An *et al. Journal of Big Data* (2024) 11:119
https://doi.org/10.1186/s40537-024-00941-6

Journal of Big Data

RESEARCH Open Access

Automated subway touch button detection using image process

Junfeng An[1*], Mengmeng Lu[3], Gang Li[2], Jiqiang Liu[3*] and Chongqing Wang[4]

*Correspondence:
21111093@bjtu.edu.cn;
97098541@qq.com;
jqliu@bjtu.edu.cn

[1] Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing 100044, China
[2] Jinan Rail Transit Group Co., LTD, Jinan 250000, China
[3] Shandong Labor Vocational and Technical College, Jinan 250000, China
[4] Jinan Municipal Bureau of Industry and Information Technology, Jinan 250000, China

Abstract

Subway button detection is paramount for passenger safety, yet the occurrence of inadvertent touches poses operational threats. Camera-based detection is indispensable for identifying touch occurrences, ascertaining person identity, and implementing scientific measures. Existing methods suffer from inaccuracies due to the small size of buttons, complex environments, and challenges such as occlusion. We present YOLOv8-DETR-P2-DCNv2-Dynamic-NWD-DA, which enhances occlusion awareness, reduces redundant annotations, and improves contextual feature extraction. The model integrates the RTDETRDecoder, P2 small target detection layer, DCNv2-Dynamic algorithm, and the NWD loss function for multiscale feature extraction. Dataset augmentation and the GAN algorithm refine the model, aligning feature distributions and enhancing precision by 6.5%, 5%, and 5.8% in precision, recall, and mAP50, respectively. These advancements denote significant improvements in key performance indicators.

Keywords: Touch button detection, Camera, YOLOV8, RTDETRDecoder, P2, DCNv2-Dynamic, Small target, NWD, GAN, Multiscale, Indicators

Introduction

High-speed railway stations require the implementation of image analysis through cameras toachieve passenger flow statistics and abnormal behavior analysis, thereby enhancing the safety of high-speed rail operations [1]. The European Union has achieved the detection of abnormal conditions, such as fighting, rogue behavior, and foreign body detection on railway tracks. Figure 1 illustrates the emergency button in the subway. Upon pressing the button, it sends a warning signal (such as a bell) to the staff and automatically initiates relevant control procedures. For instance, activating the fire alarm button triggers the subway's fire alarm system, initiating evacuation procedures and emergency measures. Pressing the emergency stop button halts the train, leading to passenger evacuation. Similarly, engaging the emergency stop button on an escalator immediately halts its operation, posing potential safety risks such as passenger falls. Touching the gas extinguishing button automatically releases fire-suppressing gas, but inadvertent button presses may result in gas wastage and economic losses.

Fig. 1 Subway Emergency button **a** Emergency stop **b** Emergency stop **c** fire alarm **d** vehicle emergency intercom **e** gas fire extinguishing

In modern metro systems, ensuring passenger safety and operational efficiency are paramount. With advancements in surveillance technology, video surveillance systems have become instrumental in monitoring passenger behavior and preventing potential terrorist activities. Particularly, the ability to detect and identify abnormal behaviors such as accidental activations of emergency buttons in real-time is crucial for maintaining the normal operation of the metro system.

Despite numerous studies focusing on video-based passenger behavior analysis, the detection of accidental button presses still presents challenges. These challenges include the uncertainty and diversity of accidental behavior, as well as the adaptability of detection algorithms in complex environments. Existing methods often require extensive annotated data for model training and may perform poorly under varying lighting conditions, background noise, and passenger flow rates.

Recent years have seen significant progress in deep learning techniques for anomaly detection. For instance, Espinosa et al. introduced the UP-Fall Detection dataset and utilized deep learning algorithms for fall detection [2]. Khraief et al. designed a four-stream CNN architecture, integrating motion, shape, RGB, and depth appearance information, as well as transfer learning and data augmentation techniques [3]. Cao et al. proposed the RailDet method, an effective railway intrusion image recognition technique using deep learning [4]. Ling et al. built the SCAU-SD dataset by introducing self-attention mechanisms and post-processing modules, with the code available on GitHub [5].

Currently, there is a lack of systems that can prompt events after a button is touched, leading to difficulties in determining the touch location and identity. This uncertainty affects rapid decision-making and response. By adopting video/image-based button touch detection, abnormal touches can be timely identified, emergency plans can be formulated, and operational efficiency and passenger safety can be improved, which is of significant importance for quickly understanding and addressing on-site situations. Camera systems can monitor and record touch events in real-time, eliminate false alarms, and ensure resource allocation, promoting an orderly operational environment. Therefore, the proposed use of cameras for touch button detection has practical and far-reaching implications for enhancing metro system operational efficiency and safety.

However, the detection of touch behavior on small buttons presents technical challenges due to increased detection difficulty (e.g., occlusion, lighting changes, complex backgrounds, and the small size of the buttons themselves). Additionally, there are currently no systems that can immediately alert to "button touch" events after an incident, leading to inefficiency and response delays as operators often need to be on-site for inspection.

This paper aims to explore how video and image analysis techniques can be utilized to detect and identify accidental button presses in metro systems. We will outline the current technical challenges, introduce the latest research advancements, and discuss future research directions, aiming to provide practical technical support for safety monitoring in metro systems.

However, the identification of button touches presents two primary challenges. Firstly, there is a deficiency in sufficient datasets and exemplary scene recognition network models. Current focus primarily revolves around single-scene detection. Secondly, the accuracy of recognition is insufficient, failing to meet the demands of actual operations and rendering application in real scenarios challenging. Experimental results indicate suboptimal detection accuracy for behaviors such as bar delivery, crossing gate interactions, and crossing fence activities, with an abnormality rate reaching up to 90%. However, the accuracy of touch button behavior detection remains below 80%.

