamW
Model 14	0.0001	Adam
Model 15	0.0001	RAdam
Model 16	0.0001	NAdam
Model 17	0.0001	Adamax
Model 18	0.0001	SGD

Table 4. Training Scheme II

Model	Type	Optimizer
Model 19	YOLO11n	AdamW
Model 20	YOLO11n	RAdam
Model 21	YOLO11n	NAdam
Model 22	YOLO11n	Adamax
Model 23	YOLO11n	SGD
Model 24	YOLO11n	Adam
Model 25	YOLO11n+GAM	AdamW
Model 26	YOLO11n+GAM	RAdam
Model 27	YOLO11n+GAM	NAdam
Model 28	YOLO11n+GAM	Adamax
Model 29	YOLO11n+GAM	SGD
Model 30	YOLO11n+GAM	Adam

2.9 Evaluation

In object detection, evaluation commonly relies on metrics like Recall (R), Precision (P), and Average

Precision (AP). Recall, as defined in Equation 4, quantifies the fraction of actual targets successfully identified. Precision, as per Equation 5, measures the fraction of identified targets that are actually correct.

$$R = \frac{TP}{TP+FN} \quad (4)$$

$$P = \frac{TP}{TP+FP} \quad (5)$$

Here, TP signifies the count of accurately identified targets, FP denotes the number of incorrectly detected targets, and FN refers to the number of missed targets. Average Precision (AP) is defined as the area underneath the Precision Recall (PR) curve, where recall serves as the x-axis and precision as the y-axis, as formulated in Equation 6. For multi-class scenarios, the mean Average Precision (mAP) is obtained by averaging the AP values across all classes, as illustrated in Equation 7.

$$AP = \int_0^1 P_i(R_i) dR_i \quad (6)$$

$$mAP = \frac{\sum_{i=1}^C AP_i}{C} \quad (7)$$

mAP is crucial in object detection because it measures the overall performance of the model by combining precision, recall, and object localization accuracy through bounding boxes [16] - [19]. Object detection not only evaluates class classification but also how accurately the object's position is found. mAP calculates the average precision at various recall and IoU thresholds, thus providing a comprehensive overview of the model's ability to correctly detect objects and accurately position bounding boxes. With mAP, we can assess the model's effectiveness in capturing all relevant objects while minimizing detection or position errors, making it the most reliable and favored metric in object detection research and development.

3. Results and Discussions

3.1 Training Results

Table 5 presents a comprehensive evaluation of various optimizers and learning rates for the YOLO11-s model, specifically augmented with the Global Attention Mechanism (GAM), over 300 epochs in Scheme I. A primary observation is the significant impact of both the optimizer choice and the learning rate on the model's performance metrics (Precision, Recall, and mAP).

At lr 0.01, SGD (Model 6: 98.3% mAP) and Adamax (Model 5: 97.7% mAP) performed strongly, while NAdam (Model 4: 91.7% mAP) and AdamW (Model 1: 95.6% mAP) showed lower efficacy, suggesting sensitivity to higher learning rates. Reducing the lr to 0.001 generally improved performance; Model 11 (Adamax) achieved the highest mAP of 98.5%, and RAdam (Model 9) closely followed with 98.4% mAP, indicating optimal convergence. Conversely, a further reduction to lr 0.0001 yielded mixed results, with a dramatic performance drop for SGD (Model 18: 56.7%

mAP), implying steps too small for effective exploration within 300 epochs. Inference speed (112-135 FPS) remained largely unaffected by optimizer or LR choice, indicating their primary influence on training convergence and accuracy. Ultimately, Adamax and RAdam at lr 0.001 emerged as the most effective optimizers for YOLO11-s with GAM in military aircraft detection.

