Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.063270

ARTICLE

Improving Hornet Detection with the YOLOv7-Tiny Model: A Case Study on Asian Hornets

Yung-Hsiang Hung, Chuen-Kai Fan and Wen-Pai Wang*

Department of Industrial Engineering & Management, National Chin-Yi University of Technology, Taichung, 411030, Taiwan

*Corresponding Author: Wen-Pai Wang. Email: wangwp@ncut.edu.tw

Received: 10 January 2025; Accepted: 13 March 2025; Published: 16 April 2025

ABSTRACT: Bees play a crucial role in the global food chain, pollinating over 75% of food and producing valuable products such as bee pollen, propolis, and royal jelly. However, the Asian hornet poses a serious threat to bee populations by preying on them and disrupting agricultural ecosystems. To address this issue, this study developed a modified YOLOv7tiny (You Only Look Once) model for efficient hornet detection. The model incorporated space-to-depth (SPD) and squeeze-and-excitation (SE) attention mechanisms and involved detailed annotation of the hornet's head and full body, significantly enhancing the detection of small objects. The Taguchi method was also used to optimize the training parameters, resulting in optimal performance. Data for this study were collected from the Roboflow platform using a 640 × 640 resolution dataset. The YOLOv7tiny model was trained on this dataset. After optimizing the training parameters using the Taguchi method, significant improvements were observed in accuracy, precision, recall, F1 score, and mean average precision (mAP) for hornet detection. Without the hornet head label, incorporating the SPD attention mechanism resulted in a peak mAP of 98.7%, representing an 8.58% increase over the original YOLOv7tiny. By including the hornet head label and applying the SPD attention mechanism and Soft-CIOU loss function, the mAP was further enhanced to 97.3%, a 7.04% increase over the original YOLOv7tiny. Furthermore, the Soft-CIOU Loss function contributed to additional performance enhancements during the validation phase.

KEYWORDS: Computer vision; object detection; YOLOv7tiny; SE; SPD; Asian hornet

1 Introduction

Bees are a keystone species in global ecosystems and play an essential role in agricultural production. They contribute to over 75% of crop pollination, which is vital for producing many fruits, vegetables, and oilseed crops [1,2]. Additionally, bees provide products such as honey, beeswax, bee pollen, and royal jelly, all of which have extensive applications in nutrition, medicine, and the cosmetics industry.

In recent years, bee populations have faced numerous crises. Environmental pollution, pesticide use, climate change, and habitat destruction have caused dramatic declines in global bee numbers. In Taiwan, these declines have profoundly affected agricultural production and ecological balance. Furthermore, the spread of invasive species, such as the Asian hornet (*Vespa mandarinia*), presents an additional threat to native bees [3]. These hornets prey on adult bees and invade hives, disrupting colony structures and significantly impacting bee health and productivity.

Addressing these challenges and protecting bee populations while ensuring the sustainability of agricultural ecosystems requires effective monitoring and protective strategies for hornets. Traditional monitoring methods typically involve manual observation, which is time-consuming and inefficient. However, with the

rapid development of artificial intelligence and machine learning technologies, computer vision offers a novel solution for hornet identification [4,5]. This approach improves monitoring efficiency and provides accurate detection and timely responses to invasive species like hornets.

YOLOv7tiny is an efficient object detection model optimized for real-time applications, balancing accuracy, speed, and lightweight deployment. It achieves competitive detection accuracy while maintaining a compact model size (6–36 MB), making it ideal for embedded systems and edge devices. Compared to Faster R-CNN, which offers higher accuracy but operates at only 5–15 FPS, YOLOv7tiny runs significantly faster at 100–200 FPS. RetinaNet and EfficientDet provide improved precision but are larger and slower, while Transformer-based models like DETR achieve superior accuracy but suffer from extremely slow inference speeds (1–10 FPS). YOLOv7tiny's single-shot detection architecture allows real-time processing, making it particularly useful for applications such as drones, surveillance, and robotics. Unlike attention-based models, it optimally balances detection performance and computational efficiency. While some frameworks provide higher accuracy, YOLOv7tiny's speed and efficiency make it the best choice for real-time object detection in constrained environments. Overall, it is a practical, lightweight, and high-speed solution for small object detection in real-world applications.

This study employed advanced deep-learning techniques to enhance detection efficiency and accuracy. A modified YOLOv7tiny model was selected, incorporating space-to-depth (SPD) convolution and squeeze-and-excitation (SE) attention mechanisms. These mechanisms enhanced the model's ability to identify small targets, such as hornets, and improved accuracy through specific annotations of their heads and full bodies without significantly increasing computational complexity. SPD-Conv enhances the detection of small objects by restructuring feature maps to retain high-resolution spatial details. This method ensures that fine-grained features essential for identifying hornets remain preserved during downsampling. SE Attention dynamically recalibrates feature importance across channels, allowing the model to assign greater focus to hornet-specific characteristics, such as body shape and coloration. The Soft-CIOU loss function was also applied to optimize the model's validation process, allowing it to adapt to complex environmental conditions. This hornet recognition system provides an innovative approach that not only aids in bee conservation but also serves as a valuable technological tool for ecosystem research and conservation. Fig. 1 presents the architecture of the YOLOv7tiny model.

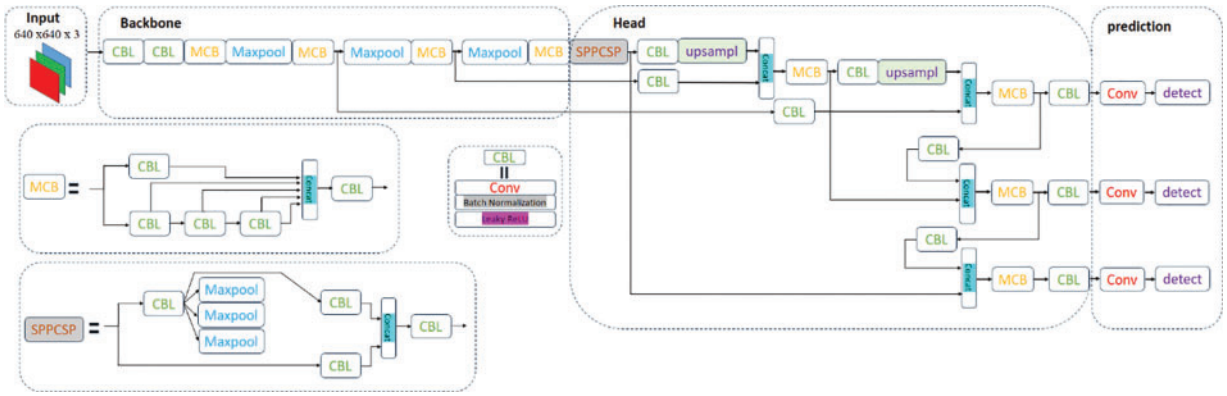

Figure 1: YOLOv7tiny architecture diagram

YOLO is chosen for its real-time object detection capability, offering high accuracy and efficiency. Unlike two-stage detectors like Faster R-CNN, YOLO's single-shot approach ensures faster detection while maintaining competitive accuracy, crucial for hornet detection in dynamic environments. YOLOv7tiny

balances accuracy and computational efficiency, making it ideal for drones and IoT-based monitoring systems in agriculture. Compared to attention-based models, which enhance feature representation but are computationally expensive, YOLOv7tiny integrates space-to-depth (SPD) and squeeze-and-excitation (SE) mechanisms to improve small object detection without high computational cost. These modifications allow fine-grained detection of hornets in cluttered scenes. YOLOv7tiny outperforms transformer-based models with faster inference, fewer parameters, and better suitability for embedded systems. The integration of SPD and SE mechanisms enhances small object detection, making YOLOv7tiny highly effective for real-world deployment.

This study focuses on modifying YOLOv7tiny to develop an effective image recognition system for detecting the Asian hornet (*Vespa mandarinia*). Previous research has primarily explored the threat posed by this species to bees and the beekeeping industry, emphasizing physical and biological methods to protect bees from hornet attacks [6]. However, the use of image recognition technology for hornet detection remains limited. Current technologies face challenges in detecting small objects; thus, this study enhances YOLOv7tiny's capability to detect small targets, such as hornets, by incorporating SPD and SE attention mechanisms. The model's accuracy and reliability are further improved by employing the Soft-CIOU loss function, while additional hornet head annotations validate the effectiveness of small object detection. Finally, the Taguchi method was used to determine optimal training parameters, maximizing training efficiency. Collectively, these four methods optimize the model.

The venom of the Asian hornet (*Vespa mandarinia*) can cause severe allergic reactions and may pose a risk of fatal outcomes. This study aims to develop a hornet recognition system using advanced image recognition technology. The system is intended for use in two primary scenarios: (1) to effectively assist beekeepers in defending against hornet attacks, thereby mitigating the threat to bees, and (2) to address the potential risks hornets pose to human health and safety.

2 Related Works

2.1 YOLOv7tiny

YOLO is a groundbreaking object detection system that was first introduced in 2016 [7]. Its primary innovation is the simplification of the object detection process into a single network computation. Unlike traditional object detection methods, YOLO integrates object localization and classification into a single, rapid scanning process, significantly enhancing detection speed. YOLO has diverse applications, including rapid object recognition and tracking in autonomous driving systems, crop and pest detection in agriculture, cancer detection and drug identification in healthcare, and object detection and classification in remote sensing. Additionally, it is used in security systems, surface inspection in manufacturing, traffic applications, wildlife detection and monitoring, and robotics. The YOLO series has undergone several iterations, with each version improving detection accuracy, speed, and efficiency. From the groundbreaking innovations of YOLOv1 to the latest YOLOv7, the series has continuously evolved to meet advancing technological demands and a wide range of application scenarios. For a detailed explanation of the YOLO recognition system flowchart, please refer to Fig. 1 in [8].

YOLOv1 introduced an innovative approach to object detection by executing the entire process in a single network pass, which is fundamentally different from previous methods that required multiple passes using sliding windows and classifiers. The YOLOv1 architecture consists of 24 convolutional layers and two fully connected layers, which are used to predict bounding box coordinates and probabilities [9]. Fig. 2 illustrates the basic architecture of YOLO.

