Application Research of Object Detection Model Based on Deep Learning in Video Surveillance Process

Xuechun Wang
Puyang Institute of Technology
Henan University, China
wxc202302@163.com

Feifei Cheng
Faculty of Engineering
Huanghe Science and Technology University, China
chengfeifei@hhstu.edu.cn

Junhao Jia
School of Computer Science and Technology, Henan Polytechnic University, China
jhstrive@home.hpu.edu.cn

Abstract: *The continuous progress of society has put forward higher requirements for the performance of current video surveillance, and object detection is an essential research content in the video surveillance process. Traditional object detection methods have disadvantages such as poor generalization ability, large limitations, and weak adaptability, which cannot meet the current social demand for object detection in video surveillance. Therefore, to promote the effectiveness of urban video surveillance, an Improved Fast Regional Convolutional Neural Network (IFast R-CNN) model is constructed. This model has been introduced with optimization strategies such as the K-Means Clustering Algorithm (K-means), optimized Non-Maximum Suppression (NMS) algorithm, weighted penalty method, and Region of Interest Align (RoIAlign) layer to raise the accuracy of object detection in video surveillance. This study conducts relevant research to assess the effectiveness of the proposed model. This model has better convergence performance compared to the other two comparative models, and it tends to stabilize after 56 iterations. The Area Under the Curve (AUC) of the model can arrive at 0.989. After verifying the effectiveness of the optimization strategy, it is found that the mean accuracy of the optimized model can reach 96.13%. Therefore, the model constructed by this research institute has good object detection performance and great potential for application in video surveillance processes.*

Keywords: *object detection, deep learning, IFasterR-CNN, video surveillance, K-means.*

Received May 26, 2025; Accepted December 08, 2025
https://doi.org/10.34028/iajit/23/4/7

1. Introduction

Currently, the construction of smart cities is constantly moving forward, and the demand for Video Surveillance (VS) is increasing. The intelligent VS system is a type of video image acquisition system based on electronic surveillance devices, which achieves automation and intelligent acquisition of events and information. Advanced intelligent VS can utilize visual computing to provide reliable real-time warnings for system managers, thereby ensuring the safety of real production and life. In addition to its significant importance in the field of public safety, intelligent VS also has a huge demand and wide application prospects. Object Detection (OD) in VS is an important research topic in computational vision. OD refers to the detection of all targets in a video or image. Currently, OD tasks have significant practical value in various production fields of society and have a significant impact on people's daily lives. Traditional OD methods mainly use histograms of oriented gradients, scale invariant feature transformation, sliding window, etc., which have poor generalization ability, large limitations, weak adaptability, and other shortcomings [1, 7, 18, 26]. As a result, video OD methods based on Deep Learning (DL) have emerged. The intelligence in DL is reflected in the reasonable utilization of neural networks, which can significantly reduce the amount of data required for processing and extract image features well. Although DL algorithms have higher detection accuracy, they still face challenges such as long training cycles and large memory consumption. To promote the target detection accuracy in the VS process, this study proposes an Improved Faster Region-based Convolutional Neural Network (IFasterR-CNN) OD model, which aims to improve the performance of VS systems to better adapt to the development of smart cities.

The main innovation of this study lies in four aspects.

1) The K-means Clustering Algorithm (K-means) is used to cluster the target scales in the training data, and the clustered scales are used as prior knowledge to initialize the anchor box. This makes the model more in line with the actual data distribution and improves the detection ability of targets at different scales.
2) By using the RoIAlign layer instead of RoIPooling and adopting the bilinear interpolation method, the rounding error of floating-point coordinates is avoided, thus more accurately extracting target features, especially when dealing with small targets,

the effect is significant.

3) The SoftNMS algorithm is introduced, and the scores of overlapping candidate boxes are reduced through linear or Gaussian weighting, instead of directly removing overlapping candidate boxes, which improves the recall rate of dense or occluded targets while maintaining detection efficiency.
4) Multiple data augmentation methods (such as cropping, color transformation, flipping, rotation, scaling, noise addition, etc.) are used to expand the original image. This increases the diversity of training samples in a low-cost manner, effectively preventing model overfitting and enhancing the adaptability of the model in complex scenes. This study is mainly composed of four sections. The first section isRelated Work, mainly analyzing the techniques used in the research. The second section is the improvement and optimization of the OD model, FasterR-CNN. The third section is the research assessment of the proposed model. The last section is a summary and outlook for the entire article.

2. Related Works

Intelligent VS is widely used in common fields such as public transportation, healthcare, and the military. The technology of intelligent VS mainly focuses on computer vision, and many scholars have conducted research on this topic. Yu and *et al*. [21]. Proposed a building architecture based on an intelligent power surveillance system to address issues such as difficulty in coordinated operation between systems in intelligent buildings. This architecture could provide determination support for building operation and maintenance management by combining big data and analyzing VS objects [15, 22]. Sharma and Sungheetha, [12]. Proposed a DL framework that combined CNN and SVM to address the phenomenon of camera-captured images being prone to data loss during dimensionality reduction. This framework detected abnormal events in continuous videos by preprocessing video data and performing dimensionality reduction on the images within it. Scholar Elhoseny, [2]. Developed a new VS technology for multi-OD and tracking. The technology used the optimal Kalman filter to achieve real-time tracking of moving objects in the video frame, and used the region growth model to convert video clips into morphological operations. The evaluation outcomes of this framework denoted that its maximum detection and tracking accuracy were 76.23% and 86.78%, respectively Pareek and Thakkarput, [9]. Forward a new VS technology and applied it to human behavior recognition to achieve recognition and surveillance of human activities. This technology utilized machine learning and DL to classify human abnormal activities, and had advantages such as data dimensionality reduction and action analysis for intelligent transportation system, Wan and *et al*. [14]. Discussed the real-time VS technology. Research found that the increasing amount of streaming media brought more inconvenience to operators in capturing suspicious actions. Therefore, a long video event retrieval method was proposed. This method could extract the required video clips from a massive amount of long videos, significantly improving the efficacy and detection accuracy of semantic description Zhang and *et al*. [24]. Constructed an intelligent poultry farming base environment VS system with the Internet of Things from the perspective of agricultural intelligence. The system used the ESP-12F WIFI module as the control core and used sensors to monitor the temperature, humidity, lighting, and other information of the breeding base Intelligent VS technology has two core contents, namely OD and behavior analysis. OD is mainly used for real-time surveillance of object status and position information. Behavioral analysis mainly focuses on operators. With the quickgrowth of computer science and technology, the intelligent method of combining DL with OD has been highly favored by scholars from all walks of life, and has also driven a research boom in the academic community. Junos and *et al*. [4]. Applied intelligent OD technology to the agricultural field and proposed a YOLO-P model optimized by YOLOv3. This model combined a densely connected lightweight neural network backbone and a multi-scale detection architecture, enabling the detection and localization of mature fruits in orchards Wei and *et al*. [17]. Raised a new feature fusion strategy to address the limitation of significant differences in the features extracted by traditional OD models. This strategy could adaptively select complementary components while avoiding the introduction of excessive redundant information Kim and *et al*. [5]. Scholars constructed a spike neural network based on the good OD performance of deep neural networks. This network had low power characteristics, but its training was relatively complex Li and *et al*. [6]. Proposed a bidirectional adversarial attack method to address security vulnerabilities caused by adversarial attacks in target detection. This method first pushed the predicted outcomes given by the detector away from the real class on the ground; Then, the foreground score was reduced according to the set confidence loss function; Finally, the confrontation samples were produced, and the model was trained by the confrontation method to speed up the rate of convergence of the algorithm Qin and *et al*. [11]. Designed a real-time salient OD network of ID-YOLO based on the OD performance in an autonomous vehicle. This target detection network could simulate the driver's selective attention mechanism to more accurately detect targets most relevant to driving safety Qin and *et al*. [10]. Constructed a dense sampling and detail enhancement network to improve the performance of small OD. The proposedmodel improved the resolution of the feature map and expanded its receptive field.