Table 5. Training Results of Scheme I

Model	Precision (%)	Recall (%)	mAP (%)	FPS
Model 1	91.2	91.5	95.6	133
Model 2	94.4	94.2	97.4	135
Model 3	94.1	94.2	97.3	135
Model 4	87.9	87.1	91.7	132
Model 5	94.9	95.6	97.7	117.65
Model 6	96.4	96.8	98.3	135
Model 7	95.6	96.8	98	116
Model 8	95.6	95.7	98	118
Model 9	96.4	96.8	98.4	113.6
Model 10	96.2	95.3	97.8	115
Model 11	96.7	96.8	98.5	116
Model 12	93.3	94.5	96.8	118
Model 13	95.2	93.3	96.8	117.6
Model 14	96.6	95	97.7	118
Model 15	95	95.4	97.4	117.6
Model 16	95.7	94.8	97.5	117.6
Model 17	92.7	90.9	95.4	116
Model 18	54	56.6	56.7	112

From the results, the combination of a 0.001 learning rate with adaptive optimizers like RAdam and Adamax in YOLO has proven highly effective in achieving good accuracy and mAP (mean Average Precision). The use of Adam with a 0.001 learning rate for breast cancer detection using YOLO resulted in a 97% hit rate and 82% mAP, demonstrating excellent performance for medical applications [20]. The AdamPlus optimizer (an Adam variant) on YOLOv5s also improved mAP@0.5 by 6% and mAP@0.95 by 13.2% compared to traditional optimizers, and accelerated convergence [21]. Furthermore, various studies on YOLO that adopt adaptive optimizers and other optimization techniques consistently show improvements in mAP and detection accuracy across different domains, such as solar cell defect detection (mAP 89.64%) [22], object detection in UAV imagery (mAP 91.4%) [23], and vehicle detection in high-resolution imagery (mAP@0.5 increased from 47.9% to 89.6%) [24]. Although not all studies explicitly mention the use of RAdam, Adamax, or NAdam, the general trend indicates that adaptive optimizers with low learning rates significantly contribute to improving YOLO's accuracy and mAP in various applications. Research indicates that using a learning rate of 0.01 with adaptive optimizers such as RAdam, Adamax, AdamW, Adam, and NAdam results in highly varied mAP (mean Average Precision) values, depending on the model, dataset, and other parameter combinations. In a study on fabric pattern detection using various adaptive optimizers, a learning rate of 0.01 was not reported as the best combination; instead, lower learning rates such as 0.001 or 0.0001 with Adamax or Adam yielded much higher mAP and accuracy (e.g., Adamax with a learning rate of 0.0001 achieved mAP 91.28%) [25]. In general, a too-high

learning rate like 0.01 with adaptive optimizers often leads to unstable training, slow convergence, or even failure to achieve optimal accuracy, resulting in highly fluctuating and inconsistent mAP. Other research also emphasizes the importance of dynamically adjusting the learning rate to keep the model stable and achieve the best performance [26]. Thus, the combination of a 0.01 learning rate with various optimizers can indeed produce highly varied and often suboptimal mAP compared to lower learning rates.

Overall, these findings underscore the sensitivity of detection models to several parameters such as learning rate and optimizer. For military aircraft detection tasks using YOLO11s with the global attention mechanism, a learning rate of 0.001 with optimizers such as RAdam or Adamax is the most promising configuration, providing an optimal balance between convergence speed and model accuracy.

Table 6 presents the performance metrics of twelve YOLO11n models within Training Scheme II, all subjected to 80 epochs and a fixed learning rate of 0.00417, to assess the impact of various optimizers and the Global Attention Mechanism (GAM). Adaptive optimizers (AdamW, RAdam, NAdam, Adamax) consistently outperformed Stochastic Gradient Descent (SGD), with Adamax (Models 22, 28) and RAdam (Model 20) frequently achieving the highest mAP scores (up to 96.8%). SGD, conversely, consistently produced the lowest metrics (e.g., Model 23: 93.5% mAP), highlighting its suboptimal performance for this configuration. While non-GAM models maintained high FPS (312-322), the integration of GAM (Models 25-30) resulted in a substantial FPS reduction (approximately 210 FPS) due to increased computational complexity. Although GAM-enabled models, such as Model 28 (Adamax, GAM), achieved a competitive 96.8% mAP with strong precision (94.8%) and recall (94.2%), this accuracy was comparable to some non-GAM models (e.g., Model 20, Model 22). This suggests that while GAM can enhance accuracy, its significant computational overhead necessitates careful consideration, particularly when alternative optimizer choices can yield similar accuracy without the FPS reduction.

