Real-time Target Detection System in Scenic Landscape Based on Improved YOLOv4 Algorithm

Cheng Pan[1], Haiyan Zhao[2], Meijiao Sun[3*]
[1]Scientific Research Office, Nanchang Vocational University, Nanchang 330500, China
[2]Office of Academic Affairs, Nanchang Vocational University, Nanchang 330500, China
[3]School of Economics and Management, Nanchang Vocational University, Nanchang 330500, China
E-mail: summeijiao@163.com
[*] Corresponding author

Keywords: YOLOv4; Image; Target detection; Landscape; Real-time; Deep learning

Received: February 6, 2024

With the rapid development of computer vision technology, the use of real-time target detection systems in scenic landscape management and services is increasingly widespread. To enhance the precision and efficiency of real-time target detection in scenic landscapes, this research integrates the fourth version of the You Only Look Once (YOLO) algorithm to construct an optimized real-time target detection system is introduced for scenic landscapes. First, adaptive spatial feature fusion to enhance the fourth version of the You Only Look Once algorithm. Then, the optimized algorithm was combined with OpenCV library, Python OS library, and other hardware and software to design a real-time image recognition system for scenic landscapes. The study results indicated that the proposed optimized algorithm had better recognition performance, and its precision value, recall rate, and F1 value were as high as 0.96, 0.97, and 0.98, respectively. The recognition system, which was developed using an optimization algorithm, demonstrated excellent practical application effect. It displayed stable system operation under four natural landscapes: sunrise, sea of clouds, maple forest, and stone monument, with a stability performance of 0.92, 0.93, 0.92, and 0.94, respectively. Moreover, the system operated remarkably fast, with low operational times of 2.3 s, 0.8 s, 2.9 s, and 1.2 s under these landscapes. In conclusion, the research institute's target detection algorithm has demonstrated excellent performance. Utilizing this algorithm in the detection system can offer technical aid for managing and intelligently detecting scenic landscape images.

Povzetek: Raziskava predstavlja izboljšan sistem za zaznavanje tarč v realnem času v slikovitih krajinskih območjih, temelječ na algoritmu YOLOv4, s prilagodljivo prostorsko združitvijo značilnosti in uporabo knjižnic OpenCV in Python OS.

1 Introduction

Under the present wave of digitization, intelligent management of tourist sites has become crucial in improving guest experience and ensuring safety [1-2]. The real-time Target Detection (TD) system plays a vital role by accurately identifying and locating various objects in the scenic area, such as tourists, natural landscapes, historical sites, and more. Furthermore, it provides technical support for image data management in the scenic area. In the field of TD, You Only Look Once (YOLO) and its derived network structures have achieved better detection results. Among them, the You Only Look Once Version 4 (YOLOv4) algorithm has received wide attention for its efficient detection speed and good accuracy [3-4]. Real-time TD of scenic landscapes is challenging due to various factors, including light variations, occlusion problems, complex backgrounds, and diverse target types. Therefore, relying solely on the traditional YOLOv4 for building recognition models is no longer sufficient to achieve high-precision TD tasks [5]. To address the aforementioned issues, this study optimizes the YOLOv4 algorithm and introduces a real-time landscape-based tourism demand system suitable for scenic locations. Building a real-time TD system for scenic spots enhances theoretical research on intelligent monitoring system applications in actual scenic spots and provides a practical technical solution. This solution is of great practical significance for promoting the development of intelligent tourism. This study is comprised of five sections. The initial section provides a concise overview of the study, while the subsequent section critically evaluates and summarizes prior research. The third section presents the research methodology, while the fourth section assesses algorithm performance. The fifth and final section offers a comprehensive summary of the entire study.

2 Related Works

Landscape image TD is a method in computer vision and deep learning designed to identify and locate specific features or objects within landscape photos or video

streams. A multitude of experts have researched this field, combining different models and algorithms to advance the technique. Jahani et al. study utilized three machine learning techniques, support vector machine, radial basis function neural network with multilayer perceptron, to simulate and evaluate landscape images of deciduous forests in northern Iran. By analyzing 13 landscape features, it was found that the multilayer perceptron model performed optimally in assessing the aesthetic quality of forest landscapes with a coefficient of determination of 0.878. In addition, the study identified the significant effect of factors such as tree species diversity and canopy density on landscape quality through the aforementioned models [6]. Peng et al. developed an image style conversion framework based on a recurrent generative adversarial network model, and used the framework to realize the transformation of landscape photos to Chinese landscape painting style. With the contour enhancement technique and edge detection operator, the conversion effect outperformed the traditional generative adversarial network model in both edge sensitivity and structural similarity. The outcomes demonstrated that the framework can enhance the landscape painting effect of landscape photographs with a comprehensive similarity score as high as 0.92. In the comparative analysis, the method outperformed several existing reference models in terms of visual quality [7]. Zhou et al. improved the traditional codebook modeling algorithm and proposed an improved codebook modeling algorithm. In addition to examining the origins and uses of the background modeling approach in motion TD, the paper also highlighted the shortcomings and suitability of the conventional approach, providing a framework for future studies on motion TD based on complicated background modeling. Furthermore, this study employed a combination of deep learning algorithms to examine the properties of fast-moving films and enhance their ability to identify features. According to study findings, the updated algorithm can analyze high-speed films efficiently and increase motion video frame feature detection [8]. Kikuchi et al. designed a method capable of performing real-time detection and virtual removal of existing buildings from a video stream, aiming to more intuitively demonstrate a future scene without these buildings. The results showed that the method was able to accurately perform real-time detection and building removal at 5.71 frames/sec when the complementary field of view was no more than 15%, which can effectively help users visualize the future environment on-site while reducing time and cost consumption [9].

As the computational speed and detection efficiency of the YOLO algorithm continue to improve, it has become increasingly prevalent in numerous real-time application scenarios, including video surveillance, autonomous driving, drone monitoring, and various industrial vision systems. Along with advancements in artificial intelligence technology, experts have conducted vast research on the YOLO algorithm's performance. To address the problem of degraded performance of deep learning techniques for cross-domain object detection in the presence of insufficient labeled data, Li et al. proposed a step-by-step domain-adaptive YOLO framework. The framework creatively constructed an auxiliary domain to narrow the gap between the source and target domains, and then utilized the newly developed domain-adapted YOLO algorithm for the cross-domain object detection task. Experimental results showed that the detection framework designed by the institute significantly improved the detection performance of the algorithm [10]. Lee and Hwang explored the service performance of the YOLO algorithm in real-time object detection in resource-constrained AI embedded systems. To address the poor performance of YOLO in webcam object detection, a novel YOLO architecture with adaptive frame control was proposed in the paper to effectively address these issues. The results showed that the proposed adaptive frame control YOLO scheme can reduce the service delay while maintaining the high accuracy and convenience of YOLO, overcoming the real-time processing limitations of pure YOLO systems [11]. Aiming at the real-time monitoring and assisted driving requirements in self-driving vehicles, Liang et al. proposed an edge-cloud cooperative object detection system called Edge YOLO. With the use of compressed feature fusion and pruned feature extraction networks, Edge YOLO developed a lightweight framework that may significantly increase multi-scale prediction efficiency while lowering the system's reliance on CPU resources. The research results showed that Edge YOLO had high reliability and detection accuracy on COCO2017 and KITTI datasets [12].