The primary contributions of this paper are outlined below:

1. We introduce an application framework for touch button detection. Upon anomaly detection, it enables functionalities including toucher identification, trajectory tracking, situational context comprehension, and event handling record storage. This proposed framework establishes an applied technological architecture to enhance subway security and ensure safe travel.

2. Addressing the constraint of limited touch button data samples, we adopted an inductive methodology to construct a dataset. Through the integration of image processing and Generative Adversarial Network (GAN) techniques, we effectively executed data augmentation. We introduced a pioneering dataset tailored for the identification of abnormal touch button interactions within subway environments. Employing an inventive image processing approach to augment the training set, we achieved a noteworthy expansion in sample diversity. This methodology played a pivotal role in elevating the overall performance of the network.

3. Our paper presents the YOLOv8-DETR-P2-DCNv2-Dynamic-NWD-DA method, signifying a noteworthy advancement. This methodology significantly boosts network performance, especially within the YOLOv8 framework. Notably, it demonstrates substantial enhancements in small target detection, establishing high accuracy as a key advantage.

4. The innovative DCNv2-Dynamic module and M_CA_Attention attention mechanism are introduced. The DCNv2-Dynamic, incorporating the M_CA_Attention attention mechanism, enhances the generation capability of the DCN mask. This innovation significantly contributes to the overall effectiveness of the network, particularly in capturing intricate details and improving the accuracy of the detection process. Through comparative experiments, it is evident that our M_CA_Attention attention mechanism outperforms other attention mechanisms.

5. The efficacy of the proposed method, module, and attention mechanism is validated across multiple datasets, providing empirical evidence for the effectiveness of the proposed architecture.

Related work

Routine method for touch-button detection

In contemporary subway stations within China, automation of emergency stop button activation is at a nascent stage, encompassing emergency stop buttons, emergency

stop escalator buttons, fire trigger buttons, and gas extinguishing buttons. Nevertheless, on trains, the installation of emergency intercom buttons enables the subway control center to receive and respond to information when activated, facilitating communication. Simultaneously, the activation of these buttons triggers nearby cameras, allowing dispatchers to monitor the interior of the train. This approach integrates technologies such as the Internet of Things (IoT) and voice communication, necessitating the transmission of alarm signals via the train-ground wireless network and the interconnection of on-board video surveillance systems, voice intercom systems, train-ground wireless systems, on-board dispatch systems, signal systems, and vehicle control management systems. This intricate network necessitates frequent information exchange, and any system malfunction may diminish the functionality's effectiveness.

Our proposed computer vision approach, in contrast, is straightforward, featuring a unified network architecture, reduced points of failure, minimal investment, and simplified construction. It is well-suited for the detection of emergency stop buttons, emergency stop escalator buttons, fire trigger buttons, and gas extinguishing buttons, extending beyond the limitations of onboard voice intercom systems. Computer vision enables the annotation of button touch regions in footage (typically at a resolution of 1920*1080 or higher), facilitating rapid issue identification by operational staff. This functionality is not inherent to traditional onboard voice intercom systems.

Touch Button Recognition Method: In China's subway stations, emergency stop button activation does not proactively link to video images, lacking intelligence. Onboard, an emergency intercom button is installed, which, upon activation, informs the control center and enables communication. The camera near the button is activated, aiding in dispatcher observation. This method employs IoT and voice communication, requiring alarm signal transmission via the vehicle-ground wireless network. Integration of the onboard video surveillance system, voice intercom, wireless network, dispatching system, signaling, and vehicle control is crucial. The network's complexity and high frequency of information exchange render it susceptible to failure, potentially diminishing functionality.

In China's subway systems, emergency stop button activation triggers a buzzer warning in the control room's IBP panel. Pressing the fire alarm button prompts a message on the fire alarm system's workstation. However, this method has limitations: it depends on the network connectivity of the buttons and the station control room, and it lacks direct on-site situational awareness.

Our machine vision approach, on the other hand, is characterized by its simplicity, a single network architecture, reduced points of failure, low resource investment, minimal system interfaces, and straightforward implementation. It is specifically designed to detect emergency stop buttons, emergency stop ladder buttons, fire trigger buttons, and air extinguishing buttons, offering broader applicability. Operators can quickly identify issues on the screen (typically at resolutions of 1920 * 1080 and above), a capability not present in traditional vehicle voice intercom systems.

Unlike the conventional approach, where network failure can prevent remote communication, the machine vision method is not reliant on network accessibility and can continuously monitor scene imagery. Alarms are triggered in response to detected anomalies.

Method of target detection

Following the research literature, the application of machine vision for subway button touch detection has not been explicitly documented. However, this domain aligns with the broader field of target detection, which has been extensively studied. Target detection is a pivotal visual task that encompasses both recognition and localization. Contemporary target detection systems are characterized by two predominant architectures: Convolutional Neural Network (CNN)-based, exemplified by YOLOV8, and Transformer-based, represented by DETR and RT-DETR. Despite DETR's computational cost drawback, the current research focus has shifted towards the remarkable precision and efficiency exhibited by RT-DETR [6]. YOLO-NAS [7] employs strategies such as knowledge distillation and Distribution Focal Loss, achieving a mAP value that surpasses YOLOV8 by over 1 percentage point.

RT-DETR stands out for its efficiency, balancing high accuracy and rapid processing speed. To mitigate the Non-Maximum Suppression (NMS) delay, an end-to-end detector is employed. Notably, on the COCO dataset, RT-DETR-X has achieved exceptional performance, attaining a substantial 54.8% Average Precision (AP) at a remarkable speed of 74 Frames Per Second (FPS). This underscores the significance of RT-DETR-X in addressing the demands of target detection tasks with both efficacy and efficiency. Li et al. [8] combined YOLOv8 and mt-DETR to achieve more accurate detection of fish. However, RT-DETR is not much used in related industrial fields.