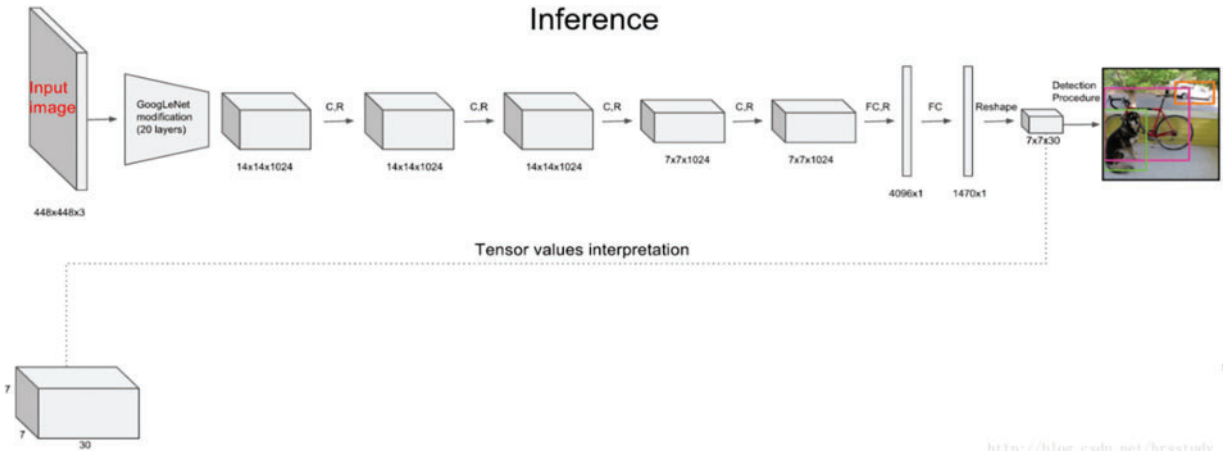

Figure 2: Basic architecture of YOLO

YOLOv2, which was released later, introduced several modifications to the original design, including the use of batch normalization on all convolutional layers, removing dense layers to adopt a fully convolutional architecture, and using anchor boxes for bounding box prediction. Additionally, DBSCAN was used to identify optimal anchor boxes. These enhancements significantly improved YOLOv2's performance [10]. YOLOv3 introduced a more extensive architecture and refined bounding box and class predictions. It also incorporated spatial pyramid pooling (SPP) and multi-scale prediction, further enhancing the model's accuracy and flexibility [11]. YOLOv4 focused on optimizing the trade-off between training strategies and inference cost, integrating the latest techniques and strategies into an efficient architecture. This version maintains high performance while prioritizing real-time detection speed [12].

YOLOv7 marked significant advancements in real-time object detection technology, offering greater accuracy and speed than its predecessors. It has been effectively applied to the real-time detection of players, soccer balls, and their movement scenarios, demonstrating its enhanced capability in dynamic environments [13]. A comprehensive comparison of YOLO performance, including accuracy and computational efficiency, is available in Fig. 1 of [13].

YOLOv7, the most recent object detector, has made significant improvements in speed, accuracy, and performance over its predecessors [13]. Its advancements are especially evident in dynamic and challenging environments, making it a preferred option for various practical applications. While YOLOv7 and YOLOv7tiny are two different object detection models, YOLOv7 achieves more precise object detection in complex environments but requires higher computational demands. In contrast, YOLOv7tiny is more lightweight and faster, making it ideal for simple applications and limited hardware. Under optimal conditions, the accuracy of both models is comparable. However, in challenging environments, such as poor lighting or when objects are extremely close or far away, YOLOv7 outperforms YOLOv7tiny. Thus, a model should be selected based on specific application scenarios, hardware resources, and performance needs [13]. YOLOv7 focuses on optimizing the training process through innovations like model reparameterization and dynamic label assignment, which enhance object detection accuracy while ensuring efficiency. It is also compatible with various hardware platforms, including edge and cloud GPUs, expanding its range of applications. By introducing these techniques, YOLOv7 achieves high accuracy without sacrificing inference speed, offering multiple configurations to cater to diverse requirements in real-time object detection.

2.2 SPD and SE Attention Mechanisms

2.2.1 SPD-Conv

Existing CNNs often struggle with low-resolution images or small objects. This challenge is attributed to the widespread use of strided convolution and pooling layers in CNN architectures, which result in the loss of detailed information and poor feature representation [14]. To address this, the authors propose a new CNN building block: SPD-Conv [15]. SPD-Conv integrates a space-to-depth (SPD) layer and dilated convolution, which can replace traditional strided convolution and pooling in most CNN architectures. The SPD layer downsamples the feature map while retaining all information, while the dilated convolution reduces the Number of Channels while preserving the detail. The researchers tested SPD-Conv on two representative computer vision tasks: object detection and image classification. Applying SPD-Conv to YOLOv5 and ResNet showed that this method showed significant performance improvements on tasks involving low-resolution images and small objects compared to existing deep-learning models. Fig. 3 presents the SPD architecture flowchart when scale = 2.

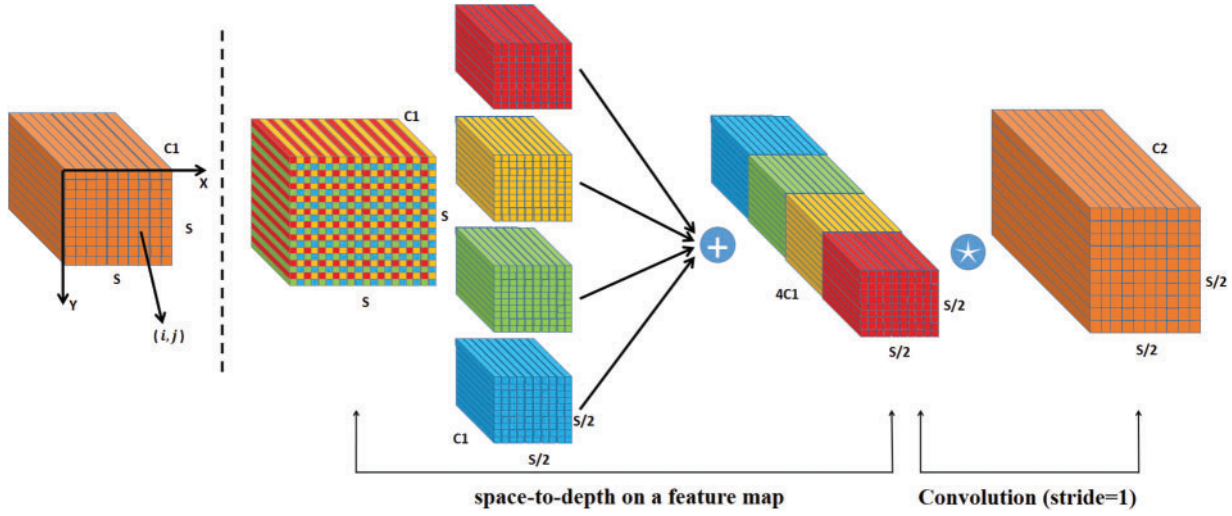

Figure 3: SPD architecture flowchart

2.2.2 SE Attention

The paper <Squeeze-and-Excitation Networks> introduces a novel neural network architecture called SE [16,17]. This architecture enhances the feature recognition capabilities of CNNs by dynamically recalibrating the feature responses of each channel to enhance overall performance. This is achieved by learning the interdependencies between channels, enabling the model to apply significant weighting to various features and better capture essential information. In the squeeze stage, SE uses global average pooling to spatially compress the output of each convolution layer, creating a channel descriptor that captures information from the entire feature map. This descriptor captures the global distribution of each channel and provides crucial contextual information for the excitation stage. In the excitation stage, SE employs a learned gating mechanism consisting of a small neural network with two fully connected layers, which uses the descriptors to generate weights for each channel. These weights scale the original convolution features, enhancing important feature responses and suppressing irrelevant ones [18].

Another notable feature of the SE block is its lightweight design. While it enhances the model's functionality, the increase in parameters and computational complexity is minimal, making the SE block

suitable for high-performance large models and applications requiring lower computational resources, such as those on mobile devices or edge computing devices. Fig. 4 illustrates the SE architecture flowchart.

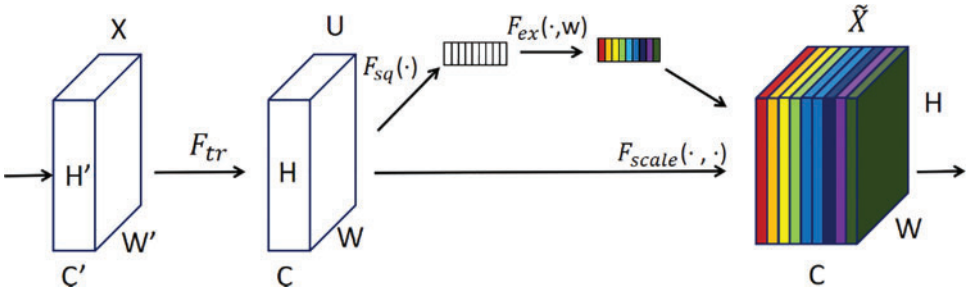

Figure 4: SE architecture flowchart

While Structural Attention enhances long-range object relationships, it primarily benefits scene-based detection tasks rather than small object detection. Its main drawbacks in this context are Computational Overhead and Limited Improvement for Small Objects. In fact, SPD and SE were chosen over Structural Attention due to their superior performance in preserving spatial details and computational efficiency, making them ideal for real-time hornet detection.

The selection of SPD convolution and SE attention mechanisms in this study is grounded in their well-documented advantages in small object detection. SPD convolution, as demonstrated by [14,15], effectively restructures feature maps to retain high-resolution spatial information, ensuring that small objects remain distinguishable even after downsampling. This technique has been widely applied in tasks requiring precise localization, proving its ability to enhance feature retention and mitigate the loss of fine details during the convolution process.

Similarly, the SE attention mechanism, validated in [16–18], dynamically adjusts the importance of each feature channel, enabling the model to focus on critical object attributes while suppressing background noise. The effectiveness of SE in improving detection accuracy has been extensively studied, with previous works highlighting its capability to amplify discriminative features and refine model predictions. These findings align with the objectives of this study, reinforcing the rationale for integrating SPD and SE mechanisms into the proposed model. By leveraging these enhancements, the model achieves superior detection accuracy while maintaining computational efficiency, making it well-suited for real-world applications requiring small object recognition.

2.3 Soft-CIOU Loss Function

In 2023, Ji et al. proposed the Soft-CIOU loss function, which modifies the traditional complete intersection over union (CIOU) loss function to optimize it specifically for small object detection. The CIOU loss function evaluates the similarity between predicted bounding boxes and ground truth boxes, factoring in the overlap area, center point distance, and aspect ratio of the bounding boxes. The key innovation of the Soft-CIOU loss function is the introduction of an aspect ratio weight factor and the application of a square root operation to the Euclidean distance. These modifications are designed to improve sensitivity and accuracy in small object detection [19].