Table 1: Summary of related work

Researchers	Year	Technology methods	Key Findings	Limitations	Literature number
Jahani et al.	2023	Machine learning techniques	Using machine learning to assess the aesthetic quality of forest landscapes	Limited to forested landscapes in specific areas, not covering a wider range of landscape types	[6]

Peng et al.	2022	Recurrent generative adversarial networks	Realising the transformation of landscape photos to Chinese landscape painting style	Conversion effects are dependent on specific styles with limited ability to generalise	[7]
Zhou et al.	2023	Modified codebook modelling algorithms	Improving feature recognition of fast-motion video sequences	Needs further optimisation to deal with extreme lighting and complex backgrounds	[8]
Kikuchi et al.	2022	Semantic segmentation and GAN	Capable of detecting and virtually removing existing buildings from video streams in real-time	Only applicable to specific landscape assessment scenarios	[9]
Li et al.	2022	Progressive domain adaptation YOLO framework	Significantly improves cross-domain object detection performance	Requires ancillary domain data, limited applicability	[10]
Lee and Hwang	2022	Adaptive frame control YOLO architecture	Improved service performance of YOLO in real-time object detection.	Primarily for resource-constrained environments, may not be applicable to all scenarios	[11]
Liang et al.	2022	Edge YOLO system	Enabling efficient real-time object detection in autonomous driving	Dependent on edge-cloud co-operation, high implementation costs	[12]

In summary, Table 1 shows that numerous experts have conducted studies on image detection and YOLO algorithm performance testing. Additionally, many experts have applied the YOLO network model to image detection and achieved superior research results. The ongoing development of the tourism industry has led to the application of various intelligent technologies in tourism management. To enhance real-time detection and retrieval of landscape images in scenic spots, this study will optimize the YOLO network and develop a real-time TD system for the said images. By doing so, tourists can acquire landscape images swiftly and locate attractions efficiently, providing significant technical assistance to intelligent tourism management.

3 Real-time TD system for landscape images of scenic spots based on YOLOv4-ASFE algorithm

In today's digital era, the real-time TD system has a significant enhancement effect on both the management of tourist attractions and the experience of tourists. In this study, the traditional YOLOv4 TD algorithm is first optimized, and the TD accuracy of YOLOv4 is optimized by introducing Adaptive Spatial Feature Fusion (ASFF). On this basis, adaptive spatial feature optimization You Only Look Once version 4 (YOLOv4-ASFF) for real-time images of scenic landscapes is designed, aiming at completing the real-time detection of landscape images by this system, so as to improve the image data management effect of the scenic management system.

3.1 TD Algorithm design based on improved YOLOv4

The traditional YOLOv4 is a popular real-time TD

algorithm commonly used for computer vision tasks. Compared to the previous three versions of YOLO algorithm, YOLOv4 brings several improvements and advantages [13]. Figure 1 depicts the conventional YOLOv4 network configuration.

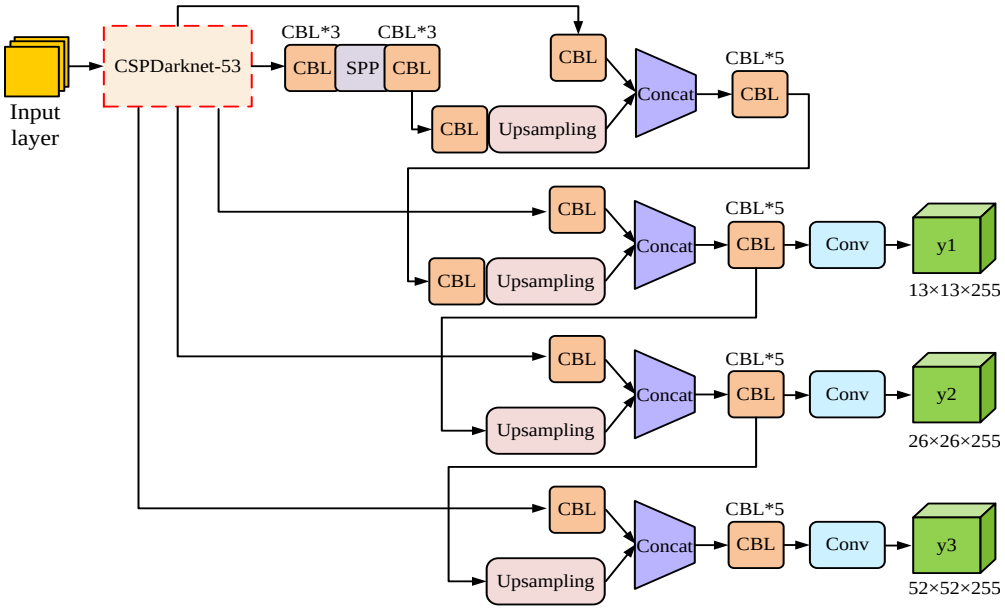


Figure 1: Traditional YOLOv4 network structure

The input layer, the additional modules, the head network, the anchors, the loss function, and the backbone network make up the six key components of the classic YOLOv4 network topology shown in Figure 1. The input layer primarily prepares the incoming picture data. Cross Stage Partial (CSP) networks are frequently used to increase the backbone network's learning capacity while lowering the computational cost of the model. The backbone network is mostly used for feature extraction. The additional module is located behind the backbone network and enhances the sensory field through pooling operations, allowing the network to recognize features at different scales and increasing the model's adaptability to the input size. The header network is dedicated to the final TD task, and this part contains a target size prediction layer, a target category prediction layer, and a target frame prediction layer. In YOLOv4, anchor points are target boxes used to predict actual features. The dimensions of the anchor points are obtained by clustering and analyzing the bounding box dimensions in the training dataset. Finally, the loss function is mainly used to train the loss process so that the network can achieve the predetermined prediction through multiple training. In the YOLOv4 network structure, the loss function is mainly divided into two parts: classification loss and location regression loss. The common classification loss functions are cross-entropy loss function and Softmax loss function. Equation (1) is the mathematical expression of the cross-entropy loss function, which is frequently employed in classification loss [14].

$$L(y)=\frac{1}{N}\sum_{i}-\left[y_i\log y_i+(1-y_i)\log(1-y_i)\right] \quad (1)$$

In equation (1), y_i denotes the probability that sample i is predicted to be a positive class. The predicted probability of all samples is divided into two categories of labels, where the positive category labels are denoted by 1 and the negative category labels are denoted by 0. N denotes the number of all labels. $L(y)$ denotes the cross-entropy loss function. The Softmax loss function is calculated as shown in equation (2).

