Sensors and Materials, Vol. 37, No. 7 (2025) 2935–2948
MYU Tokyo

S & M 4095

Enhancing YOLOv8 by Adding Global Attention Mechanism to Identify Targets in Complex Backgrounds

Ming-Te Chen,* Cheng-Hui Chen, Chun-Ting Lin, and Yi-Ying Chang

Department of Computer Science and Information Technology,
National Chin-Yi University of Technology, Taichung, Taiwan

(Received April 17, 2025; accepted June 30, 2025)

Keywords: ecological conservation, invasive species control, feature extraction, object detection

In recent years, the population of green iguanas in Taiwan has grown rapidly, causing significant damage to the local ecosystem and agricultural crops. Because their coloration closely resembles the natural environment, accurately detecting green iguanas remains a challenging task. We propose an enhanced You Only Look Once (YOLO)v8-based image recognition framework tailored for green iguana detection. By integrating a global attention mechanism into the Backbone of the YOLOv8 architecture and incorporating color feature weighting, the model's feature extraction capabilities are significantly improved. These enhancements allow for a more accurate and reliable identification of green iguanas in complex natural settings. The proposed method offers a fully automated detection solution that supports agricultural and environmental experts in developing effective management strategies to control the green iguana population, thereby mitigating their ecological and agricultural impacts.

1. Introduction

With the rapid advancement of technology, image recognition techniques have demonstrated significant value across various fields, including agriculture, industry, and medicine. Among these techniques, You Only Look Once (YOLO) has emerged as a highly efficient object detection framework, renowned for its speed and accuracy, making it widely adopted for real-time object detection tasks. With continuous iterations, the newer versions of YOLO have shown notable improvements in detection performance, establishing it as a crucial tool in modern object detection applications. Despite these advances, the application of image recognition technology in ecological conservation still faces many challenges. In recent years, green iguanas have had a profound impact on Taiwan's ecosystem. As an invasive species, they have caused various environmental problems. Their voracious feeding behavior has significantly reduced vegetation cover, destroying habitats of native species. Their rapid reproduction and strong adaptability have further threatened native wildlife by reducing available food resources and living space, even leading to starvation and population decline. Additionally, their burrowing activity damages farmland, reduces crop yields, and negatively affects the livelihoods of

*Corresponding author: e-mail: mtchen@ncut.edu.tw
https://doi.org/10.18494/SAM5696

ISSN 0914-4935

farmers. YOLO, as a one-stage object detection method, eliminates the need for multistage region proposal generation, offering fast and accurate detection capabilities. Applying YOLO to green iguana monitoring facilitates the effective tracking of their movements. However, under the pursuit of efficiency, YOLO's performance in complex or natural environments often suffers in terms of precision. To address this limitation, in this study, we enhance the YOLOv8 model by integrating the global attention mechanism (GAM) along with color feature weighting. The improved model, equipped with the modified GAM, shows a slight trade-off in accuracy but increases detection quantity in complex scenes. Furthermore, it effectively reduces false positives and false negatives in situations where the target and background share similar colors, thereby enhancing the model's detection robustness and stability.

2. Background

Green iguanas are an invasive species in Taiwan, and detecting them in complex natural environments presents significant challenges owing to their camouflage coloring and environmental similarity. To enhance target detection performance under such conditions, we integrate an attention mechanism into the YOLOv8 architecture, aiming to improve the stability and robustness of feature extraction and object recognition in complex backgrounds.

2.1 YOLO evolution

With the continuous advancement of visual imaging technology, its applications have expanded to various fields such as face recognition, object detection, medical image analysis, and image segmentation, becoming increasingly integrated into everyday life. The YOLO series, as a real-time object detection framework, has undergone numerous iterations from YOLOv1 to YOLOv8, each version bringing notable improvements to both performance and efficiency.

YOLOv1[(1)] was proposed by Redmon *et al.* in 2016. It divides the input image into X × X grids, where it only needs one forward propagation to complete the prediction. The recognition speed is very high and is especially suitable for the real-time recognition of large objects. However, the recognition effect of small objects and complex scenes is poor.

YOLOv2[(2)] improves upon the defects of YOLOv1, adding an anchor box function to improve the detection of objects of different sizes and using the Darknet-19 network architecture to improve the feature extraction capabilities of the model, thus improving the model's inference speed and accuracy. Furthermore, through multiscale training, the model can adapt to different image resolutions, and regularization is introduced to improve the stability of model training and reduce overfitting in model training.