The aspect ratio weight factor ensures that the loss function emphasizes the shape and size of small objects, which is crucial in small object detection. Due to their small size, small objects are more likely to

be overlooked during detection. This weight factor helps the model better capture and learn the features of small objects.

Meanwhile, applying a square root to the Euclidean distance between the center points of the bounding boxes increases the loss function's sensitivity to positional differences in small objects. The traditional CIOU loss function uses the squared distance directly, which may be sufficient for larger objects but is not sensitive enough for small objects. By applying the square root, the exaggeration of distance is reduced, allowing the model to learn the precise positioning of small objects more accurately [19].

2.4 Optimize Training Parameter Adjustments Using the Taguchi Method

When training deep learning models for object detection, the selection of training parameters significantly impacts the detection performance of the model. This method evaluates the impact of different parameters through a designed experimental plan to identify the optimal configuration. In the past, the learning rate has significantly affected training effectiveness in machine learning. A high learning rate can make the model unstable, while a low learning rate may slow down the training process. In contrast, the YOLO model uses optimizers and decay parameters for optimization. Optimizers (such as SGD and Adam) determine how to update the model's weights, while learning rate decay gradually reduces the learning rate to ensure stable convergence as the model approaches the optimal solution. Combining these techniques enhances the detection performance of the YOLO model, achieving higher accuracy and stability. The core of the Taguchi method lies in obtaining high-quality and stable results with fewer experimental trials. The method typically includes three stages: System Design, Parameter Design, and Tolerance Design.

In practical applications, the Taguchi method first identifies the most critical factors in a specific engineering problem through a series of predetermined screening experiments. By designing interaction experiments, it evaluates the interrelationships among these factors. Finally, a main experiment is conducted to determine the optimal combination of these factors for process optimization. This method emphasizes integrating quality factors during the design phase rather than conducting quality inspections after production. A key tool in the Taguchi method is the orthogonal array design, which systematically explores combinations of different factor levels, thereby reducing the number of experiments and increasing efficiency. Additionally, this method uses the signal-to-noise ratio (S/N ratio) to evaluate experimental outcomes, helping researchers identify factor settings that can resist noise and maintain stable performance.

3 Methods

3.1 Research Design

This study developed an improved YOLOv7tiny model-based hornet detection system to address the threat posed by Asian hornets to Taiwan's bee populations and agricultural ecosystems. During the data collection phase, a dataset from Roboflow was used to train images. Due to varying resolutions across datasets, preprocessing was applied prior to use, and images from multiple angles were included. Image processing techniques were also employed to enhance feature visibility, aiming to improve the precision of subsequent analyses and computational speed. The YOLOv7tiny model was then enhanced by integrating SPD and SE attention mechanisms, which significantly improved its detection capabilities for small hornet targets. The Taguchi method was applied to identify optimal training parameters, and the Soft-CIOU Loss Function was used to optimize the training process further, enhancing the model's performance in complex environments. Given that the parameters affect overall detection efficacy, experiments were conducted using the Taguchi method to determine the best training parameter factors. The evaluation and testing phases employed metrics such as precision, recall, F1 score, and mAP to comprehensively assess model performance.

The final results demonstrate that the improved YOLOv7tiny model outperforms the original in hornet detection, offering a new technological solution for bee conservation efforts. Fig. 5 presents the research framework flowchart.

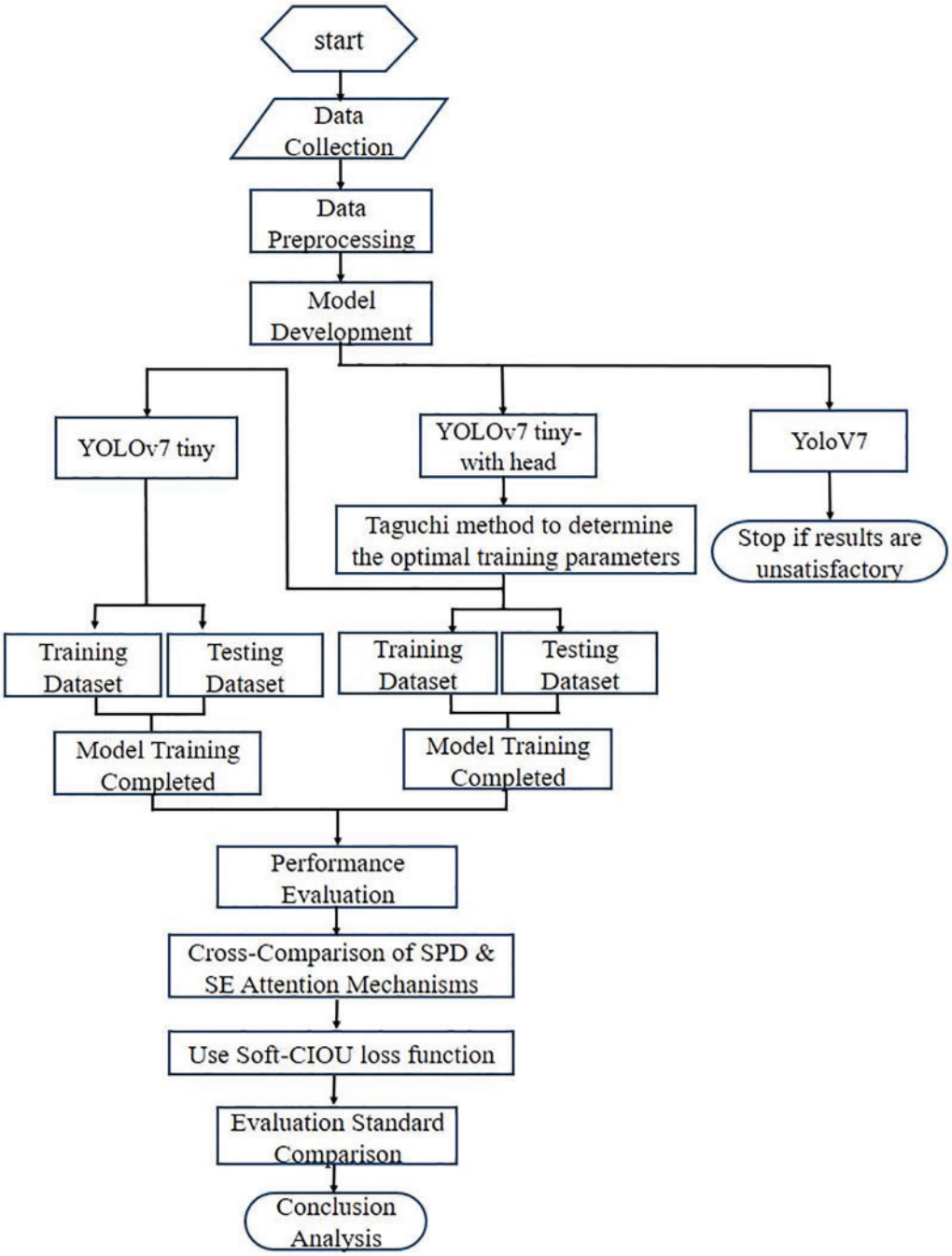

Figure 5: Research framework flowchart

3.2 Datasets

During the data collection phase, this study used data from Roboflow as the basis for the experiment, selecting 541 images of hornets and 302 images of bees. To address data imbalance and diversify the backgrounds, additional images of bees from various angles were included, with some images rotated at different orientations. This resulted in a total of 459 bee images and 1397 annotations, which included 694 bees and 703 hornets. The dataset is available for download from Roboflow's database. The data was split into training/validation and test sets for the experiment. The training data includes 567 annotations for bees and 567 for hornets, while the test data includes 127 bee annotations and 136 hornet annotations [20–22].

In future work, adding recognition features such as hornet head or eye characteristics is expected to enhance recognition rates in practice. Hornet heads possess unique and distinctive features, such as color patterns and eye shapes, which are crucial for improving the accuracy of the recognition system. Therefore, an additional label category was created for hornet heads, comprising a total of 705 annotations, including 567 for training data and 138 for validation data. This dataset includes images of hornet heads captured from various angles and in diverse environments and lighting conditions to ensure adaptability in recognition. Table 1 presents the data training, valiation and testing data splite. Despite efforts to balance the dataset, some imbalances remain that may impact model performance. The limited representation of nighttime and low-light images could reduce detection accuracy in dim environments, which will be mitigated by expanding the dataset with artificially enhanced and real-world nighttime images. Additionally, the dataset lacks diversity in non-hornet objects, as it contains few negative samples beyond bees and flying insects, necessitating the inclusion of more insect species to lower false positives. Lastly, the dataset is biased toward natural settings, potentially reducing accuracy in urban or indoor environments, which will be addressed by collecting more images near human dwellings and artificial structures.

Table 1: Data distribution

Dataset	Class	Training	Validation & Test	Total
YOLOv7tiny	Bees	567	127	1397
	Hornet	567	136	
YOLOv7tiny with head	Bees	567	127	2102
	Hornet	567	136	
	Hornet head	567	138	

Data Preprocessing

In this study, the images collected from Roboflow initially did not meet the unified specifications required for the experiment, specifically a resolution of 640 × 640 pixels. This inconsistency in dimensions could pose challenges for subsequent image processing and analysis. Therefore, image preprocessing was performed to resize all images to a standardized size, ensuring data consistency and enhancing model training efficiency. Given the imbalance in the number of hornet and bee images in the original dataset, this sample imbalance could cause the model to overfit to one category during training, potentially impacting the model's accuracy and generalization ability in real-world applications.

3.3 Model Development

This study primarily used the YOLOv7 model with modifications. After testing, the YOLOv7 model was found too large and not practical for real-world applications, and the results after training were not

satisfactory. Therefore, the development was later switched to YOLOv7tiny, a lightweight model that is faster and more suitable for simple applications and limited hardware. In favorable environments, the performance of YOLOv7tiny is similar to the original YOLOv7 model in terms of basic detection. The results indicate that for certain application scenarios, using a lightweight model saves computational resources while maintaining a comparable level of detection performance, making it suitable for embedded systems and mobile devices. The primary objects of detection are bees and hornets, both considered small objects in object detection. In the original YOLOv7tiny, the detection of small objects may not perform well. Hence, this study further incorporated SPD and SE attention mechanisms to improve small object detection. SPD enhances the model's sensitivity to small objects by performing multi-scale decomposition on feature maps, while the SE mechanism adjusts feature responses to improve the model's discriminative ability. Fig. 6 illustrates the integration of SPD and SE attention mechanisms.

