Journal of Quality Measurement and Analysis
e-ISSN: 2600-8602
https://doi.org/10.17576/jqma.2103.2025.16

JQMA 21(3) 2025, 247-269
http://www.ukm.my/jqma

OBJECT DETECTION AND NEAR-MISS ANALYSIS IN TRAFFIC VIDEOS OF HIGH AND LOW QUALITY BASED ON THE YOLO MODEL

(*Pengesanan Objek dan Analisis Hampir Terlepas dalam Video Trafik Berkualiti Tinggi dan Rendah Berdasarkan Model YOLO*)

ZHU WEN, LEK MING LIM, MAJID KHAN MAJAHAR ALI*, LILI WU, AMR ELSHAHED & LEE JIAN AUN

ABSTRACT

The development of intelligent transportation systems has made traffic detection technology crucial for preventing near-miss events and enhancing road safety. You Only Look Once (YOLO), a popular target detection model, is widely used in traffic target detection. However, the performance of different YOLO models varies significantly on the same quality video data. Furthermore, different video qualities affect near-miss event detection differently. This paper conducts experiments on different YOLO models using high-quality and low-quality video data to assess their detection performance. Considering the expressway scenario in this study, we use manual reporting (based on human detection), the bird's-eye view method (based on image processing), and the distance neighbor method (based on data analysis) for near-miss detection. Results show that YOLOv5m has the highest precision, recall, and mAP@0.5 (0.990, 0.977, and 0.994) for high-quality video detection. YOLOv8m has the highest mAP@0.5 - 0.95 (0.908) for low-quality video detection. In high-quality video near-miss detection, manual reporting, the bird's-eye view method, and the distance neighbor method show high consistency, indicating that the latter two can replace manual calculation in specific scenarios. In low-quality video near-miss detection, all three methods yield identical results, highlighting the significant impact of clarity and fluency on near-miss detection.

Keywords: traffic object detection; YOLO model performance comparison; video data quality; near-miss detection; distance detection

ABSTRAK

Pembangunan sistem pengangkutan pintar telah menjadikan teknologi pengesanan trafik sangat penting untuk mencegah kejadian nyaris kemalangan dan meningkatkan keselamatan jalan raya. You Only Look Once (YOLO), sebuah model pengesanan lanjutan yang popular, digunakan secara meluas dalam pengesanan sasaran trafik. Namun, prestasi model YOLO yang berbeza menunjukkan variasi yang ketara pada data video dengan kualiti yang sama. Selain itu, kualiti video yang berbeza turut mempengaruhi pengesanan kejadian nyaris kemalangan secara berbeza. Kajian ini menjalankan eksperimen ke atas pelbagai model YOLO menggunakan data video berkualiti tinggi dan rendah untuk menilai prestasi pengesanan mereka. Dengan mengambil kira senario lebuh raya dalam kajian ini, kami menggunakan pelaporan manual (berdasarkan pengesanan manusia), kaedah pandangan mata burung (berdasarkan pemprosesan imej), dan kaedah jarak jiran (berdasarkan analisis data) untuk pengesanan kejadian nyaris kemalangan. Eksperimen ini menunjukkan bahawa YOLOv5m mempunyai ketepatan, kadar panggilan semula (recall), dan mAP@0.5 tertinggi (masing-masing 0.990, 0.977, dan 0.994) dalam pengesanan video berkualiti tinggi. YOLOv8m pula mencatatkan mAP@0.5 - 0.95 tertinggi (0.908) untuk pengesanan video berkualiti rendah. Dalam pengesanan kejadian nyaris kemalangan menggunakan video berkualiti tinggi, kaedah pelaporan manual, pandangan mata burung, dan jarak jiran menunjukkan tahap keserasian yang tinggi, menunjukkan bahawa dua kaedah terakhir boleh menggantikan pengiraan manual dalam situasi tertentu. Bagi pengesanan kejadian nyaris kemalangan menggunakan video berkualiti rendah, ketiga-tiga kaedah

memberikan keputusan yang sama, menekankan kesan ketara kejelasan dan kelancaran video terhadap pengesanan kejadian nyaris kemalangan.

Kata kunci: pengesanan objek lalu lintas; perbandingan prestasi model YOLO; kualiti data video; pengesanan nyaris; pengesanan jarak

1. Introduction

Serious traffic congestion has resulted from the rise of cars on the road, which has worsened traffic conditions (Chen 2023). Traffic congestion usually leads to slow vehicle movement and increased waiting time for parking. In this case, the distance between vehicles is reduced, increasing the possibility of near-miss events (Retallack & Ostendorf 2020). Near-miss events in traffic refer to those situations that almost lead to traffic accidents but ultimately do not cause actual harm or loss (Lim *et al*. 2024). These incidents usually involve close contact or near collisions between vehicles, pedestrians, or other traffic participants, but the real collision is avoided due to some reasons (such as timely response, speed control, or other intervention measures). By detecting near-miss events, timely warnings can be achieved, effectively reducing the probability of accidents (Liu 2024) and providing support for urban traffic flow optimization. Currently, there is extensive research on the detection of near-miss events, and each method has its applicability and limitations. Manual reporting is a traditional method that provides basic data for traffic safety management through human observation and recording. It is ineffective, though, and susceptible to subjective influences. The bird's-eye view method analyzes traffic flow and vehicle spacing from a global perspective, allowing for clearer detection of partially occluded vehicles, but it involves complex computational conditions. The distance neighbor method assesses collision risk based on the distance and relative speed between vehicles and quantifies potential dangers through traffic conflict indicators, demonstrating high scientific validity and practicality. However, it requires high accuracy in target detection. Therefore, to achieve the identification and prevention of near-miss events using automated methods, traffic target detection technology will play a vital role. The use of target detection technology involves recognizing and locating various targets in traffic scenes, such as vehicles, pedestrians, bicycles, traffic signs, and traffic signals (Zhang *et al*. 2023b). This is one of the most important technologies to make traffic on the roads more effective and safer. It monitors and predicts traffic by detecting and predicting real-time traffic flow data (Jia *et al*. 2023). Once a traffic accident happens, target detection immediately and accurately identifies traffic targets at the accident spot, thus saving time for emergency services response, easing traffic congestion, raising traffic safety, and making fewer accidents happen (Meng 2024).

Technical specifications and the quality of road surveillance have a direct effect on video quality in traffic videos. Some high-end cameras may have larger image sensors, higher resolution, and better image processing, leading to clearer video quality (Liu *et al*. 2024c). Meanwhile, there is a chance that some low-end or dated cameras might see a fall in video quality as a result of technical limitations or wear and tear (Zheng 2024). Additionally, videos for surveillance should be compressed for storage and transmission, and compression techniques and their rates have different effects on video quality. Video quality may decrease as a result of high compression rates (Jin *et al*. 2024). Thus, for traffic video detection, we have data of varying qualities. The You Only Look Once (YOLO) is a real-time object detection system that rapidly identifies and locates objects in videos. Different YOLO models perform differently with video quality. In low-quality videos, YOLO's detection accuracy decreases, and false detections and missed detections increase, as was found by Li *et al*. (2025). YOLO works better in high-quality videos with clear details, brightness, and contrast because it can extract object features more easily.

For this work, we conducted experiments on how well different YOLO models performed on

good-quality traffic video data and bad-quality traffic video data. Finally, we chose the best performing model for object detection. Next, we detected both high-quality and low-quality video data on YOLO models. For object distance detection, we picked the best performing model across different video qualities. The detection results were combined with three methods: the manual reporting, the bird's-eye view method, and the distance to neighbor method to detect near-miss events. This approach is intended to provide technological aid in generating intelligent transportation systems and achieving accurate and efficient automated traffic monitoring analysis. The main contributions of this paper are:

(1) Using high-quality traffic data for detection in YOLO models to obtain experimental results for each model, and analyzing each model through model evaluation metrics.

(2) Using low-quality traffic data for detection in YOLO models to obtain experimental results for each model, and analyzing each model through model evaluation metrics.

(3) The distance calculation method was compared and analyzed using the manual report, bird's-eye view, and distance neighbor (DN) methods to provide method selection and reference for Near Miss event research.

2. Literature Review

This paper's literature review section is mostly split into two sections. Section 2.1 focuses on research related to YOLO, while Section 2.2 looks at distance detection. By systematically reviewing these two key areas, we aim to provide a complete picture of the current research in these fields and lay a theoretical foundation for the follow-up research.

2.1. *YOLO*

Every iteration of the algorithm, from You Only Look Once version 1 (YOLOv1) to You Only Look Once version 8 (YOLOv8), has been enhanced and refined in accordance with its predecessor to satisfy the requirements of object detection in traffic situations. As shown in Table 1, a YOLO model comparison is given.

Table 1: YOLO model comparison

Version	Anchor	Framework	Backbone	AP/%	Dataset
YOLOv1	NO	Darknet	Darknet-24	63.4	PASCAL VOC2007
YOLOv2	YES	Darknet	Darknet-19	78.6	PASCAL VOC2007
YOLOv3	YES	Darknet	Darknet-53	36.2	COCO
YOLOv4-tiny	YES	PyTorch	CSPDarknet-53	43.5	COCO
YOLOv5s,YOLOv5n,YOLOv5m	YES	PyTorch	Modified CSP v7	55.8	COCO
YOLOv6s,YOLOv6n	NO	PyTorch	EfficientRep	52.5	COCO
YOLOv7x,YOLOv7	NO	PyTorch	RepConvN	56.8	COCO
YOLOv8s,YOLOv8n,YOLOv8m	NO	PyTorch	YOLOv8	53.9	COCO

As we can see from Table 1, the YOLO series has been evolving. By introducing or abandoning anchor boxes and improving the framework and backbone network, it has gradually increased detection accuracy. The dataset has also changed from the PASCAL VOC2007 dataset with 10,000 simple images to the COCO dataset with about 330,000 complex images. This shows that the YOLO series is constantly improving to meet more complex and diverse real-world application needs.