Table 6. Training Results of Scheme II

Model	Precision(%)	Recall(%)	mAP(%)	FPS
Model 19	94.2	93.8	96.3	313
Model 20	94.7	94	96.8	322.58
Model 21	95.5	93.4	96.5	312.5
Model 22	95	93.5	96.8	322.58
Model 23	91.6	88.2	93.5	294.12
Model 24	95.1	92.1	96.4	322.58
Model 25	95	92.9	96.3	212.77
Model 26	92.9	92.3	95.9	212.77
Model 27	93.9	93.4	96.3	212.77
Model 28	94.8	94.2	96.8	212.77
Model 29	91.3	89.4	94.2	217.39
Model 30	93.8	92	96.1	208.33

An in-depth exploration of the factors influencing confidence scores and overall performance in Tables 5 and 6 highlights a complex relationship between model architecture, hyperparameter selection, and specific architectural enhancements. A primary determinant of performance is the foundational model architecture, where a clear trade-off between speed and accuracy is evident. Table 5, utilizing the YOLO11s+GAM models, demonstrates the capability to achieve very high accuracy, with Model 11 reaching a peak mean Average Precision (mAP) of 98.5%. However, this higher accuracy comes at the cost of inference speed, with Frames Per Second (FPS) generally ranging from 112 to 135. Conversely, the YOLO11n models in Table 6 are engineered for efficiency, achieving significantly faster inference speeds between 294 and 322 FPS, while offering a slightly lower, yet still robust, maximum mAP of 96.8%. This dynamic is a well-established principle in the YOLO series, where different model variants are designed to cater to diverse application needs, balancing computational efficiency with detection accuracy [9].

Furthermore, the integration of architectural modifications, such as the Global Attention Mechanism (GAM), introduces another layer to this trade-off. A direct comparison within Table 6 between the base YOLO11n models (19-24) and the GAM-enhanced YOLO11n+GAM models (25-30) reveals that the attention mechanism did not consistently improve mAP for this specific task. For instance, both Model 22 (without GAM) and Model 28 (with GAM) registered an identical mAP of 96.8% using the Adamax optimizer. Critically, the addition of GAM led to a substantial decrease in computational efficiency, with FPS dropping from approximately 322 to around 212. This illustrates that while attention mechanisms are designed to enhance feature extraction by allowing the model to focus on critical regions, their implementation can introduce significant computational overhead that may not translate into superior accuracy, a critical consideration for real-time applications [27][28]. The YOLO11 architecture itself incorporates advanced attention mechanisms like the C2PSA block to improve focus on important image regions, indicating that the value of adding further attention modules like GAM must be empirically validated against its impact on speed.

Finally, the pivotal role of hyperparameter tuning is consistently demonstrated across both tables, where the choice of optimizer and learning rate is crucial for unlocking a model's potential. The data shows that adaptive optimizers such as Adamax, Adam, and RAdam generally provide strong, reliable performance across different learning rates, with Model 11 (Adamax, lr 0.001) and Model 9 (RAdam, lr 0.001) achieving the highest mAP scores in Table 5. The learning rate itself is a critical factor; for the YOLO11s+GAM models, a learning rate of 0.001 proved to be the sweet spot for achieving peak performance. The sensitivity to this hyperparameter is most dramatically illustrated by the

Stochastic Gradient Descent (SGD) optimizer. In Table 5, SGD delivered excellent performance with a 0.01 learning rate (Model 6, 98.3% mAP), but its effectiveness plummeted at a rate of 0.0001 (Model 18, 56.7% mAP), underscoring that even traditional optimizers can excel but require more meticulous tuning compared to their adaptive counterparts [29]. This necessity for task-specific tuning, balancing the choice of model, architectural features, and hyperparameters, is essential for optimizing the dual objectives of accuracy and efficiency in object detection.

3.2 Training Model Performance

Figure 7 presents a 3D scatter plot (from Table 5) with precision on the x-axis, recall on the y-axis, and mAP on the z-axis, overlaid on the dataset of trained models; here are the three best performing models (Models 6, 9, and 11) to compare optimizer effects in a three-dimensional performance space. All three models occupy the upper-right-rear region of the plot, indicating simultaneously high precision, high recall, and high mAP, but subtle differences are visible: Model 11 (Adamax, learning rate 0.001) is positioned slightly ahead on the precision axis and marginally higher on the mAP axis (P 96.7%, R 96.8%, mAP 98.5%), making it the best overall performer among the three; Model 9 (RAdam, learning rate 0.001) closely follows with virtually identical recall and mAP (P 96.4%, R 96.8%, mAP 98.4%), suggesting comparable convergence characteristics; Model 6 (SGD, learning rate 0.01) attains competitive performance (P 96.4%, R 96.8%, mAP 98.3%) but is slightly lower in mAP relative to Model 11.