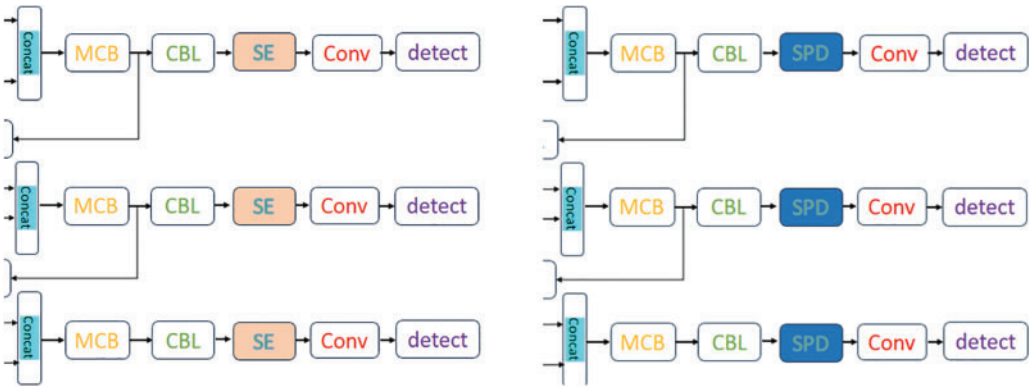

Figure 6: Add the SPD and SE attention mechanisms after the Head

3.4 Add Head Labels for the Asian Hornet

In the process of hornet recognition, this study proposes an innovative approach: enhancing the identification of hornet head features, particularly their unique characteristics such as color patterns and eye shapes. These features play a crucial role in improving the accuracy of the recognition system. During the data annotation phase, special attention was given to labeling hornet head features, allowing the model to more accurately identify this target species.

In the field of object detection research, focusing on recognizing specific parts of an object, such as the head or eyes, is crucial for improving the recognition rate of small objects or those in complex backgrounds. By refining the labels and training specifically for these features, even lightweight object detection models can achieve high detection accuracy. This strategy is significant for the accurate identification of hornets in practical applications, especially when dealing with small creatures or visually complex environments in natural settings. Through the implementation of these strategies, this study provides an effective methodology for the precise detection of hornets. Labeling was used as the annotation method in this study. Fig. 7 shows the XML display after annotation.

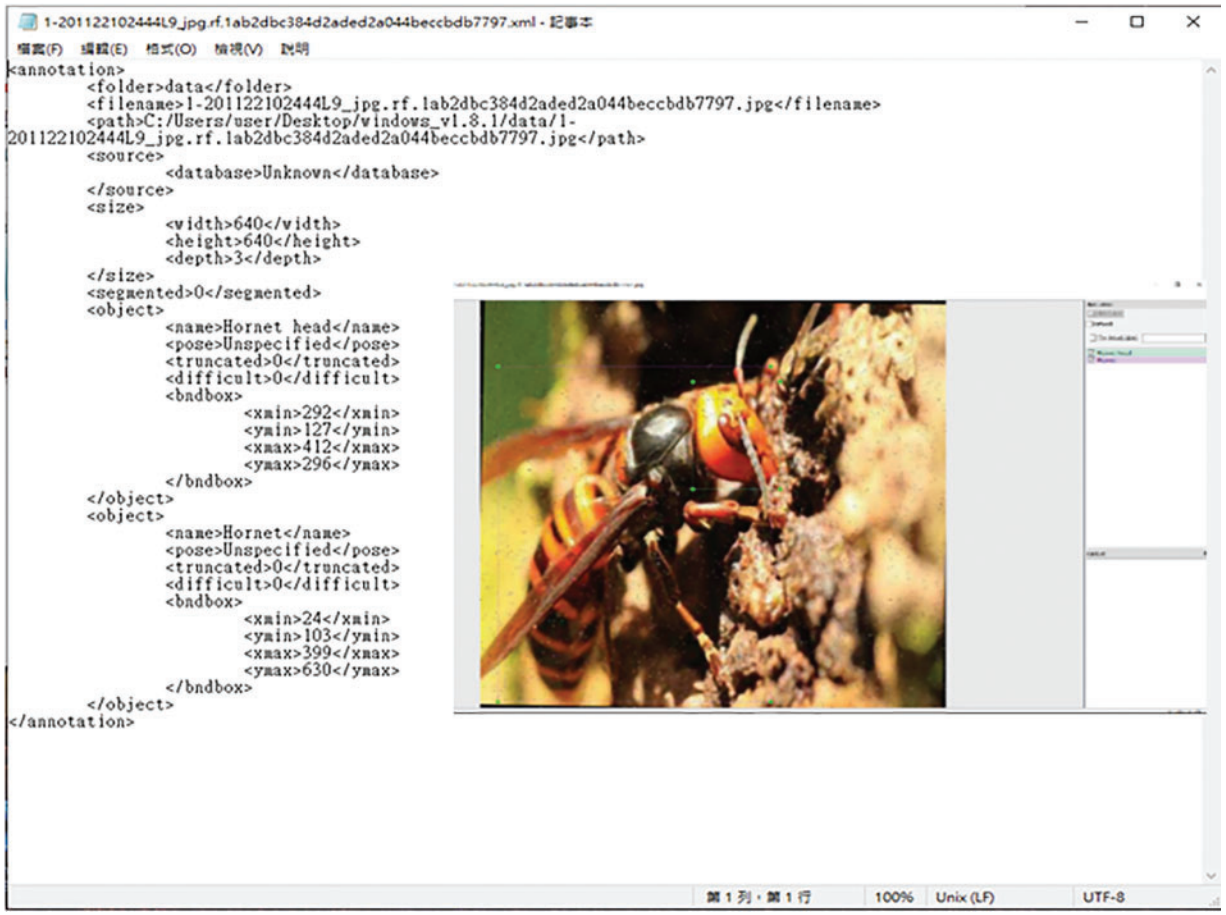

Figure 7: XML display after annotation

3.5 SPD

SPD-Conv stands for space-to-depth convolution, and its operation is mainly divided into four parts: sub-feature map slicing, sub-map formation, downsampling factor, and concatenation and transformation. The following formula describes how sub-feature maps are extracted from the original feature map Y during the SPD-Conv operation. Here, T represents the spatial dimensions of the feature map, while scale represents the downsampling factor. The feature map Y is sliced into multiple sub-feature maps according to a certain scale, with each sub-feature map obtained by extracting elements from the original feature map at specific intervals. These sub-feature maps are then concatenated along the channel dimension, thereby reducing the spatial size of the feature map while increasing the channel depth.

Extracting more abstract, high-level features from images is particularly useful for classification, recognition, and detection tasks. Specifically, YOLOv7tiny may not perform well in complex environments, such as poor lighting or when the object is too close or far away. After adding the SPD module, the detection results for small objects or in complex environments can be improved. When scale = 2, four sub-feature maps are obtained: $f_{(0,0)} f_{(1,0)} f_{(0,1)} f_{(1,1)}$. After concatenation, a new feature map is formed. Since each sub-feature map is downsampled from the original feature map in each direction by a factor of scale = 2, the spatial dimensions (height and width) of each sub-feature map are 1/2 of the original feature map dimensions. For example, if the original feature map is 640 × 640, it becomes 320 × 320 after SPD. The following are the

formulas for the four features. Fig. 8 shows the SPD diagram.

$$
\begin{aligned}
f_{(0,0)} &= Y\left[0{:}\,T{:}\,scale, 0{:}\,T{:}\,scale\right], \\
f_{(1,0)} &= Y\left[1{:}\,T{:}\,scale, 0{:}\,T{:}\,scale\right], \\
f_{(0,1)} &= Y\left[0{:}\,T{:}\,scale, 1{:}\,T{:}\,scale\right], \\
f_{(1,1)} &= Y\left[1{:}\,T{:}\,scale, 1{:}\,T{:}\,scale\right]
\end{aligned} \tag{1}
$$

Y = Original feature map; T = Spatial dimension of the feature map; *scale* = Downsampling factor.

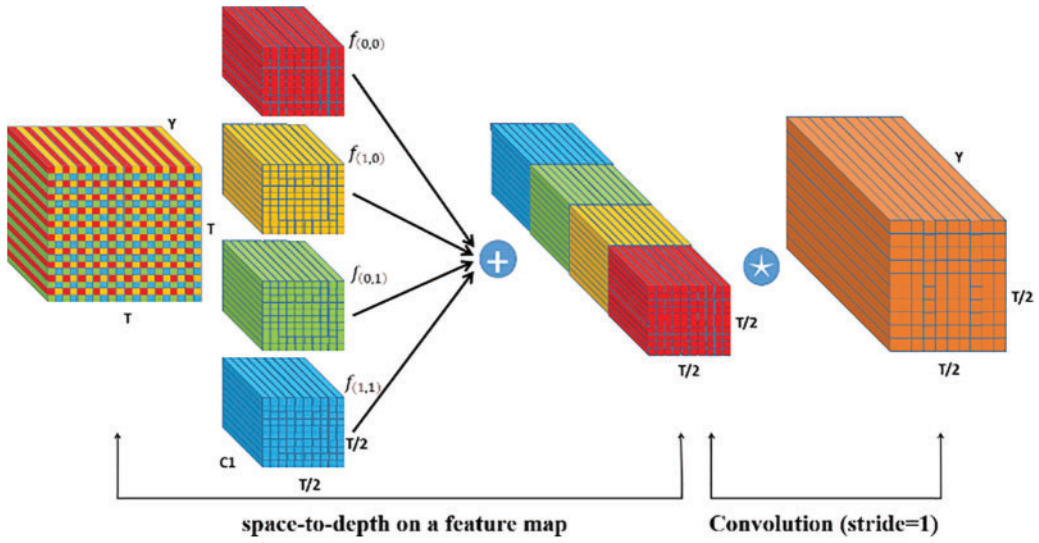

Figure 8: SPD diagram

3.6 Taguchi Method to Identify the Optimal Training Factors

This study also employed the Taguchi method to identify the experimental factors that significantly impact the YOLOv7tiny model's performance. By designing a series of experiments, an in-depth analysis was conducted on how different parameters affect the performance of the bee and hornet detection model based on the YOLOv7tiny algorithm. In the initial stage, this study referred to previous research to select several parameters that are considered crucial to model performance, including batch size, weight initialization, optimizer, and data ratio. These parameters were incorporated into the Taguchi method experimental design to systematically evaluate their specific impacts on model performance. This method allows for the precise identification of the parameters most critical for improving detection accuracy and model performance, thereby providing strong guidance for optimizing the model configuration. Through these experiments, this study aims to maximize the detection accuracy for bees and hornets while maintaining the model's performance.