YOLOv3[(3)] uses Darknet-53 in place of the Darknet-19 network architecture in YOLOv2 to enhance feature extraction. YOLOv3 combines the ResNet residual network to prevent gradient explosion or disappearance problems caused by deepening, and adds a new network, the feature pyramid network (FPN). Object detection is performed on the basis of feature maps of different sizes to improve the detection of small objects. However, the recognition accuracy of YOLOv3 will decrease when dealing with a large number of targets or complex scenes.

YOLOv4[4] was proposed to address the shortcomings of YOLOv3. The Backbone part extracts image features and combines Darknet-53 with cross-stage partial DenseNet (CSPNet) to form CSPDarknet-53. CSPDarknet-53 improves the learning capabilities of convolutional neural networks (CNNs). CSPNet allows the model to obtain richer gradient fusion information and reduce computing time. The Neck part fuses the obtained shallow and deep features using spatial pyramid pooling (SPP) and the path aggregation network (PANet). The SPP part enhances the multiscale feature capturing by the model, and its functionality is to process objects with clear differences in size. PANet adds a layer of series connection to the FPN to fuse the original features to improve the detection of small objects while avoiding feature degradation. The Head part converts the features after the fusion by the Neck part to generate the final detection results, including bounding box prediction, target classification, and confidence evaluation. YOLOv4 continues to be used with YOLOv3.

YOLOv4 also uses Bag of Freebies (BoF) and Bag of Specials (BoS). BoF includes CutMix, Mosaic data enhancement, and DropBlock. CutMix combines parts of two different pictures to improve the model's adaptability to different scenes. Mosaic data enhancement combines four different images to improve the model's learning of targets at different scales. DropBlock partially shields the feature map, allowing the model to learn a partially occluded object. BoF improves the model accuracy without increasing the cost of inference time. BoS includes CSPNet and Mish activation functions, which slightly increase the inference time and improve the accuracy.

YOLOv5[5] uses the PyTorch framework to facilitate development and use by users. YOLOv5 is an optimized version of YOLOv3 and includes functions such as adaptive anchor frame calculation, Mosaic data enhancement, and Focus structure. The adaptive anchor frame can automatically calculate the anchor frame size suitable for the dataset to improve the accuracy of detection. The Focus structure uses the Focus layer in the Backbone. The Backbone extracts image features and cuts the image through Focus and increases the number of channels, reducing the extraction time of feature maps.

YOLOv8[6] has an improved network architecture to address the shortcomings of previous generations of models and inherits the advantages of YOLOv5 for light weight, maintaining accuracy and reducing inference time. YOLOv8 has model versions of different sizes (YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x) that users can use in accordance with different scenarios. Among them, YOLOv8n is the smallest model (lightweight model) in the YOLOv8 series, having the lowest amount of calculation, although the accuracy of identification will be reduced; hence, it is suitable for use on devices with limited computing.

A 640 × 640 RGB image is input to the Backbone for target feature extraction. The Backbone consists of three main components: the convolution + batch normalization + sigmoid linear unit (SiLU) (CBS) module, C2f module, and spatial pyramid pooling fast (SPPF). The CBS module comprises a standard 2D convolutional layer (Conv2D), batch normalization (BN), and the SiLU activation function. The SiLU function introduces nonlinearity during feature extraction, thereby enhancing the representational capacity of the extracted features and improving the model's overall detection performance.

The C2f module first processes the initial features through a CBS block. A portion of these features is temporarily held for fusion via Concat, while the remainder is passed to a Split

operation that divides the feature map into two branches. One branch bypasses further processing and is directly sent to Concat for subsequent feature fusion, preserving shallow features. The other branch undergoes multilayer bottleneck processing to perform deep feature extraction, enabling the model to capture semantic information at different scales. After this, the shallow and deep features are fused via Concat, followed by another CBS layer to refine the fused features and enhance the model's recognition capability.

In addition, the improved SPP module—SPPF—is incorporated to enhance feature extraction efficiency and accelerate model inference. This module integrates multiscale contextual information by applying a sequence of max-pooling operations. The process begins with a convolutional layer that reduces the dimensionality of the input feature map. The reduced feature map is then passed through several max-pooling layers with different receptive field sizes to capture spatial features at multiple scales. Finally, another convolutional layer restores the feature map to its original dimensions, facilitating effective multiscale feature fusion. This design improves the model's ability to detect targets in complex environments by capturing both local and global contextual information more efficiently.