YOLOv1 is the pioneering work of the YOLO model, published by Joseph Redmon and others (Cai 2024). In order to produce a feature map of a predetermined size, it originally suggested the idea of single-stage object detection employing 24 convolutional layers and 2 fully linked layers. Although YOLOv1 has the advantage of speed, accuracy, and the ability to detect small targets, it can still be improved (Shao *et al.* 2022).

YOLOv2 builds on YOLOv1 to further improve model performance. It introduces batch normalization to accelerate training and enhance performance (Shi *et al*. 2024a), using anchor boxes to predict bounding boxes, improving detection accuracy, and introducing a high-resolution classifier, while also employing multi-scale input during training (Gao *et al*. 2019). Although YOLOv2's multi-scale input feature is innovative, it still has a long way to go to reach the ideal state and needs further optimization.

To better address the issue of detecting objects at multiple scales, YOLOv3 uses feature pyramid networks (FPN) to predict objects of different sizes (Ren & Liu 2020). Nevertheless, its performance to detect small objects still needs to improve. The spatial pyramid pooling (SPP) module in YOLOv4 uses multi-scale max pooling procedures to extract deeper contextual information beyond the bounding box that is produced by YOLOv3. YOLOv4 further enhances the feature extraction ability and deep information propagation of the network with the Mish activation function (Shuai *et al*. 2021). Recent studies comparing YOLOv4 with YOLOv3 indicate that they are more accurate and perform better on object detection tasks (Li *et al*. 2022). YOLOv4's complex model structure, however, comes with a large model size. On the other hand, YOLOv4-tiny, the lightweight version of YOLOv4 (listed in Table 1), is strictly smaller in size than YOLOv4 as it contains fewer parameters. In particular, it utilizes a simplified version of CSPDarknet as the backbone network (Cheng *et al*. 2023), which is reduced in depth and width.

To reduce model complexity even more, YOLOv5 has optimized its network structure. In YOLOv5, the C3 module is adopted as the backbone part that is able to learn residual characteristics very well. YOLOv5 also applies the spatial pyramid pooling fast (SPPF) structure that combines features through several 5×5 pooling layers to better capture more feature information and alleviate training parameters for multiple detection layers (Chai 2024). While YOLOv5 has shown some improvements in model size, detection speed, and accuracy, we still have space to optimize inference efficiency and scale fusion.

In order to satisfy both high precision and high efficiency, YOLOv6 introduces backbone, using EfficientRep for feature extraction to enhance inference efficacy greatly. For feature fusion, the neck part uses Rep-PAN, a feature pyramid structure, to decrease latency even further on the neck part. The head part uses a decoupled head structure to do position regression and classification on feature maps with different scales (Xu *et al*. 2024a). Nonetheless, YOLOv6 has some shortcomings in terms of detection accuracy and overall model performance. For extracting features of small objects in complex scenes, the EfficientRep Backbone is to be improved.

Further, YOLOv7 has introduced the E-ELAN efficient network architecture, in which the backbone network extracts features in processed images through modules such as CBS, ELAN, and MPConv, then using the SPPCSPC module, composed of SPP and CSP, to enlarge the target receptive field through maximum pooling at different sizes (Wen 2024), simultaneously reducing computational load and improving computation speed. Despite the presence of efficient architectures such as E-ELAN, YOLOv7 still fails to fully leverage the feature extraction ability when confronted with complex textures or shapes.

YOLOv8 consists of three structures—Conv, C2f, and SPPF—that extract image features and work together to form the backbone of YOLOv8. In the neck, it adopts a path aggregation network (PAN) structure design to fuse different scale feature maps so the network can better detect objects of different sizes. At the output end, a decoupled head (DH) structure is employed to extract the target's position and category information (Shi & Wang 2024). The YOLOv8 brings substantial improvement in the detection of small objects, but the performance of the model as a whole leaves room for advancement.

Depending on the network width and depth, model names in different versions of YOLO often contain suffixes like "n", "s", "m", and "x", denoting different scales. The smallest model is called the "n" (nano) because it can run on devices with limited resources (Tan & Wang 2024). The "s" (small) is a smaller model, balancing speed and precision, and is suited to more exacting scenarios that still need less time to process (Wang *et al*. 2024a). A medium

(i.e., "m") version is used, with higher detection accuracy but slower inference speed, which is appropriate for scenarios requiring higher accuracy but with sufficient device resources to capture its computational needs (Yang *et al*. 2022). A larger model, the "x" (extra large), has even higher detection accuracy but slower inference speed (Li *et al*. 2024a) and is typically used in situations where high accuracy is required even at the expense of device resources. These different scale models offer researchers choices in detection performance and results with the capability to select an appropriate model version matching the project or hardware conditions.

2.2. *Distance detection*

Over the years, while in the field of transportation, researchers improved the performance of YOLO through optimization of models in object detection. In YOLOv8, we provided the adaptive downsampling (ADown) module, the deformable attention transformer (DAT) mechanism, and the wise and efficient IOU loss function, which addresses the low detection accuracy, high model complexity, and slow response speed problems in the existing traffic sign detection methods (Zheng *et al*. 2025). Sun *et al*. (2025) proposed the SDF-YOLO model based on the improved YOLOv9 algorithm, using selective kernel networks (SKN), distribution shifting convolution (DSConv), and focaler inner IOU (FIIOU) loss function, effectively addressing the challenges of vehicle detection in flood scenarios and significantly improving detection accuracy.

Building on object detection, there have also been many advances in the research of distance calculation. Liang *et al*. (2023) proposed a monocular camera-based ranging method, using the YOLOv3 network model and COCO dataset to train weights for extracting vehicle detection boxes and solving the camera parameters through calibration points to achieve ranging, addressing the issue of precise and fast vehicle distance measurement. Zhang *et al*. (2024a) suggested a real-time monocular vision-based ranging method for surface targets of unmanned boats, using the YOLOv8 model to detect surface targets and obtain high-precision ranging reference points, and combining a dual-model ranging algorithm based on category prior division, solving the problem of monocular image ranging for unmanned boats in maritime navigation scenarios. Wang *et al*. (2024b) presented a vehicle distance warning technique that utilizes the enhanced YOLOv4-tiny algorithm, further extracting target features by adding a spatial pyramid pooling feature (SPPF) layer, effectively realizing vehicle distance warning, and providing a reference for the design of other warning systems.

In addition to improving the YOLO model itself for distance detection, some researchers have combined YOLO with other algorithms to achieve more accurate distance detection. Tan *et al*. (2024) designed an object detection and ranging system based on YOLOv5 and disparity calculation. With object detection using YOLOv5 and calculating the true distance of the target object utilizing stereo matching and the perspective difference function of a binocular camera, they create a system program that does both object detection and ranging. Xu & Li (2024) proposed a traffic object recognition and ranging method based on the YOLO model. The problem they solved is to recognize and range traffic objects in real time, and they solved this by dividing traffic images into grids and performing object recognition and distance prediction for each grid.

The bird's-eye view approach has also been used for distance detection in the field of autonomous driving. In the field of autonomous driving, Liu *et al*. (2024b) proposed an efficient view conversion method for 3D target detection tasks, where existing camera-to-bird's-eye view conversion methods are unsuitable for deployment complexity and real-time performance. Wang *et al*. (2024c) proposed a three-dimensional target detection algorithm based on bird's-eye view space fusion of infrared and lidar, solving the problems of data structure representation and spatial coordinate system differences in multi-sensor fusion, and verified the effectiveness of the algorithm through actual scene tests.

In summary, YOLO has been widely studied and applied in the field of traffic object detection. However, research comparing the performance of different YOLO models on traffic videos

of varying qualities is still relatively limited. This kind of research is of great significance for improving the robustness and generalization ability of models in real complex environments. With the iteration of the YOLO algorithm, each new version usually improves in detection speed, accuracy, robustness, etc. By comparing different versions of YOLO models, the performance improvement of the new version of the algorithm when processing traffic videos can be evaluated (Xu *et al*. 2024b), providing guidance for target detection in practical applications. At the same time, in the field of distance detection, there are few studies on distance detection using bird's-eye view and distance neighbor methods in combination with the YOLO model. Therefore, this study will further explore and improve methods for detecting the distance between vehicles. It will use the bird's-eye view method based on image processing and the distance-to-neighbor method based on data analysis to calculate the distance between vehicles in videos, providing more comprehensive and accurate distance information. To verify the accuracy and reliability of these two methods, this study also introduces manual reporting. Researchers manually measure the distance between vehicles in videos, and the recorded results are used as a reference to evaluate the performance of YOLO models in near-miss detection.

3. Methodology

To achieve efficient vehicle detection, we conducted the following experiments on both high-quality and low-quality videos: We compared the performance of different YOLO models and selected the best one using evaluation metrics to determine the performance. Next, we measure the distance between detected vehicles in the videos for near-miss detection using three different methodologies: manual report, bird's-eye view, and distance neighbor method. We compared the results from these three methods. The goal of this work is to apply these methods together to improve distance estimation accuracy and robustness.

3.1. *YOLO*

The YOLO series models have been applied in traffic data, which is one of the important research directions in the field of deep learning (Wang *et al*. 2022). The YOLO workflow diagram is shown in Figure 1 (Ma *et al*. 2024).