$$Soft\max(x)=-\log\frac{e^{x_i}}{\sum_{i}^{C}e^{x_i}} \quad (2)$$

In equation (2), C denotes the number of categories. x_i denotes the output of the correct category. $Soft\max(x)$ denotes the Softmax loss function. Smooth L1 is a kind of location regression loss function and its expression is shown in equation (3).

$$SmoothL_1(x)=0.5x^2 \quad (3)$$

In equation (3), Smooth L1 uses the expression in equation (3) when $|x|<1$. x denotes the positional regression value. When the positional regression value $|x|\geq 1$, equation (3) becomes equation (4).

$$SmoothL_1(x)=|x|-0.5 \quad (4)$$

Based on equation (3) and equation (4), it is possible to obtain the border regression task loss calculation equation for real TD as shown in equation (5).

$$L_{loc}(t^u,v)=\sum_{i\in\{x,y,w,h\}}SmoothL_1(t_i^u-v_i) \quad (5)$$

In equation (5), $L_{loc}\left(t^u, v\right)$ denotes the loss value of the border regression task for the actual TD. t^u and v denote the predicted and actual coordinates, respectively, and their specific expressions are shown in equation (6). i denotes the sample, whose coordinates are denoted by $\{x, y, w, h\}$.

$$\begin{cases} t^u = \left(t_x^u, t_y^u, t_w^u, t_h^u\right) \\ v = \left(v_x, v_y, v_w, v_h\right) \end{cases} \tag{6}$$

In equation (6), $\left(t_x^u, t_y^u, t_w^u, t_h^u\right)$ and $\left(v_x, v_y, v_w, v_h\right)$ denote the true value coordinates and predicted value coordinates, respectively. Equation (6) is viewed as a whole for regression, and its calculation equation is obtained as shown in equation (7).

$$IoULoss = -\ln \frac{Inter\,\sec tion\left(box_{gt}, box_{pre}\right)}{Union\left(box_{gt}, box_{pre}\right)} \tag{7}$$

In equation (7), box_{gt} and box_{pre} denote the true and predicted frames, respectively. When there is no overlapping region between the real and predicted boxes in equation (7), it will lead to equation (7) being equal to 0, which does not reflect the distance between the predicted and real values in depth. Based on this, the distance measurement equation is introduced as shown in equation (8).

$$GIoULoss = IoU - \frac{\left|A_c - U\right|}{\left|A_c\right|} \tag{8}$$

In equation (8), IoU is an abbreviation for $IoULoss$, which denotes the intersection and merger ratio loss. A_c denotes the area of the smallest closed region shared by the two boxes. U denotes the concatenation of the two boxes. Based on equation (8), the distance between the centroids of the two boxes is further considered to obtain the $DIoULoss$ loss function in equation (9).

$$DIoULoss = IoU - \frac{\rho^2\left(c^{pre}, c^{gt}\right)}{d^2} \tag{9}$$

In equation (9), c^{pre} and c^{gt} denote the centroid positions of the prediction frame and the real frame, respectively. ρ denotes the Euclidean distance between the two centroids. d denotes the diagonal distance between the prediction frame and the real frame. A penalty factor is added to equation (9) to obtain equation (10).

$$CIoULoss = IoU - \frac{\rho^2\left(c^{pre}, c^{gt}\right)}{d^2} - \alpha v \tag{10}$$

In equation (10), αv denotes the penalization factor. Where α and v denote the weight function and aspect ratio measurement parameters, respectively. The specific equation for the weight function is shown in equation (11).

$$\alpha = \frac{v}{\left(1 - IoU\right) + v} \tag{11}$$

In equation (11), IoU denotes the cross-merger ratio loss. The equation for the aspect ratio measurement parameter is shown in equation (12).

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w^{pre}}{h^{pre}}\right)^2 \tag{12}$$

In equation (12), w^{gt} and h^{gt} denote the width and length of the true frame, respectively. w^{pre} and h^{pre} denote the width and length of the predicted frame, respectively. According to the above equation (10) is able to calculate the loss function of YOLOv4.

In the scenic landscape TD problem, since the distant buildings may be very small and the near sculptures may be very large therefore the traditional YOLOv4 cannot better detect the real-time landscape. Furthermore, the study incorporates the ASFF to enhance the stability and detection precision of the conventional YOLOv4 in light of the potential impact of shifting lighting conditions, dynamic target objects, and complex terrain backgrounds on its detection. ASFF enables effective information exchange at the feature layer and enhances the model's recognition ability for targets at different scales by intelligently adjusting weights between multi-scale feature maps. The ASFF mechanism dynamically adjusts fusion weights by learning weight parameters between different feature maps, allowing the algorithm to adaptively strengthen the response to important features. In addition, ASFF can simultaneously suppress background noise, significantly improving YOLOv4's ability to detect small targets in complex landscapes and accurately recognize large targets. The incorporation of ASFF into the YOLOv4 model enhances the feature extraction process and improves the model's recognition accuracy in various and challenging landscapes. This is particularly true when dealing with scenes that have strong lighting variations and significant differences in target size. Figure 2 depicts the ASFF's organizational structure.

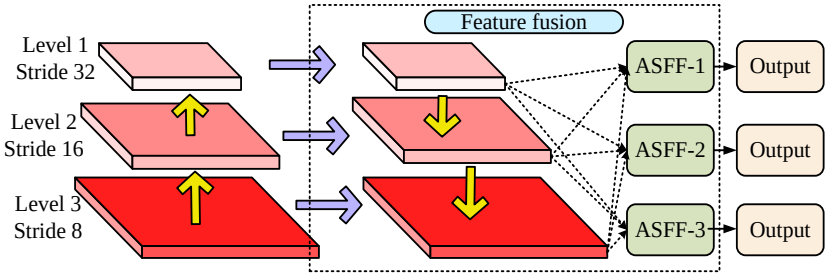


Figure 2: ASFF structure diagram

In Figure 2, ASFF is mainly composed of multi-scale feature maps, adaptive weight learning, and special fusion mechanism. With the use of ASFF, feature maps with varying scales will be able to communicate information more effectively, which will enhance tiny TD performance and preserve high identification accuracy for large targets. In ASFF, the input features are fused through three layers of adaptive fusion to increase the richness of information, and the fusion process is shown in equation (13).

$$y_{ij}^{l} = \alpha_{ij}^{l} x_{ij}^{1\to l} + \beta_{ij}^{l} x_{ij}^{2\to l} + \gamma_{ij}^{l} x_{ij}^{3\to l} \quad (13)$$

In equation (13), α_{ij}^{l}, β_{ij}^{l}, γ_{ij}^{l} denote the weight parameters of the first, second, and third layers, respectively. $x_{ij}^{1\to l}$, $x_{ij}^{2\to l}$, $x_{ij}^{3\to l}$ denote the features of the first, second, and third layers, respectively. y_{ij}^{l} denotes the fused features. The structure of YOLOv4-ASFF is obtained by adding ASFF into YOLOv4 as shown in Figure 3.