With the rapid development of deep learning, target detection technology has become a focal point of research. Common two-stage algorithms include R-CNN, Fast R-CNN, and Faster R-CNN. On the other hand, single-stage methods include the You Only Look Once (YOLO) series and the Single Shot MultiBox Detector (SSD). The YOLO algorithm predicts both object boundaries and classifications for each grid within a multitude of grids. The evolution of YOLO from versions YOLOV1 to YOLOV8 has seen enhancements in various components. This includes improvements in the backbone network, such as the introduction of Darknet19 in V2, feature extraction modules like FPN, C3, and SPPF in V3, V5, and V7, and activation functions like the wish function in V4. These developments aim to boost feature extraction capabilities and enhance the expression of semantic information, ultimately optimizing target detection. Numerous studies have been conducted on the realization and improvement of YOLOV8. Some notable enhancements include the adoption of the Global Attention Mechanism (GAM) to address occluded foreign bodies and the replacement of the SPPF module with the SPPCSPC module for automatic detection of power grid anomalies, thereby improving inspection efficiency [9]. Yaping et al. [10] provided a comprehensive summary of the single-stage YOLO series detection algorithm, its improvement methods, and an analysis of its advantages and disadvantages.

Improvements in YOLOV8 mainly focus on performance and model parameters. These enhancements encompass the addition of attention mechanisms, changes in network architecture, improvements in loss functions, and modifications to convolution types. Model parameter improvements include model pruning and enhancements to convolution functions. Table 1 provides a summary of the methods and ideas for algorithmic improvement.

Table 1 The improvement scheme of YOLOV8

Year, the author	Detection object	Data set	Method	Index ascension	Parameters and computations
2023,Yang, G [11]	Tomato	Self-built tomato data set	DSConv, Design of dual-path attention gate module (DPAG), feature enhancement module (FEM)	The accuracy increased by 2% and the recall increased by 0.8%	The model size was significantly reduced from 22 to 16 M, with FPS: 138.8
2023,Wenjie Yang [12]	Cattle	Self-built	Variable convolution, coordinate attention mechanism (CA)	mAP:62.5%	72.9 frames per second (FPS)
2023, Li Song [13],	Road damage	RD D2022,Road Damage	Lightweight algorithm, BOT, CA,C2fghost	mAP increase 2%, 3.7%	reduce 6.7%, 8.5%
2023,Yuan Hongchun [14]	Fish	FishNet	YOLOv8n-GCBlock-GSConv	The mAP was improved by 2. 0%	reduce0. 5 GFLOPS
2023, Geng Huantong [15]	Road cracks	RDD20(Road Damage Detection 20)	C2f-Faster module is followed by anotherSE(Squeeze-andExcitation)	The F1 value was increased by 0.8%	Model reduce42%, FPSraise292
2023, Zhouying [16]	Photovoltaic cell defects	Self-built	GauGAN Data enhancement, embed the context aggregation module (CAM), and build a multi-attention detection head (MADH) to replace the decoupling head	The average accuracy reached 89.90%	The number of model parameters is given in 13.13 M
2023, Xiong Enjie [17]	Traffic sign	China Traffic Signs Testing Data Set TT100K	GhostConv, GAM, loss function Giou	Precision and mAP were improved by 9.5%, 6.5%	Number of parameters and model reduction of 0.223G,0.2 MB
2023, Chen Yifang [18]	SAR to image the aircraft targets	SADD(SAR Aircraft Detection Dataset)	deformable convolutional networks, DCN), GAM attention, GAM), (Wise-IoU)	The highest accuracy was increased to 98.1%	parameters decreased 61.18%, and the calculated amount decreased 18.29%
2023, Gao ang [19]	Indense pedestrians	CrowdHuman and WiderPerson	Occlusion of perceptual attention mechanism, Dynamic decoupling head, and regression loss of Wise-IoU combined with distributed focus loss	The AP50 reached 90.6% and 92.3%, respectively	
2023 , Wang, G, [20]	Drone aerial shots of small targets	UAV dataset	Wise-IoU (WIoU) v3 serves as a bounding box for regression loss, gradient allocation strategy, attention mechanism of BiFormer, and Focal FasterNet feature processing module	The average detection accuracy was 7.7% higher than the baseline model	

Table 1 (continued)

Year, the author	Detection object	Data set	Method	Index ascension	Parameters and computations
2024, Siyu Chen [21]	Corn seed crack	Corn seed crack DATASET	COTnet,EMA,GSConv,siou,image enhancement	The average accuracy (AP) value was 3.1% higher than the original model	The average single-frame image detection time was 7.49 ms
2023, Zheng Wang [22]	Non-standard miners' behavior	Self-built	SimAM, dcn,Collaboration loss function	95.7% mAP, 95.3% accuracy and 95.1% recall	
2023, Shizhong Yang [23]	Strawberry maturity	Self-built	LW-Swin Transformer	The mAP 0.5 on the validation set was improved by 1.6%, 33.5% and 3.4% compared to YOLOv5s, CenterNet and SSD, respectively	parameters was 51.93%,speed was 19.2
2023, Hongyang Zhao [24]	Fire	Self-built	The VQ-VAE integrates with the YOLOv8	A 94.2% F score and 93.5% accuracy were achieved	
2023, Cao, S [25]	Unmanned aerial vehicle (UAV) detection	Self-built	BiFPN,CARAFE,BiFormer	Precision of 97.1% and a recall rate of 94.9%, with an FPS of 110	
2023, Fan, J [26]	Surface defects detection	Surface defects	Anchor optimization,cam,DSConv	Mean average precision of 79.3%	FPS:72
2023, Ji, Y [27]	Textile defect detection	Textile	MPAM,CMFFN,EWIoU	Up by 6.5 points	
2024, Wenkai Xu[28]	FISH	Self-built	MHSA,wIOU		
2023, Yu hangliu [29],	surface of wind turbine blades	Self-built	Soft-NMS,RBiFPN,CAFL	Better than YOLOv8	
2024, Liuyang [30]	fish	SeaCLEF2017	Vision Transformer,Caraffe,	AP50:95.2%	The fops are about one-third lower
2023, Yifan Bai [31]	detecting flowers and fruits on strawberry	Self-built	GSConv,Swin Transformer	The mAP was 4.6% higher	45 FPS
2024, Yanming Hui [32]	A small target	VisDrone	CNeB,Swin Transformer,SimAM,CWDHead	3.9%mAP, 2.0%AP50 higher	

Method

Application architecture for the button detection

The depicted Fig. 2 illustrates the application architecture for button detection. Initially, the camera adjusts its focus to the location of the buttons, and the AI model is employed to detect the occurrence of button touches. If a button touch is detected, three valuable applications ensue.