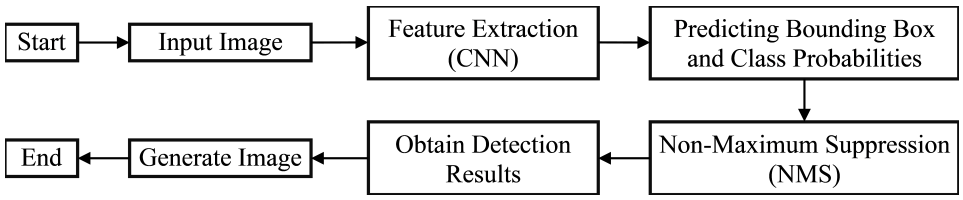
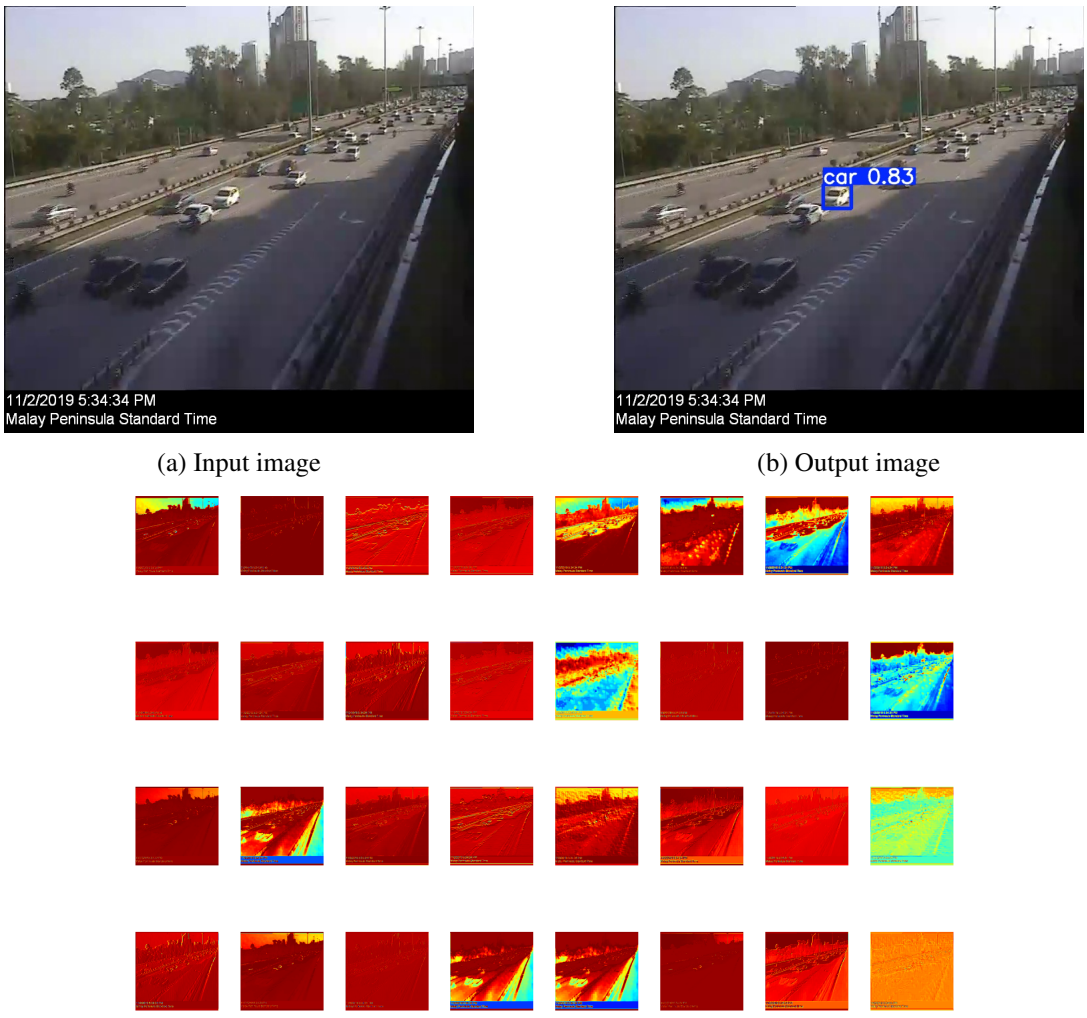


Figure 1: YOLO workflow diagram

The first stage in the object detection part of YOLO is to preprocess the input image.The image will be resized to the fixed dimensions desired by the YOLO model, and then pixel values will be normalized to [0, 1]. It helps in optimizing data processing efficiency. As shown in Figure 2(a), this is an input image used in this study.

A convolutional neural network (CNN) is then used to extract features from the preprocessed image and to create feature maps. Object detection then builds on this process, using this semantic information captured in the image. Figure 2(c) shows this feature map after it has undergone feature extraction of the input image.

From the extracted feature maps, the YOLO model predicts multiple bounding boxes and their corresponding class probabilities. Each bounding box contains the position information of the object (height, width, and center coordinates) and the confidence score, while the class probabilities are used to determine the category of the object inside the bounding box.



(a) Input image

(b) Output image

(c) Feature map

Figure 2: Illustration of the input image, output image, and feature map

The model use the non-maximum suppression (NMS) algorithm to choose the best outcomes from the several projected bounding boxes. This algorithm removes overlapping bounding boxes and retains the detection results with the highest confidence scores, ensuring that each object is accurately identified by only one bounding box.

Finally, the output image is generated based on the filtered detection results, clearly showing the position and category information of the objects. As shown in Figure 2(b), this is the final image generated after YOLO detection.

3.2. *YOLO algorithm principle*

Since the release of YOLOv1, the YOLO model has gone through multiple iterations. Among them, one of the YOLO series' representative versions is YOLOv5. Its network structure and design philosophy have had a far-reaching impact on subsequent versions. Moreover, YOLOv5 and YOLOv8 share a high degree of similarity in their overall architecture. Thus, in order to illustrate the YOLO algorithm idea, this paper uses YOLOv5 as an example.

3.2.1. *Network architecture*

Figure 3 is the network structure diagram of YOLOv5 (Chai 2024). It mainly includes three parts: the Backbone, Neck, and Head.

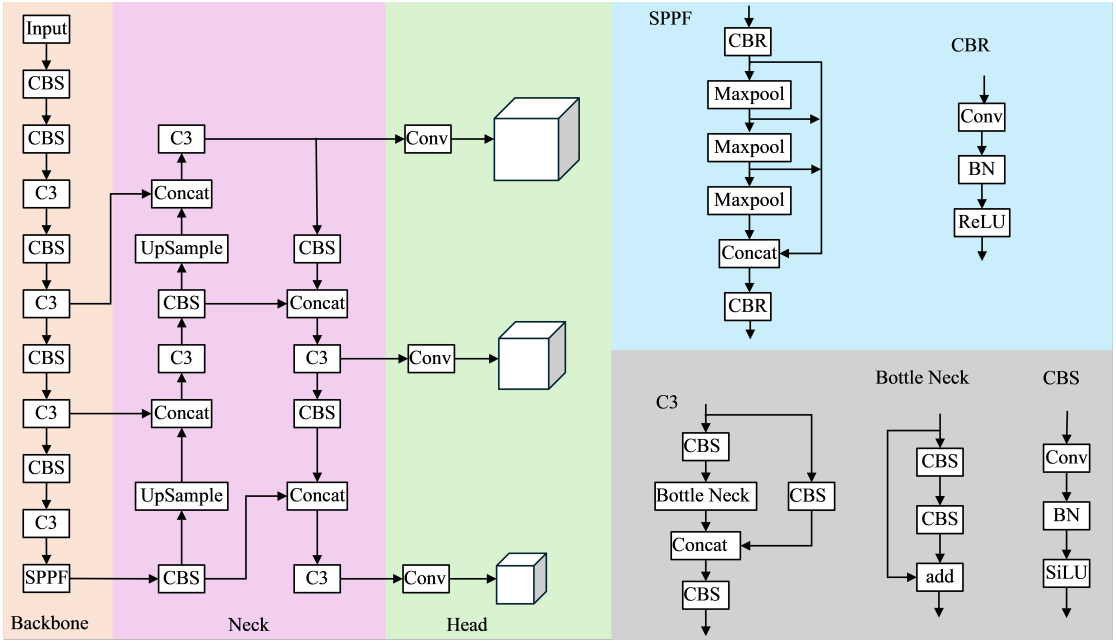


Figure 3: YOLOv5 network architecture diagram

The Backbone of YOLOv5 employs the CSPDarknet53 structure, which is an efficient feature extraction network that uses cross-stage partial connections to improve the network's computational efficiency and feature extraction capabilities (Sun *et al*. 2022). The Conv, C3, and SPPF modules are the three primary building blocks of the Backbone. The Conv module refers to the convolutional layer. It extracts features from images. Through down sampling operations, while increasing the number of channels (Sun *et al*. 2024). The standard convolution outputs a feature matrix as follows:

$$Y(P_0) = \sum_{P_n \in R} \theta(P_n) \cdot X(P_0 + P_n). \tag{1}$$

where $Y(P_0)$represents the value of the output feature map at position P_0, $\theta(P_n)$ represents the weights of the convolution kernel at position P_n, and $X(P_0 + P_n)$ represents the value of the input feature map at position $P_0 + P_n$. The summation is performed over all positions P_n of the convolution kernel.