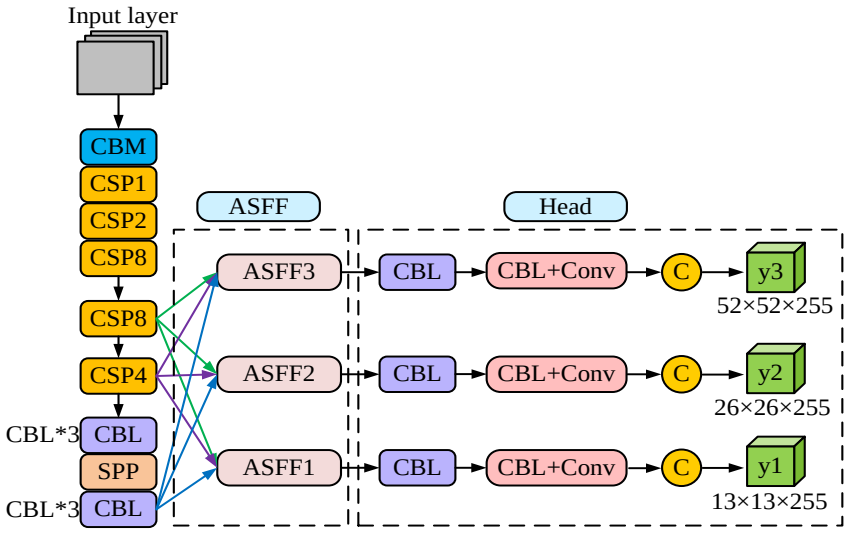


Figure 3: YOLOv4-ASFF structure diagram

In Figure 3, the optimized YOLOv4-ASFF is mainly composed of input layer, feature fusion layer, and decoupling header module. Different from the traditional path aggregation network, YOLOv4-ASFF adopts the ASFF network, which fuses and weights the three layers of features output from the backbone network, thus further enriching the feature information of the scenic landscape image, and avoiding the introduction of too many parameters. To further increase the model's detection accuracy, the decoupled detection header is also adopted by the model to optimize edge regression and classification, respectively.

3.2 Design of Real-time Image Recognition System for Scenic Landscapes

In addition to optimizing YOLOv4 to improve the detection accuracy of the target, it is necessary to further combine various types of hardware and software to build a complete scenic landscape image recognition system. The designed real-time image recognition system for scenic landscapes can not only support real-time image recognition, but also process static images, so as to provide tourists with instant and rich scenic area information, enhance the tourists' experience, and also support the digital management of scenic areas. By building a real-time image recognition system for scenic landscapes, the cultural value and natural beauty of scenic spots can be better demonstrated, and at the same time provide a scientific basis for the protection and management of scenic resources [15]. The traditional scenic landscape recognition system has shortcomings such as insufficient recognition accuracy, slow processing speed, poor generalization ability, limited real-time monitoring ability, and poor user interactivity, etc. The combination of optimized YOLOv4-ASFF algorithm to build a real-time image recognition system for scenic landscapes can effectively improve the above shortcomings. The structure of the scenic landscape image recognition system designed in this research is shown in Figure 4.

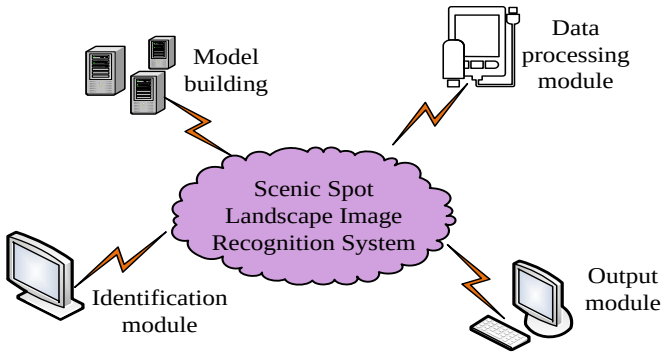


Figure 4: Structural diagram of scenic landscape image recognition system

In Figure 4, the designed landscape image recognition system for scenic spots mainly consists of four modules: model construction, data acquisition and input, model recognition, and recognition result output. The core of the model building module is to create an accurate landscape recognition model. First, a large amount of scenic landscape image data collection is carried out. These images need to contain a variety of landscapes in the scenic area, such as natural landscapes, buildings, sculptures, and so on. Next, these images were accurately labeled using an annotation tool, including the categories and locations of the objects. Then, these labeled data are trained using the YOLOv4-ASFF algorithm. The model's parameters are adjusted during training to increase recognition speed and accuracy. The model is retained for use in later courses once the training is over. The data acquisition and input module mainly uses the OpenCV library to realize the acquisition of real-time video streams, which can acquire real-time images from cameras set up in scenic spots. For non-real-time image recognition, an interface is provided to allow users to upload image files, and Python's os library is used to process file paths and read image data. In the model recognition module, it is first necessary to import the previously trained YOLOv4-ASFF model. When image data is received from the data acquisition module, the model detects and recognizes the landscapes in the image. The recognition process involves extraction of image features, inference using the model, and deriving category and location information for each landscape in the image. The recognition result output module focuses on visualizing the recognition results of the model on the user interface, such as marking the recognized landscapes on the image through a bounding box and displaying the category name next to it. At the same time, the recognition results, including the images, the recognized landscape information and the associated confidence level, are stored in a local folder for further analysis or archiving. Figure 5 depicts the precise workflow of the landscape image recognition system.

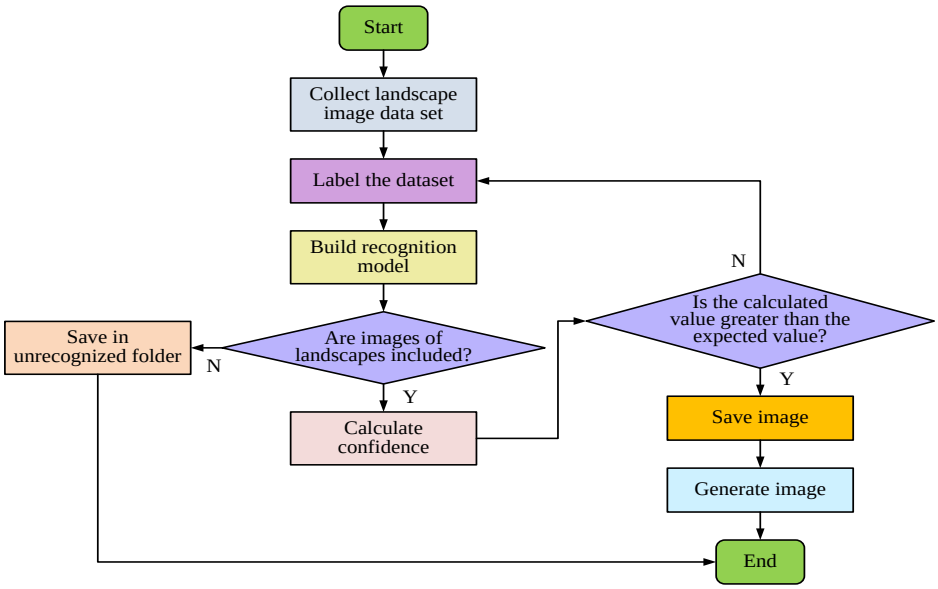


Figure 5: Flowchart of scenic landscape image recognition

In Figure 5, firstly, a complete landscape image dataset needs to be built in the designed landscape image recognition system, and then the image dataset is labeled. Then the optimized YOLOv4-ASFF algorithm is used to build a recognition model to detect the input landscape images. The model's confidence can be computed by

comparing the difference between the actual and expected confidence values, provided that the detected content includes the target landscape. If the detected content does not contain the target landscape, then the image needs to be saved in the computer and labeled as unrecognized. When the actual confidence value is greater than the expected value, this image can be saved in the corresponding landscape file, thus completing the whole image recognition process. Continuously repeating the above steps, then the performance of the YOLOv4-ASFF algorithm can be optimized, so that the output value of this algorithm is getting closer and closer to the preset value, and then all TD tasks can be completed. The final recognition content will be displayed in the UI visualization page, as shown in Figure 6.