The Neck module integrates both FPN and the path aggregation network (PANet) to achieve comprehensive multilevel feature fusion. FPN enhances detection capabilities by transmitting deep semantic features from the Backbone to shallower layers, thereby constructing a hierarchical multiscale feature pyramid. This top-down fusion strategy allows the model to effectively detect objects of various sizes by combining high-level contextual information with low-level spatial details. However, the unidirectional nature of FPN may lead to the loss of fine-grained details during downward propagation, potentially hindering the detection of small or subtle targets. To overcome this limitation, PANet is employed to introduce a bottom-up information flow. By channeling precise localization cues from lower layers back to higher layers, PANet complements FPN's top-down semantics with enhanced spatial accuracy. This bidirectional feature fusion ensures that both detailed positional information and abstract semantic features are preserved across all scales. As a result, the model's overall detection accuracy is significantly improved, particularly in complex environments and scenarios involving small object detection.

The Head module is responsible for the final stage of target detection and comprises three detection layers, each corresponding to multiscale fused feature maps. These layers are designed to detect objects of various sizes by extracting features at different spatial resolutions. High-resolution feature maps are dedicated to identifying smaller objects, whereas low-resolution maps are optimized for detecting larger ones. This multiscale strategy enables the model to simultaneously detect targets across a broad size spectrum, thereby enhancing detection accuracy and robustness. YOLOv8 adopts a decoupled head architecture in which the processes of bounding box regression and classification are separated. This decoupling mitigates the risk of mutual interference during training, allowing each task to be optimized more effectively. To further improve bounding box prediction accuracy—especially for small targets—distribution focal loss (DFL) is employed. DFL models the bounding box regression task as a probability distribution, enabling more precise localization. In addition to the conventional anchor-based detection paradigm, YOLOv8 integrates an anchor-free detection framework, which directly

regresses object positions and sizes without relying on predefined anchor boxes. This dual detection strategy—combining both anchor-based and anchor-free approaches—enhances the model's flexibility and detection capability across diverse object scales. As a result, YOLOv8 demonstrates superior performance in detecting small objects and maintaining high accuracy in complex and cluttered environments.

2.2 GAM

GAM[7] mainly captures the global features in the image and relies on the global context to adjust each position in the feature map so that the model can clearly identify the target and focus on important positions. Furthermore, the semantics of the overall image are extracted by calculating the global average, and dynamic weight allocation is performed on the basis of local features, thereby achieving accurate global contextual attention guidance. GAM as shown in Fig. 1 mainly implements the weighted processing of features through two attention mechanisms, namely, channel attention and spatial attention. First, provide the input feature map $\boldsymbol{F}_1$ to the channel attention $\boldsymbol{M}_c$ to generate the channel-weighted weight $\boldsymbol{M}_c(\boldsymbol{F}_1)$. Multiply the channel-weighted weight $\boldsymbol{M}_c(\boldsymbol{F}_1)$ with the feature map $\boldsymbol{F}_1$ element by element to obtain the channel-weighted feature map $\boldsymbol{F}_2$. Then, provide the channel-weighted feature map $\boldsymbol{F}_2$ to the spatial attention $\boldsymbol{M}_s$ to generate the spatial weight $\boldsymbol{M}_s(\boldsymbol{F}_2)$, and multiply the channel-weighted feature map $\boldsymbol{F}_2$ and the spatial weight $\boldsymbol{M}_s(\boldsymbol{F}_2)$ element by element to obtain the feature map $\boldsymbol{F}_3$ of the final target position.

$$\boldsymbol{F}_1 = \boldsymbol{M}_c\left(\boldsymbol{F}_0\right) \otimes \boldsymbol{F}_0 \tag{1}$$

$$\boldsymbol{F}_2 = \boldsymbol{M}_s\left(\boldsymbol{F}_1\right) \otimes \boldsymbol{F}_1 \tag{2}$$

2.2.1 Channel attention

Figure 2 shows that the spatial information of each channel is first compressed into a single global feature value through the global pooling layer, the global semantic information of each channel is next extracted, and then the compressed features are processed through the multilayer

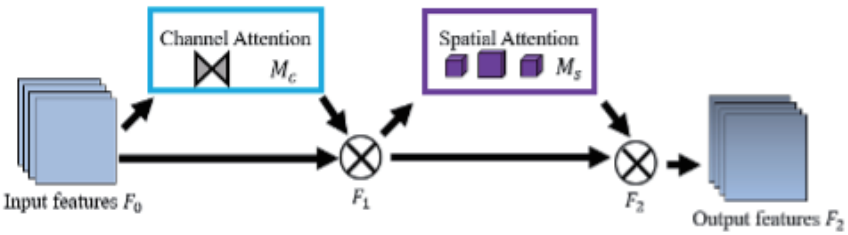


Fig. 1. (Color online) Global attention mechanism.