After the Conv module, batch normalization and activation function operations are typically conducted. Batch normalization accelerates the convergence of the network and enhances the stability of the model by normalizing the distribution of feature values within the neural network (Xie *et al*. 2024). By normalizing each mini-batch of data, it lessens the problems of vanishing and exploding gradients and stabilizes the input distribution to each network layer (Zhang *et al*. 2024b). The following is the computation formula for batch normalization:

$$\mu_B = \frac{1}{m} \sum_{i=1}^{m} x_i \tag{2}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \tag{3}$$

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{4}$$

$$y_i = \gamma\hat{x}_i + \beta \tag{5}$$

where m represents the batch size, $B = [x_1, x_2, \ldots, x_m]$ represents the input vector to the Batch Normalization layer, μ_B and σ_B represent the mean and standard deviation of each batch of neurons, respectively, x_i represents the value of the i-th input node in the batch, and $\hat{x}_i$ represents a normal distribution with a mean of 0 and a variance of 1, ϵ is the minimum value to avoid division by zero, γ and β are scaling and shifting parameters, and y_i is the output value of the batch normalization layer.

By adding non-linearity, activation functions enable the network to learn and model increasingly intricate functional mappings (Zhang *et al*. 2023a). The activation function SiLU in YOLOv5 is calculated as follows:

$$\mathrm{SiLU}(x) = \frac{x}{1 + e^{-x}} \tag{6}$$

Su *et al*. (2024) suggest that the C3 module can enhance the network's feature extraction ability with residual connections and feature fusion. The SPPF module, an enhanced version of SPP, performs multi-scale feature fusion. It pools operations of different sizes to produce a static scale that is coherent across many scales, boosting the network's robustness (Han *et al*. 2024). The maximum pooling (max pooling) operation of the output of the t-th filter in the i-th dimension of the convolutional layer can be expressed as:

$$p_i(j) = \max_{(j-1)z \le t \le jz} \{q_i(t)\} \tag{7}$$

In this case, $q_i(t)$ is the input to the pooling layer, $p_i(j)$ is the j-th output of the pooling layer, and z is the width of the pooling kernel.

According to Chang *et al*. (2024), YOLOv5's neck uses hybrid feature pyramid networks (FPN) and path aggregation network (PAN) to fuse and transfer multi-scale features. As shown in Fang & Huang (2024), through a top-down pathway, FPN provides the high-level semantic features to the lower layers to supplement the semantic information of the feature maps from the lower layers. In a bidirectional fusion of layout features, PAN utilizes a bottom-up path in which layout features in low-level tiles are asynchronously passed to higher layers (Guan *et al*. 2024).

The YOLOv5 network's head is the output component. One of its responsibilities is to use the extracted feature maps to generate bounding boxes and class predictions for object detection. YOLOv5 uses a multi-scale detection mechanism and has 3 detection heads using feature maps of different sizes (Yang *et al*. 2024). With this design, YOLOv5 could detect objects of varying sizes simultaneously, thus giving the model greater flexibility and versatility.

3.2.2. *Loss function*

According to Sun & Gao (2022), the loss function computes localization loss, confidence loss, and classification loss with respect to differences between the predicted bounding boxes, the class probabilities, and true annotations. The total loss function of YOLOv5 can be expressed as:

$$L_{\text{total}} = \lambda_1 L_{\text{cls}} + \lambda_2 L_{\text{box}} + \lambda_3 L_{\text{obj}} \tag{8}$$

L_{cls}, L_{box}, and L_{obj} indicate the classification loss, localization loss, and confidence loss, respectively, and λ_1, λ_2 and λ_3 represent the weights of each part of the loss (Zhao *et al*. 2023). The classification loss, confidence loss, and localization loss are all binary cross-entropy (BCE) loss functions, but the localization loss uses the complete intersection over union (CIOU) loss function (Liu *et al*. 2024a). The calculation formula for CIOU is:

$$L_{\text{box}} = L_{\text{CIOU}} = 1 - IOU + \frac{\rho^2(b, \hat{b})}{c^2} + \alpha v \tag{9}$$

$$\alpha = \frac{v}{(1 - IOU) + v} \tag{10}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w}{h} - \arctan\frac{\hat{w}}{\hat{h}}\right)^2 \tag{11}$$

IOU is the value of intersection over union, b is the center of the true bounding box, $\hat{b}$ is the center of the predicted bounding box, $\rho(\cdot)$ is the Euclidean distance, c is the diagonal distance of the smallest enclosing box of predicted and true bounding boxes, w and h are the width and height of the true bounding box, and $\hat{w}$ and $\hat{h}$ are the width and height of the predicted bounding box. The calculation formula for the binary cross-entropy loss function is:

$$\text{BCE} = \frac{1}{N}\sum_{i=1}^{N}(y_i \times \log(\sigma(p_i)) + (1 - y_i) \times \log(1 - \sigma(p_i))) \tag{12}$$

where $\sigma(x)$ is the Sigmoid function, log is the natural logarithm, N is the number of samples, with each sample x_i having a binary label y_i (where y_i is the true label of the i-th sample, typically taking a value of 0 or 1), and p_i is the probability for each sample with $p_i \in [0, 1]$.

3.3. *YOLO evaluation metrics*

Precision, recall, mean average precision (mAP), frames per second (FPS), number of parameters, and floating-point operations (FLOPs) are all evaluated in this experiment to compare the models' performance.

The percentage of samples that are accurately predicted to be positive out of all samples that are anticipated to be positive is known as precision (Yan 2024), and the calculation formula is:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{13}$$

where TP stands for true positives and FP stands for false positives.

The percentage of samples that are accurately anticipated to be positive out of all actual positive samples is known as recall (Yang & Qian 2024), and the calculation formula is:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{14}$$

where FN stands for false negatives.

Average precision (AP) measures the area under the precision-recall curve, thus giving a single statistic that quantifies the model's performance in terms of precision and recall. Mean

average precision (mAP) is the average of AP over all classes and is:

$$\mathrm{mAP} = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{15}$$

Intersection over union (IOU) is how we compute the overlap between true and predicted bounding boxes:

$$\mathrm{IOU} = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{16}$$

mAP@0.5–0.95 is the average of mAP computed at IOU thresholds from 0.5 to 0.5-0.95 with a step size of 0.05, while mAP@0.5 is the average precision at an IOU threshold of 0.5. By examining detection accuracy for different levels of overlap, mAP measures the performance of the model thoroughly.

The term frames per second (FPS) is used to measure the model's performance on that specific piece of hardware in terms of the processing it is able to run per second, and the calculation formula is:

$$\mathrm{FPS} = \frac{No.Images}{No.Time} \tag{17}$$

Implementation of real-time detection is under stress on FPS value.

Here 'the number of parameters' is the total amount of all learnable parameters in the model (Zhang *et al*. 2025). It usually refers to how much computational resources a model requires and is connected to the complexity of the problem.

We can measure how complex a model is to compute, also known as floating point operations per second (FLOPs), which is how many floating point operations are needed to process one input image (Cheng *et al*. 2025).

3.4. *Manual report*

Manual reporting refers to manually annotating information to extract the position of targets from video data and directly calculating the distance between targets based on geometric principles or formulas. In order to locate video frames with a target, we need to identify where targets are in the image. Based on the coordinates of targets within each group, the Euclidean distance between two points is then computed using a formula. The Euclidean distance is computed and is given by the formula:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{18}$$

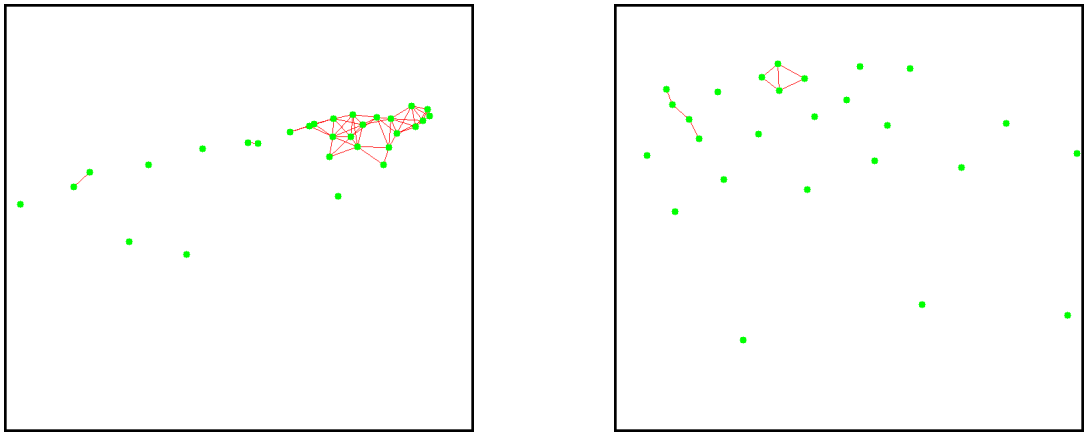
The distance d is the distance between the points (x_1, y_1) and (x_2, y_2).

Due to the lack of relevant camera parameters, this paper studies the pixel distance and sets the threshold to 50, and counts the number of target pairs with a distance between the targets less than 50 in all frames.

Since manual labeling takes a lot of time, the manual report method is suitable for small-scale data. At the same time, since manual labeling can ensure high accuracy of the target position, it can be used as a benchmark for comparison when it is necessary to verify the automated method. However, for long videos or high frame rate videos, frame-by-frame labeling and calculation costs are extremely high, so it is very difficult to process long videos, which limits the scope of application of the method.

3.5. *Bird's-eye view*

The bird's-eye view is calculated by mapping the front view of the vehicle into the bird's-eye view to help the driving system fully understand the road scene (Shi *et al*. 2024b). This method can retain the position and size information of the vehicle, thereby more accurately assessing the distance between vehicles and potential dangerous events (Yu *et al*. 2024). As shown in Figure 4, the bird's-eye view in this experiment, where the green dots are detected vehicles and the red lines indicate that the distance between the two targets is less than 50. Figure 4 (a) is a bird's-eye view of a high-quality video, and Figure 4 (b) is a bird's-eye view of a low-quality video.