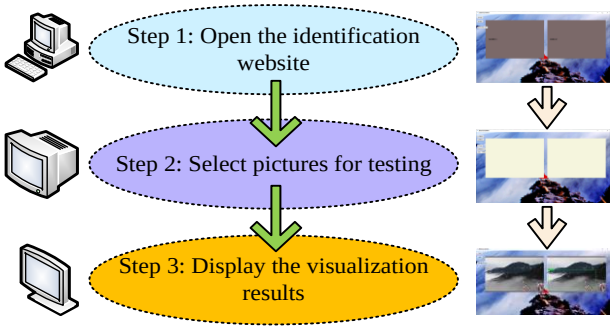


Figure 6: Visual page diagram

In Figure 6, the user is able to get the recognition visualization result of the image through the designed scenic landscape image recognition system. Users first need to enter the browser to open the recognition website, and then open the image to detect the image, when the recognition system completes the detection will be displayed in the UI visualization page to detect the results.

4 Evaluation of performance and application effect of scenic landscape image detection algorithm based on improved YOLOv4-ASFF

In order to prove that the algorithms and recognition systems designed in this research have better performance and application results, three different detection algorithms were selected to compare their performance, so as to find that the YOLOv4-ASFF algorithm's detection accuracy and stability in TD are better than the other compared algorithms. In addition, the application of YOLOv4-ASFF algorithm to the recognition system can also achieve better recognition results.

4.1 Performance test of scenic landscape image detection algorithm

To comprehensively evaluate the performance of the improved YOLOv4-ASFF algorithm in real-world application scenarios, this study carefully selects and processes two types of datasets, namely the publicly available cityscapes dataset and the homemade scenic landscape image dataset collected specifically for the needs of this research. During the data pre-processing stage, the data quality is first ensured through a series of standardized steps, including image resizing, contrast enhancement, and denoising, in order to simulate the various environmental factors that
may be encountered in TD in scenic landscapes. The Cityscapes dataset is selected for the dataset selection criteria because of its rich urban street view images and accurate pixel-level annotation, in order to test the algorithm's ability to detect targets in complex urban environments. The homemade scenic landscape image dataset, on the other hand, covers a wide range of natural landscapes, reflecting the specific application scenarios of scenic landscape TD, ensuring the practicality and wide applicability of the experimental results. The two datasets are partitioned randomly into training and validation sets in a 9:1 ratio after preprocessing to ensure fairness in the training process and reliability in the validation results. Performance evaluation in this study comprehensively considered several metrics, such as precision, recall, and F1 score, chosen based on their wide application and recognition in the field of TD. The precision metric reflects the model's ability to recognize positive class samples, while the recall measures the proportion of positive class samples recognized by the model to the total positive class samples. The F1 score is the reconciled average of precision and recall, providing a comprehensive performance evaluation. These performance metrics allow for a thorough evaluation and demonstration of the YOLOv4-ASFF algorithm's performance under various conditions and its superiority over other algorithms. Table 2 displays the precise makeup of the two datasets.

Table 2: Data set information

Data set composition	Cityscapes	Scenic spot landscape image data set

category	50 different street scenes	8 different natural landscapes
Number of samples	3100	6500
Source	Images taken in different urban street view environments	Image taken at a tourist attraction in the city
Annotation information	19 categories annotated for image detection	Image detection
Data Format	JPEG	JPEG

In Table 2, the specific information of the dataset is given, including the number of dataset samples, categories, labeling information and so on. The specific hardware and software environment of the experiment is shown in Table 3.

Table 3: Experimental environment

Environment	Set up	Parameter configuration
Hardware environment	CPU	AMD Ryzen7 4800H
	GPU	NVIDIA GeForce RTX2060, 6GB RAM
Software Environment	Programming system	Pytorch
	Operating system	Windows 10, 64-bit

In Table 3, the specific hardware environment and software environment for this experiment are given. The variation of loss function of YOLO, Single Shot MultiBox Detector (SSD), YOLOv4-ASFF, and You Only Look Once version 5 (YOLOv5) is tested under the dataset information in Table 1 and the experimental environment in Table 2 as shown in Figure 7.

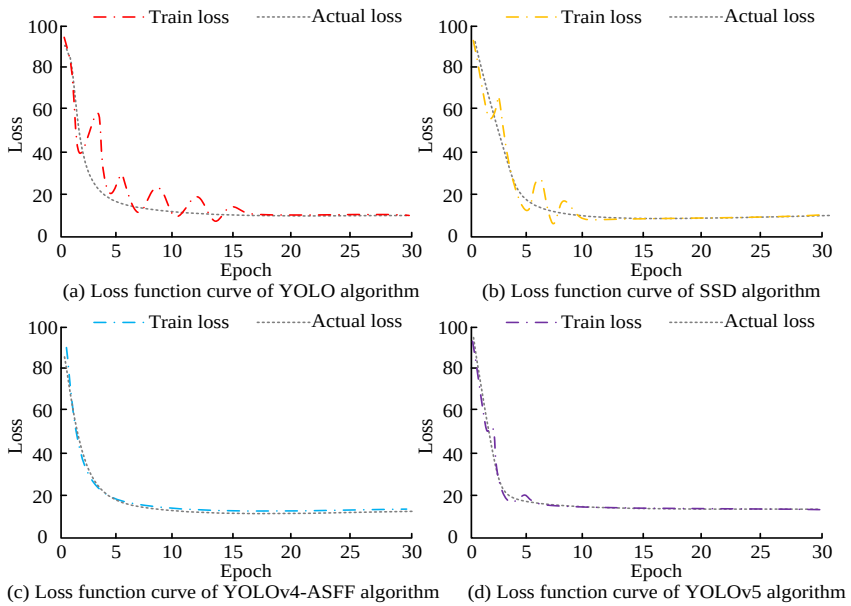


Figure 7: Loss function curve changes of each algorithm

Figure 7 shows the variation of loss function curves for the four detection algorithms. Figure 7(a), Figures 7(b), (c), (d) show the loss function curves of the four algorithms YOLO, SSD, YOLOv4-ASFF, and YOLOv5, respectively. Taken together, the training loss curve of YOLOv4-ASFF can overlap well with the actual loss curve, and when the value of epoch is 7, the training loss curve of YOLOv4-ASFF starts to stabilize. On the contrary, YOLO, SSD, and YOLOv5 need to traverse 17, 12, and 9 epochs, respectively, in order to reach a stable training loss value.