The shape, size, and appearance of the target in the video can all affect the detection accuracy of the model.

The above OD techniques can solve the detection limitations caused by occlusion and blurringto some extent, but there are still shortcomings, such as complex operation and difficulty in control. Therefore, the IFasterR-CNN OD model is proposed in this study to improve the performance of VS systems to better adapt to the development of smart cities.

3. OD Model Based on IFASTERR-CNN and its Application in VS Process

The OD model plays an essential role in VS, and this study enhances the FasterR-CNN model. The overall optimization strategy includes the following aspects:Firstly, K-meanisintroduced to cluster the detection targets proportionally. Afterwards, the traditional NMS algorithm in the network is improved and the Soft Non-Maximum Suppression (NMS) algorithm is proposed to reduce the score of overlapping candidate boxes. To address the phenomenon of sample imbalance, weighted penalties are introduced to enhance the model's generalization ability. Finally, the RoIAlign layer is studied to replace the RoiPooling layer to improve the accuracy of target feature mapping.

3.1. OD Model Based on FASTERR-CNN

Under the speedadvancement of DL technology and significantly improved computer computing power, the performance of OD models based on DL in practical OD tasks is quite excellent. The OD model based on DL usually divides the OD task into two stages: The first stage is to use the algorithm to obtain candidate box regions on the image, and the second stage is to classify and correct the candidate box regions Zhang and *et al.* [25]. The structure of the Region-based Convolutional Neural Network (R-CNN) is displayed in Figure 1.

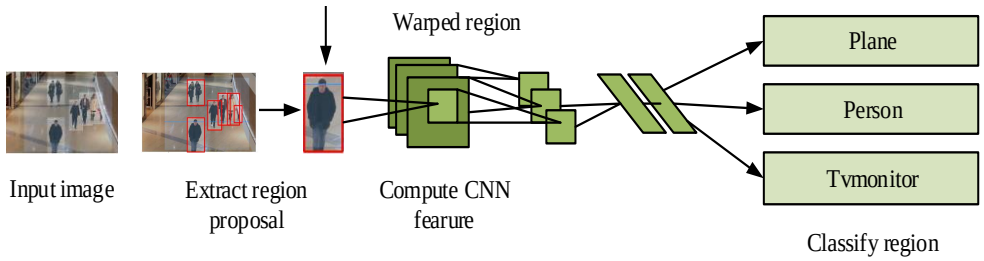


Figure 1. R-CNN structure.

As shown in Figure 1, R-CNN can be mainly divided into three steps. Firstly, a selective search algorithm is used to extract candidate box regions and normalize them to ensure input consistency. Then, CNN is used to extract features from each candidate region for subsequent classification tasks. Finally, R-CNN classifies the target and determines its precise position in the image through boundary regression Zhang, [23]. Although R-CNN has made significant breakthroughs compared to traditional algorithms, it still has many limitations. For example, in image normalization, it is easy to cause stretching or truncation, resulting in significant distortion and loss of important image information Natarajan and Periyasamy, [8]. In addition, the normalized region requires a large amount of separate calculations, which can easily lead to repeated extraction of the same feature and cause more time consumption. FastR-CNN replaces the SVM classifier in R-CNN with Softmax, which has more efficient classification performance. In addition, FastR-CNN replaces the bounding box regression loss with SmoothL1Loss. The model integrates classification and bounding box regression, and can complete feature extraction and those of candidate box regions with just one model. The structure of FastR-CNN is expressed in Figure 2.

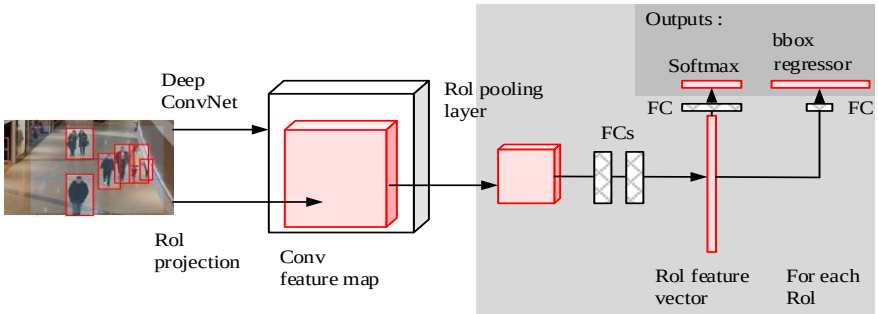


Figure 2. FastR-CNN structure.

As shown in Figure 2, in the OD process based on FastR-CNN, the input image is first subjected to feature extraction by a deep convolutional network to generate a feature map. Then, the Region of Interest Pooling (ROIPooling) layer maps the candidate regions onto the feature map, unifying the size of the feature vectors. The feature vectors of each candidate region are passed to two branches: The Softmax classifier and the bounding

box regressor. The former is used to determine the target category, while the latter is used to accurately predict the bounding box coordinates of the target. Ultimately, FastR-CNN can simultaneously perform target classification and localization in a single forward propagation. This method adopts an end-to-end approach and has made significant progress compared to traditional methods Giveki, [3]. The main drawback of FastR-CNN is that it uses a selection search algorithm, resulting in a longer model solving time. To further reduce the calculation time of FastR-CNN, FasterR-CNN is proposed. FasterR-CNN reduces the computational complexity of the overall model by obtaining relevant information from feature maps. The unique Region Proposal Network (RPN) in FasterR-CNN has successfully replaced the candidate box generation algorithm based on selection search, significantly improving the detection efficiency of the overall model. The basic structure of FasterR-CNN is denoted in Figure 3.

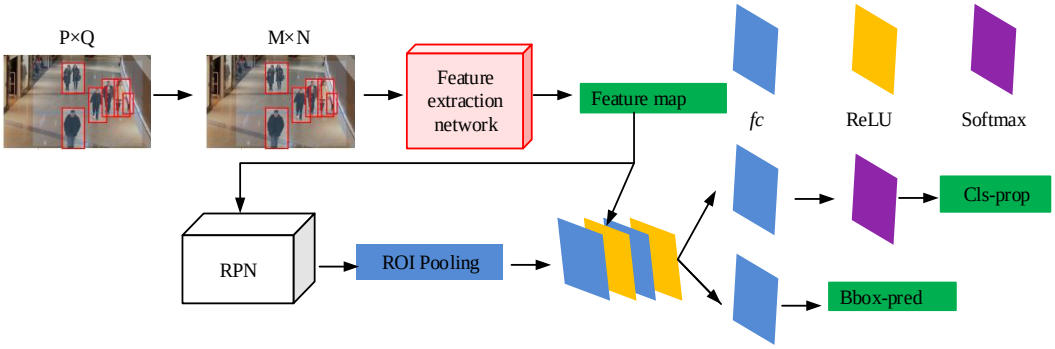


Figure 3. FasterR-CNN structure.