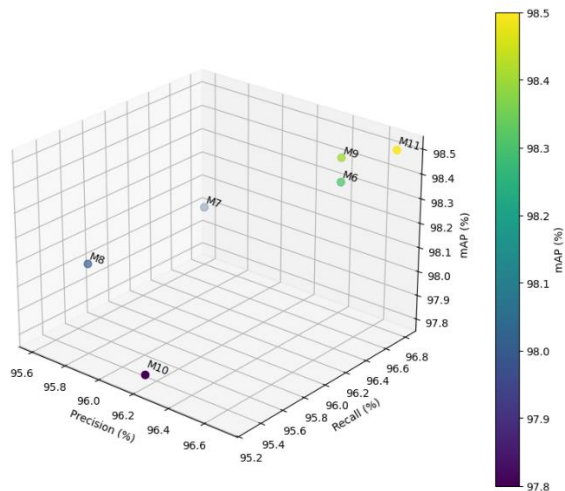


Figure 7. 3D Scatter Plot of the Six Best-Performing Models: Precision (%), Recall (%), and mAP (%) from Table 5 (M=Model)

The 3D visualization clarifies that while precision and recall are tightly clustered (reflecting their positive correlation), small separations along the mAP dimension distinguish optimizer effectiveness: adaptive methods such as Adamax and RAdam yield the highest aggregated detection quality, whereas a well-tuned SGD can approach similar results. These observations

indicate that, for the YOLO11 s-version, optimizer and learning rate selection leads to modest yet meaningful differences in overall detection performance. Consequently, when choosing a training strategy, researchers should balance these marginal improvements in mAP against other factors, including training stability and computational cost.

Figure 8, a 3D scatter plot of Precision (%), Recall (%), and mAP (%), visually encapsulates the performance analysis, highlighting the impact of different optimizers and the Global Attention Mechanism (GAM) on YOLO11n object detection models. The plot clearly illustrates that adaptive optimizers (AdamW, RAdam, NAdam, Adamax) consistently position models in the upper-right quadrant of the P-R plane, corresponding to higher mAP values, which are visually represented by brighter, warmer colors on the mAP scale. Conversely, models employing SGD (e.g., Model 23 and Model 29) are distinctly located in the lower-left region of the plot, characterized by lower precision, recall, and consequently, darker, cooler mAP colors. This visual separation strongly supports the finding that adaptive optimizers are significantly more effective for this architecture and dataset.

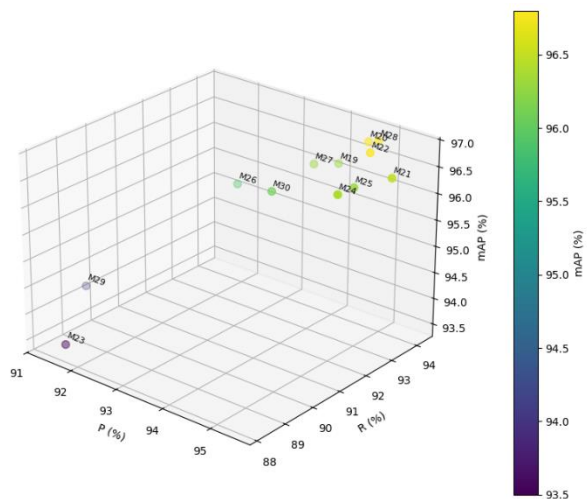


Figure 8. 3D Scatter Plot of Precision (%), Recall (%), and mAP (%) from Table 6 (M=Model)

3.3 Testing Model Performance

From the analysis of training performance, the three best models were identified for each scheme. For Scheme I, the top models were 6, 9, and 11. For Scheme II, the best models were 20, 22, and 28.