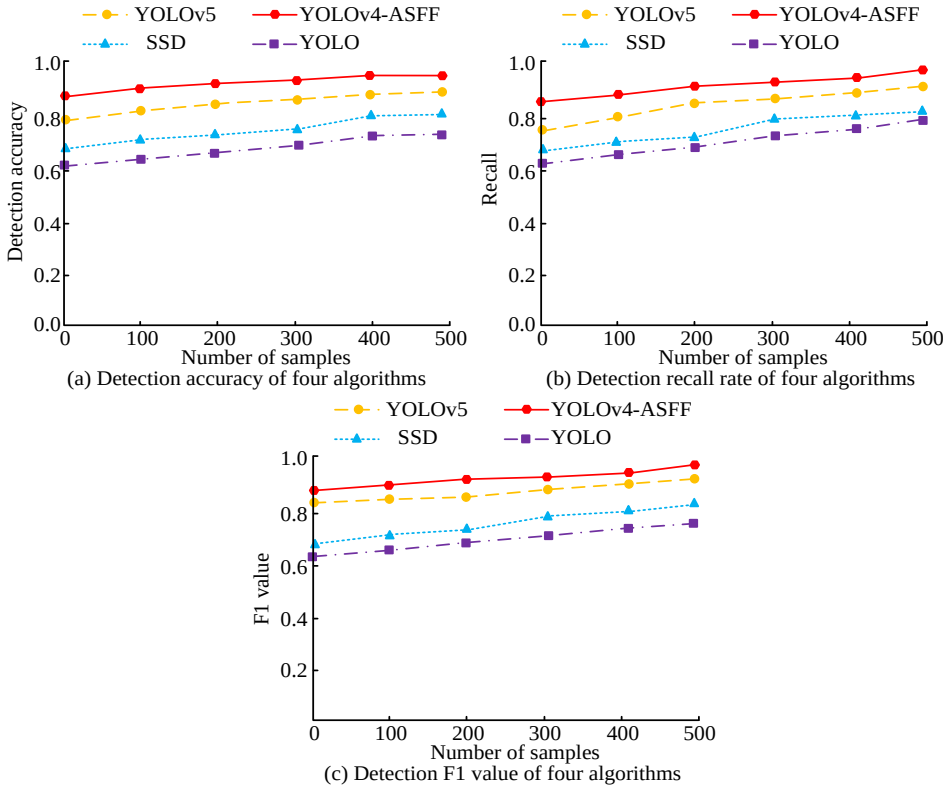


Figure 8: Detection accuracy, recall rate and F1 value changes of each algorithm

The detection precision, recall and F1 value variation of the four detection algorithms are shown in Figure 8. From Figure 8(a), when the number of samples is 500, the detection precision values of the four algorithms YOLO, SSD, YOLOv4-ASFF, and YOLOv5 are 0.69, 0.78, 0.96, and 0.91, respectively. From Figure 8(b), when the number of samples is 500, the detection recall values of the four algorithms YOLO, SSD, YOLOv4-ASFF, and YOLOv5 algorithms have detection recall values of 0.71, 0.77, 0.97, and 0.90, respectively. From Figure 8(c), when the number of samples is 500, the four algorithms YOLO, SSD, YOLOv4-ASFF, and YOLOv5 have detection F1 values of 0.70, 0.78, 0.98, and 0.90, respectively.

The variation of frames per second for the four detection algorithms is shown in Figure 9. In Figure 9(a), the four algorithms, YOLO, SSD, YOLOv4-ASFF, and YOLOv5, are finally able to achieve frame rate values of 22, 25, 29, and 34 under the training dataset, respectively. In Figure 9(b), the four algorithms YOLO, SSD, YOLOv4-ASFF, and YOLOv5 are finally able to achieve frame rate values of 23, 25, 30, and 35 under the validation dataset, respectively. Compared to YOLO and SSD, YOLOv4-ASFF and YOLOv5 are able to reach stable frame rate values faster, thus indicating that these two algorithms are more efficient in detecting images.

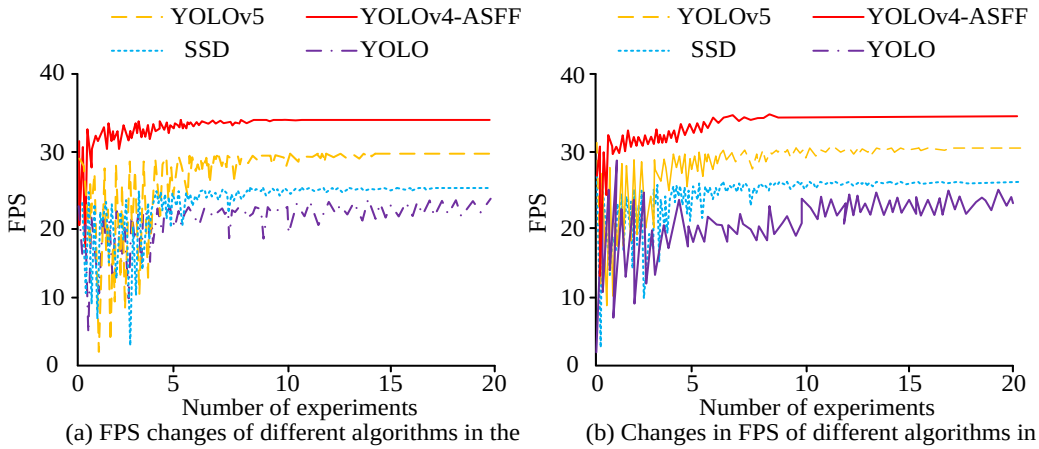
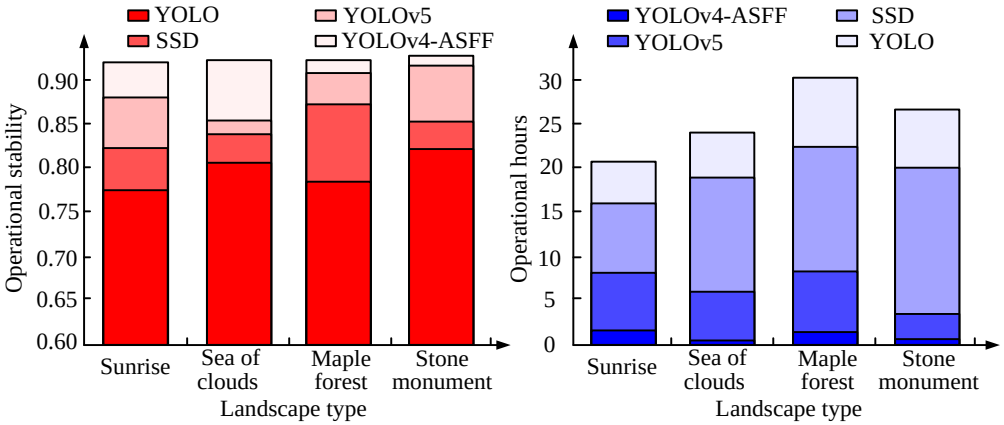


Figure 9: Changes in frames per second for each algorithm

4.2 Scenic landscape recognition system application effect analysis

In addition to verifying that the YOLOv4-ASFF algorithm has a better performance advantage in image detection, the study further utilized the above detection algorithms to build a real-time recognition system for scenic landscape images respectively, and compared the image recognition effect of each system in practical applications (see Figure 10).