1. Real-time Scene Understanding: Upon identifying a button touch, the system captures a segment of video before and after the incident. This video data is then transmitted to mobile terminals, enabling on-the-go work for operational personnel. It eliminates the necessity of being in the control room to receive alerts and video sequences.

2. Identification and Trajectory Tracking of Button Touchers: Upon detecting a button touch event, facial recognition algorithms are employed to capture the face region. The individual's identity is confirmed against existing subway identity systems, such as card systems or facial entry systems. Concurrently, the algorithm extracts the person's body region and utilizes pedestrian re-identification algorithms to track the individual's movements, establishing a trajectory. This application enables rapid identity confirmation, expediting information dissemination to law enforcement, thereby reducing search time and mitigating secondary impacts.

3. Archiving Event Processes: The system archives the process of the event, storing images and information on the handling process. This allows for quick searches and replays of the event. Moreover, it automatically generates event reports, enriches the abnormal image sample library, retrains the model, and enhances network performance. This iterative process elevates the accuracy of button touch detection.

The system architecture encompasses multiple application functionalities, and effective alarm processing methods are integral to its operation. As depicted in the Fig. 3, the alarm processing workflow involves pre-configuring AI settings rules (such as detection locations, areas, timeframes, etc.). Following the diagnosis of anomalies by AI algorithms, the alarm management process is executed, encompassing alarm notifications, storage, utilization of alarm data, and post-event handling. This workflow constitutes a series of tasks and functions related to monitoring, security, and event processing.

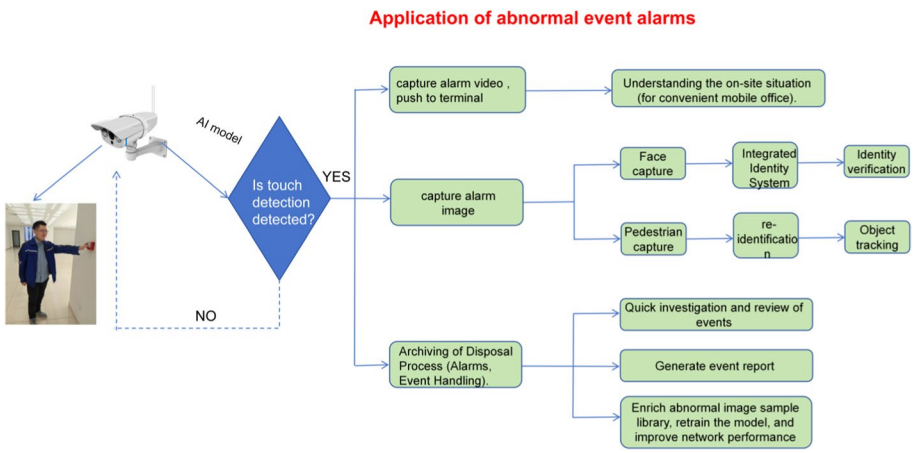


Fig. 2 Application architecture of the button detection

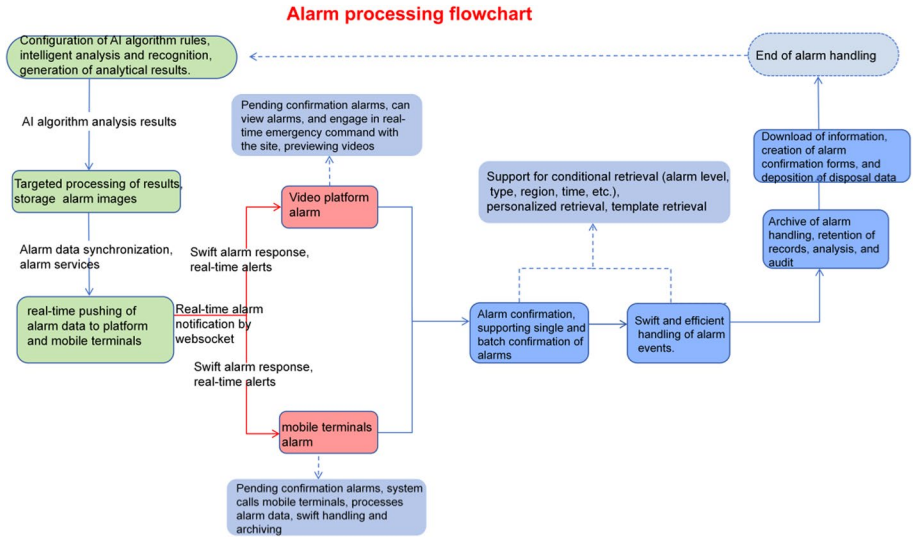


Fig. 3 The process of alarm disposal

In summary, these tasks involve the alarm processing flow of the system, including the collection, analysis, confirmation, disposition, and archiving of alarm data. Real-time alarm notifications are pushed to the platform and mobile terminals, supporting both conditional and personalized retrieval. The functionalities include handling alarm data, generating alarm confirmation forms, and storing disposition materials. These tasks and functions are designed to enhance the efficiency of the monitoring system, strengthen security management, and enable rapid response and handling of various events and alarms, thus realizing the functionalities depicted in the application architecture diagram.