Table 7 shows the results of a study testing Scheme I, consisting of three baselines (without GAM) and three modified models (Models 6, 9, 11). In the first scenario with the SGD optimizer and a learning rate of 0.01, Baseline A achieved a precision of 96.8%, recall of 96%, and an mAP of 97.9%. When the Global Attention Mechanism (GAM) was activated (Model 6), there was an increase in precision to 96.9% and mAP to 98.1%, although recall slightly decreased to 95.7%. The increase in mAP indicates that GAM began to contribute to improving the model's detection accuracy.

Table 7. Performance comparison of Scheme I testing models

Model	Optimizer	Lr	GAM	P (%)	R (%)	mAP (%)
Baseline A	SGD	0.01	-	96.8	96	97.9
Model 6	SGD	0.01	v	96.9	95.7	98.1
Baseline B	RAdam	0.01	-	97.3	96.4	98.1
Model 9	RAdam	0.01	v	96.8	96.4	98.4
Baseline C	Adamax	0.001	-	95.8	97.3	97.9
Model 11	Adamax	0.001	v	97.5	96.1	98.3

Furthermore, with the RAdam optimizer and a learning rate of 0.01, Baseline B showed strong initial performance with a precision of 97.3%, recall of 96.4%, and an mAP of 98.1%. Interestingly, when GAM was added (Model 9), the mAP further increased to 98.4%, which is the highest mAP value overall in this study, although precision slightly decreased to 96.8% and recall remained stable. This demonstrates that the combination of GAM with the RAdam optimizer and a learning rate of 0.01 is highly effective in comprehensively improving object detection capabilities.

Lastly, in the configuration with the Adamax optimizer and a learning rate of 0.001, Baseline C recorded a precision of 95.8%, recall of 97.3%, and an mAP of 97.9%. With the implementation of GAM (Model 11), there was a substantial increase in precision to 97.5% (the highest among all models), and mAP to 98.3%, although recall slightly decreased to 96.1%. The significant increase in precision in Model 11 indicates that this model is very good at minimizing false positive detections.

Table 8 summarizes the performance of Scheme II testing models, highlighting subtle but meaningful differences across optimizers, learning rates, and GAM integration. Model 22 (Adamax, lr 0.00417, no GAM) achieved the highest precision (94.6%), recall (94.6%), and mAP (97.9%), demonstrating robust detection. Model 28 (Adamax, lr 0.00417, with GAM) showed a marginal increase in recall (94.9%) but a slight decrease in precision (94.4%), resulting in a 97.4% mAP, suggesting GAM's incremental impact. Model 20 (RAdam, no GAM) recorded competitive performance with a 97.5% mAP. Across both Scheme I and II, attention/GAM-based enhancements generally improved detection. In Scheme I, GAM consistently boosted mAP, with Model 9 (RAdam, LR=0.01) achieving the best overall mAP of 98.4%, and Model 11 attaining the highest precision (97.5%). For Scheme II, Adamax-based configurations, particularly Model 22, offered the best precision-recall balance and highest mAP. While GAM variants showed potential, their impact was metric-dependent.

Table 8. Performance comparison of Scheme II testing models

Model	Optimizer	Lr	GAM	P (%)	R (%)	mAP (%)
Model 20	RAdam	0.00417	-	93.6	93.8	97.5
Model 22	Adamax	0.00417	-	94.6	94.6	97.9
Model 28	Adamax	0.00417	v	94.4	94.9	97.4

3.4 Model Evaluation

After analyzing the training and testing performance of both scheme 1 and scheme 2 experiments, we identified the two best models for each scheme in Table 9. Model 11, utilizing the YOLO11-s version with a learning rate of 0.001, and Model 28, employing the YOLO11-n version with a learning rate of 0.00417, represent the optimal configurations derived from our comparative analysis.