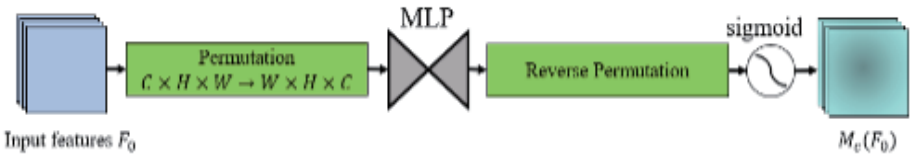


Fig. 2. (Color online) Channel attention.

perceptron (MLP). Linear transformation maps the generated channels to attention weights. These weights are then aligned with the original feature map through the inverse arrangement operation, and the weights are compressed to between 0 and 1 through the sigmoid activation function, which is applied to each channel of the original feature map. This processing can strengthen important channels, suppress interference from unimportant channels, and help the model focus on important features.

2.2.2 Spatial attention

After channel attention processing, spatial attention, as shown in Fig. 3, is then used to process the spatial position in the feature map. Spatial attention uses two layers of 7 × 7 convolutions to capture spatial contextual information in images. The first layer of convolution is responsible for expanding the detection range and capturing a wide range of spatial information in the feature map. The second layer of convolution is responsible for refining spatial features and will generate attention weights corresponding to spatial positions. After passing these weights through the sigmoid activation function, the values are compressed to between 0 and 1 and applied to the spatial position of the feature map. Such processing can emphasize local features in the image, suppress interference from the background or irrelevant targets, and help the model focus on local features in the image.

3. Proposed Method

3.1 Network architecture

We use the YOLO target detection method, which has the advantages of high immediacy and accuracy, so the YOLOv8 model is chosen as the basis for modification.

We take this thesis[8] as the building block reference and adopt the 8.2.90 version of YOLOv8n to introduce GAM into YOLOv8. In the Backbone, the feature map is weighted to enhance the ability to identify important features in the image. As shown in Fig. 4, we add the modified GAM to the C2f modules of 40 × 40 pixels × 512 channels and 20 × 20 pixels × 512 channels in the Backbone, respectively. To clearly identify the target in the natural environment, color feature weighting is added to GAM. This model enhances the color of the green iguana to highlight the color features of the target, so that subsequent spatial features can identify the target from the natural environment. A new 160 × 160 pixel detection head is added to the Head part to ensure that large and small targets can be identified and also to enhance detection in complex environments.

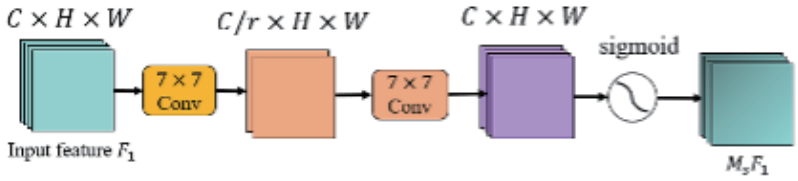


Fig. 3. (Color online) Spatial attention.

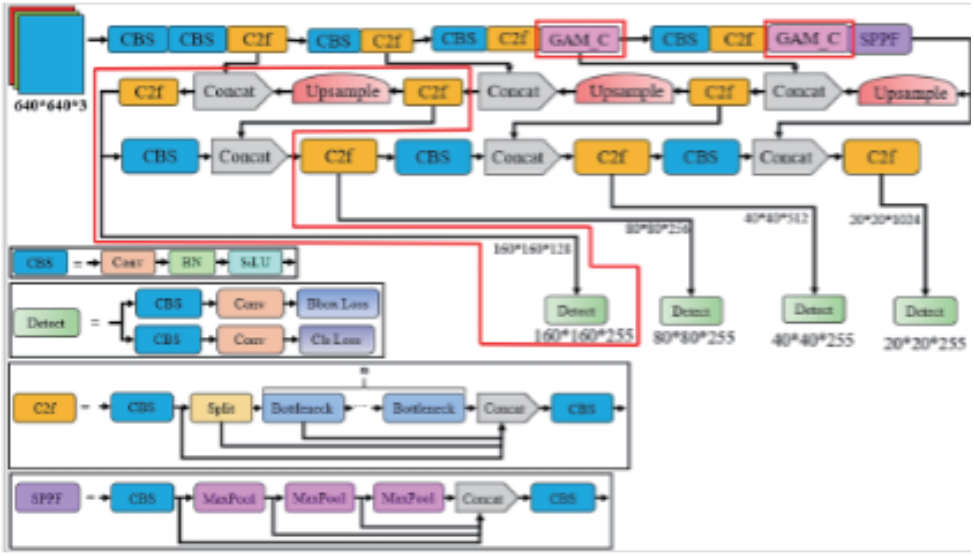


Fig. 4. (Color online) Overall structure of the identification method.