(a) High-quality video (b) Low-quality video

Figure 4: Bird's-eye view of high-quality and low-quality videos

This method visualizes and calculates the distance between objects in a video by creating a bird's-eye view. Figure 5 shows a flowchart for the calculation of the bird's-eye view. First, the YOLO model is loaded, and video frames are read. Vehicles in the video frames are detected to obtain the center points of the targets and calculate the Euclidean distance between vehicles. If the distance between vehicles is less than 50 pixels, a red line segment is drawn; if the distance is greater than 50 pixels, no red line segment is drawn. Based on the detection results, a bird's-eye view is created to display the center points of the vehicles and the drawn distances between them. In order to finish visualizing the experimental findings, the detection results are finally saved.

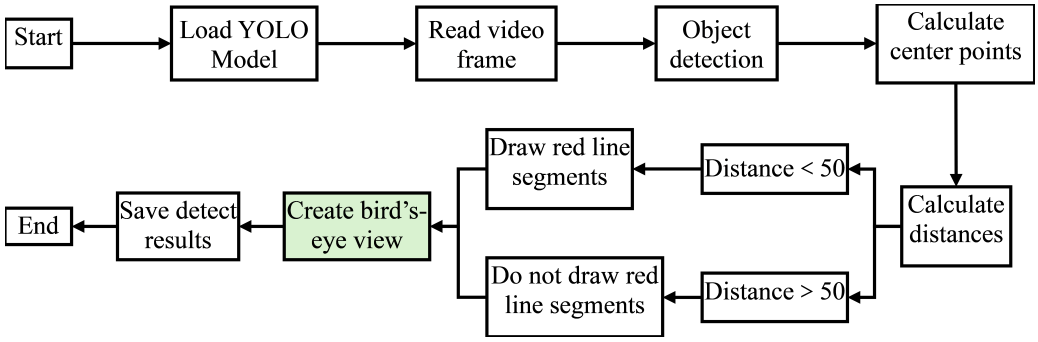


Figure 5: Bird's-eye view calculation flow chart

3.6. *Distance neighbor*

Distance neighbor (DN) is a method for analyzing and processing spatial data, particularly in the area of tracking and target detection. The core idea is to determine the proximity relationship between targets by calculating the distance between them (Li *et al*. 2024b). This method can be used in a variety of application scenarios, including neighboring target detection, abnormal behavior detection, etc.

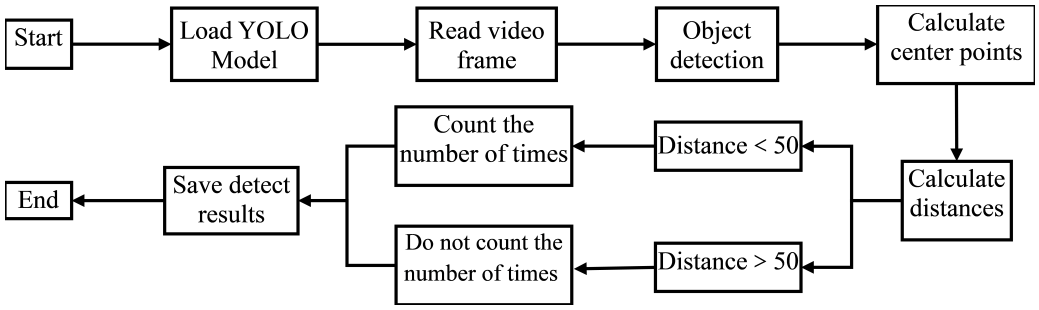


Figure 6: Distance neighbor calculation flow chart

This study uses the Distance Neighbor method to detect the distance between targets in the video and counts the cases where the distance between vehicles is less than 50 pixels. Figure 6 shows the calculation flowchart of the Distance Neighbor method. First, the YOLO model is loaded, and the experiment video is read frame by frame for vehicle detection. The center points of the vehicles are calculated based on the detection results. The distance between vehicles is calculated according to the center points. If the distance is less than 50 pixels, it is included in the count. If the distance is more than 50 pixels, it is not included. Finally, the detection results are saved to complete the experiment.

4. Experimental Results and Analysis

In order to assess how well the YOLO model performed under various video quality situations, this experiment used multiple iterations of the model to train and test high-quality and low-quality films, respectively. The experimental results cover the number of parameters, floating point numbers, detection accuracy, recall rate, and inference speed, providing data support for near-miss event detection.

4.1. *Dataset and model selection*

This experiment uses two different experimental data sets: high-quality video and low-quality video, based on different frame rates and bit rates. The high-quality video is the traffic data of Malay Peninsula Standard Time on 11/2/2019, as shown in Figure 7 (a). The bit rate of this movie is 1702 kbps, and its frame rate is 24.84 frames per second. The low-quality video is the traffic data of Malay Peninsula Standard Time on 7/2/2019, as shown in Figure 7 (b). The bit rate of this movie is 1124 kbps, and its frame rate is 16.61 frames per second.

In terms of model selection, this paper did not conduct model training for YOLOv1 and YOLOv2 because their average precision calculation methods are different from those of the later versions (Ali & Zhang 2024). Instead, we only performed comparative experiments for YOLOv3 to YOLOv8. Due to limitations in computational resources, for the YOLOv4 model, YOLOv4-tiny was chosen for the experiment, and for the YOLOv5 to YOLOv8 models, smaller width and depth factor models were selected for the experiment.

(a) High-quality video (b) Low-quality video

Figure 7: Dataset illustration

4.2. *Experimental environment*

The experimental environment is configured with the Windows 11 (64-bit) operating system, Python 3.8, and CUDA version 12.6. The batch size was set to 1 in order to help the model better understand the nuances in the pictures. To better observe the model's performance changes during training and promptly detect and prevent overfitting, the epoch was set to 350. In the detection of low-quality videos, all three models of YOLOv8 exhibited early stopping phenomena. Specifically, YOLOv8s, YOLOv8n, and YOLOv8m stopped early at 279, 304, and 317 iterations, respectively. The condition for triggering early stopping was based on the weighted calculation of model fitness using mAP@0.5 and mAP@0.5:0.95 with weights of 0.1 and 0.9, respectively. Early stopping was triggered when the fitness did not improve after 100 epochs.

4.3. *Results analysis*

Floating-point operations (FLOPs) and parameter count are key metrics for assessing computing cost and model complexity in object detection tasks. Moreover, these variables determine whether a model is feasible and efficient for use in practice. Therefore, in this section, we first introduce the number of parameters and FLOPs for the different YOLO models. Based on the experimental findings, it then analyzes how well these models perform under varying video quality settings. Table 2 shows how the number of parameters and the computing capacity in FLOPs change among various YOLO models used for the object detection domain.

As shown in Table 2, YOLO models vary significantly in terms of parameters and FLOPs. YOLOv5n and YOLOv6n have fewer parameters and FLOPs, allowing them to run on devices with low resources. Big models with more parameters and FLOPs (such as YOLOv3 and YOLOv7x) are more accurate at detecting, but you require more powerful hardware. When considering a model, it is necessary for the model to possess detection precision, processing speed, and hardware resources to select the best model for the task.

Next, we will analyze the experimental results for both high-quality and low-quality videos in detail. We will also discuss the performance of different models under various conditions in combination with the data on the number of parameters and FLOPs.

4.3.1. *Model comparison with high-quality data*

As shown in Table 3, which presents the experimental results for high-quality data, YOLOv4-tiny achieved a precision value of 1, but its recall value is very low (0.000215), indicating that

Table 2: FLOPs and parameters for several YOLO models

Model	Params/M	FLOPs/G
YOLOv3	61.524	85.393
YOLOv4-tiny	6.057	9.056
YOLOv5s	7.022	8.770
YOLOv5n	1.765	2.323
YOLOv5m	20.871	26.521
YOLOv6s	20.112	27.422
YOLOv6n	5.040	6.949
YOLOv7x	70.813	103.884
YOLOv7	37.195	57.812
YOLOv8s	11.136	15.756
YOLOv8n	3.011	4.507
YOLOv8m	25.857	43.486

it correctly predicted very few positive instances. Therefore, YOLOv4-tiny is not suitable for traffic detection with high-quality data in this experiment, and its data will not be discussed further.

Table 3: Experimental results of YOLO on high-quality data

Model	Precision	Recall	mAP@0.5	mAP@0.5-0.95	FPS
YOLOv3	0.989	0.974	0.994	0.808	2.77
YOLOv4-tiny	1.0	0.000215	0.998	0.815	17.06
YOLOv5s	0.969	0.946	0.987	0.818	31.11
YOLOv5n	0.943	0.897	0.967	0.743	31.32
YOLOv5m	**0.990**	**0.977**	**0.994**	**0.882**	**28.42**
YOLOv6s	0.875	0.884	0.942	0.702	19.64
YOLOv6n	0.878	0.876	0.938	0.686	19.1
YOLOv7x	0.906	0.917	0.970	0.753	21.52
YOLOv7	0.911	0.913	0.969	0.740	27.06
YOLOv8s	0.969	0.955	0.990	0.895	33.1
YOLOv8n	0.948	0.918	0.979	0.833	33.52
YOLOv8m	0.971	0.965	0.992	0.920	23.77

As shown in Table 3, in the object detection of high-quality videos, the two models with the highest precision are YOLOv5m and YOLOv3, with values of 0.990 and 0.989, respectively. Next are YOLOv8m, YOLOv8s, and YOLOv5s, with values around 0.970, which are about 0.02 lower than the best-performing YOLOv5m. This indicates that these models can effectively distinguish between positive and negative samples in high-quality data, have strong feature-capturing abilities for positive samples, and are less likely to be disturbed by negative samples. The worst-performing models are YOLOv6n and YOLOv6s, with precision values of 0.878 and 0.875, respectively, both below 0.9. This suggests that the network structure of YOLOv6 was designed to focus more on other performance metrics, with insufficient optimization for classification accuracy.