(a) Operational stability of different recognition systems (b) Running time of different recognition systems

Figure 10: Operation stability and identification time of each system

Figures 10(a), (b) show the operation stability and recognition time of the four recognition systems, respectively. Four natural landscapes, namely sunrise, sea of clouds, maple forest and stone monument, are selected as test objects. In Figure 10(a), the recognition system built by the YOLOv4-ASFF algorithm has an operational stability as high as 0.92, 0.93, 0.92, 0.94 under the four kinds of natural landscapes, which is much higher than that of the recognition system built by the other three algorithms. In Figure 10(b), the recognition time of the recognition system built by the YOLOv4-ASFF algorithm under the four natural landscapes is 2.3s, 0.8s, 2.9s, and 1.2s, respectively, which is much lower than that of the recognition system built by the other three algorithms.

The recognition of three natural landscape images by the traditional YOLO recognition system and the recognition system built by the YOLOv4-ASFF algorithm are shown in Figure 11, respectively. Combined with Figure 11(a), (b), the optimized recognition system is able to better recognize the details in the natural landscape images, including sunrise, inscription text, pedestrians, and so on. The recognition system constructed using the YOLOv4-ASFF algorithm has better practical application results.

(a) Optimized recognition effect of landscape recognition system

(b) Recognition effect of traditional landscape recognition system

Figure 11: Actual landscape recognition situation of the two-recognition system

5 Discussion

To enhance the accuracy of detecting landscape images in scenic areas, this study optimized the traditional YOLOv4 TD algorithm by introducing ASFF. The optimized YOLOv4 algorithm is then combined with ASFF to create the fourth generation of adaptive spatial feature optimization primary TD system for real-time images of scenic landscapes, known as YOLOv4-ASFF. This approach improves the TD accuracy of YOLOv4. The system's ability to detect and manage landscape images in scenic areas is confirmed by its real-time detection capabilities. In the comparative analysis, the YOLOv4-ASFF proposed in this study demonstrates significant performance advantages compared with existing related work. The optimized YOLOv4 algorithm achieved not only higher scores in terms of precision, recall, and F1 value through the introduction of ASFF, but also provided a significant improvement in real-time performance when dealing with complex landscape environments. The YOLOv4 algorithm achieved high precision, recall, and F1 scores of 0.96, 0.97, and 0.98, respectively. In comparison, Jahani A et al. achieved a coefficient of determination of 0.878 in assessing the aesthetic quality of forest landscapes using machine learning techniques, and Peng X et al. achieved a composite similarity score of 0.92 in transforming landscape photos based on recurrent generative adversarial network models. Although these studies performed well in their respective domains, the precision and recall rates are lower than those of the algorithm proposed in this study for the complex task of detecting scenic landscapes. Additionally, this study demonstrated exceptional performance in the frame rate test, achieving processing speeds of up to 30 fps without compromising detection accuracy. This was critical for real-time surveillance systems. Compared to the other two approaches in the related work, YOLOv4-ASFF outperformed mainly because the ASFF mechanism and the efficient network architecture design significantly improve the algorithm's ability to detect multi-scale targets, especially in complex scenic environments, which enables more accurate identification and localization of targets of different sizes. In addition, optimizing the YOLOv4 algorithm improved not only the accuracy of TD, but also significantly increased the processing speed, enabling the algorithm to meet the dual requirements of speed and accuracy for real-time TD systems.

In this study, three comparison models (YOLO, SSD, and YOLOv5) were introduced to test the performance of YOLOv4-ASFF. The YOLOv4-ASFF algorithm-based recognition system for scenic landscape images achieved an operational stability of 0.92, 0.93, 0.92, and 0.94 under four natural landscapes, namely sunrise, cloud sea, maple forest, and stone monument, and a recognition time of 2.3 s, 0.8 s, 2.9 s, and 1.2 s, respectively, surpassing the performance of the other three models. The analysis below explained why YOLOv4-ASFF outperforms other models in specific usage scenarios. YOLOv4-ASFF optimized the YOLOv4 framework with ASFF, which significantly improves the model's recognition ability for targets of different sizes. The ASFF mechanism can dynamically adjust the weights of feature fusion according to the target sizes, which is especially important in multi-scale TD in scenic landscapes. Furthermore, YOLOv4-ASFF utilized efficient backbone network and feature fusion techniques, including CSPNet and PANet, to improve both detection speed and accuracy. In comparison, SSD was less accurate in processing small-size targets due to its limited method of detecting directly on feature maps at different scales, especially in complex landscape environments. YOLOv5, while improved in speed and accuracy, lacked the adaptive feature fusion mechanism found in YOLOv4-ASFF and did not perform as well as YOLOv4-ASFF for highly complex backgrounds and multi-scale targets. The YOLOv4-ASFF architecture had been optimized to provide a significant performance advantage in real-time TD scenarios in scenic landscapes, particularly in dealing with multi-scale TD tasks under changing light and complex background conditions.

In summary, this study has improved the YOLOv4 algorithm, achieving breakthrough performance in real-time TD in scenic landscapes. It also outperforms existing related work in key performance metrics, such as precision, recall, F1 value, and frame rate. This result demonstrates the superiority of the improved algorithm and provides a new direction for subsequent research on real-time TD in complex environments. It has important academic value and practical application potential.