3.2 Data preprocessing and enhancement

We use the green iguana part of Kaggle's public dataset.[9] Since this target has high alertness and is difficult to photograph, the collected public datasets are organized. The public dataset used lacks multitarget images, so some images are captured from the Internet[10,11] to obtain more images to increase the number. The sorted dataset will be preprocessed and enhanced through the Roboflow platform to ensure the image diversity of the dataset. Since targets are difficult to identify in the natural environment, the model's identification capabilities are improved through data preprocessing and enhancement. In the preprocessing part of this work, because the number of image pixels in the collected dataset is 512 × 512, the number of pixels of the images is uniformly adjusted to 640 × 640 to ensure the consistency of the image input into YOLOv8 and to avoid inconsistency in training results caused by different image sizes. The data enhancement involves horizontal and vertical flipping, rotation (−15–+15%), hue adjustment (−15°–+15°), brightness adjustment (−15–+15%), and noise (up to 0.1%), allowing the model to clearly identify targets at different viewing angles and in various environments.

3.3 Channel attention

The YOLOv8 model has the Backbone, Neck, and Head structures, which correspond to different tasks. We import the modified GAM into the Backbone of YOLOv8 to strengthen the model's important features in green iguana identification. Compared with the original YOLOv8, the modified GAM is added after the 40 × 40 pixel × 512 channel and 20 × 20 pixel × 512 channel C2f modules in the YOLOv8 Backbone. This GAM addition improves the model's ability to capture global and local features by combining channel attention and spatial attention.[12–15] Channel attention dynamically adjusts global information to each channel for weight distribution, strengthens the model's focus on important channels, and suppresses the effect of unimportant channels. Spatial attention captures the characteristics of each spatial location in the image through 7 × 7 convolution and generates corresponding weight maps to strengthen the characteristics of important areas. To improve the recognition effect of targets in the natural environment, as shown in Fig. 5, we add color feature weighting to GAM to highlight the color features of the green iguana, and combine spatial attention to focus on the local features of the target. This allows the model to enhance its ability to identify targets when the background and target colors are similar. Such improvements are particularly helpful for target detection in natural environments, ensuring the model's recognition accuracy in complex environments.

3.3.1 Improved GAM

As shown in Fig. 6, the feature map is converted from the RGB image to the HSV color space and the color feature information in the image is extracted. The HSV color space can divide colors into hue (H), saturation (S), and brightness (V), capturing subtle differences in target color characteristics, particularly in complex backgrounds where the target is similar to the natural environment. Then, an adaptive global pooling layer is applied to the HSV feature map to compress the spatial information of each channel into a single global feature value and extract the global semantic information of color. The global color feature value is nonlinearly transformed through the MLP to generate a dynamic color feature weight for the environment. The color feature weight represents the importance of the target color channel. It is compressed by the sigmoid function to limit the color feature weighting to between 0 and 1. Finally, the generated color feature weight is applied to each channel of the feature map to strengthen the target color feature.

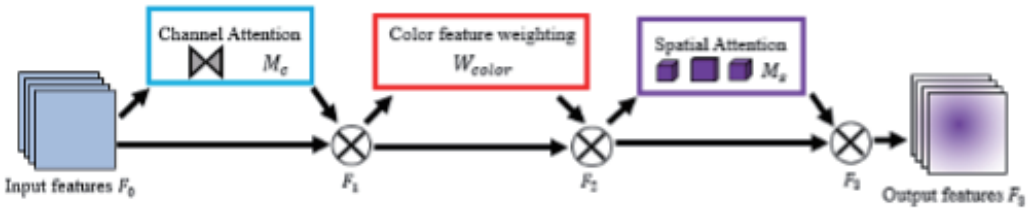


Fig. 5. (Color online) Improved GAM.

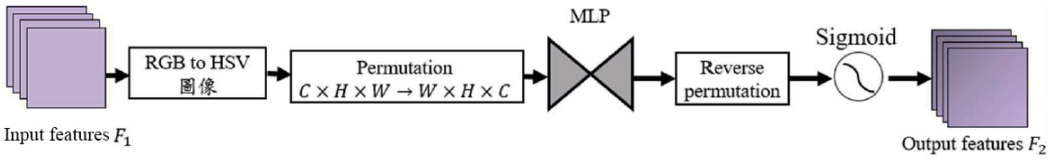


Fig. 6. (Color online) Color feature weighting.