Table 9. The best performance of scheme I and scheme II

Model	P (%)	R (%)	mAP (%)	FPS	Description
Model 11	96.7	96.8	98.5	116	Scheme I
Model 28	94.8	94.2	96.8	212.77	Scheme II

Figure 9 presents the confusion matrix for Model 11 (Adamax, Learning Rate 0.001, with Global Attention Mechanism), illustrating its robust performance in military aircraft differentiation. High diagonal values (e.g., 592 FA-18s, 326 F-15s, 215 F-16s, 283 P-3Cs correctly identified) demonstrate excellent classification capabilities, underscoring the Global Attention Mechanism's efficacy in focusing on discriminative features. While overall performance is strong, the matrix reveals areas for improvement, such as the misclassification of 44 true KC-10 instances as FA-18 and 4 true TU-95 instances as TU-22. These off-diagonal entries highlight challenges in distinguishing visually similar aircraft and opportunities for future refinement through expanded training data or architectural adjustments. The model also effectively differentiates aircraft from background, though minor misclassifications exist. In summary, the confusion matrix confirms Model 11's high accuracy in military aircraft detection and classification tasks, largely attributed to the Adamax optimizer and Global Attention Mechanism.

Figure 10 presents the confusion matrix for Model 28, an optimized YOLO11n network with GAM enhancements, demonstrating robust military aircraft detection. Trained over 80 epochs, the model achieves a precision of 94.8%, recall of 94.2%, and an mAP of 96.8%. Diagonal dominance in the matrix confirms high accuracy across various aircraft types (B-1B, B-52, C-130, E-3, F-16, KC-series, P-3C, SU-series, TU-series) and the background class. While minor off-diagonal entries indicate some confusion among visually similar aircraft variants, the overall high performance metrics validate Model 28's efficacy in distinguishing military aircraft from backgrounds and within diverse categories, underscoring its suitability for reconnaissance and defense applications.

The visual data presented in Figure 11 (ground-truth labels), Figure 12 (predictions from Model 11, Scheme I), and Figure 13 (predictions from Model 28, Scheme II) offer crucial insights into the efficacy of integrating the Global Attention Mechanism (GAM) and the impact of hyperparameter tuning for high-fidelity military aircraft detection, as further substantiated by the

accompanying training results from the best models in Table 9.