In the Taguchi method, quality characteristics refer to the numerical values used to measure engineering problems, such as dimensions, weight, and temperature. These characteristics represent the performance indicators that are the focus of the product or process. The ideal function represents the expected target setting for these quality characteristics. For example, if a particular characteristic is intended to remain at a constant ideal value, this situation is referred to as nominal-the-best. If the goal is to maximize the value, it

is termed larger-the-better. Conversely, if a smaller value is preferred, it is called smaller-the-better. These target settings help define the direction for improving the quality of a product or process.

The S/N ratio is an indicator that measures the strength of the signal relative to the background noise level and serves as an important method for quality assessment. In quality management, a higher S/N ratio indicates better product or process quality. The Taguchi method employs three types of S/N ratio calculation formulas, each corresponding to different quality objectives: nominal-the-best, larger-the-better, and smaller-the-better. These formulas involve statistical concepts such as sample size (n), standard deviation (S), and mean value (y). This analysis helps quantify the variability and consistency of quality. The following formulas explain the calculation.

$$\text{larger-the-better S/Nratio: S/N} = -10\log\left(\frac{\sum_{i=1}^{n}\frac{1}{y_i^2}}{n}\right) \tag{2}$$

$$\text{smaller-the-better S/Nratio: S/N} = -10\log\left(\frac{\sum_{i=1}^{n}y_i^2}{n}\right) = -10\log\left(\overline{y}^2 + S_n^2\right) \tag{3}$$

$$\text{nominal-the-best S/Nratio: S/N} = -10\log\left(\frac{S^2}{\overline{y}^2}\right) \tag{4}$$

In conducting quality control experiments using the Taguchi method, an experimental orthogonal array is employed to arrange the experimental design. This table is constructed based on the number of selected factors and their possible interactions, aiming to efficiently combine different levels of factors to systematically explore their overall impact on quality. After completing these experimental combinations, the impact of each combination on product quality is evaluated through data analysis, particularly by calculating the S/N ratio. In this study, [A, B, C, D] represent batch size, weight initialization, optimizer, and data ratio, respectively. Table 2 presents the L8 orthogonal array.

Table 2: L8 Orthogonal array

EXP	A	B	C	D
1	1	1	1	1
2	1	1	2	2
3	1	2	1	2
4	1	2	2	1
5	2	1	1	2
6	2	1	2	1
7	2	2	1	1
8	2	2	2	2

3.7 Add Soft-CIOU Loss Function

The innovation of the Soft-CIOU loss function lies in the introduction of aspect ratio weighting factors and the application of the square root operation to the Euclidean distance to the center point of the bounding box. These two modifications enhance the sensitivity and accuracy of small object detection. First, the introduction of the aspect ratio weighting factor ensures that the loss function places greater emphasis on

the shape and size of small objects, enabling the model to more accurately learn and capture the features of these small objects, thereby improving detection accuracy.

Secondly, applying the square root operation on the Euclidean distance to the center point of the bounding box is designed to increase the sensitivity of the loss function to positional differences in small objects. In the traditional CIOU loss function, the square of the center point distance is directly used in the calculation, which may be effective for larger objects but less sensitive for small objects. The square root operation reduces the amplification effect of distance differences, making the model more accurate in locating small objects and preventing location bias. The Soft-CIOU loss function is a modification of the traditional CIOU loss function, primarily aimed at improving the accuracy of small object detection. The following formula represents the components of the CIOU formula.

$$\mathrm{CIOU} = \mathrm{IOU} - \frac{p^2\left(b, b_{gt}\right)}{c^2} - \alpha v \tag{5}$$

$p^2\left(b, b_{gt}\right)$ = The squared Euclidean distance between the centers of the predicted and ground truth boxes.

c = The diagonal length of the smallest box enclosing both the predicted and ground truth boxes.

v = A measure of aspect ratio consistency.

α = A factor for adjusting the aspect ratio term.

In the Soft-CIOU modification, the Euclidean distance term is adjusted by taking the square root to reduce its influence, particularly for small objects. This adjustment makes the loss function more sensitive to small positional variations. The following formula represents the components of Soft-CIOU.

$$\mathrm{Soft-CIOU} = \mathrm{IOU} - \frac{\sqrt{p^2\left(b, b^{gt}\right)}}{c^2} - av \tag{6}$$

4 Experiments and Results

The operating system used in this experiment is Windows 10, and the network model was built using Python 3.8 and PyTorch 2.1.1. The images were sourced from the Roboflow platform, which provided images representing hornets and bees as the basis for the experimental data. The input size is 640 × 640 × 3. The system processor used in this study is an Intel(R) Core(TM) i7-9700 CPU @ 3.00 GHz, paired with an NVIDIA GeForce RTX 3080 graphics card. Table 3 presents the experimental setup.

Table 3: Experiment configuration

	Item	Specification
Software	Operating System	Wimdows10
	CUDA	11.0
	cuDNN	8.0
	IDE	VS Code 1.84.2
	Programming Language	Python 3.8
Hardware	Processor (CPU)	NVIDIA GeForce RTX 3080
	Graphics Card (GPU)	Intel(R) Core(TM) i7-9700 CPU @ 3.00 GHz

4.1 Data Preprocessing

In this study, the Roboflow platform was selected as the data source, and images representing hornets and bees were chosen as the basis for the experimental data. Considering the imbalance in the number of hornet and bee images in the original dataset, several measures were taken to enhance the diversity and balance of the dataset. These measures included adding bee images with different backgrounds and images of bees taken from various angles. To further increase the diversity of the data, image rotation was applied to simulate various observation angles that might be encountered in natural environments, helping to avoid bias during the training process.

To address this imbalance, data augmentation techniques were applied, including adding bee images with different backgrounds and increasing the diversity of bee images. This strategy helps balance the dataset and improves the model's ability to recognize target species under various environments and angles. These steps are expected to enhance the model's performance in real-world applications. Fig. 9 illustrates the results after data preprocessing.

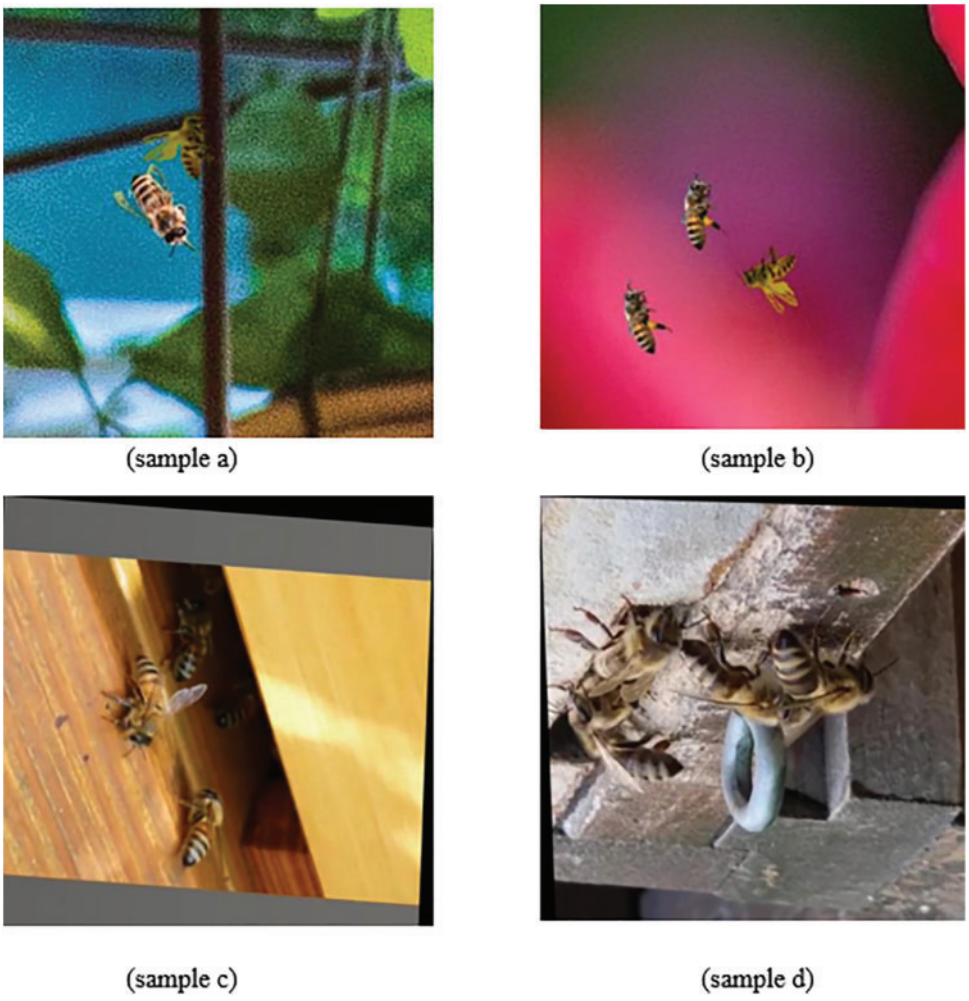

Figure 9: Preprocessing results for hornet and bee image dataset

4.2 Taguchi Parameter Setting

The Taguchi method, initially developed for industrial optimization, is widely used in engineering and deep learning for its efficiency and robustness in hyperparameter tuning. Unlike Grid Search, which tests all parameter combinations, and Bayesian Optimization, which refines results iteratively but may converge to local optima, Taguchi requires fewer experiments while maintaining accuracy. The advantages of the Taguchi method in hyperparameter tuning include: 1. Efficient Experimental Design—Taguchi systematically selects key parameter combinations using orthogonal arrays, significantly reducing computational cost while ensuring broad coverage of potential optimal solutions. 2. Robustness and Stability—Proven in industrial applications for minimizing variation, Taguchi enhances deep learning by improving model stability and ensuring better generalization across datasets and environmental conditions. and 3. Optimized Factor-Level Analysis—Unlike trial-and-error methods, Taguchi employs signal-to-noise (S/N) ratio analysis to prioritize hyperparameters that have the greatest impact on performance, refining tuning in a structured way.

To optimize hyperparameters systematically, we used the Taguchi experimental design with four key factors. Batch size affects training efficiency and generalization, with 8 and 16 chosen to balance stability and computational feasibility. Optimizer selection compared SGD and Adam, where SGD provides stable convergence but needs careful tuning, while Adam adapts learning rates dynamically for noisy datasets. Weight initialization impacts training speed and gradient stability, so we tested pretrained **vs.** random initialization, with pretrained weights improving feature extraction. Data split ratio determines model generalization, and we tested 70:30 vs. 80:20, balancing training data size against overfitting risks. The Taguchi method's orthogonal array ensures efficient evaluation with minimal experiments. SGD and Adam were selected based on prior studies in object detection. Batch sizes of 8 and 16 were chosen for their balance of noise control and efficiency. Pretrained weights were expected to enhance small object recognition performance. The final model configuration was determined by analyzing results from the experimental trials.