In Figure 3, FasterR-CNN replaces traditional candidate box generation methods with RPN to improve detection efficiency. The input image first undergoes a feature extraction network to generate a feature map. Subsequently, candidate regions are extracted through RPN, and the feature map size is standardized by ROIPooling. Finally, OD is achieved through classification and regression modules. The variation of feature maps in FasterR-CNN after the convolution operation is shown in Equation (1).

$$\left[\frac{M-m+2p}{s}\right]+1 \quad \left[\frac{N-n+2p}{s}\right]+1 \tag{1}$$

In Equation (1), *M*, *N* represent the width and height of the feature map, respectively. *m*, *n* denote the size of the convolutional kernel. *p* indicates the filling quantity. *S* means the grid size. After convolution operation, the input feature map remains the same size. The candidate box area is obtained by the RPN. After convolution, the RPN is divided into two paths, one of which is to classify candidate regions through 1×1 convolution and softmax classifier, and divide them into front backgrounds. The boundary box regression offset obtained by the 1×1 convolution of the other path. The losses of RPN and FastR-CNN together constitute the losses of FasterR-CNN. The loss function of RPN is expressed in Equation (2).

$$L_{cls}(p_i,p_i^*) = -log_{cls}[p_i p_i^* + (1-p_i)(1-p_i^*)] \tag{2}$$

In Equation (2), L_{cls} indicates the classification loss of RPN N_{cls}. represents the anchor box; *i* means the index of the anchor box. *i* refers to the probability that the *i*-$_{th}$ anchor box is the foreground target. *1*-P_i stands for the probability of the background. P_i^* refers to the true label of the *i*-$_{th}$ anchor box. The value of the label is 1 or 0, that is:

$$p_i^* = f(x) = \begin{cases} 0, & negative \\ 1, & positive \end{cases} \tag{3}$$

In Equation (3), 0 means the label of the background of the negative sample. 1 denotes the background label of the positive sample. The loss of FastR-CNN is usually a multi-category cross-entropy loss function, which is similar to RPN in other places. Therefore, the specific method of FasterR-CNN loss function is expressed in Equation (4).

$$L(p_i,t_i) = \frac{1}{N_{cls}}\Sigma_i L_{cls}(p_i,p_i^*) + \lambda\frac{1}{N_{reg}}\Sigma_i p_i^* L_{reg}(t_i,t_i^*) \tag{4}$$

In Equation (4), L_{reg} indicates the loss function of FasterR-CNN. N_{reg} means the anchor box. t_i refers to the probability that the *i*-$_{th}$ anchor box is the foreground target. t_i^* expresses the true label of the *i*-$_{th}$ anchor box. By continuously optimizing the regression model in RPN, the position of the target detection box can be continuously corrected. *A* position transformation is required, which is applied to the target prior box G^* obtained from the feature map through an RPN network, to obtain a regression box *G'*. *G'* should be as close as possible to the real target box *G*. The mathematical expression for positional changes is expressed in Equation (5).

$$\begin{aligned} &G'_x = A_w.d_x(A) + A_x \quad G'_y = A_w.d_y(A) + A_y \\ &G'_w = A_w.exp(d_w(A)) \quad G'_h = A_h.exp(d_h(A)) \end{aligned} \tag{5}$$

In Equation (5), $d_x(A)$, $d_y(A)$, $d_w(A)$, $d_h(A)$ stands for the transformation of unknown parameters. If G^* and *G* are relatively close, the transformation can be approximated as a linear transformation. Assuming the region of the feature map is Φ and the training parameter is *W*, then

there is Equation (6).

$$d_*(A) = W_*^T . \Phi(A) \tag{6}$$

In Equation (6) represents the parameter training process that is close to the true value. The expression for calculating the translation between G^* and G' is shown in Equation (7).

$$t_x^* = \frac{x^* - x_a}{w_a} \quad t_y^* = \frac{y^* - y_a}{h_a}$$
$$t_x = \frac{x - x_a}{w_a} \quad t_y = \frac{y - y_a}{h_a} \tag{7}$$

In Equation (7), G^* refers to the center horizontal axis, vertical axis, width, and height of the regression box, respectively. G^* corresponds to the regression box, the target prior box, and the real target box, respectively. To make the predicted values closer to the actual values, the SmoothL1 function is chosen for this study, as shown in Equation (8).

$$Smooth_{L1}(x) = \begin{cases} 0.5^* x^2 & |x| < 1 \\ |x| - 0.5 & otherwize \end{cases} \tag{8}$$

In the target detection task, the algorithm ultimately predict multiple candidate boxes, which need to be filtered out. Usually, NMS algorithm is used to suppress the generation of redundant boxes.

3.2. Optimization of OD Model Based on IFASTERR-CNN and Its Application in VS

FasterR-CNN has significantly improved recognition performance and detection accuracy compared to traditional R-CNN and FastR-CNN, which can meet most target detection needs. To further optimize the detection accuracy and operational efficiency of FasterR-CNN in complex scenarios, this study proposes relevant strategies. The proposed strategy aims to reduce redundant target boxes, accurately detect targets that are similar to the background, and detect occluded targets. The improvement strategy is divided into two parts: Improving the scale range of the detected object depends on the design of the initialization anchor box. There is a mismatch problem in the process of optimizing target feature extraction. For the former, the anchor box size in FasterR-CNN is usually used to determine the final detection box size. However, in VS, the targets vary widely and contain a large number of smaller targets. If the scale in the initial anchor box is too large, the size of the scale will be ignored due to its size, resulting in the omission of smaller scales Unver and *et al*. [13]. For the latter, since FasterR-CNN needs to quantify the candidate box coordinates twice during the target feature extraction stage, there is a certain deviation between the obtained candidate box coordinates and the initial regression position, resulting in a deviation in the target features extracted based on the candidate box coordinates. This offset will be more pronounced in small target detection, which in turn affects displacement regression and classification.

To overcome the shortcomings of the first point, this study adopts K-means. K-means clusters different types of objects proportionally to assist the FasterR-CNN algorithm, setting a scale for the initial anchor box to better match the size of the anchor box and the object in the same scene Yang and Song, [20]. To solve the mismatch problem of the RoIPooling layer in FasterR-CNN during target feature extraction, an improved RoIAlign method is put forward to raise the accuracy of target feature mapping. This study aims to address the two key scientific issues mentioned above and promote the recognition accuracy and efficiency of FasterR-CNN in VS. The proposed method can be divided into the following modules: basic network, candidate box coordinate extraction, target feature representation based on candidate boxes, detection box classification, and offset regression Wardana and *et al*. [16] and Yang, [19]. The IFasterR-CNN is shown in Figure 4.

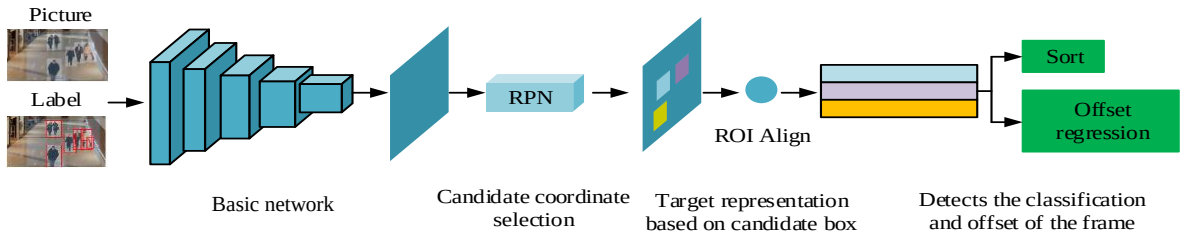


Figure 4. IFasterR- CNN structure.