4. Experiment Analysis

4.1 Experimental software and hardware

The experiments were conducted on the Windows 10 operating system, using Anaconda to create a virtual environment for deep learning model training and testing. The detailed configurations of the software and hardware used in the experiment are summarized in Tables 1 and 2, respectively. We use the preprocessed and enhanced dataset, and YOLOv8n is used as the basis for the training and testing of the dataset. The image size is uniformly 640 × 640 pixels for training and testing. The dataset data are divided into approximately 8:1:1 corresponding to 3156 images in the training set and 345 images each in the test and verification sets to detect the target and test the effect of the model.

4.2 Experimental values

We obtained the following data through experiments; they are the model results of the original YOLOv8 version 8.2.90 and the model results of YOLOv8 with the modified GAM and the new hierarchical detection head. These two models were compared and analyzed in terms of their green iguana detection performance: F1–confidence curve, precision–confidence curve, precision–recall curve, recall–confidence curve, and various loss (loss) curves during

Table 1
Software device configuration.

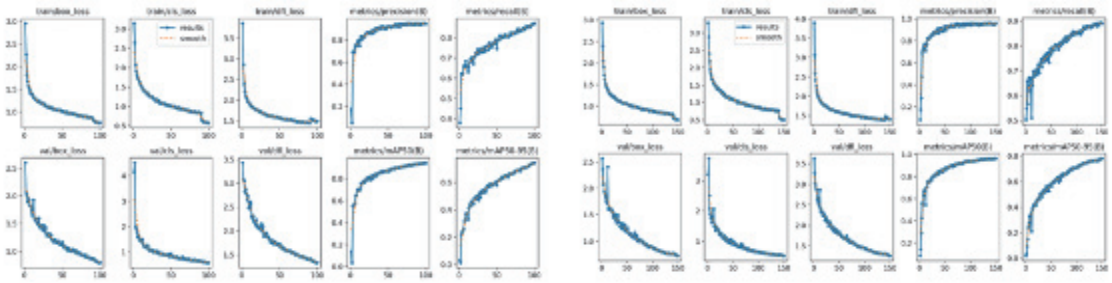
Software	Version
Python	3.10.14
PyTorch	2.4.1
CUDA	11.8
CuDNN	8.9.7

Table 2
Hardware device configuration.

Hardware	Specification
CPU	Intel i5-12400
GPU	Nvidia-RTX-3060-Ti
RAM	DDR4-32GB (16GBx2)
Storage	1TB SSD

the training process. These indicators were used to evaluate the performance of the model in training. Figures 7(a) and 7(b) show the various loss values of YOLOv8 before and after modification, respectively. The box_loss term is the accuracy of the position of the target bounding box when using the slowest model. The smaller the value, the more accurate the model is in predicting the target. Partly because we only targeted green iguanas, the lower the classification accuracy, the more accurate the targeting of the target. The goals for all metrics were the same as those in real applications. There are 95% average accuracies across multiple IoU metrics (0.5 to 0.95), and we also show the performance of the main models at different detection accuracies in Fig. 7.

We compared the results for YOLOv8n before and after modification. F1-Score, Recall, and mAP (@0.5) were used to compare model benchmarks. Most people directly use YOLOv8 for training. We added a modified GAM and a 160 × 160-pixel detection head to the model, compared it with the original YOLOv8, and evaluated the advantages and disadvantages of the modified YOLOv8. The Accuracy, Precision, Recall, and F1-Score of the two models are calculated using confusion matrices. The number of confusion matrices used for performance evaluation is summarized in Table 3 to facilitate subsequent comparison and analysis. As shown in Table 4, the Recall of the modified YOLOv8 increases from the original 0.9 to 0.93, indicating that the model effectively reduces the number of missed detections of targets. Precision drops from the original 0.91 to 0.87, indicating that the model captures more potential candidate frames when the target is very similar to the background. At the same time, the number of false detections still increases when the colors are very similar. The F1-Score still remains at 0.9, indicating that the model had not declined in terms of comprehensive values. mAP(@0.5) increases from the original 0.945 to 0.964, indicating that the model demonstrates higher accuracy when there is greater overlap between the predicted frame and the ground truth annotation (IoU > 0.5), and it consistently identifies targets even in complex background environments.



(a) (b)

Fig. 7. (Color online) (a) Loss values of unmodified YOLOv8. (b) Various loss values of YOLOv8 after modification.

Table 3
Number of confusion matrices used for YOLOv8 before and after modification.

	TP	FP	FN	TN
YOLOv8 model before modification	3163	296	347	0
The proposed method	3291	489	219	0

Table 4
Performance evaluation indicators of YOLOv8 before and after modification.