The recall and precision of the experimental models show similar trends. The two models with the highest recall are still YOLOv5m and YOLOv3, with values of 0.977 and 0.974, respectively. This is because they have a strong ability to cover positive samples and can detect most of the actual positive samples with fewer misses. YOLOv5n, YOLOv6s, and YOLOv6n perform the worst, with values of 0.897, 0.884, and 0.876, respectively, all below 0.9.

In terms of mAP@0.5, YOLOv3 and YOLOv5m have the highest values, both at 0.994.

This indicates that they perform the best on average when the threshold is 0.5, finding a healthy balance between precision and recall. They are quite good at identifying the target objects' position and type. The worst performers are still YOLOv6s and YOLOv6n, with values of 0.942 and 0.938, respectively. This is consistent with their performance in precision and recall, showing that YOLOv6 has significant room for improvement in overall detection performance.

In terms of mAP@0.5-0.95, the three top-performing models are YOLOv8m, YOLOv8s, and YOLOv5m, with values of 0.920, 0.895, and 0.882, respectively. This means these three models maintain high detection accuracy across different levels of IOU thresholds. The lowest values for YOLOv6s and YOLOv6n are around 0.7, at 0.702 and 0.686, respectively.

In terms of FPS, YOLOv8n, YOLOv8s, YOLOv5n, YOLOv5s, YOLOv5m, and YOLOv7 have values of 33.52, 33.10, 31.32, 31.11, 28.42, and 27.06, respectively. Given that the frame rate of the high-quality video is 24.84, these five models can all achieve real-time detection on high-quality data. The value for YOLOv3 is the smallest, at only 2.77, which is only one-twelfth of the FPS value of the fastest detecting model, YOLOv8n.

In summary, YOLOv8n, YOLOv8s, YOLOv5n, YOLOv5s, and YOLOv5m perform the best in terms of FPS. YOLOv3 and YOLOv5m successfully strike a balance between recall and precision, and YOLOv5m and YOLOv8m show excellent performance in mAP. Therefore, in the detection of high-quality video data, YOLOv5m can maintain high precision, recall, and mAP values while also having a faster detection speed. Additionally, YOLOv5m has a moderate number of parameters and FLOPs among all models. It is neither too computationally intensive to run on ordinary hardware nor too simple to compromise detection accuracy. It can well adapt to the hardware limitations and task requirements of most practical application scenarios. Thus, in practical applications such as traffic monitoring that require high precision and speed, YOLOv5m is undoubtedly a more ideal choice.

4.3.2. *Model comparison for low-quality data*

As shown in Table 4, which presents the experimental results for low-quality data, similar to the high-quality data experimental results, YOLOv4-tiny is not suitable for traffic detection with low-quality data in this experiment, and its data will not be discussed further.

Table 4: YOLO low-quality data experimental results

Model	Precision	Recall	mAP@0.5	mAP@0.5-0.95	FPS
YOLOv3	0.962	0.928	0.954	0.846	2.66
YOLOv4-tiny	1.0	0.000306	0.971	0.629	16.62
YOLOv5s	0.956	0.925	0.969	0.865	29.71
YOLOv5n	0.954	0.922	0.968	0.838	30.15
YOLOv5m	0.960	0.919	0.969	0.892	27.65
YOLOv6s	0.949	0.938	0.977	0.803	16.9
YOLOv6n	0.958	0.926	0.978	0.806	18.11
YOLOv7x	0.953	0.939	0.972	0.846	20.42
YOLOv7	0.959	0.928	0.979	0.846	24.15
YOLOv8s	0.955	0.930	0.978	0.906	19.11
YOLOv8n	0.953	0.930	0.978	0.892	31.77
YOLOv8m	**0.950**	**0.938**	**0.978**	**0.908**	**30.39**

The four models with the highest precision are YOLOv3, YOLOv5m, YOLOv7, and YOLOv6n, with values of 0.962, 0.960, 0.959, and 0.958, respectively, all around 0.960. This indicates that these models have strong robustness against noise and interference during training, and they generalize well to data of different qualities, making them adaptable to various complex real-world scenarios. The performance of YOLOv8m and YOLOv6s is poorer, with

values of 0.950 and 0.949, respectively, but there is just a 0.013 difference between the maximum and minimum numbers.

In terms of recall, the best-performing model is YOLOv7x, with a value of 0.939. Next are YOLOv6s and YOLOv8m, both with a value of 0.938, only differing from YOLOv7x by 0.001. This means that these three models can effectively recognize the features of the targets, and they can capture the key information of the targets even when the images are blurry or lack detail, thereby improving the recall rate of detection. The worst scoring is YOLOv5m, 0.919, indicating that YOLOv5m is more sensitive to low image quality.

YOLOv7 had the highest mAP@0.5 of 0.979, while YOLOv6n, YOLOv8s, YOLOv8n, and YOLOv8m had an mAP@0.5 of 0.978 (0.001 lower than YOLOv7). YOLOv6s value is 0.977, which is 0.002 lower than YOLOv7's value. This demonstrates that these models can continue to perform well in high detection performance despite low-quality data, and they are still quite robust towards different data conditions. Compared to the other models, YOLOv3 has the smallest value, 0.954, which is much lower than the others, showing that the model has a huge room for improvement in terms of detection of low-quality videos and should improve its versatility and detection capability for such videos.

The three YOLOv8 models perform very well in terms of mAP@0.5-0.95. The values for YOLOv8m, YOLOv8s, and YOLOv8n are 0.908, 0.906, and 0.892, respectively. Moreover, the value of YOLOv5m is the same as that of YOLOv8n. That means that the three YOLOv8 models and YOLOv5m can somehow offset that image quality and frame rate impact and achieve good detection results. The two models of YOLOv6 have similar performance and the smallest values, with YOLOv6n at 0.806 and YOLOv6s at 0.803, showing that YOLOv6 struggles to achieve precise object detection in low-quality video data.

In terms of FPS, YOLOv8n and YOLOv8m perform the best, with FPS values of 31.77 and 30.39, respectively. The YOLOv5 series models also show good performance, with YOLOv5n, YOLOv5s, and YOLOv5m having FPS values of 30.15, 29.71, and 27.65, respectively. This indicates that these five models can still quickly detect and analyze each frame in videos with low bit rates and low frame rates and promptly output detection results. The FPS value of YOLOv3 is the smallest, only 2.66, which is one-twelfth of the best-performing YOLOv8m.

Overall, although the precision value of YOLOv8m is 0.950, it only differs from YOLOv3, which has the highest precision value, by 0.012. Moreover, YOLOv8m performs well in other aspects, enabling it to more effectively detect and recognize targets when processing low-quality data. Additionally, YOLOv8m has a moderate number of parameters and FLOPs among all models. It maintains good performance while being relatively hardware-friendly, making it well-suited for the hardware conditions and task requirements of most practical application scenarios. Considering detection accuracy, speed, and model applicability, YOLOv8m is a more ideal choice.

4.3.3. *High-quality video near-miss detection*

Based on the experimental results of Section 4.3.1, this paper uses the YOLOv5m model to detect targets in high-quality videos and then uses the target detection data to estimate the distance of the target vehicle in the video using three different methods: manual report, bird's-eye view method, and distance neighbor method. Table 5 shows the near-miss detection results of high-quality videos.

According to the results calculated by the three methods, this study also calculated the absolute error and relative error of the three methods, respectively. The experiment found that in videos of different lengths, the absolute error values were all less than 1%, and the relative error values were all less than 5%. This shows that the three methods can provide high-precision results when calculating the distance between targets, and the three methods have consistent performance in this scenario. The data selected were 15, 30, 45, and 60 seconds, covering different time scales, and the error remained small, indicating that the results are highly robust and the methods have good adaptability to the expansion of the time dimension. All three methods

Table 5: High-quality video near-miss detection results

Times	Frames	Manual Report	Bird-Eye	Distance Neighbor	Absolute Error	Relative Error
15s	373	9.4650%	9.5765%	9.5985%	0.1335%	1.3984%
30s	746	9.8115%	10.0194%	10.048%	0.2365%	2.3746%
45s	1118	9.5773%	9.8838%	9.9111%	0.3338%	3.4093%
60s	1490	9.7105%	9.9717%	10.0024%	0.2919%	2.95%

meet the tolerance range common in the industry (absolute error $< 1\%$, relative error $< 5\%$), indicating that automated methods (bird's-eye view and distance neighbors) can replace manual calculations in specific scenarios.

4.3.4. *Low-quality videos near-miss detection*

Based on the experimental results of Section 4.3.2, this study uses the YOLOv8m model to detect targets in low-quality videos. Three different methods, namely the manual report, the bird's-eye view method, and the distance neighbor method, are used to estimate the distance of target vehicles in the video. Table 6 shows the near-miss detection results of low-quality videos.

Table 6: Low-quality video near miss detection results

Times	Frames	Manual Report	Bird-Eye	Distance Neighbor
15s	250	3.155%	3.155%	3.155%
30s	500	3.3529%	3.3529%	3.3529%
45s	748	3.3481%	3.3481%	3.3481%
60s	997	3.4999%	3.4999%	3.4999%

From the experimental results, we can see that in low-quality videos of different lengths, the three methods get exactly the same results. This could be as a result of low bit rate and frame rate overall in low-quality videos, which directly impair the video's clarity and smoothness, both of which have a major effect on the target detection algorithm's performance. The target detection algorithm may find it difficult to distinguish the target in low-quality videos, resulting in identical findings from various computation techniques.