In Figure 4, in IFasterR-CNN, the input image and label are first extracted as feature maps through the basic network. Then, RPN is used to generate candidate box coordinates and perform feature representation on each candidate box. Subsequently, the ROIAlign method is adopted instead of the traditional ROIPooling to solve the mismatch problem in the feature extraction process. Finally, the processed feature vectors are input into the fully connected layer, the target is classified through the Softmax layer, and the offset of the bounding box is predicted to accurately locate the target. The entire structure clusters different types of objects proportionally using the K-means method, assisting in setting the scale for the initial anchor box to better match the size of objects in the same scene. The original FasterR-CNN is an initial anchor box with a preset fixed size, which may not be sensitive to the scale of various types of targets in the VS environment. Based on this, the detection box calculated is not accurate enough. K-means can prune the tree through a small number of

known cluster samples, thereby classifying a certain cluster sample. In addition, it also has an optimization iteration function, which can iterate and modify the clustering of some samples once again to decide the clustering of some samples. Besides, for some smaller samples, this algorithm can effectively reduce the complexity of clustering. Therefore, this study utilizes the K-means to cluster the scales of targets in VS training data, and introduces the obtained clustering results as prior experience into the scale design of the initial anchor box, so that the algorithm can better match the target scales in traffic scenarios. The measurement methods of clustering algorithms often use Euclidean distance and cosine distance. Equation (9) is the mathematical expression for Euclidean distance.

$$d(x,y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2} \tag{9}$$

In Equation (9), *x, y* means two points on the plane. The expression for cosine distance is shown in Equation (10).

$$d=\frac{\sum_{i=1}^{n}x_i y_i}{\sqrt{\sum_i^n x_i^2}\cdot\sqrt{\sum_i^n y_i^2}} \tag{10}$$

In OD, the *IOU* between boxes is used as the metric. Assuming the midpoint coordinate of the box in the data annotation is (x_i, y_i, w_i, h_j) $j \in \{1,2,\ldots,k\}$, and the anchor width and height of the initial cluster center are (W_i, H_j) $i \in \{1,2,\ldots,k\}$, then the distance between each annotation box and each cluster center is shown in Equation (11).

$$\begin{aligned} &d_{anc}=1-IOU\left(\left(x_i,y_i,w_i,h_i\right),\left(x_i,y_i,W_i,H_i\right)\right) \\ &i\hat{I}\{1,2,\ldots,k\},j\hat{I}\{1,2,\ldots,k\} \end{aligned} \tag{11}$$

In Equation (11), *IOU* expresses the overlap between the cluster center and the annotation box. After all the annotation boxes are allocated, the calculation expression of the average width and average height of the annotation boxes in each cluster is shown in Equation (12).

$$\dot{W_i}=\frac{1}{N_j}\sum w_j,\dot{H_i}=\frac{1}{N_j}\sum h_j \tag{12}$$

In Equation (12), N_j means the amount of annotation boxes in the *i-th* cluster. The average width and height calculated by Equation (12) are used as the new cluster center point. The above process is repeated until the change in the cluster center is minimal. In the target detection task, FasterR-CNN will ultimately predict multiple candidate boxes. These candidates are relatively redundant and require appropriate methods to filter them. Usually, the NMS algorithm is used for deletion. However, traditional NMS tend to miss the detection of close-range targets in complex scenarios. Therefore, this study improves NMS and proposes the Soft Non-Maximum Suppression (SoftNMS) algorithm to reduce the score of overlapping candidate boxes. The changes made by SoftNMS are expressed in Equation (13).

$$s_i \leftarrow s_i f(iou(T,c_i)) \tag{13}$$

In Equation (13), *T* refers to the candidate box with the highest score. c_i denotes candidate boxes for all *i* classes. The calculation methods of SoftNMS mainly include linear weighting and Gaussian weighting. The calculation for linear weighting is shown in Equation (14).

$$s_i=\begin{cases} s_i & , iou(T,c_i)<N_t \\ s_i\left(1-iou(T,c_i)\right), & iou(T,c_i)\geq N_t \end{cases} \tag{14}$$

The Gaussian weighting calculation method is shown in Equation (15).

$$s_i=s_i e^{-\frac{iou(T,c_i)^2}{\sigma}}, \quad \forall c_i \notin D \tag{15}$$

The uneven distribution of sample data in actual detection tasks is highly likely to result in imbalanced samples. Sample imbalance can lead to a large number of types of samples in the overall model, weakening the model's generalization ability. Therefore, to alleviate this phenomenon, this study introduces weighted punishment to improve it. Weighted punishment is optimized by assigning weights to different categories of losses, and its function expression is indicated in Equation (16).

$$FL(p_t)=-\alpha_t(1-p_t)^{\gamma}log(p_t) \tag{16}$$

In Equation (16), a_t is used to adjust the category weight. *y* is used to adjust the weights of different hard-to-detect samples to realize the model's focus on hard-to-detect samples. *t* refers to different categories.

Finally, the RoIPooling layer in FasterR-CNN undergoes two quantization steps during target feature extraction. During this process, there is a certain deviation between the candidate box position and the initially regressed position, which is a mismatch problem and has a significant impact on the final detection accuracy. RoIAlign can cancel quantization operations and utilize bilinear interpolation to gain image values. Therefore, this study proposes an improved method of using the RoIAlign layer instead of the RoIPooling layer to improve the accuracy of target feature mapping. ROIAlign does not have quantization operations, so there is no error caused by rounding floating-point coordinates. Therefore, the construction of the IFasterR-CNN target monitoring model has been completed.

The OD model is an essential component of VS systems, and VS systems play an irreplaceable and important role in contemporary society. The OD model in VS systems mainly analyzes and provides feedback on video content for timely warning and decision-making. The intelligent VS system, combined with IFasterR-CNN, is shown in Figure 5.

As shown in Figure 5, the intelligent VS system combined with IFasterR-CNN inputs the monitoring video into the entire system for quality inspection. If the quality is poor, then it gives up and skips to the next frame. When the number of discarded frames exceeds the threshold, the system generates an alarm. When the quality of the image reaches a certain standard, the image is input into the OD model for OD.