Algorithmic architecture for the push-button detection

In this paper, we introduce a novel and improved model, YOLOv8-DETR-P2-DCNv2-Dynamic-NWD-DA. A key modification is the replacement of the last layer of YOLOv8 with RTDETRDecoder. To enhance the accuracy of small target detection, we incorporate the P2 small target detection layer, enabling YOLOv8 to more effectively detect small target objects. The innovative DCNv2-Dynamic algorithm, along with the NWD loss function (Normalized Gaussian Wasserstein Distance) and the DA algorithm, are employed to further optimize the model.

The feature map layers of the neck network are transformed from the original P3, P4, P5 to P2, P3, P4, P5. The detection head is then replaced with the detection head from RT-DETR. The SPPF layer preceding the C2f module in the backbone network is swapped with the C2f-DCNv2-Dynamic module, where the Dynamic component employs the M_CA_Attention attention mechanism. This attention mechanism adjusts the generation parameters of DCNv2, which is further augmented with the NWD loss function to enhance small object extraction capabilities. To mitigate the distribution discrepancy between the training and test sets, data augmentation algorithms are incorporated into the dataset. Owing to prevalent necessity for button detection, the

incorporation of each module within the algorithm has been innovatively applied. We delineate the underlying rationale for this approach.

Theoretically, the improvement of the network structure, which includes the adoption of a detection head and the capability for small object detection, enhances the model's capacity and generalization ability. The NWD loss function is more effective in learning boundaries, which can help the model better detect small objects, maintaining good performance even in low-resolution or occluded scenarios. This encourages the model to learn more robust feature representations. Data augmentation techniques enhance the model's generalization ability by increasing the diversity of samples. Data indicate that while maintaining a similar distribution, the diversity of the generated data is significantly improved.

Our approach also integrates advanced techniques such as data set enhancement through image processing and the Generative Adversarial Network (GAN) algorithm. This combination of methods contributes to improved performance in detecting small target objects. The main network architecture diagram is provided below in Fig. 4 for reference.

The pseudo code is as follows in Table 2.

1) RTDETRDecoder: In our proposed model, YOLOv8-DETR-P2-DCNv2-Dynamic-NWD-DA, a significant enhancement is made by replacing the last layer of YOLOv8 with RTDETRDecoder. The RTDETRDecoder is a decoder module equipped with auxiliary predictive headers, which iteratively refines the object queries to generate box coordinates and confidence scores. This decoder module is designed using the Transformer architecture and incorporates deformable convolution. This combination facilitates the more efficient prediction of button touch behaviors. The module predicts bounding boxes and class labels for objects in the image, integrating information from multiple layers and passing through a series of Transformer decoder layers to produce the final predictions. This strategic modification enhances the model's capability to accurately detect and classify objects in the images, particularly optimizing the prediction of button touch behaviors.

The decoder component within the model, as depicted in Eq. 3, is responsible for processing feature map and generating object bounding boxes and classes.

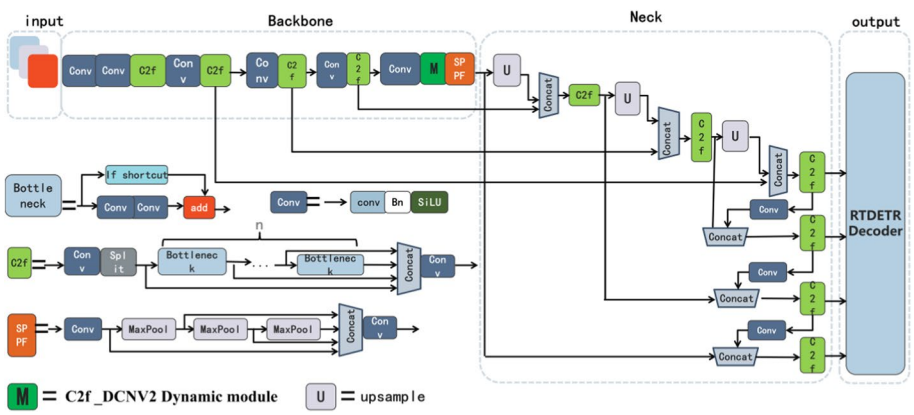


Fig. 4 Architecture diagram

Table 2 Pseudo code

pseudo code
/* Define the Backbone Architecture */ Let B be the backbone network, which takes an input image and processes it through a sequence of convolutional layers and C2f modules with increasing filter sizes and strides, terminating in a SPPF module, and producing a feature map . /* Define the Head Architecture */ Let H be the head network, which takes the feature map from the backbone and processes it through a series of upsampling layers, concatenation modules, C2f modules, and a final RTDETRDecoderModule to produce bounding boxes and class probabilities. /* Define the Complete Model */ Let M be the model, consisting of the backbone B and head H, which takes an input image and outputs bounding boxes and class probabilities. /* Data and Loss */ Input images are augmented using function DA, and the model's loss is computed as a combination of giou and normalized weighted distance (nwd) losses. /* Training and Testing */ The model M is trained using the augmented data and loss function to learn to predict accurate bounding boxes and class probabilities. The trained model is then tested on a separate set of images to evaluate its performance.

RTDETRDecoder(P2, P3, P4, P5) refers to the utilization of the decoder part within the model to handle four distinct feature map at various resolutions: P2, P3, P4, and P5. These feature map are sourced from the encoder part, representing different levels of abstraction within the image.

$$Decoder(X1, X2, ..., XN) = TransformerDecoder(Encoder(X1), Encoder(X2)...) \quad (1)$$

X1, X2, …, XN denote the feature map at different resolutions, with Encoder representing the encoder part and TransformerDecoder representing the decoder part.