	Accuracy	Precision	Recall	F1-Score	mAP(@0.5)
YOLOv8 model before modification	0.831	0.91	0.9	0.9	0.945
Proposed method	0.822	0.87	0.93	0.9	0.964

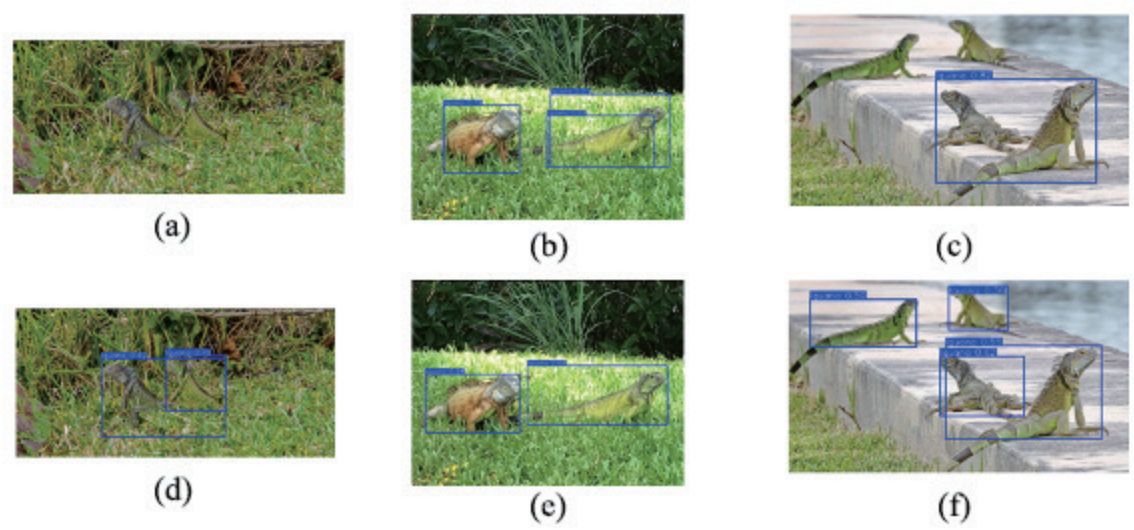


Fig. 8. (a)–(c) Detection images obtained using YOLOv8 before modification. (d)–(f) Modified YOLOv8 detection maps.

4.3 Experimental results

After the experiment, we compared the detection effects of the original YOLOv8n model and YOLOv8n with the modified GAM and the 160 × 160-pixel detection head. In Fig. 8, we can see the difference between the models in green iguana detection. (a) Compared with (d), the original YOLOv8 model cannot identify the target within a complex background. The modified model can effectively distinguish the target from the background with similar colors and has better background suppression. (b) Compared with (e), the original YOLOv8 model makes repeated judgments and misjudgments. The modified model can accurately locate the target. Comparing (c) and (f), we can see that the original model cannot distinguish or clearly identify the targets when there are multiple targets. The modified model can accurately identify the targets in the same situation.

5. Conclusions

We improved YOLOv8 with the addition of GAM to the Backbone of the YOLOv8 model, as well as an attention mechanism to the Backbone's 40 × 40 pixel × 512 channels and 20 × after C2f with 20 pixels × 512 channels. The model effectively improved the accuracy of target and false detection in natural environments. An additional 160 × 160-pixel detection head was added to the model to enhance the recognition accuracy of targets within complex backgrounds and the recognition accuracy of small targets. The experimental results showed that the modified model's indicators improved; the two indicators Accuracy and Precision declined slightly. Such a decline allows the model to identify more targets from the environment. While the F1-Score was maintained, indicators such as Recall and mAP (@0.5) were slightly improved compared with those of the original YOLOv8 model, indicating that the feature extraction function of the modified model can capture the color, texture, and shape characteristics of green iguanas in complex backgrounds, thereby enhancing the model accuracy for detecting targets and improving the stability and accuracy of identification in natural environments. In low-contrast or complex background environments, the green iguana target and background can be effectively identified, reducing erroneous detection and missed detection situations.

The YOLO model is constantly being updated. It is currently at the YOLOv11 version. YOLOv11 is improved upon the YOLOv8 version. The accuracy is slightly improved and the CPU execution speed is considerably improved. Compared with the GPU, it is slightly slower but maintains the same accuracy. At the same time, using fewer parameters reduces the running time, but the new model will still be facing difficulties depending on the environment in which it is used. Future research will further the understanding of the attention mechanism in complex environments and modify it to improve the recognition accuracy, especially target detection in complex backgrounds and natural environments. It is hoped that low-cost equipment can be used for real-time identification and incorporating more ecological data, such as temperature changes and humidity in the ecological environment, to achieve the long-term monitoring and data accumulation of species. In future research, we will further optimize the adaptability of the model in various seasons and climate conditions, evaluate the stability of the model, and evaluate the performance of the model under long-term environmental changes. As the changes in the ecological environment intensify, it is necessary to protect the ecological environment and provide early warning to enable timely responses to ecological changes.