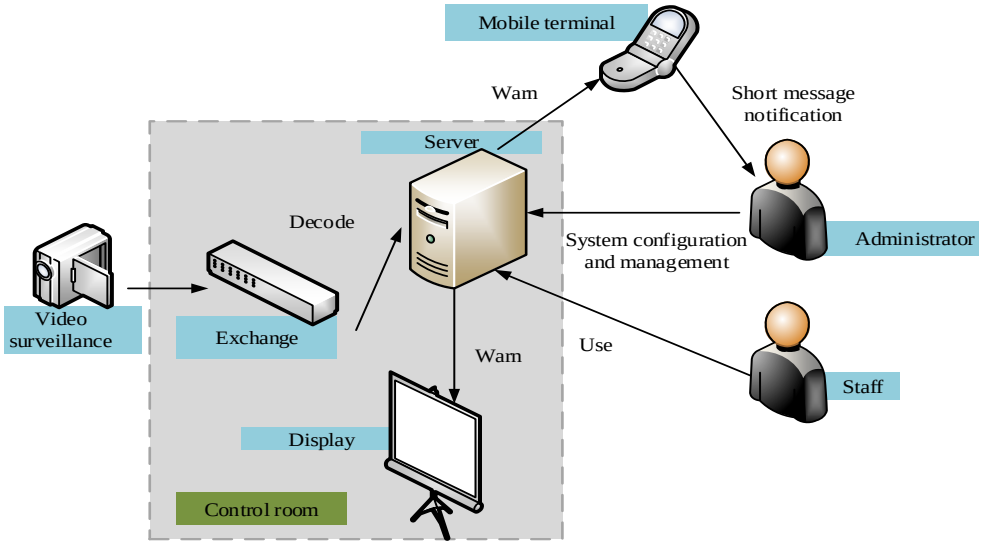


Figure 5. Intelligent VS system combined with IFasterR-CNN.

Finally, it analyzes the detection results to determine if they meet the triggering conditions, provides warnings, and records the detected objects and detection results. If it has not yet been met, it proceeds to the next screen.

4. Performance Verification of OD Model Based on IFASTER R-CNN

This section verifies the effect of the proposed IFasterR-CNN OD model. The experiment is conducted from two aspects, first verifying the operational performance of IFasterR-CNN, and then analyzing its application effect in VS.

4.1. Performance Verification of IFASTERR-CNN OD Model

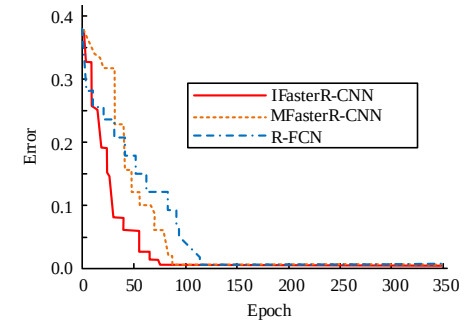
The data used in this study come from daily monitoring of various areas of urban street traffic, using cameras to capture images of various areas and various time periods. After analysis, the monitored objects are mainly pedestrians. Therefore, when establishing a dataset, it is necessary to ensure that the pedestrian morphology and scale in the dataset are different. This study collects a total of 4,260 photos of 1920×1080. To improve the generalization effect of the model, a large amount of data needs to be used for training, which requires data augmentation. Data augmentation can extract new training samples from the original image at a minimal cost, while expanding the dataset. Data augmentation can effectively prevent overfitting of the model by cutting, changing the color difference of the image, flipping, rotating, changing the size of the image, and adding noise and blur. Building OD models using DL methods often requires extremely high hardware requirements and a large amount of computation. When training IFasterR-CNN, its SHIYANP feature extraction network could adopt different pre-training modes, but its parameter settings were basically consistent. Table 1 lists the specific configuration options and parameter settings based on the server in this study.

Table 1. Experimental configuration and parameter settings.

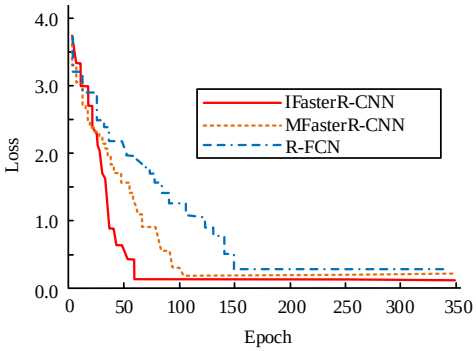
Experimental environment configuration		**Parameter setting**	
Item	Set value	Parameter	Value
GPU	Teslat4	Optimizer	SGD
Pytorch	1.2	Initialize the learning rate	0.001
Hard disk	16GB	Epochs	8
Paython	3.7	Momentum	0.9
CUDA	10.1	Weight attenuation coefficient	0.0001
Ubuntu	18.04LTS	RPN Number of samples	256
CPU	Intel(R)CPU@2.00GHz	IFaster-RCNN sample number	128
/	/	Loss function	CrossEntropyLoss

To better highlight the performance of IFasterR-CNN, this study selects the other two popular target detection models for comparative analysis. The two models are the Multi-scale sensing Fast-RCNN model (MFasterR-CNN)

and the R-FCN model. In Figure 6, the convergence performance of the three is first compared. When the iteration reaches 56 times, the Error and Loss values of IFasterR-CNN both drop to the lowest point, and the convergence of the model tends to be stable with the increase of iterations. Compared to IFasterR-CNN, MFasterR-CNN and R-FCN are trained 76 times and 110 times, respectively. IFasterR-CNN model has the best convergence performance.



a) Eeeor.



b) Loss.

Figure 6. Convergence performance of the three models.

The performance of three different models is verified through the image data collected through experiments. Figure 7 shows the F1 values for three models. In Figure 7, after iterative processing, the F1 values of each model have been improved. However, when it reaches a critical point, the amplitude of F1 change is very small and close to a constant value. After stabilization, the F1 value of the IFasterR-CNN model is 0.957, which is 0.045 higher than that of MFasterR-CNN and 0.068 higher than that of R-FCN.

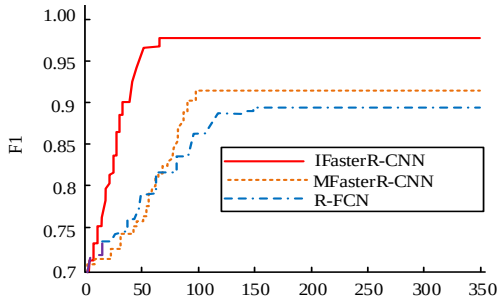
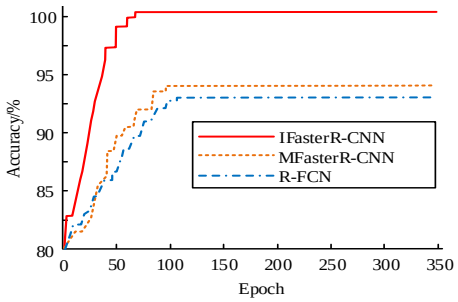
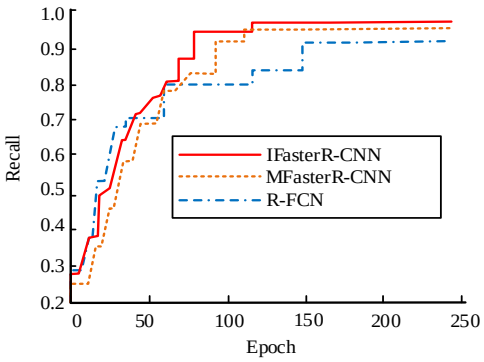


Figure 7. F1 values of the three models.

To evaluate the recognition accuracy of the model, the three models are tested using the enhanced data. Figure 8 shows the comparison of recognition accuracy and recall value of the three models after data enhancement. In Figure 8-a), the recognition accuracy of the IFasterR-CNN algorithm after data enhancement is 99.06%. Compared with MFasterR-CNN and R-FCN, the prediction accuracy of the IFasterR-CNN algorithm is 5.12% and 6.01% higher, respectively. In Figure 8-b), the Recall value of the IFasterR-CNN model after image enhancement is 97.56%, which is about 2.12% higher than that of MFasterR-CNN and 3.11% higher than that of R-FCN. Therefore, IFasterR-CNN has a better target detection effect.



a) Accuracy.



b) Recall.