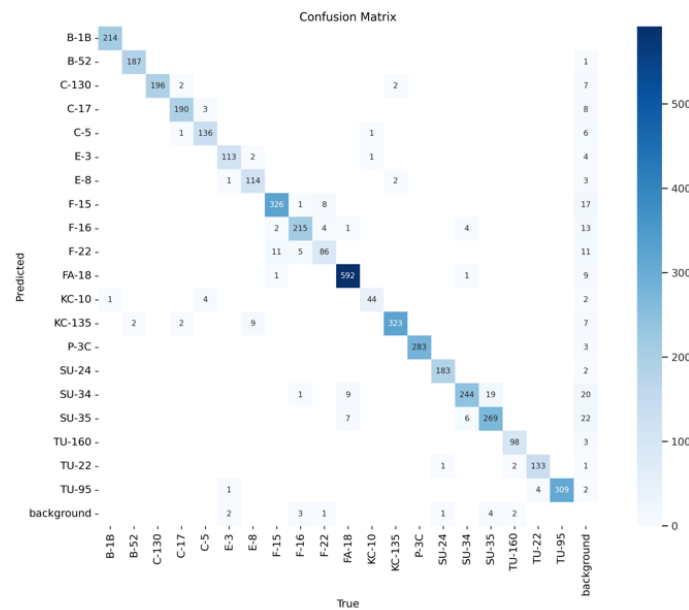


Figure 9. Confusion matrix of Model 11

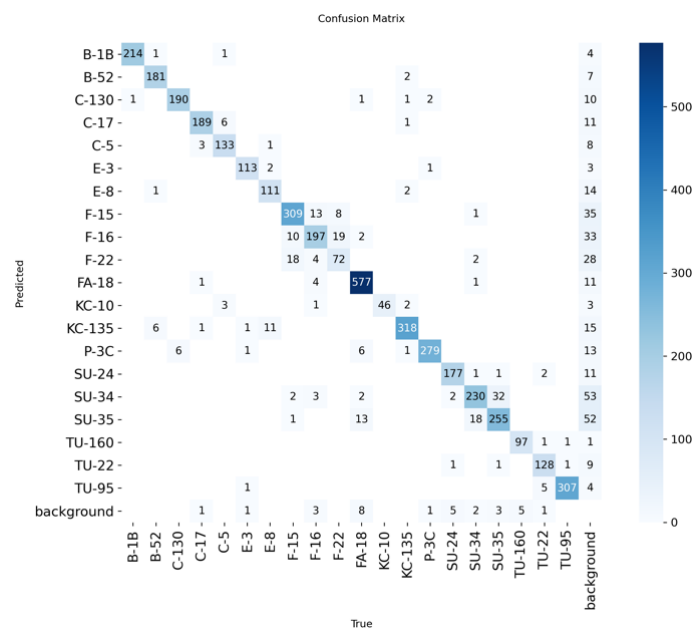


Figure 10. Confusion matrix of Model 28

Figure 11 serves as the indisputable benchmark, meticulously outlining the true class identities and precise bounding box coordinates for each military aircraft, forming the foundation for quantitative performance assessment. Comparing this to Figure 12, which showcases Model 11's performance (from Scheme I, trained with Adamax, lr 0.001, and GAM), we observe a generally competent detection capability. Model 11 effectively identifies most aircraft instances, and its quantitative metrics (P 96.7%, R 96.8%, mAP 98.5%) confirm a high level of overall accuracy. However, visual inspection reveals instances of inter-class confusion, particularly among visually similar aircraft types (e.g., subtle differentiation between F-15

and F-22, or KC-135 and C-130 variants). While localization is largely accurate, some bounding boxes might appear slightly looser or less precisely fitted compared to the ground truth, indicating a potential area for refinement in spatial accuracy.

Figure 13, presenting predictions from Model 28 (from Scheme II, also trained with Adamax and GAM, but with a higher lr 0.00417), illustrates a distinct shift in performance. Visually, Model 28 appears to exhibit superior fine-grained discrimination and often provides tighter, more accurate bounding box localizations for the specific samples shown. It seems more adept at resolving ambiguities between closely related aircraft, suggesting that its internal feature representations are

more robust and discriminative. This visual improvement could be attributed to the combined influence of GAM, which enhances the model's ability to focus on salient features, and the hyperparameter settings within Scheme II, leading to a potentially different optimization landscape.



Figure 11. Military aircraft labels



Figure 12. Military aircraft detection results using Model 11

However, a critical observation emerges when integrating these visual insights with the quantitative data from the table 9: despite the apparent visual enhancements in Figure 13 for specific instances, Model 28 (Scheme II) records a slightly lower overall performance (P 94.8%, R 94.2%, mAP 96.8%) compared to Model 11 (Scheme I). This discrepancy suggests that while Model 28 may excel in certain challenging scenarios or specific class detections as visually sampled, Model 11, with its lower learning rate (lr 0.001), achieved a more optimal generalization across the entire test set, leading to a higher aggregate

mAP. This highlights the intricate balance between visual perception of performance on select samples and the comprehensive statistical evaluation across a diverse dataset. Both models leverage GAM, affirming its general benefit in enhancing feature extraction. The difference in mAP between Model 11 and Model 28, despite both incorporating GAM, underscores the profound impact of hyperparameter tuning (specifically the learning rate in this context) and potentially subtle differences in their respective scheme configurations (e.g., dataset specifics, training methodologies) on the ultimate detection fidelity of YOLO11 for military aircraft.



Figure 13. Military aircraft detection results using Model 28

Table 10 presents a comparative analysis between the proposed Model 28 (Ours) and an existing method, DET-YOLO [3], highlighting significant performance distinctions. Model 28 demonstrates superior accuracy across all evaluated metrics: achieving a Precision (P) of 94.8% compared to DET-YOLO's 91.2%, a Recall (R) of 94.2% versus 89.3%, and a higher mean Average Precision (mAP) of 96.8% against 94.7%. Beyond accuracy, Model 28 exhibits a remarkable advantage in processing speed, operating at 212.77 Frames Per Second (FPS), which is substantially faster than DET-YOLO's 68.97 FPS. This comprehensive improvement in both accuracy and inference speed underscores the efficacy and practical deployability of Model 28, suggesting its architectural enhancements contribute to a more robust and efficient solution for the task at hand compared to the referenced DET-YOLO method.