This study employs the Taguchi experimental design to identify the optimal parameters. The data used includes those with added head labels, and after identifying the best training parameters, they are applied to each improved model to enhance the recognition accuracy. The study defines recognition accuracy as the quality characteristic, with the ideal goal being "larger-the-better". The selected experimental factors include batch size, weight initialization, optimizer, and data ratio. Each factor is assigned two representative levels based on previous experience. An L8 orthogonal array is used to design the experiment, examining the effects of individual factors and their potential interactions with other factors on recognition accuracy. Table 4 presents the experimental factor settings. These parameters were selected based on prior research and practical constraints, ensuring maximum performance improvement while minimizing computational cost. The Taguchi hyperparameters experimential show as Fig. 10.

Table 4: Experimental factor settings

Factors	Explain
A	Batch size
B	Weight initialization
C	Optimizer
D	Data ratio

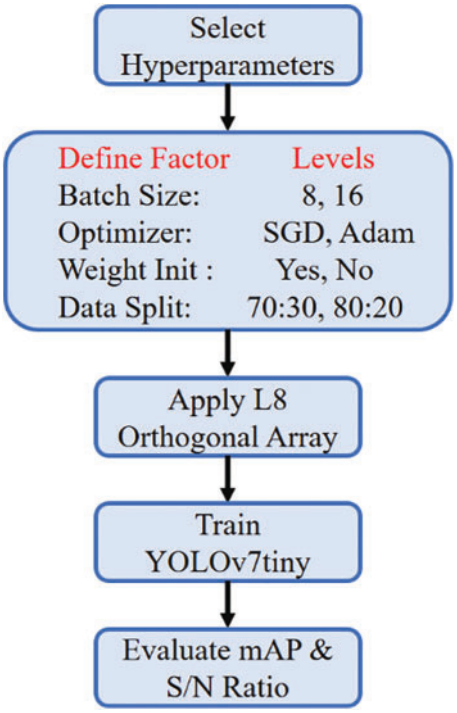

Figure 10: Taguchi hyperparameter experimental procedure

4.2.1 Taguchi Method L8 Main Experiment

To precisely identify the key factors for optimizing model performance, this study selected several important factors and levels based on prior expertise and experimental data. These factors were then evaluated in the main experiment using the L8 orthogonal array in the Taguchi method. This approach focused on evaluating four parameters considered to have a decisive impact on model training and recognition accuracy. These parameters include batch size [A], weight initialization [B], optimizer algorithm [C], and data split ratio [D]. Due to hardware limitations and based on past research, batch sizes of 8 and 16 were chosen as the experimental benchmarks. Table 5 presents the main experiment factor and level settings. Each factor is assigned two levels to examine their independent and interaction effects on model performance. The application of the L8 orthogonal array systematically analyzes and evaluates these parameters within a limited number of experiments, thereby identifying the optimal parameter combination to further improve the model's accuracy and reliability. Table 5 presents the setup of the L8 orthogonal array as the main experiment factor.

Table 5: Main experiment factor and level setting table

Factor	Explain	Level 1	Level 2
A	Batch size	8	16
B	Weight initialization	No	Yes
C	Optimizer	SGD	Adam
D	Data ratio	70:30	80:20

4.2.2 Taguchi Method L8 Experiment Results

This section presents the results of the Taguchi experiment using the L8 orthogonal array. To achieve optimal recognition accuracy, this study employed the Taguchi method to select appropriate model parameter levels and identify the key factors that most impact accuracy through systematic experiments. By analyzing Taguchi experiments 1 through 8 from the L8 orthogonal array, the factor settings were further refined to ensure the model achieves the best stability and performance. Table 6 presents the results of the Taguchi experiment conducted using the L8 orthogonal array.

Table 6: Accuracy of mAP@0.5 in Taguchi-L8 experiment

Taguchi	All	Bees	Hornet	Hornet head
Exp#1	69.9	77.1	55.9	76.7
Exp#2	82.5	85.6	78.3	83.5
Exp#3	79.1	80.1	74.8	82.3
Exp#4	94.6	95	93.8	95
Exp#5	76.1	81.1	68.8	78.4
Exp#6	79.1	74.9	79.1	83.3
Exp#7	89.2	92	86.8	88.7
Exp#8	83.5	87.4	80.2	82.8

4.2.3 Analysis of Key Factors in L8

The experimental results are analyzed by calculating the S/N ratio. In the Taguchi method, SNR is an indicator used to evaluate quality characteristics, helping to identify factors that significantly impact experimental outcomes. By calculating the average SNR for each factor at different levels, the influence of each factor can be determined. In this assessment, factors with higher SNRs have a greater effect on the experiment, while those with lower SNRs have a lesser impact. This analysis aids in identifying the optimal combination of factors to ensure the quality of the experimental results. By comparing quality characteristic charts, the factors that most significantly affect model performance can be identified, providing insights for further experimentation.

This study used the L8 orthogonal array to select key factors for experimentation. During the analysis, line plots were created to show the level values of each factor [A, B, C, D] and their calculated ranges. Based on the range sizes, the factors were ranked accordingly. Table 7 presents the S/N ratio response for each factor, showing the relative impact levels and their ranking .

Table 7: Factor response table for S/N ratio

Quality characteristic	A (batch size)	B (weight)	C (optimizer)	D (data ratio)
Level 1	38.175	37.70	37.87	38.34
Level 2	38.257	38.73	38.56	38.08
Range	0.082	1.03	0.69	0.26
Rank	4	1	2	3
Significant	No	Yes	Yes	No

After a series of experimental tests, Fig. 11 shows that factors B and C significantly impact the results, indicating that adjustments to these factors are crucial for enhancing system performance. Although factors A and D have a smaller impact, they still play an important role in the selected optimal combination. Based on the experimental results, the optimal factor combination was determined as [A2, B2, C2, D1], which improves recognition performance while reducing variability. These findings reinforce the effectiveness of the Taguchi method in practical applications. Table 8 presents the final experimental factor selection.

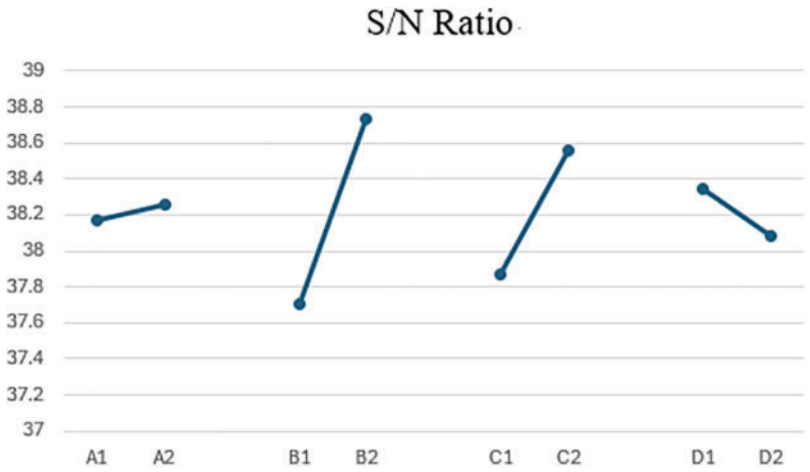

Figure 11: Factor response plot for S/N ratio

Table 8: Final experimental factor selection

Factor	Explain	Level 1	Level 2
A	Batch size	8	16
B	Weight initialization	No	Yes
C	Optimizer	SGD	Adam
D	Data ratio	70:30	80:20

4.3 Evaluation Indicators

This study utilized the data provided by the confusion matrix to evaluate the model's performance. Key performance indicators were calculated to determine the best model, including Precision, Recall, F-measure, AP (Accuracy Precision), and mAP (mean Average Precision). The confusion matrix is an essential tool for assessing the performance of classification models. It displays the correlation between the model's predicted and actual results in a tabular format. Each row in the matrix corresponds to an actual class, while each column represents a predicted class, with each cell showing the level of match between them. The confusion matrix includes four basic elements: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), which reflect the correspondence between the model's predictions and the actual conditions. By analyzing these data, this study explores which model performs better in handling specific tasks.

Recall: Recall refers to the ratio of correctly detected positive samples (True Positives) to the total actual positive samples (True Positives plus False Negatives). The calculation formula is:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

Precision: Precision refers to the ratio of the correctly detected positive examples (True Positives, TP) to all the positive examples detected by the model (True Positives, TP, and False Positives, FP). The formula is as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}$$

F1-score: The F1 score denotes the harmonic mean of precision and recall, providing a single metric that balances both. The formula is as follows:

$$\text{F1} = 2\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recal}} \tag{9}$$

Accuracy Precision (AP): AP is a standard metric for evaluating the performance of object detection models across different classes and thresholds. It calculates the precision (AP) for each class at different recall thresholds, which is equivalent to the area under the precision-recall curve. The formula is as follows:

$$\text{AP} = \int_0^1 \text{p}(\text{r})\text{dr} \tag{10}$$

Mean Average Precision (mAP): mAP averages the AP values across all classes to obtain the mAP. This calculation provides a comprehensive evaluation of the model's overall detection ability, whether for single-class or multi-class object detection tasks. The formula is as follows:

$$\text{mAP} = \frac{1}{\text{N}} \sum_{\text{i}=1}^{\text{N}} \text{AP}_\text{i} \tag{11}$$

N = Number of classes.

4.4 Result Analysis

After identifying the best training parameters using the Taguchi method, the model was trained with the modified parameters and tested on the data with added head labels. Once the training was complete, the best model was tested, and comparisons of different combinations helped to select the optimal model. YOLOv7 was not used in this study due to its long training time and large number of parameters, so it is not included in the results. The conclusion is divided into two parts to assess whether adding head labels has an impact.

4.4.1 YOLOv7tiny

Table 9 shows that adding the SPD module to YOLOv7tiny achieves an mAP of 98.7%, making it the best model, with an improvement of approximately 8.58% compared to the original YOLOv7tiny, demonstrating significant performance gains. Although the model size slightly increased to 36.6 MB, the training time did not increase significantly, indicating that this model is practical. In theory, adding Soft-CIOU should result in an improvement. However, in this experiment, it did not show a significant effect. This may be because Soft-CIOU is more suitable for multi-object detection, as mentioned in later experiments. Overall,

YOLOv7tiny-spd outperforms the original YOLOv7tiny by a large margin. The Soft-CIOU outperforms CIOU and DIOU by enhancing small object sensitivity and bounding box shape prioritization. Unlike CIOU and DIOU, Soft-CIOU applies a square root transformation to the Euclidean distance between bounding box centers, reducing the impact of large distance variations and improving localization accuracy. Additionally, it introduces an aspect ratio weighting factor, allowing the model to better adapt to small object shapes, a feature not explicitly considered in CIOU and DIOU. Experimental results (Table 9) show that Soft-CIOU provided a consistent improvement in detection accuracy, contributing to a 7.04% mAP increase when combined with SPD attention.