Figure 8. Recognition accuracy of the three models.

TheROCof the three models are compared with the AUC value, as shown in Figure 9. The outcomes indicate that the AUC value of IFasterR-CNN is 0.989. The AUC of MFasterR-CNN is 0.968, 0.021 lower than that of the IFasterR-CNN model. The AUC of the R-FCN model is 0.960, 0.029 lower than that of the IFasterR-CNN model. This result further confirms that IFasterR-CNN has higher target detection accuracy.

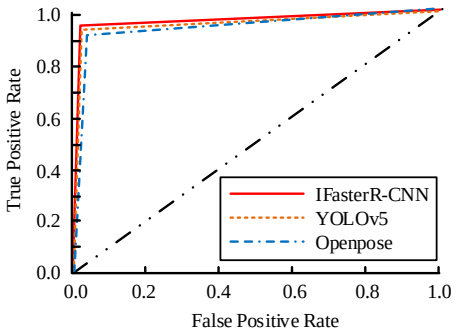
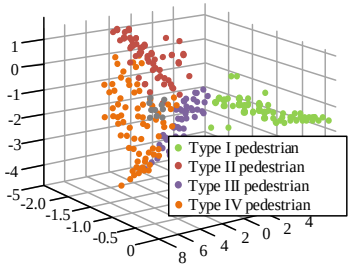


Figure 9. AUC values for the three models.

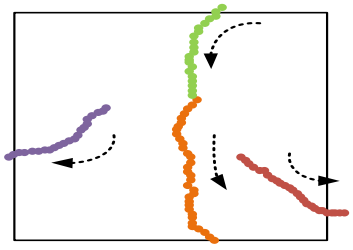
To verify the degree of optimization of model

performance by the introduction of K-means, this study discusses the clustering effect of target scales in VS training data. Figure 10 shows the moving track fragments, clustering results, and pedestrian behavior rules of the area, and confirms that the clustering results of K-means can reflect the movement rules of pedestrians in the area. In Figure 10-a), correlation refers to the locus fragment being located near the area of focus. Since each track fragment contains only 127 track points, the track fragment length is limited, and the boundary is expanded by 10pixel to include more nearby tracks during screening. In Figure 10-b), the four clusters contain 179, 37, 280, and 101 locus fragments, respectively. Figure 10-c) reflects the direction of motion containing the central locus in each cluster. Green clusters indicate that pedestrians enter the area from the top right. The inter-cluster distance here is small because the pedestrian has not yet begun to select.

a) Pedestrian track segments.



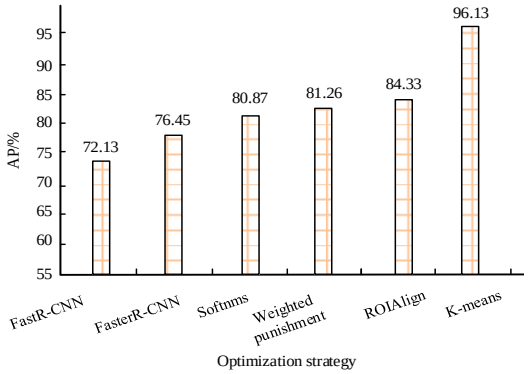
b) Clustering result.



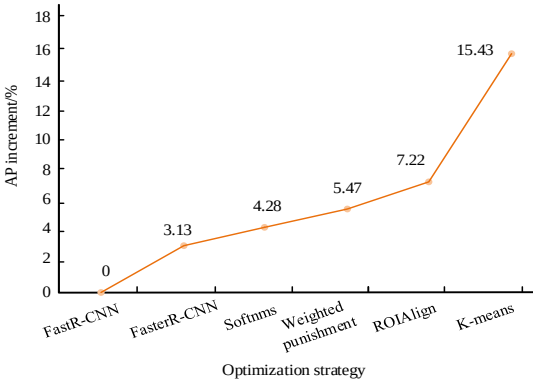
c) The law of pedestrian behavior.

Figure 10. Clustering effect of IFasterR-CNN model.

On this basis, the Average Precision (AP) of each optimization scheme is tested to verify the impact of each optimization scheme on the overall model target accuracy of the system. The AP value is the result obtained by averaging the accuracy corresponding to all Recall values. IFasterR-CNN combines the optimal strategy proposed in this project with the optimal strategy proposed in this project, transforming the optimal strategy into a cumulative equation. Figure 11 shows the specific strategy optimization effects in this model. After improving the optimization strategy proposed by the research institute, the average accuracy reaches 96.13%. Using the accuracy of ResNet50 as the standard, the graph shows the degree of improvement for each optimal strategy compared to the previous AP. In Figure 11, the optimization of K-means has a great influence on the average accuracy of the model trained with the current dataset. The SoftNMS algorithm, weighted penalty application, and ROIAlign algorithm all have certain optimization effects, but their optimization effects are relatively limited.



a) The influence of optimization strategy on AP.



b) The degree of impact of the optimization strategy.

Figure 11. Optimization effect of policies in IFasterR-CNN.

4.2. Analysis of the Application Effect of IFASTERR-CNN Model in VS Systems

The above verification results show that IFasterR-CNN is an effective target detection network. To verify the application effect of IFasterR-CNN in full scene surveillance of actual cities, this study conducts ablation experiments on complex scene surveillance datasets (https://gitcode.com/open-source-toolkit/d45a3/?utm_source=tools_gitcode&index=top&type=card&). The research results are shown in Table 2. The FrameSize in the table is the resolution size of the test video. In Table 2, after introducing K-means and ROIAlign algorithms, the mAP of the key technology models for DL-based surveillance video motion OD and pedestrian structured description on the full scene monitoring dataset increases by 2.68% and 1.78%,

respectively. The mAP of IFasterR-CNN, which simultaneously introduced two methods, reaches 90.12% on the entire scene dataset, and can process an average of 155 FPS.

Table 2. Experimental results on complex urbansurveillance dataset.

Model	Frame size	mAP(%)	FPS
FasterR-CNN	1920×1080	85.68	156
K-FasterR-CNN	1920×1080	86.97	154
R-FasterR-CNN	1920×1080	88.31	161
IFasterR-CNN	1920×1080	90.12	155

This study uses the above three modes to conduct experiments on pedestrian components in monitoring, and the test outcomes are displayed in Table 3. Among them, the R-FCN model has the best performance in detecting the head, upper body, lower body, and feet of pedestrians. MFasterR-CNN has the best performance for the inspection of pedestrian package parts. The IFasterR-CNN model has the highest mAP value for detecting the entire component.

Table 3. Test results of pedestrian parts.

Model	MAP	Head	The upper part of the body	The lower part of the body	Foot	Package
R-FCN	0.8741	0.9166	0.9368	0.8999	0.8129	0.8011
MFasterR-CNN	0.8945	0.9533	0.9544	0.9323	0.8547	0.8501
IFasterR-CNN	0.9112	0.9761	0.9811	0.9512	0.8712	0.8045

In the test dataset, three images captured by urban surveillance are selected and used for OD using MFasterR-CNN and IFaster R-CNN, respectively. The detection comparison outcomes are shown in Figure 12. The recognition rate of IFasterR-CNN is significantly higher than that of MFasterR-CNN. The two motorcycle pedestrians and some distant cars in the middle on the right side of Figure 12-a), as well as the bicycle pedestrians in the left corner of Figure 12-b), all of these missed targets can be successfully detected by IFasterR-CNN. In Figure 12-c), IFasterR-CNN also has good detection performance for nighttime targets.