4.3.5. *Comparison with other experiments*

In addition to comparing the experimental results with this paper, this paper also conducts comparative analysis on the research of other authors. Table 7 and Table 8 show the related research on object detection and distance detection.

Table 7: Related research on object detection

Authors	Model	Research Methods	Conclusions	Limitations
Sundaresan *et al.* (2024)	YOLOv8, YOLOv10	Data augmentation	YOLOv10 outperforms in detecting motorcycles and trucks, while YOLOv8 outperforms in detecting cars	Environmental conditions (such as light and weather changes) were not evaluated
Alif (2024)	YOLOv8, YOLOv10, YOLOv11	Data augmentation	YOLOv11 outperforms YOLOv8 and YOLOv10 in vehicle detection tasks	The performance of YOLO11 may be affected in extreme environments

Table 8: Related research on distance detection

Authors	Model	Research Methods	Conclusions	Limitations
Zheng & Cui (2024)	YOLOv8	Cross-matching feature points, binocular vision principle	When the distance is short, the ranging accuracy is high, and the accuracy decreases as the distance increases	The distance measurement of distant objects depends on the camera resolution
Shen *et al*. (2024)	YOLOv5	Karhunen-Loeve Transform (KLT), Constructing the Kalman filter equation	The proposed algorithm improves the ranging accuracy by 20% within 2-8 m	The ranging accuracy and tracking effect depend on the calibration of the camera and single-line lidar

As shown in Tables 7 and 8, Sundaresan *et al*. (2024) compared YOLOv8 and YOLOv10, while Alif (2024) compared YOLOv8, YOLOv10, and YOLOv11. Both studies used data augmentation techniques to improve the models' generalization ability. However, none of them conducted a systematic comparison of video quality (such as resolution and clarity). This indicates that in comparative studies of object detection using different YOLO models, video quality is an overlooked but important factor. In the research on distance detection, Zheng & Cui (2024) used cross-matching feature points combined with the principles of binocular vision to calculate target distances. Shen *et al*. (2024), on the other hand, used the KLT algorithm to match feature points between consecutive frames and built a Kalman filter equation to correct the tracking results. All these studies used YOLO for object detection, but the subsequent distance calculation methods varied. These different methods have differences in ranging accuracy and scope of application. This shows that the processing methods after object detection significantly affect the final distance detection results. Therefore, it is necessary to choose the appropriate distance detection algorithm according to the specific application scenario and requirements.

4.4. *Conclusion*

The experimental results show that in high-quality data, the medium-scale YOLOv5m, as a target detection model, has an FPS value of 28.42, which is greater than the frame rate of 24.84 in high-quality videos, and can provide real-time detection of traffic data. The precision value reaches 0.99, indicating that the detection results of YOLOv5m have little effect on the final distance calculation, and its results are highly reliable. YOLOv5m can efficiently and accurately extract the bounding box information of the target, providing an accurate data basis for subsequent distance calculation so that the experimental results can be directly applied to subsequent distance analysis, indicating that the model has strong generalization ability and practical value. When processing the same set of data, the calculation results of manual report, the bird's-eye view method, and the distance neighbor method are highly consistent, with an absolute error of less than 1% and a relative error of less than 5%, indicating that these methods are reliable and consistent in the current scenario. Manual report is a high-precision benchmark method, and the errors of the three methods are all within a very small range, indicating that the automated method (bird's-eye view method, distance neighbor method) can be more efficient while ensuring accuracy. In short, by using YOLOv5m for target detection and then using the bird's-eye view method or the distance neighbor method for near-miss detection, we can achieve fully automated analysis of detection and distance calculation. This improves efficiency and reduces labor costs. It has significant practical value for the large-scale application and long-term stable operation of intelligent transportation systems. It can effectively enhance the efficiency and intelligence level of traffic control, promoting the development and popularization of transportation systems with intelligence.

This demonstrates the reliability of the experimental methods in current video data analysis. It also shows that image blurring and noise in low-quality videos have some impact on the detection of near-miss events. Therefore, in future research, it is necessary to further explore the interference factors of video quality on detection results and their potential impact on near-miss event analysis to increase the detection's accuracy and resilience.

In low-quality videos, YOLOv8m, a medium-sized model, has the best overall performance. It has a high mAP value and the FPS value reaches 30.39, which is much higher than the frame rate of low-quality videos and fully meets the real-time detection of low-quality videos. In near miss detection, the calculation results of the three methods are completely consistent This demonstrates the reliability of the experimental methods in current video data analysis. It also shows that image blurring and noise in low-quality videos have some impact on the detection of near-miss events. Therefore, in subsequent research, further attention should be paid to the interference of video quality on detection results and its potential impact on near-miss event analysis to increase the detection's accuracy and resilience.

In future research, we can explore the following three aspects. First, model fusion is an important research direction. Combining YOLO with Spatio-Temporal Graph Convolutional Networks (STGCN) and Long Short-Term Memory (LSTM) networks can fully utilize YOLO's object detection capabilities, STGCN's spatio-temporal modeling capabilities, and LSTM's time series analysis capabilities. This can achieve more comprehensive and accurate object detection and near-miss detection. Second, validation in real-world applications is crucial. We need to deploy monitoring equipment in real traffic environments and run the selected YOLO models to verify the effectiveness and reliability of near-miss detection methods. In this way, we can not only improve traffic safety and efficiency but also promote the advancement of intelligent transportation technology, laying the foundation for a smarter and safer future traffic environment. Finally, considering that the current focus is mainly on vehicle detection and distance calculation in expressway scenarios with relatively simple video scenes. In future research, more complex scenes can be further introduced to explore the applicability and robustness of target detection and near-miss event analysis in diverse traffic environments (such as urban roads, congested areas, or mixed traffic flows).

References

Ali M.L. & Zhang Z. 2024. The YOLO framework: A comprehensive review of evolution, applications, and benchmarks in object detection. *Computers* **13**(12): 336.

Alif M.A.R. 2024. YOLOv11 for vehicle detection: Advancements, performance, and applications in intelligent transportation systems. *arXiv preprint arXiv:2410.22898* .

Cai W.Y. 2024. Improvement of traffic sign detection method based on YOLOv8s. *Computer Science and Application* **14**(10): 33–43.

Chai Y.X. 2024. Research on YOLOv5 detection algorithm for jujube in complex natural environment. Master's thesis. Xi'an University of Technology.

Chang F., Wang B., Zhang X. & Wang Z. 2024. Study on small target detection method based on improved YOLOv5. *Smart Rail Transit* **61**(2): 7–13.

Chen H. 2023. Research on urban road traffic layout and planning design. *Architectural Design and Research* **4**(1): 133–135.

Chen Z.J., Li H.P. & Xu H.R. 2024. Lightweight detection algorithm for small target insects based on LID-YOLO. *Semiconductor Optoelectronics* **45**(6): 945–952.

Cheng H.R., Wang X.T., Li J.R., Guo Z.Y. & Liu W. 2023. Improved YOLOv4-tiny for epidemic collaborative mask wearing detection. *Computer Engineering and Applications* **59**(20): 208–218.

Cheng S., Li J., Wang Z., Liu S. & Wang M. 2025. Lightweight underwater optical image recognition algorithm based on YOLOv8. *Laser & Optoelectronics Progress* **62**(4): 0437008.

Fang X. & Huang L. 2024. SAR ship detection algorithm based on global position information and fusion of residual feature. *Systems Engineering & Electronics* **46**(3): 839–848.

Gao L., Chen N. & Fan Y. 2019. Vehicle object detection based on fusing multi-scale context convolutional features. *Opto-Electronic Engineering* **46**(4): 180331.

Guan H., Ling Y. & Wang S. 2024. A review of deep learning algorithms for single-stage helmet detection. *Computer Engineering and Applications* **60**(16): 61–75.

Han Y., Li H., Zhu J., Xiao S. & Si P. 2024. Danger sign recognition algorithm based on YOLOv8. *Electronic Communication and Computer Science* **6**(6): 193–195.

Jia Z., Li M.J. & Li W.T. 2023. Real-time vehicle detection at intersections based on improved YOLOv5+DeepSort algorithm model. *Computer Engineering & Applications* **45**(4): 674–682.

Jin W., Ning M., Wang X. & Liang F. 2024. Experiment on the impact of quantization parameters on 4k video quality of the H.266 compression standard. *Experiment Science and Technology* **22**(3): 9–14.

Li P., Su Y.L., Ning H., Meng Q.W. & Wei Q. 2024a. Detection of self-explosion defects in power insulators based on embedded YOLO network. *Transactions of China Electrotechnical Society* **2110**.

Li W., Xu G., Kong W., Guo F. & Song Q. 2022. Research on plant leaf-stem intersection target detection based on improved YOLOv4. *Computer Engineering and Applications* **58**(4): 221–228.

Li Y., Li X., Chen L., Yang Q., Xu C. & Xu S. 2025. FDLIE-YOLO: Frequency domain enhanced end-to-end low-light image target detection method. *Infrared and Laser Engineering* **54**(1): 20240376.

Li Y., Zhang D. & Zhang X. 2024b. An improved local mean pseudo-nearest neighbor algorithm. *Computer Engineering and Applications* **60**(5): 88–94.

Liang Z., Huang Z., Hu X. & Chen J. 2023. Research on distance measurement using vehicle four-point calibration based on YOLO network. *Journal of Mechanical Engineering* (10): 226–235.

Lim L.M., Sathasivam S., Ismail M.T., Mohamed A.S.A., Ibidoja O.J. & Ali M.K.M. 2024. Comparison using intelligent systems for data prediction and near-miss detection techniques. *Pertanika Journal of Science & Technology* **32**(1): 365–394.