Acknowledgments

This research was funded by the National Science and Technology Council (NSTC) of the Republic of China under Grant No. NSTC 114-2222-E-167-004.

References

1 J. Redmon, S. Divvala, R. Girshick, and A. Farhadi: arXiv (2016). https://doi.org/10.48550/arXiv.1506.02640
2 J. Redmon and A. Farhadi: arXiv (2016). https://doi.org/10.48550/arXiv.1612.0824
3 J. Redmon and A. Farhadi: arXiv (2018). https://doi.org/10.48550/arXiv.1804.02767
4 A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao: arXiv (2020). https://doi.org/10.48550/arXiv.2004.10934
5 G. Jocher and Ultralytics: GitHub Repository, YOLOv5 (2022). https://github.com/ultralytics/yolov5
6 G. Jocher: GitHub Repository, YOLOv8 by Ultralytics (2023). https://github.com/ultralytics/ultralytics
7 Y. Liu, Z. Shao, and N. Hoffmann: arXiv, Global Attention Mechanism: Retain Information to Enhance Channel-Spatial Interactions (2021). https://doi.org/10.48550/arXiv.2112.05561
8 C. T. Lin and Y. Y. Chang: Master Thesis, Research on Strengthening Green Iguana Target Recognition Through Global Attention Mechanism (2024). National Chin-Yi University of Technology, Taichung, Taiwan.
9 A. Therrien: Kaggle Dataset, Image Classification – 64 Classes – Animal (2023). https://www.kaggle.com/datasets/anthonytherrien/image-classification-64-classes-animal
10 iNaturalistTW: Website, Iguana iguana. https://taiwan.inaturalist.org/taxa/35342-iguana-iguana/browse_photos?term_id=17&term_value_id=18
11 miesiao: Website, South-South Corner (2020). https://sousoucorner.org/media/green-iguana-around-caribbean/.
12 R. Khanam and M. Hussain: arXiv (2024). https://arxiv.org/abs/2410.17725.
13 H. Wu, Y. Zhu, and L. Wang: Appl. Sci. **13** (2023) 11760. https://doi.org/10.3390/app132111760
14 Z. Yafeng, Y. Junyang, W. Yuanyuan, T. Shuang, L. Han, X. Zhiyi, W. Chaoyi, and Z. Ziming: IET Image Process. **17** (2023) 3986. https://doi.org/10.1049/ipr2.12912
15 J. Fu and Y. Tian: Eng. Lett. **32** (2024) 1377. https://www.engineeringletters.com/issues_v32/issue_7/EL_32_7_13.pdf

About the Authors

Ming-Te Chen received his M.S. and Ph.D. degrees in computer science and information engineering from National Sun Yat-sen University, Taiwan, in 2005 and 2012, respectively. In 2018, he joined the faculty of the Department of Computer Science and Information Engineering, National Chin-Yi University Technology, Taichung, Taiwan. His current research interests include information security, applied cryptographic protocols, digital signatures, IoT security, and electronic commerce (mtchen@ncut.edu.tw)

Cheng-Hui Chen received his Ph.D. degree in computer science from National Chung Hsing University. He is currently an assistant professor in the Department of Computer Science and Information Engineering at National Chin-Yi University of Technology. His research interests include image processing, signal processing, pattern recognition, and intelligent manufacturing. (chchen@ncut.edu.tw)

Chun-Ting Lin received his B.S. and M.S. degrees in computer science and information engineering from National Chin-Yi University of Technology, Taichung, Taiwan, in 2022 and 2024, respectively. He has experience in model optimization, data annotation, and applying attention mechanisms to improve recognition accuracy. He is currently conducting his research on deep-learning-based object detection, with a focus on improving the recognition accuracy of ecologically impactful species in natural environments. (rock693412136@gmail.com)

Yi-Ying Chang received his bachelor's degree in electronics engineering (1982) and his master's degree in computer information science (1997) from the Department of Computer Information & Science Engineering of the Knowledge System Institute, Skokie, Illinois, USA. He is currently pursuing his Ph.D. degree at the Data Compression and Multimedia Communication Laboratory, Electrical Engineering Department of National Cheng Kung University, Tainan, Taiwan. His research interests include image enhancement and segmentation. He joined National Chin-Yi University of Technology in 1982, and he is currently an assistant professor in the Department of Computer Science and Information Engineering.