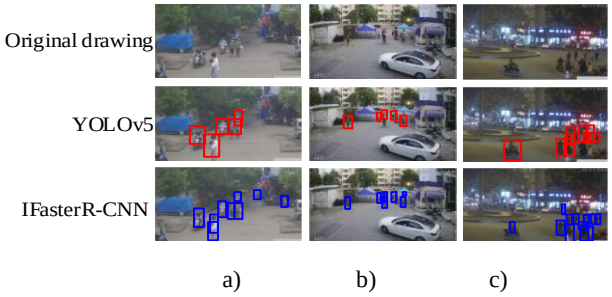


Figure 12. Comparison of detection results.

To further validate the generalization and superiority of the IFasterR-CNN model, the Common Objects in Context (COCO) dataset is used for testing and compared with four baseline models: YOLOv5, YOLOv8, EfficientDet-D1, and Detection Transformer (DETR). The COCO dataset is a large image dataset primarily used for computer vision tasks, containing over 330,000 daily scene images covering 80 object categories and semantic segmentation annotations. The experiment adopts mAP@0.5, mAP @[0.5:0.95], and FPS as evaluation metrics. The comparison results of the indicators of the five models are shown in Table 4. The proposed IFasterR-CNN model still demonstrates good performance in the COCO dataset, and the mAP@0.5 and mAP@[0.5:0.95] are the highest, at 70.82% and 52.84%, respectively, which are better than all other baseline models. This proves that through a series of strategies such as K-means anchor box optimization, RoIAlign, and SoftNMS, the IFasterR-CNN model has achieved more accurate classification and localization in complex scenes. The IFasterR-CNN model can process an average of 52 Frames Per Second (FPS), which is slightly lower than the YOLOv5 and YOLOv8 models, but still within an acceptable range, with the best overall performance. The results indicate that the proposed IFasterR-CNN model still has good generalization performance in large image datasets.

Table 4. Comparison results of indicators for five models

Models	mAP@[0.5:0.95](%)	mAP@0.5(%)	FPS
YOLOv5	48.47	66.85	98
YOLOv8	51.22	69.51	91
EfficientDet-D1	50.14	68.50	65
DETR	41.87	62.39	28
IFasterR-CNN	52.84	70.82	52

5. Conclusions

In response to the shortcomings of traditional OD methods, such as poor generalization ability, significant limitations, and weak adaptability, this study proposed an IFaster R-CNN to improve the performance of urban VS. To improve the accuracy of OD in VS, IFasterR-CNN was introduced with optimization strategies such as K-means, SoftNMS, weighted penalty method, and RoIAlign layer. This study conducted relevant experiments to verify the performance of IFasterR-CNN. In addition, two models, MFasterR-CNN and R-

FCN, were introduced for comparative verification. IFasterR-CNN had better convergence performance than MFasterR-CNN and R-FCN, and it became stable after 56 iterations. The F1 value of the IFasterR-CNN model was 0.957, which was 0.045 higher than that of MFasterR-CNN and 0.068 higher than that of R-FCN. The recognition accuracy of IFasterR-CNN was 99.06%, which was 5.12% and 6.01% higher than that of MFasterR-CNN and R-FCN, respectively. The Recall value of the IFasterR-CNN model is 97.56%, which was 2.12% and 3.11% higher than that of MFasterR-CNN and R-FCNR, respectively. The AUC value of IFasterR-CNN could reach 0.989, which was higher than the other two models. After verifying the effectiveness of the optimization strategy, it was found that the average accuracy of the optimized model could reach 96.13%. IFasterR-CNN performed well in VS systems. Therefore, IFasterR-CNN has good OD performance and can be applied in VS processes. However, this study did not conduct in-depth optimization on the training time, inference speed, and Graphics Processing Unit (GPU) memory requirements of the model. Complex feature extraction networks and RPN structures may result in high computational costs, which are not conducive to rapid deployment in resource-constrained environments. In addition, although the current model has a high processing speed in complex scenarios, it has not undergone stress testing in extremely high-resolution or multi-channel video streaming scenarios. Therefore, future research should explore model lightweighting and acceleration techniques, such as using more efficient backbone networks and applying knowledge distillation to transfer the capabilities of large models to small models. Thus, it is possible to significantly improve inference speed and reduce computational resource consumption while maintaining accuracy. At the same time, efforts should also be made to develop optimization frameworks for real-time applications, such as designing an adaptive frame rate adjustment mechanism that enables the system to dynamically adjust analysis strategies based on server load, ensuring system stability during large-scale deployment.

Acknowledgment

The research is supported by Postgraduate Education Reform and Quality Improvement Project of Henan Province, (No.YJS2023JD67); Private Education Brand Professional Construction Project of Henan Province Education Department, Computer Science and Technology, (No.ZLG201903); Industry-university Cooperative Education Project of Ministry of Education, Teaching Reform and Practice of Python Language Course Based on CDIO Mode, (No.220601221223311); Zhengzhou Local Universities Urgent (special) Major Construction Project-Data Science and Big Data Technology, Data Science and Big Data Technology, (No.ZZLG202301).

References

[1] Dhillon A. and Verma G., "Convolutional Neural Network: A Review of Models, Methodologies and Applications to Object Detection," *Progress in Artificial Intelligence*, vol. 9, no. 2, pp. 85-112, 2020. https://doi.org/10.1007/s13748-019-00203-0

[2] Elhoseny M., "Multi-Object Detection and Tracking (MODT) Machine Learning Model for Real-Time Video Surveillance Systems," *Circuits, Systems, and Signal Processing*, vol. 39, pp. 611-630, 2020. https://doi.org/10.1007/s00034-019-01234-7

[3] Giveki D., "Robust Moving Object Detection Based on Fusing Atanassov's Intuitionistic 3D Fuzzy Histon Roughness Index and Texture Features," *International Journal of Approximate Reasoning*, vol. 135, pp. 1-20, 2021. https://doi.org/10.1016/j.ijar.2021.04.007

[4] Junos M., Mohd Khairuddin A., Thannirmalai S., and Dahari M., "An Optimized YOLO-Based Object Detection Model for Crop Harvesting System," *IET Image Processing*, vol. 15, pp. 2112-2125, 2021. https://doi.org/10.1049/ipr2.12181

[5] Kim S., Park S., Na B., and Yoon S., "Spiking-Yolo: Spiking Neural Network for Energy-Efficient Object Detection," *in Proceedings of the AAAI Conference on Artificial Intelligence*, New York, vol. 34, no. 7, pp. 11270-11277, 2020. https://doi.org/10.1609/aaai.v34i07.6787

[6] Li X., Jiang Y., Liu C., Liu S., and et al., "Playing Against Deep-Neural-Network-Based Object Detectors: A Novel Bidirectional Adversarial Attack Approach," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 1, pp. 20-28, 2022. https://doi.org/10.1109/TAI.2021.3107807