Table 10. Comparison of DET-YOLO and Model 28

Methods	Epochs	P (%)	R (%)	mAP (%)	FPS
DET-YOLO[3]	80	91.2	89.3	94.7	68.97
Model 28 (Ours)	80	94.8	94.2	96.8	212.77

Based on the findings obtained in this research regarding the effectiveness of the Global Attention Mechanism and hyperparameter optimization on the YOLO11 model for military aircraft detection, there are

- vol. 38, no. 1, pp. 142–158, 2016, doi: 10.1109/TPAMI.2015.2437384.
- [17] L. Peng, H. Wang, and J. Li, “Uncertainty Evaluation of Object Detection Algorithms for Autonomous Vehicles,” *Automot. Innov.*, vol. 4, no. 3, pp. 241–252, 2021, doi: 10.1007/s42154-021-00154-0.
- [18] L. Shen, H. Tao, Y. Ni, Y. Wang, and V. Stojanovic, “Improved YOLOv3 model with feature map cropping for multi-scale road object detection,” *Meas. Sci. Technol.*, vol. 34, no. 4, p. 45406, 2023, doi: 10.1088/1361-6501/acb075.
- [19] X. Zhang et al., “Automatic learning for object detection,” *Neurocomputing*, vol. 484, pp. 260–272, 2022, doi: <https://doi.org/10.1016/j.neucom.2022.02.012>.
- [20] C.-T. Hu et al., “HARNESSING YOLO ALGORITHMS FOR EFFICIENT BREAST CANCER DETECTION IN MAMMOGRAPHY,” *J. Med. Imaging Radiat. Sci.*, vol. 54, no. 3, p. S7, Sep. 2023, doi: 10.1016/j.jmir.2023.06.025.
- [21] C. Meng, S. Liu, and Y. Yang, “DMAC-YOLO: A High-Precision YOLO v5s Object Detection Model with a Novel Optimizer,” in *2024 International Joint Conference on Neural Networks (IJCNN)*, 2024, pp. 1–8. doi: 10.1109/IJCNN60899.2024.10650645.
- [22] M. Zhang and L. Yin, “Solar Cell Surface Defect Detection Based on Improved YOLO v5,” *IEEE Access*, vol. 10, pp. 80804–80815, 2022, doi: 10.1109/ACCESS.2022.3195901.
- [23] X. Zhao et al., “G-YOLO: A Lightweight Infrared Aerial Remote Sensing Target Detection Model for UAVs Based on YOLOv8,” 2024. doi: 10.3390/drones8090495.
- [24] Z. Chen, L. Cao, and Q. Wang, “YOLOv5-Based Vehicle Detection Method for High-Resolution UAV Images,” *Mob. Inf. Syst.*, vol. 2022, no. 1, p. 1828848, Jan. 2022, doi: <https://doi.org/10.1155/2022/1828848>.
- [25] I. Amelia Dewi and M. A. Negara Ekha Salawangi, “High performance of optimizers in deep learning for cloth patterns detection,” *IAES Int. J. Artif. Intell. (IJ-AI)*, Vol 12, No 3 Sept. 2023, 2023, doi: 10.11591/ijai.v12.i3.pp1407-1418.
- [26] W. Cheng, R. Pu, and B. Wang, “AMC: Adaptive Learning Rate Adjustment Based on Model Complexity,” 2025. doi: 10.3390/math13040650.
- [27] N. Jegham, C. Y. Koh, M. Abdelatti, and A. Hendawi, *YOLO Evolution: A Comprehensive Benchmark and Architectural Review of YOLOv12, YOLO11, and Their Previous Versions*. 2025. doi: 10.13140/RG.2.2.15952.83201.
- [28] W. Zhu, X. Han, K. Zhang, S. Lin, and J. Jin, “Application of YOLO11 Model with Spatial Pyramid Dilation Convolution (SPD-Conv) and Effective Squeeze-Excitation (EffectiveSE) Fusion in Rail Track Defect Detection,” *Sensors (Basel)*, vol. 25, no. 8, Apr. 2025, doi: 10.3390/s25082371.
- [29] A. M. Moraes, L. F. Pugliese, R. F. Santos, G. B. Vitor, R. A. Braga, and F. R. Silva, “Effectiveness of YOLO Architectures in Tree Detection: Impact of Hyperparameter Tuning and SGD, Adam, and AdamW Optimizers,” 2025. doi: 10.3390/standards5010009.