6 Conclusion

To enhance the detection effectiveness of scenic landscape images, this study utilizes the upgraded YOLOv4 algorithm to optimize and evaluate the real-time TD system intended for scenic landscapes. The study's results indicated that by comparing the changes in the loss function curves of YOLO, SSD, YOLOv4-ASFF and YOLOv5 algorithms, it was found that the YOLOv4-ASFF algorithm performed the best, and its training loss started to stabilize at the 7th epoch, while the other algorithms required 17, 12 and 9 epochs, respectively. When the sample size was 500, YOLOv4-ASFF achieved high scores of 0.96, 0.97, and 0.98 for detection accuracy, recall rate, and F1 value, respectively, outperforming the other algorithms significantly. Furthermore, YOLOv4-ASFF and YOLOv5 demonstrated exceptional performance in the frame rate test, achieving 29 fps and 34 fps, respectively. Conversely, YOLO and SSD exhibited a lower frame rate of 22 fps and 25 fps, respectively. During the analysis of the scenic landscape recognition system's performance, it was found that the system developed using the

YOLOv4-ASFF algorithm demonstrated high operational stability rates of 0.92, 0.93, 0.92, and 0.94 for four natural landscapes: sunrise, sea of clouds, maple forest, and stone monument. Moreover, the recognition time for these landscapes was low, ranging from 0.8 s to 2.9 s, with an average of 1.6 s. Furthermore, the enhanced recognition system demonstrates the ability to detect landscape image attributes with greater accuracy than the conventional recognition solution. To summarize, the TD algorithm developed in this research presents superior proficiency and yields improved outcomes in real-world scenarios. However, there are still some limitations in this study, such as the need for improved recognition performance under extreme lighting and complex backgrounds, and the absence of coverage for all possible types of natural landscapes in the current test. Future research should expand its focus onto more landscape types.

7 Future work

The optimization model YOLOv4-ASFF, designed in this research, has achieved significant results in real-time TD in scenic landscapes. However, the significance of this research extends beyond the field of intelligent monitoring of scenic landscapes. Future work will explore the algorithm's potential in other areas, such as intelligent transport systems, unmanned surveillance security, automated agricultural monitoring, and rapid response to natural disasters. These areas require efficient and accurate real-time TD techniques. The challenge of balancing computational efficiency, real-time performance, and resource consumption of algorithms is particularly relevant for practical deployments. This is especially true in resource-constrained environments, such as the use of UAVs for on-site monitoring during natural disasters. Ensuring algorithm performance while reducing energy consumption is a major challenge. In addition, future research should focus on improving the model's generalization ability in diverse environmental conditions and complex backgrounds through in-depth adaptive improvements. Furthermore, future research is planned to investigate the utilization of multimodal data sources, such as infrared and radar fusion techniques, to further improve the model's ability to detect targets in extreme weather conditions and low-light environments. Additionally, the latest advances in deep learning, such as self-supervised learning and meta-learning strategies, can be combined with model training methods that require only a small amount of labeled data. This approach can help reduce the cost of large-scale data labeling and improve the adaptability of models. Finally, with the importance of AI ethics and privacy protection in mind, future research will focus on ensuring the interpretability and fairness of algorithms to promote the sustainability and social responsibility of the technology. In summary, the improved YOLOv4 algorithm and related technologies will play an important role in a wider range of fields, promoting the development of intelligent monitoring technologies and bringing positive impacts in practical applications.

References

[1] Syamimi Abdul Khalil, Shuzlina Abdul Rahman, Sofianita Mutalib, and Nurin Mirza Afiqah Andrie Dazlee. Object detection for autonomous vehicles with sensor-based technology using YOLO. International journal of intelligent systems and applications in engineering, 10(1):129-134, 2022. https://doi.org/10.18201/ijisae.2022.276

[2] Muhammed Enes Atik, Z. Duran, and Roni ÖZGÜNLÜK. Comparison of YOLO versions for object detection from aerial images. International journal of environment and geoinformatics, 9(2):87-93, 2022. https://doi.org/10.30897/ijegeo.1010741

[3] Yunyun Song, Zhengyu Xie, Xinwei Wang, and Yingquan Zou. MS-YOLO: object detection based on YOLOv5 optimized fusion millimeter-wave radar and machine vision. IEEE Sensors journal, 22(15):15435-15447, 2022. https://doi.org/10.1109/JSEN.2022.3167251

[4] Xin Shen, Xudong Sun, Huibing Wang, and Xianping Fu. Multi-dimensional, multi-functional and multi-level attention in YOLO for underwater object detection. Neural computing and applications, 35(27):19935-19960, 2023. https://doi.org/10.1007/s00521-023-08781-w

[5] Sugiarto Wibowo, and Indar Sugiarto. Hand symbol classification for human-computer interaction using the fifth version of YOLO object detection. CommIT (communication and information technology) journal, 17(1):43-50, 2023. https://doi.org/10.21512/commit.v17i1.8520

[6] Ali Jahani, Maryam Saffariha, and Pegah Barzegar. Landscape aesthetic quality assessment of forest lands: an application of machine learning approach. Soft computing, 27(10):6671-6686, 2023. https://doi.org/10.1007/s00500-022-07642-3

[7] Xianlin Peng, Shenglin Peng, Qiyao Hu, Jinye Peng, Jiaxin Wang, Xinyu Liu, and Jianping Fan. Contour-enhanced CycleGAN framework for style transfer from scenery photos to Chinese landscape paintings. Neural computing and applications, 34(20):18075-18096, 2022. https://doi.org/10.1007/s00521-022-07432-w

[8] Kai Zhou, Zhendong Zhang, Rui Yuan and Enqing Chen. A deep learning algorithm for fast motion video sequences based on improved codebook model. Neural ccomputing and applications, 35(6):4353-4368, 2023. https://doi.org/10.1007/s00521-022-07079-7

[9] Takuya Kikuchi, Tomohiro Fukuda, and Nobuyoshi Yabuk. Diminished reality using semantic segmentation and generative adversarial network for landscape assessment: evaluation of image

inpainting according to colour vision. Journal of computational design and engineering, 9(5):1633-1649, 2022. https://doi.org/10.1093/jcde/qwac067

[10] Guofa Li, Zefeng Ji, Xingda Qu, Rui Zhou, and Dongpu Cao. Cross-domain object detection for autonomous driving: A stepwise domain adaptative YOLO approach. IEEE Transactions on intelligent vehicles, 7(3):603-615, 2022. https://doi.org/10.1109/TIV.2022.3165353

[11] Jeonghun Lee, and Kwang-il Hwang. YOLO with adaptive frame control for real-time object detection applications. Multimedia tools and applications, 81(25):36375-36396, 2022. https://doi.org/10.1007/s11042-021-11480-0

[12] Siyuan Liang, Hao Wu, Li Zhen, Qiaozhi Hua, Mohammad Mehedi Hassan, and Keping Yu. Edge YOLO: real-time intelligent object detection system based on edge-cloud cooperation in autonomous vehicles. IEEE Transactions on intelligent transportation systems, 23(12):25345-25360, 2022. https://doi.org/10.1109/TITS.2022.3158253

[13] Raidah Salim Khudeyer, and Noor Mohammed Almoosawi. Fake Image Detection Using Deep Learning. Informatica, 47(7):115-120, 2023. https://doi.org/10.31449/inf.v47i7.4741

[14] Xiaojian Wang, Xiaoye Sun, and Zixuan Wang. Construction of visual evaluation system for building block night scene lighting based on multi-target recognition and data processing. IET Circuits, devices and systems, 17(3):149-159, 2023. https://doi.org/10.1049/cds2.12154

[15] Mehdi Gheisari, Hooman Hamidpour, Yang Liu, and Peyman Saedi. Data mining techniques for web mining: a survey. Artificial intelligence and applications, 1(1):3-10, 2022. https://doi.org/10.47852/bonviewAIA2202290