[7] Nasir V. and Sassani F., "A Review on Deep Learning in Machining and Tool Monitoring: Methods, Opportunities, and Challenges," *The International Journal of Advanced Manufacturing Technology*, vol. 115, pp. 2683-2709, 2021. https://doi.org/10.1007/s00170-021-07325-7

[8] Natarajan S. and Periyasamy K., "A Silicosis-Focused Hybrid Architecture Leveraging Enhanced Segmentation and Feature Based Classification," *The International Arab Journal of Information Technology*, vol. 23, no. 2, pp. 283-296, 2026. DOI: 10.34028/iajit/23/2/09

[9] Pareek P. and Thakkar A., "A Survey on Video-Based Human Action Recognition: Recent Updates, Datasets, Challenges, and Applications," *Artificial Intelligence Review*, vol. 54, pp. 2259-

2322, 2020. https://doi.org/10.1007/s10462-020-09904-8

[10] Qin H., Wu Y., Dong F., and Sun S., "Dense Sampling and Detail Enhancement Network: Improved Small Object Detection Based on Dense Sampling and Detail Enhancement," *IET Computer Vision*, vol. 16, no. 4, pp. 307-316, 2022. DOI: 10.1049/cvi2.12089

[11] Qin L., Shi Y., He Y., Zhang J., and et al., "ID-YOLO: Real-Time Salient Object Detection Based on the Driver's Fixation Region," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15898-15908, 2022. https://doi.org/10.1109/TITS.2022.3146271

[12] Sharma R. and Sungheetha A., "An Efficient Dimension Reduction Based Fusion of CNN and SVM Model for Detection of Abnormal Incident in Video Surveillance," *Journal of Soft Computing Paradigm*, vol. 3, no. 2, pp. 55-69, 2021. https://doi.org/10.36548/JSCP.2021.2.001

[13] Unver M., Olgun M., and Turkarslan E., "Cosine and Cotangent Similarity Measures Based on Choquet Integral for Spherical Fuzzy Sets and Applications to Pattern Recognition," *Journal of Computational and Cognitive Engineering*, vol. 1, no. 1, pp. 21-31, 2022. https://doi.org/10.47852/bonviewJCCE2022010105

[14] Wan S., Xu X., Wang T., and Gu Z., "An Intelligent Video Analysis Method for Abnormal Event Detection in Intelligent Transportation Systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4487-4495, 2021. https://doi.org/10.1109/TITS.2020.3017505

[15] Wang Z., Xu P., Liu B, Cao Y., and et al., "Hyperspectral Imaging for Underwater Object Detection," *Sensor Review*, vol. 41, no. 2, pp. 176-191, 2021. DOI:10.1108/SR-07-2020-0165

[16] Wardana A., Hu S., Shimasaki K., and Ishii I., "Development of a Multi-User Remote Video Monitoring System Using a Single Mirror-Drive Pan-Tilt Mechanism," *Journal of Robotics and Mechatronics*, vol. 34, no. 5, pp. 1122-1132, 2022. https://doi.org/10.20965/jrm.2022.p1122

[17] Wei J., Wang S., and Huang Q., "F³Net: Fusion, Feedback and Focus for Salient Object Detection," *in Proceedings of the AAAI Conference on Artificial Intelligence*, New York, pp. 12321-12328, 2020. DOI:10.1609/aaai.v34i07.6916

[18] Wu X., Sahoo D., and Steven C., "Recent Advances in Deep Learning for Object Detection," *Neurocomputing*, vol. 396, pp. 39-64, 2020. https://doi.org/10.1016/j.neucom.2020.01.085

[19] Yang M., "Research on Vehicle Automatic Driving Target Perception Technology Based on Improved MSRPN Algorithm," *Journal of Computational and Cognitive Engineering*, vol. 1, no. 3, pp. 147-151, 2021. https://doi.org/10.47852/bonviewJCCE20514

[20] Yang Y. and Song X., "Research on Face Intelligent Perception Technology Integrating Deep Learning under Different Illumination Intensities," *Journal of Computational and Cognitive Engineering*, vol. 1, no. 1, pp. 32-36, 2022. https://doi.org/10.47852/bonviewJCCE19919

[21] Yu L., Nazir B., and Wang Y., "Intelligent Power Monitoring of Building Equipment Based on Internet of Things Technology," *Computer Communications*, vol. 157, pp. 76-84, 2020. https://doi.org/10.1016/j.comcom.2020.04.016

[22] Zhang J., Shi Y., Zhang Q., Liu C., and et al., "Attention Guided Contextual Feature Fusion Network for Salient Object Detection," *Image and Vision Computing*, vol. 117, pp. 104337, 2022. https://doi.org/10.1016/j.imavis.2021.104337

[23] Zhang L., "Enterprise Employee Work Behavior Recognition Method Based on Faster Region-Convolutional Neural Network," *The International Arab Journal of Information Technology*, vol. 22, no. 2, pp. 291-302, 2025. https://doi.org/10.34028/iajit/22/2/7

[24] Zhang X. and Zhi Y., "Design of Environment Monitoring System for Intelligent Breeding Base Based on Internet of Things," *Open Access Library Journal*, vol. 8, no. 10, pp. 1-9, 2021. DOI:10.4236/oalib.1108044

[25] Zhang X., Liu Y., Huo C., Xu N., and et al., "PSNet: Perspective-Sensitive Convolutional Network for Object Detection," *Neurocomputing*, vol. 468, pp. 384-395, 2022. https://doi.org/10.1016/j.neucom.2021.10.068

[26] Zhang Y., Song C., and Zhang D., "Deep Learning-Based Object Detection Improvement for Tomato Disease," *IEEE Access*, vol. 8, pp. 56607-56614, 2020. https://doi.org/10.1109/ACCESS.2020.2982456

Xuechun Wang, earned her Bachelor of Science in Mathematics and Applied Mathematics from Henan University of Science and Technology in 2004 and her Master of Engineering in Computer Application Technology from Daqing Petroleum Institute in 2007. Specializing in digital image processing. From 2007 to 2025, she served as a full-time faculty member at Huanghe Science and Technology University; from 2025 to present, she has been working at Puyang Institute of Technology, Henan University. Her scholarly contributions include 15 peer-reviewed journal articles, 2 academic monographs/textbooks, and leadership in 10 scientific research projects. She holds 5 granted invention patents and has received 8 academic honors for her research achievements.

Fiefei Cheng, is a lecturer at Huanghe Science and Technology University. She earned her Bachelor of Science in Electronic Information Science and Technology from the PLA Information Engineering University in 2011 and a Master of Science in Communication and Information System from Guizhou University in 2016. Cheng has contributed to the field through her role as a full-time lecturer while actively engaging in research. Her academic portfolio includes 1 peer-reviewed journal publication, 1 co-authored academic textbook, and participation in 6 scientific research projects. Prior to her academic career, she served as a Data Development Engineer at Beijing Venustech Inc. from 2016 to 2018, where she gained practical industry experience.

Junhao Jia, earned his B.S. degree in Software Engineering from Huanghe University of Science and Technology in 2022, and his M.S. degree in Software Engineering from Henan Polytechnic University in 2025. He is currently preparing to pursue a Ph.D. degree at the School of Computer and Artificial Intelligence, Wuhan University of Technology. His research interests include Image Processing, Multi-Source Data Fusion, and Their Applications.