Table 9: YOLOv7tiny model results before and after modification

Model	mAP (%)	Bees AP (%)	Hornet AP (%)
YOLOv7tiny	90.9	92.7	92.7
YOLOv7tiny_SPD	98.7	99.2	98.2
YOLOv7tiny_SPD_SE	98.6	98.9	98.3
YOLOv7tiny_SPD_SE with Soft-CIOU	97.7	98.2	97.1
YOLOv7tiny_SE	90.2	93.9	86.5
YOLOv7tiny_SE with Soft-CIOU	90.7	94.7	86.7
YOLOv7tiny with Soft-CIOU	89.9	95.6	84.2
YOLOv7tiny_SPD with Soft-CIOU	98.3	99.0	97.5

4.4.2 YOLOv7tiny-with Head

According to Table 10, when head labels are added, there is a slight improvement of 5.39% compared to the original YOLOv7tiny (see Table 9), indicating that adding head labels helps the model better identify small object detection targets. Further comparisons were made with the addition of SPD and SE modules. This experiment found that after adding head labels and using the SPD module and Soft-CIOU loss function, the best model was trained, resulting in an improvement of approximately 7.04%, demonstrating significant effects. The model size only slightly increased to 36.6 MB, and the training time did not significantly increase, indicating that this model is practical and feasible for real-world use. The experiment also found that adding the SE module did not have a noticeable effect on improving the model. This further confirmed that adding Soft-CIOU improved multi-object detection, but the effect was not very significant. It can be observed that after adding head labels, the overall accuracy improved from the original mAP of 90.9% to 95.8%, indicating that adding head labels leads to an improvement without significantly increasing the model size, making it easier to implement in practical applications.

Table 10: YOLOv7 tiny-with head model results before and after modification

Model	mAP (%)	Bees AP (%)	Hornet AP (%)	Hornet head AP (%)
YOLOv7tiny-with head	95.8	97.0	97.0	93.5
YOLOv7tiny-with head _SPD	96.4	99.0	97.8	92.4
YOLOv7tiny-with head _SPD_SE	96.5	97.7	96.9	94.9
YOLOv7tiny-with head _SPD_SE with Soft-CIOU	96.8	99.2	97.4	93.9

(Continued)

Table 10 (continued)

Model	mAP (%)	Bees AP (%)	Hornet AP (%)	Hornet head AP (%)
YOLOv7tiny-with head _SE	96.2	98.3	96.0	94.3
YOLOv7tiny-with head _SE with Soft-CIOU	96.2	97.8	96.9	93.8
YOLOv7tiny-with head with Soft-CIOU	95.1	97.9	93.0	94.4
YOLOv7tiny-with head _SPD with Soft-CIOU	97.3	99.2	97.7	95.1

To enhance the robustness and generalization ability of the proposed model, various data preprocessing and augmentation techniques were applied, including image flipping, background variation, and contrast adjustments. To further validate the model's robustness and generalization ability, an additional experiment was conducted using Gaussian Blur augmentation as a validation method. By training the model on blurred images, we aimed to assess its adaptability to complex environments where image clarity may be compromised.

As shown in Table 11, while the introduction of Gaussian Blur resulted in a slight decrease in overall accuracy (mAP: 95.8% to 95.4% for the baseline model, 97.3% to 96.8% for the improved model), the model maintained strong detection performance, demonstrating its ability to generalize effectively under challenging conditions. This suggests that combining head labels with SPD and Soft-CIOU not only improves small object detection but also enhances the model's robustness, ensuring reliable performance in real-world applications.

Table 11: Effect of Gaussian Blur on model generalization performance

Model	mAP (%)	Bees AP (%)	Hornet AP (%)	Hornet head AP (%)
YOLOv7tiny-with head	95.8	97.0	97.0	93.5
YOLOv7tiny-with head Gaussian Blur	95.4	96.1	94.5	95.5
YOLOv7tiny-with head _SPD with Soft-CIOU	97.3	99.2	97.7	95.1
YOLOv7tiny-with head _SPD with Soft-CIOU Gaussian Blur	96.8	98.4	96.8	95.2

4.4.3 Comparison of With and Without Head Labels

Adding head labels significantly improved the overall performance of the YOLOv7tiny model, particularly in metrics such as Average Precision, Bees AP, and Hornet AP, with substantial improvements observed. After adding head labels, the performance of the models became more stable, with no noticeable decline, and their stability was further enhanced. This provides a more efficient and accurate object detection solution for practical applications. Future research could further optimize this strategy by exploring additional label and feature combinations or adaptive label generation techniques to further enhance the model's performance

and generalization capabilities, aiming for optimal performance. Fig. 12 shows the performance comparison between models with and without head labels.

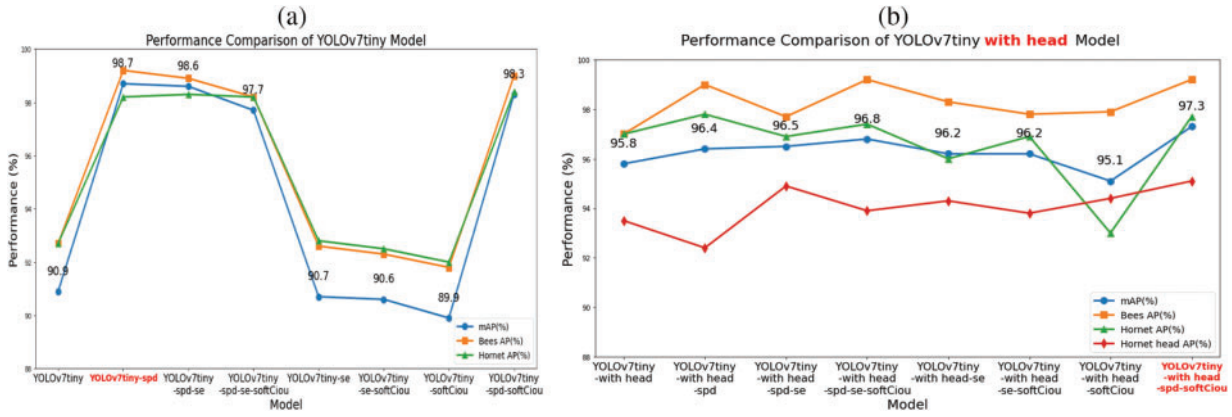

Figure 12: (**a**) Line chart without head labels. (**b**) Line chart with head labels

According to Table 12, adding the SPD module significantly improves the detection accuracy. Due to its large model size, the original YOLOv7 model is less practical for real-world applications and has longer training times, yielding unsatisfactory final results. Therefore, this study focused on the lighter YOLOv7tiny model. After adding the SPD module, significant improvements were observed compared to the original YOLOv7tiny, with mAP increasing from 90.9% to 98.7%, a total improvement of about 8.58%. Both precision and recall were notably enhanced, and the F1 score showed a remarkable increase, rising from 0.85 to 0.96, a total improvement of 12.94%.

Table 12: Performance results before and after modification for each model

Model	Class	AP	Precision	Recall	F1-Score Precision	mAP_0.5
YOLOv7	Bees	87	83.9	80.3	0.82	86.05
	Hornet	85.1				
YOLOv7Tiny	Bees	92.7	87.1	82.9	0.85	90.9
	Hornet	89.2				
YOLOv7Tiny_SPD	Bees	99.2	96.9	96.1	0.96	98.7
	Hornet	98.2				
YOLOv7Tiny with head	Bees	97	93.8	91.7	0.93	95.8
	Hornet	97				
	Hornet head	93.5				
YOLOv7Tiny-with head_SPD with Soft-CIOU	Bees	99.2	97.1	92.9	0.95	97.3
	Hornet	97.7				
	Hornet head	95.1				

Adding the hornet head labels led to a slight improvement compared to the original YOLOv7tiny, with mAP increasing from 90.9% to 95.8%, a total improvement of about 5.39%. Both precision and recall significantly improved, and the F1 score also increased slightly, rising from 0.85 to 0.93, a total improvement of 9.41%, which is a noticeable enhancement. In the Table 12 highlight, the effectiveness of the proposed modifications to the YOLOv7tiny model, demonstrating significant improvements in detection accuracy through the integration of SPD, head labeling, and Soft-CIOU loss. Below is a structured response summarizing the

findings and justifying the enhancements. The Space-to-Depth (SPD) module was introduced to enhance small-object feature extraction, which significantly boosted detection accuracy. Experimental results show as below:

1. mAP improved from 90.9% to 98.7% (+8.58%).
2. Precision and recall significantly increased.
3. F1-score improved from 0.85 to 0.96 (+12.94%).

This study demonstrates that SPD effectively retains spatial information, making the model more robust in detecting small and occluded hornets. To further improve classification accuracy, head labeling was added to enhance the model's ability to differentiate hornets from similar insects. The results show as below:

1. mAP increased from 90.9% to 95.8% (+5.39%).
2. Precision and recall improved.
3. F1-score increased from 0.85 to 0.93 (+9.41%)

While the performance gain from head labeling alone is not as significant as SPD, it still enhances feature representation, particularly when objects are partially occluded or in cluttered backgrounds. When all three enhancements were combined (SPD + Head Labeling + Soft-CIOU Loss), further improvements were observed:

1. mAP increased from 90.9% to 97.3% (+7.04%).
2. Precision and recall were significantly enhanced.
3. F1-score rose from 0.85 to 0.95 (+11.76%).

When the hornet head labels, the SPD module, and the Soft-CIOU loss function were combined, there was a slight improvement in multi-object detection, with mAP increasing from the original 90.9% to 97.3%, a total improvement of about 7.04%, demonstrating the effectiveness of our proposed enhancements. Both precision and recall showed significant improvements, the precision increased from 87.1% to 97.1% (a 11.4% improvement), indicating that the number of false positives (FPs) has significantly decreased, the recall improved from 82.9% to 92.9% (a 12.06% increase), meaning the model is now better at detecting all hornets in an image and the F1 score notably increased from 0.85 to 0.95, a total improvement of 11.76%. This metric is crucial because a high precision but low recall would mean the model is overly conservative (missing detections), while a high recall but low precision would mean the model is too permissive (many false alarms). Overall, the YOLOv7tiny-with head-spd outperforms the original YOLOv7tiny in terms of performance, making it suitable for a lightweight and efficient object detection model, as shown in Fig. 13. Our optimized YOLOv7tiny model significantly improves small object detection, certain challenging scenarios still lead to misdetections or reduced accuracy. The primary sources of error are related to lighting conditions, object occlusion, and class misclassification, which are common challenges in real-world object detection tasks.