2) P2 layer: Additionally, to enhance the accuracy of small target detection, we introduce the P2 small target detection layer in our YOLOv8-DETR-P2-DCNv2-Dynamic-NWD-DA model. This integration transforms the original three-layer pyramid architecture into a four-layer pyramid architecture, creating a deeper multi-scale feature pyramid network. This modification significantly improves the model's feature extraction capabilities, leading to more accurate detection of small target objects. The expanded feature pyramid network captures more detailed feature information across multiple scales, encompassing both the high-resolution details and global characteristics of low-resolution features. This approach enhances the model's adaptability and robustness, providing a comprehensive understanding of the scene. The lower resolution of the feature pyramid can be particularly useful for accurately locating the bounding box of small targets. This precise localization improves the accuracy of detection results, ensuring more precise target positioning and identification for touch button behaviors.

As the pixel size of an object decreases, the error in feature extraction increases. Therefore, enhancing the accuracy of small object detection is essential. In datasets containing small objects, issues such as missed detections or suboptimal detection results may arise. feature map are typically generated at different resolution levels, denoted as p2, p3, p4, p5, etc., with P2 being one of these levels. Each level represents a feature map at a different scale, with P2 typically referring to a higher resolution level capable of capturing finer-grained details. The incorporation of a P2 small object detection layer in feature fusion enables more effective detection of small objects, which typically occupy fewer pixels in the final image and are more prone to being overlooked or misclassified. A dedicated P2 layer allows the network to more acutely detect and localize small objects, thereby enhancing the accuracy of small object detection.

$$y = \text{RTDETRDecoder}(P2, P3, P4, P5) \tag{2}$$

y represents the discriminative information regarding the target's location and category by the network, with P2 denoting the small object detection layer.

(3) DCNv2-Dynamic module and M_CA_Attention

In our model YOLOv8-DETR-P2-DCNv2-Dynamic-NWD-DA, we leverage the innovative DCNv2-Dynamic algorithm. A notable modification involves replacing the c2f module in front of the SPPF with the C2f_DCNv2_Dynamic module. This modification is designed to enhance the capabilities of the feature extraction process. The schematic diagram of this module is depicted below in Fig. 5 for reference.

The DCNv2 introduces additional parameters in a specific order. First, the data stream, denoted as x, is processed, followed by the convolution kernel or filter. Subsequently, in sequential order, we have the offset and mask. The offset signifies the displacement of each convolution window, while the mask represents the sampling weights at various positions within the window.

The integration of DCNv2 into YOLO contributes to an enhanced understanding of image content, particularly in complex scenes. This incorporation enables the model to discern subtle structures and patterns within the image, providing a more comprehensive and abstract representation of features. DCNv2 significantly augments the model's capacity to express image features. To bolster the robustness of the generated mask,

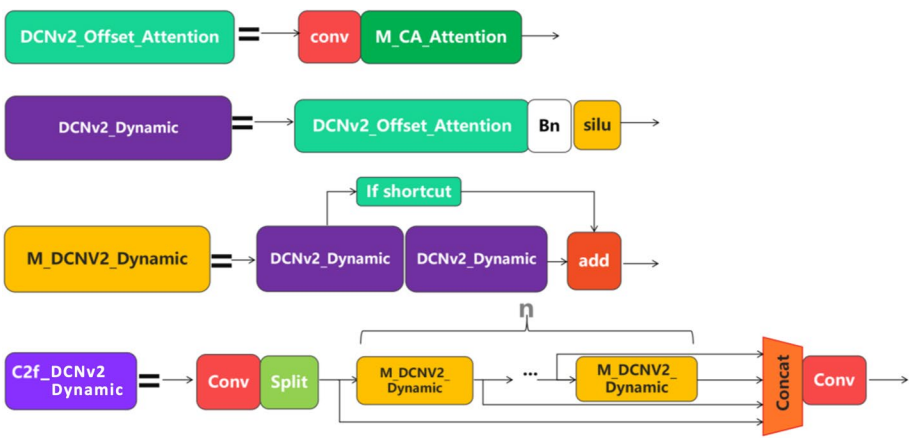


Fig. 5 C2f _DCNv2 Dynamic module

DCNv2 fully extracts the weight coefficients, optimizing the convolution operation at each sampling position to facilitate the extraction of superior features. The incorporation of a flexible, dynamically adjustable feature sampling location, along with the inclusion of an attention mechanism responsible for offsets and mask generation, synergistically enhances the overall effectiveness of the model.

M_CA_Attention: In the original DCNv2 [33], the mask generation method is relatively basic, utilizing conventional convolution, resulting in limited capabilities. To enhance this, we employ the convolution M_CA_Attention with multiplex feature extraction to generate the mask. This involves introducing the proposed attention mechanism after the convolution process to bolster mask generation, providing a more robust outcome. Features are extracted at each location, contributing to a more effective result.

We enhance the attention mechanism by incorporating additional attention mechanisms, such as CPCA [34] et al. Building upon the CoordAt attention [35], we further refine the attention mechanism by introducing a channel branch. This involves connecting different-dimensional pooling data to facilitate information exchange and amplifying the weight of the channel branches and pooling data alterations. Additionally, we reinforce the horizontal and vertical data dimension weights, resulting in a collaborative attention mechanism across multiple levels. The main architecture of M_CA_Attention is depicted in Fig. 6.