Liu C.H. 2024. Criminal liability for traffic accidents in the scenario of AI-assisted driving. *Journal of Beijing University of Technology (Social Sciences Edition)* **24**(6): 166–176.

Liu H., Zhang Y., Zhou K., Zhang Y. & Lü S. 2024a. Improved YOLOv5-S for traffic sign detection algorithm. *Computer Engineering and Applications* **60**(5): 200–209.

Liu J.H., Guan J.C., Fang H.Q. & Chao J.S. 2024b. Efficient view transformation for autonomous driving. *Computer Systems & Applications* **34**(2): 246–253.

Liu W., Zhang X. & Zhou P. 2024c. The application of network video surveillance technology in forest fire prevention. *Electronic Communication and Computer Science* **6**(2): 172–174.

Ma Z.Y., Li H. & Yang G.Y. 2024. Research on key technologies of intelligent tea picking machine based on YOLOv3 algorithm. *Journal of Chinese Agricultural Mechanization* **45**(4): 199–204.

Meng X. 2024. Research on the improvement of YOLOv7 algorithm for industrial safety multi-target detection. *Pure Mathematics* **14**(7): 1–14.

Ren X. & Liu X. 2020. Mask wearing detection based on improved YOLOv3. *Journal of Physics: Conference Series* **1678**: 012089.

Retallack A.E. & Ostendorf B. 2020. Relationship between traffic volume and accident frequency at intersections. *International Journal of Environmental Research and Public Health* **17**(4): 1393.

Shao Y., Zhang D., Chu H., Zhang X. & Rao Y. 2022. A review on YOLO object detection based on deep learning. *Journal of Electronics & Information Technology* **44**(10): 3697–3708.

Shen J., Zhang Y., Liu F. & Liu C. 2024. Ranging and tracking algorithm for surface targets based on fusion of single-line laser radar and vision. *Maritime Safety* **2024**(11): 58–60.

Shi J. & Wang Y. 2024. Research on vehicle small target recognition technology based on YOLOv8. *Journal of Taiyuan University (Natural Science Edition)* **2024**(03): 72–77.

Shi M., Dong Q., Guo N. & Xu M. 2024a. YOLOv5-W bridge crack real-time detection algorithm. *Modeling and Simulation* **13**(1): 290–303.

Shi P., Dong X., Yang A. & Qi H. 2024b. Research progress of BEV perception algorithms for autonomous driving: A review. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4790559 (12 November 2024).

Shuai K., Jianwu Z., Zunjie Z. & Guofeng T. 2021. An improved YOLOv4 algorithm for pedestrian detection in complex visual scenes. *Telecommunications Science* **37**(8): 46–56.

Su T., Wang Y., Deng Q. & Li Z. 2024. Improved foggy pedestrian and vehicle detection algorithm based on YOLOv5. *Journal of System Simulation* **36**(10): 2413–2422.

Sun B., Yang C.C., Xu L., Shang H.B. & Yu S.L. 2024. An object tracking method based on multi-stage features and asymmetric-dilated convolution. *Geomatics & Information Science of Wuhan University* **49**(9): 1712–1722.

Sun F.G., Wang Y.L., Lan P., Zhang X.D., Chen X.D. & Wang Z.J. 2022. Identification of apple fruit diseases using improved YOLOv5s and transfer learning. *Transactions of the Chinese Society of Agricultural Engineering* **38**(11): 171–179.

Sun J., Xu C., Zhang C., Zheng Y., Wang P. & Liu H. 2025. Flood scenarios vehicle detection algorithm based on improved YOLOv9. *Multimedia Systems* **31**(1): 74.

Sun Y. & Gao J. 2022. Defect detection of wheel set tread based on improved YOLOv5. *Laser & Optoelectronics Progress* **59**(22): 2215003.

Sundaresan Geetha A., Alif M.A.R., Hussain M. & Allen P. 2024. Comparative analysis of YOLOv8 and YOLOv10 in vehicle detection: Performance metrics and model efficacy. *Vehicles* **6**(3): 1364–1382.

Tan B. & Wang T. 2024. Design of target detection and ranging system based on YOLOv5 and parallax computing algorithm. *Science, Technology and Engineering* **24**(21): 9015–9024.

Tan H.S., Ma W.H., Tian Y., Zhang Q., Li M.Y., Li M.Q. & Yang X.H. 2024. Improved YOLOv8n for pear target detection. *Transactions of the Chinese Society of Agricultural Engineering* **40**(11): 178–185.

Wang H.Z., Song M.X., Cheng C. & Xie D.X. 2024a. Vehicle distance warning method based on improved YOLOv4-tiny algorithm. *Journal of Jilin University (Engineering and Technology Edition)* **54**(3): 741–748.

Wang K.T., Chen S.F., Chen G., Wei J., Meng L.W. & Chen Q.C. 2024b. Detection method for spherical hedges in gardens based on YOLOv5s. *Journal of Chinese Agricultural Machinery* **45**(8): 262.

Wang W., Xu Z., Qu C., Lin Y. & Liao J. 2024c. BEV space 3d object detection algorithm based on fusion of infrared camera and LiDAR. *Acta Photonica Sinica* **53**(1): 0111002.

Wang X.P., Wang X.Q., Lin H., Li L.X., Yang Y.Y., Meng C. & Gao J. 2022. A survey on improvements of typical deep learning object detection algorithms. *Journal of Computer Engineering & Applications* **58**(6): 1–9.

Wen B. 2024. Improved YOLOv7 automatic driving object recognition algorithm. *Internal Combustion Engines & Parts* **2024**(14): 124–126.

Xie H.T., Chen J.X., Zhang L. & Zhu N.N. 2024. Lightweight SAR image ship recognition algorithm based on spiking neural network. *Journal of Northeast University (Natural Science Edition)* **45**(4): 474–482.

Xu B. & Li Z. 2024. A traffic target recognition and ranging method and device based on YOLO model. Patent CN202410170002.9.

Xu D.G., Wang Z.Q., Xing K.J. & Guo Y.X. 2024a. Improved YOLOv6 remote sensing image object detection algorithm. *Computer Engineering & Applications* **60**(3): 119–128.

Xu Y.W., Li J., Dong Y.F. & Zhang X.L. 2024b. A survey on YOLO series object detection algorithms. *Journal of Frontiers of Computer Science & Technology* **18**(9).

Yan X.H. 2024. Intelligent decision-making support algorithm for construction waste governance in the context of low-altitude economy. *Computer Science and Application* **14**(1): 1.

Yang M. & Qian S.R. 2024. Indoor fire detection based on improved YOLOv8. *Modeling and Simulation* **13**: 4863.

Yang M.L., Zhang X., Guo Y., Yu X.W., Hou Y.N. & Gao J.J. 2022. Wildlife image recognition from infrared cameras based on YOLOv5. *Laser & Optoelectronics Progress* **59**(12): 1215015.

Yang X., Wang H.B. & Dong M.G. 2024. Improved YOLOv5 for traffic sign detection. *Journal of Guilin University of Technology* **44**(2).

Yu S.H., Wei G.L., Zhang S. & Liu M. 2024. Laser SLAM loop detection method with multivariate information. *Information and Control* **40**(5): 594–602.

Zhang F.Z., He J.Z., Hu J.Q., Li E.L. & Wang J.S. 2024a. Real-time ranging method for surface targets of unmanned boats based on monocular vision. *Journal of Ship Science and Technology* **13**: 126–131.

Zhang J.W., Lu Z.T., Zhang Q.H. & Zhan M. 2024b. Research on deep reinforcement learning based on quasi-Newton method in vehicular edge computing. *Journal of Communications* **45**(5): 90–100.

Zhang K., Zhang N., Jiang Z., Dai W. & Zou X. 2023a. Performance analysis of different activation functions in deep learning models. *Cybersecurity and Data Governance* **42**(S01): 149–156.

Zhang X., Yu H., Wu Z.J., Cheng Z.A., Gao C.C., Yang Z.Y. & Wang Y. 2025. Lightweight identification of farmed fish schools based on FasterYOLOv9-Slim. *Modern Fisheries* **52**(1): 99–109.

Zhang Y.T., Huang D.Q., Wang D.W. & He J.J. 2023b. A survey on research and applications of object detection algorithms based on deep learning. *Journal of Computer Engineering & Applications* **59**(18).

Zhao K.X., Qin L. & Xie B.L. 2023. Butterfly species detection based on YOLOv5s. *Modeling and Simulation* **12**(6):

5834–5842.
Zheng H. & Cui Y. 2024. Study on sea surface long-range ranging based on binocular system. *Modern Electronics Technique* **2024**(15): 151–156.
Zheng X., Guan Z., Chen Q., Lu X. & Xia L. 2025. YOLOv8-ADDA: An enhanced algorithm for real-time traffic sign detection. *Engineering Research Express* **7**(1): 015277.
Zheng Z. 2024. Research on intelligent video surveillance systems. *Mechanical and Electronic Control Engineering* **6**(14): 200–202.

School of Mathematical Sciences
Universiti Sains Malaysia
11800 USM Penang
Pulau Pinang, MALAYSIA
E-mail: wenzhu2024@student.usm.my, limlekming@gmail.com, majidkhanmajaharali@usm.my[*]

Department of Computer Science
Xinzhou Normal University
034000 Xinzhou
Shanxi, CHINA
E-mail: 17735009692@163.com

Springtech Ventures Sdn Bhd,
Unit 333, Block A2,
Leisure Commerce Square, Jln PJS 8/9,
Bandar Sunway
46150 Petaling Jaya
Selangor, MALAYSIA
E-mail: elriclee@springtech.ai

Received: 18 February 2025
Accepted: 16 May 2025

[*]Corresponding author