In addition to evaluating the performance of the YOLOv7tiny model, this study also compared the model size before and after adding head labels to assess their impact on model accuracy and size. The results show that adding head labels significantly improved the average precision of most YOLOv7tiny models, increasing from 90.9% to 95.8%, an overall improvement of 5.39%. Furthermore, the impact on model size was minimal, with a substantial improvement in model performance without a significant increase in resource consumption. Fig. 14 illustrates the bubble chart comparison. To further analyze this, we conducted a comparative study between models trained with and without head labels, assessing their impact on small object detection and classification robustness. This study evaluated the model's performance across mAP, precision, recall, and F1 score under both conditions. The results are summarized in Table 13. The addition of head labels improved mAP by 5.39%, with precision increasing by 7.69% and recall improving by 10.61%.

These enhancements indicate that labeling the hornet head provides additional discriminative features that contribute to both better localization and classification.


(sample 1)


(sample 2)


(sample 3)

Figure 13: Three samples of multi-target detection for Asian hornets and Asian hornet heads

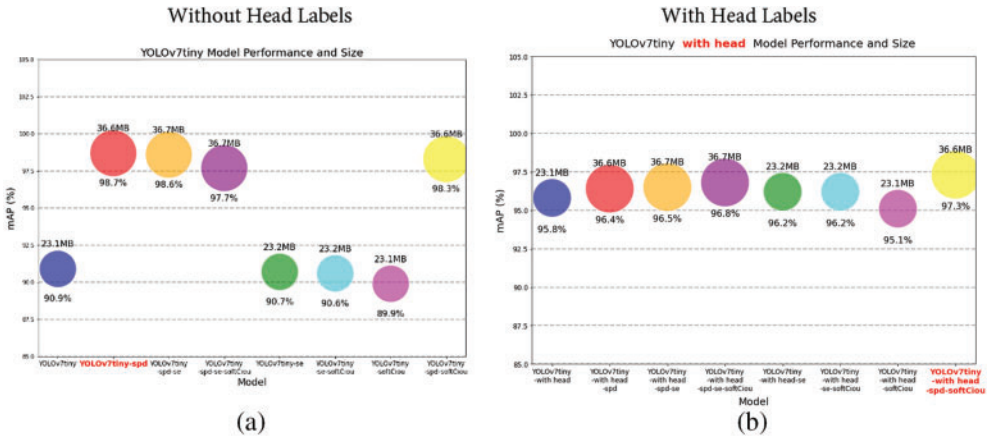

Figure 14: (**a**) Bubble chart without head labels. (**b**) Bubble chart with head labels

Table 13: Performance results with and without head labels

Model	Precision	Recall	F1 score	mAP_0.5
Without Head Labels	87.1	82.9	0.85	90.9
With Head Labels	93.8	91.7	0.93	95.8

5 Conclusions

This study developed an improved YOLOv7tiny model for real-time detection of hornet species to protect bee colonies. It incorporated SPD and SE attention mechanisms, the Taguchi method to find optimal training parameters, and the Soft-CIOU loss function to improve the recognition and accuracy of small targets such as hornets. The results indicated that incorporating the SPD attention mechanism without the hornet head label achieved the highest model mAP of 98.7%, an improvement of about 8.58% over the original YOLOv7tiny. When the hornet head label was added, the SPD attention mechanism and Soft-CIOU

loss function significantly improved the mAP to 97.3%, a 7.04% increase over the original YOLOv7tiny. This suggests that the Soft-CIOU loss function improves performance in both multi-target detection and small object detection. In conclusion, This study introduces a unique combination of three key modifications to YOLOv7tiny: 1. SPD-Conv: Improves feature retention for small object detection, boosting detection accuracy by up to 8.58% over the baseline YOLOv7tiny. 2. SE Attention: Enhances channel-wise feature importance, allowing better differentiation between hornets and background noise. and 3. Soft-CIOU Loss Function: Optimizes bounding box regression, improving detection precision, particularly for small, non-uniformly shaped objects. Additionally, a new hornet head labeling strategy was introduced, leading to a further 7.04% mAP improvement in our best model.

This study successfully applied and modified the YOLOv7tiny model, integrating the SPD attention mechanism to achieve high-accuracy recognition of hornets. Adding hornet head labels to the model significantly improved the detection of small objects, which is crucial for early warning and management of agricultural pests. The Taguchi method was utilized to systematically identify the optimal training parameters, enhancing model performance without requiring extensive computational resources. The key Parameters Optimized (using the L8 orthogonal array) reducing experimental trials while maximizing efficiency, Improved mAP and stability without increasing inference time and Enhanced model adaptability to varying environmental conditions. In brief, the study utilized the Taguchi method to systematically optimize the model's training parameters, ensuring its robustness and efficiency under various environmental conditions. This study is novel in its integration of SPD, SE, and Soft-CIOU while leveraging the Taguchi method for optimal hyperparameter tuning. Rather than using traditional grid search or Bayesian optimization, we employ the Taguchi experimental design, which systematically identifies the most impactful training parameters with minimal computational overhead.

Acknowledgement: The authors would like to thank the editors and reviewers for their detailed review and insightful advice.

Funding Statement: This research received no external funding.

Author Contributions: Conceptualization, Yung-Hsiang Hung, Chuen-Kai Fan and Wen-Pai Wang; Visualization, Yung-Hsiang Hung and Chuen-Kai Fan; Methodology, Yung-Hsiang Hung and Chuen-Kai Fan; Validation, Yung-Hsiang Hung and Chuen-Kai Fan; Formal analysis, Yung-Hsiang Hung and Chuen-Kai Fan; Investigation, Yung-Hsiang Hung and Chuen-Kai Fan; Writing, original draft preparation, Yung-Hsiang Hung and Chuen-Kai Fan; Writing, review and editing, Yung-Hsiang Hung and Wen-Pai Wang; Supervision, Yung-Hsiang Hung; Software, Yung-Hsiang Hung and Chuen-Kai Fan; Data curation, Yung-Hsiang Hung and Chuen-Kai Fan; Resources, Yung-Hsiang Hung and Chuen-Kai Fan. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data are contained within the article.

Ethics Approval: Not applicable.

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

References

1. Klein AM, Vaissière BE, Cane JH, Steffan-Dewenter I, Cunningham SA, Kremen C, et al. Importance of pollinators in changing landscapes for world crops. Proc R Soc B. 2007;274(1608):303–13. doi:10.1098/rspb.2006.3721.
2. Calderone NW. Insect pollinated crops, insect pollinators and US agriculture: trend analysis of aggregate data for the period 1992–2009. PLoS One. 2012;7(5):e37235. doi:10.1371/journal.pone.0037235.

3. Mattila HR, Otis GW, Nguyen LTP, Pham HD, Knight OM, Phan NT. Honey bees (*Apis cerana*) use animal feces as a tool to defend colonies against group attack by giant hornets (*Vespa soror*). PLoS One. 2020;15(12):e0242668. doi:10.1371/journal.pone.0242668.
4. Jordan MI, Mitchell TM. Machine learning: trends, perspectives, and prospects. Science. 2015;349(6245):255–60. doi:10.1126/science.aaa8415.
5. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436–44. doi:10.1038/nature14539.
6. Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data. 2021;8(1):53. doi:10.1186/s40537-021-00444-8.
7. Terven J, Cordova D. A comprehensive review of YOLO: from YOLOv1 to YOLOv8 and beyond. arXiv:2304.00501. 2023. doi:10.48550/arXiv.2304.00501.
8. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016 Jun 27–30; Las Vegas, NV, USA: IEEE; 2016. p. 779–88.
9. Redmon J, Farhadi A. YOLO9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017 Jul 21–26; Honolulu, HI, USA: IEEE; 2017. p. 7263–71.
10. Redmon J, Farhadi A. YOLOv3: an incremental improvement. arXiv.1804.02767. 2018. doi: 10.48550/arXiv.1804.02767.
11. Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: optimal speed and accuracy of object detection. arXiv.2004.10934. 2020. doi:10.48550/arXiv.2004.10934.
12. Gillani IS, Munawar MR, Talha M, Azhar S, Mashkoor Y, Sami Uddin M, et al. YOLOv7 performance comparison: a survey. Comput Sci Inf Technol. 2022;12:17–28. doi:10.5121/csit.2022.121602.
13. Wang CY, Bochkovskiy A, Liao HYM. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023 Jun 18–22; Vancouver, BC, Canada: IEEE; 2023. p. 7464–75.
14. Sharma N, Jain V, Mishra A. An analysis of convolutional neural networks for image classification. Procedia Comput Sci. 2018;132(2):377–84. doi:10.1016/j.procs.2018.05.198.
15. Sunkara R, Luo T. No more strided convolutions or pooling: a new CNN building block for low-resolution images and small objects. In: Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD); 2022 Sep 19–23; Grenoble, France. Cham, Switzerland: Springer Nature; 2022. p. 443–59.
16. Hu J, Shen L, Sun G. Squeeze-and-excitation networks. Proc IEEE Conf Comput Vis Pattern Recognit. 2018;7132–41. doi:10.1109/CVPR.2018.00745.
17. Hu J, Shen L, Albanie S, Sun G, Wu E. Squeeze-and-excitation networks. IEEE Trans. Pattern Ana. Mach Intell. 2020;42(8):2011–23. doi:10.1109/TPAMI.2019.2913372.
18. Wang J, Lv P, Wang H, Shi C. SAR-U-Net: squeeze-and-excitation block and atrous spatial pyramid pooling-based residual U-Net for automatic liver segmentation in computed tomography. Comput Methods Programs Biomed. 2021;208(3):106268. doi:10.1016/j.cmpb.2021.106268.
19. Ji SJ, Ling QH, Han F. An improved algorithm for small object detection based on YOLO v4 and multi-scale contextual information. Comput Electr Eng. 2023;105(1):108490. doi:10.1016/j.compeleceng.2022.108490.
20. Project-vrljw, hornet_only dataset, roboflow universe, roboflow [Internet]. 2023 [cited 2025 Mar 12]. Available from: https://universe.roboflow.com/project-vrljw/hornet_only.
21. Bees and hornets, bees dataset, roboflow universe [Internet]. 2023 [cited 2025 Mar 12]. Available from: https://universe.roboflow.com/bees-and-hornets-20yku/bees-twjei.
22. Si gn, insect_detection dataset, roboflow universe [Internet]. 2023 [cited 2025 Mar 12]. Available from: https://universe.roboflow.com/si-gn/insect_detection-qsh9u.