Utilizing a multi-path coordinate attention mechanism for feature map processing, this algorithm employs several steps. Firstly, it employs global average pooling to acquire overall features. Subsequently, adaptive average pooling in the vertical and horizontal directions captures information about rows and columns independently. After integrating this information, a convolution operation introduces a multi-path attention mechanism, separately weighting the height and width. Finally, the weights obtained through a sigmoid function are applied to the input image, providing weighted representation in the channel, row, and column directions to attain a more representative feature representation. The algorithm possesses the following advantages: (1) Comprehensive integration of multi-path information enables a more holistic understanding of features in different directions within the image. (2) The coordinate attention mechanism selectively

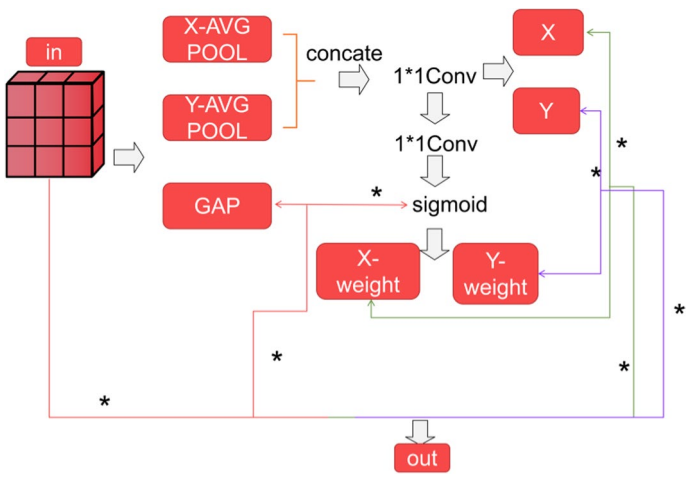


Fig. 6 Architecture diagram of M_CA_Attention

emphasizes or diminishes specific regions of the image, thereby better capturing the inherent structures and patterns. (3) Collaboration between global and local features aids in maintaining flexibility when dealing with features of different scales, adapting well to various image contents. (4) Enhanced model expressiveness enables better adaptation to the nonlinear relationships in the data. The spatial attention mechanism is applied by channel, and the final calculation involves element-by-element multiplication, followed by Sigmoid activation. Pseudocode is presented in the Table 3.

The pseudo-code uses mathematical symbols to describe a deep learning model's operations. "$\in$" indicates membership in a set, "Let" defines a term, "Concatenate" joins tensors, "$\times$" denotes element-wise multiplication, "$\sigma(\cdot)$" is the Sigmoid activation function, "T_global$(\cdot)$" represents global average pooling, "Mean$(\cdot)$" calculates averages, and "permute" changes dimension order. These symbols are used in deep learning to precisely and succinctly describe our algorithms.

Efficient feature extraction is crucial for enhancing model performance and robustness. Common techniques include channel attention, spatial attention, feature map stacking, up-sampling, multi-scale feature fusion, residual connections, advanced feature down-sampling, and attention residual networks. These techniques offer distinct advantages but also possess weaknesses. For instance, channel and spatial attention may increase computational complexity; feature map stacking may lead to an increase in model parameters; up-sampling and down-sampling operations may introduce additional computational overhead; and while residual connections and attention residual networks can enhance model performance, their design and implementation are more intricate. Given our task's requirement for small object detection, we have devised a strategy that balances these advantages and weaknesses to design the attention mechanism presented in this paper.

M_CA_Attention is an enhanced attention mechanism that achieves more effective fusion of feature information at different levels by adding channel branches and enhancing weights. This mechanism facilitates information exchange between data from different dimension pooling operations, heightens the model's attention to critical features, and enables collaborative attention across multiple levels, thereby improving the model's capability for recognition and detection in complex scenarios.

$$y1 = \sigma(\mathrm{cat}([\mathrm{avgx}(F), \mathrm{avgy}(F)])) \tag{3}$$

$$y2 = \mathrm{gap}(F) * \sigma(y1) \tag{4}$$

$$y3, y4 = \mathrm{split}(y1), \mathrm{split}(y2) \tag{5}$$

$$\mathrm{out} = F * \mathrm{gap}(F) * [y3(1) * y4(1)] * [y3(2) * y4(2)] \tag{6}$$

In the context of the attention mechanism, 'avg' denotes the pooling operation, 'Sigmoid' represents the activation function, 'F' denotes the input feature, and 'GAP' signifies global average pooling.

(4) NWD loss function. We adopt the NWD [36] loss function, specifically the Normalized Gaussian Wasserstein Distance. The primary advantage of Wasserstein

Table 3 Pseudocode of the proposed attention mechanism

```
Input: x ∈ ℝ^{C×H×W} - a feature mAP with C channels, height H, and width W
Output: x1 ∈ ℝ^{C×H×W} - the feature mAP adjusted by the spatial attention mechanism

Function forward(x):
    /* Spatial Attention Mechanism */
    Let φ_spatial(·) denote the spatial attention mechanism.
    Define φ_spatial(x):
        /* Adaptive Average Pooling */
        Let ψ_vertical(·) and ψ_horizontal(·) denote adaptive average pooling along the vertical and horizontal dimensions, respectively.
        vertical_pool = ψ_vertical(x) ∈ ℝ^{C×1×W}
        horizontal_pool = ψ_horizontal(x) ∈ ℝ^{C×H×1}

        /* Concatenation */
        pooled_features = Concatenate(vertical_pool, horizontal_pool) ∈ ℝ^{C×(H+1)×(W+1)}

        /* Convolution */
        Let Φ_conv(·) denote a convolution operation with a kernel size of 1×1 and stride 1.
        attention_weights = Φ_conv(pooled_features) ∈ ℝ^{C×H×W}

        /* Activation */
        Let σ(·) denote the Sigmoid activation function.
        attention_weights = σ(attention_weights) ∈ ℝ^{C×H×W}

        Return attention_weights

    /* Apply Spatial Attention */
    attention_weights = φ_spatial(x)

    /* Compute Adjusted Feature mAP */
    Let Γ_global(·) denote the global average pooling operation.
    global_pool = Γ_global(x) ∈ ℝ^{C×1×1}
    mean_attention_weights = Mean(attention_weights, dim=1, keepdim=True) ∈ ℝ^{1×H×W}

    /* Element-wise Multiplication and Activation */
    x1 = x × attention_weights[:, :x.size(2)] × attention_weights[:, x.size(2):].permute(0, 1, 3, 2) × (global_pool * mean_attention_weights).sigmoid()

    /* Return Adjusted Feature mAP */
    Return x1
```

distance lies in its ability to measure distribution similarity even in cases where there is no overlap or the overlap is minimal. This characteristic makes it particularly suitable for measuring the similarity between small objects. Given that the proposed touch button behavior involves smaller targets, this loss function is well-suited for the task. Consequently, the detection performance is consistently enhanced, facilitating the estimation of position deviations for small objects.

Inspired by the literature, this paper explores the fusion of the GIOU loss function with the NWD loss. The combined loss function is formulated as follows in (7)–(9). This integrated loss function leverages the strengths of both GIOU and NWD, aiming to further improve the model's ability to accurately estimate the position deviation of small objects, particularly beneficial for our touch button detection task.

$$giou = iou - \frac{|C \backslash (A \cup B)|}{|C|} \tag{7}$$

$$iou = \frac{|A \cap B|}{|A \cup B|} \tag{8}$$

$$loss\,(giou, nwd) = a * loss_{giou} + (1 - a) * loss_{nwd} \tag{9}$$

In the formulation below, A and B represent the true and predicted target boxes, respectively, while C represents the maximum external rectangle encompassing both A and B:

Here, Loss NWD represents the loss function corresponding to NWD, and a is the relative weight coefficient that balances the GIOU loss and the NWD loss. This combined loss function allows for a flexible and weighted integration of the GIOU loss and the NWD loss, optimizing the trade-off between the two for improved accuracy in estimating the position deviation of small objects.

It is imperative to establish the definition of the Normalized Wasserstein Distance (NWD). NWD is a normalized variant of the Wasserstein distance, employed to quantify the discrepancy between two probability distributions. In the context of object detection, the NWD loss function serves as a guide for the model to learn how to predict bounding boxes more effectively.

The Wasserstein distance is defined as:

$$\mathrm{W}(\mathrm{P}, \mathrm{Q}) = \inf_{\pi \in \Pi(\mathrm{P},\mathrm{Q})} \int_{\mathrm{X}\times\mathrm{Y}} \left|\mathrm{x} - \mathrm{y}\right| \mathrm{d}\pi\left(\mathrm{x}, \mathrm{y}\right) \tag{10}$$

where (P) and (Q) are two probability distributions, and (P,Q) represents the set of all probability measures that can be formed by mixing P and Q. The Euclidean distance between two points, x and y, is denoted as $||\mathrm{x} - \mathrm{y}||$.

The Normalized Wasserstein Distance (NWD) represents an improvement upon the Wasserstein distance by normalizing it through division by the size of the supports of the two distributions. This normalization serves to mitigate the bias towards larger distributions when comparing distributions of different sizes. The definition of NWD is as follows:

$$\mathrm{NWD}(\mathrm{P}, \mathrm{Q}) = \frac{\mathrm{W}(\mathrm{P}, \mathrm{Q})}{\sqrt{\mathrm{H}(\mathrm{P})\mathrm{H}(\mathrm{Q})}} \tag{11}$$

where H(P) and H(Q) are the entropies of P and Q, respectively.

In object detection, the NWD loss function is utilized to direct the model's learning in the prediction of bounding boxes. Specifically, the Normalized Gaussian

Wasserstein Distance (NGWD) loss function is applied to ensure consistency between the predicted and the actual bounding boxes. This approach leverages the Wasserstein distance from optimal transport theory to compute the distance between distributions.

In practice, we utilize the 'wasserstein_loss' function to calculate the Wasserstein distance loss between the predicted and the actual bounding boxes. This involves computing the Euclidean distance (centroid distance) and the width-height difference (wh_distance) between the two centers of the bounding boxes. These are then summed to obtain the Wasserstein distance (wasserstein_2). Finally, the loss is returned as the negative exponential function of the square root of wasserstein_2 divided by a constant (i.e., -torch.sqrt(wasserstein_2)/constant). This is the final form of the Wasserstein loss. The formula is represented as:

$$\text{loss} = -\exp\frac{-\sqrt{\text{wasserstein\_2}^2}}{\text{constant}} \tag{12}$$

where sqrt denotes the square root, and exp denotes the exponential function.

(5) DA using GAN and image processing. The DA (Data Augmentation) algorithm employed in this study integrates both image processing and the Generative Adversarial Network (GAN) algorithm. The primary ideas behind this approach are as follows:

(1) Acquisition of New Target Sample Sets:

Begin with the original samples labeled as targets.

Extract the targets from these samples and perform image processing.

First, identify positions corresponding to human skin color.

Next, identify positions corresponding to the red area.

Merge the two identified parts to create a region of interest (ROI) representing the target after image processing.

Utilize this processed sample set in the DCGAN algorithm to expand the sample set and generate a new target sample set, denoted as Set A.

(2) Data Enhancement of the Training Sample Set:

From the test set, randomly select 1000 samples.

For each sample in this subset, randomly select data from the sample set A and append it to the original sample.

The augmented samples are then added to the training set, while the test set remains unchanged.

This process effectively augments the dataset, introducing variations and diversifying the training samples to improve the model's ability to generalize and perform well on unseen data.

The image change process is shown in the Fig. 7, and we can see the extraction of the skin color area and the red area.

It appears that the figure above illustrates the results of image transformations applied to a sample image. The sequence involves operations such as skin color positioning, red

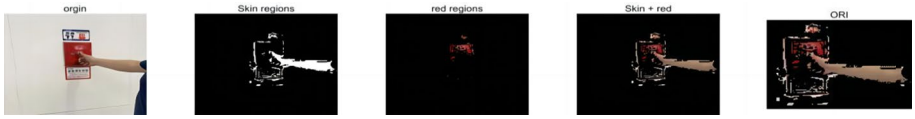


Fig. 7 Image changes and ROI region extraction

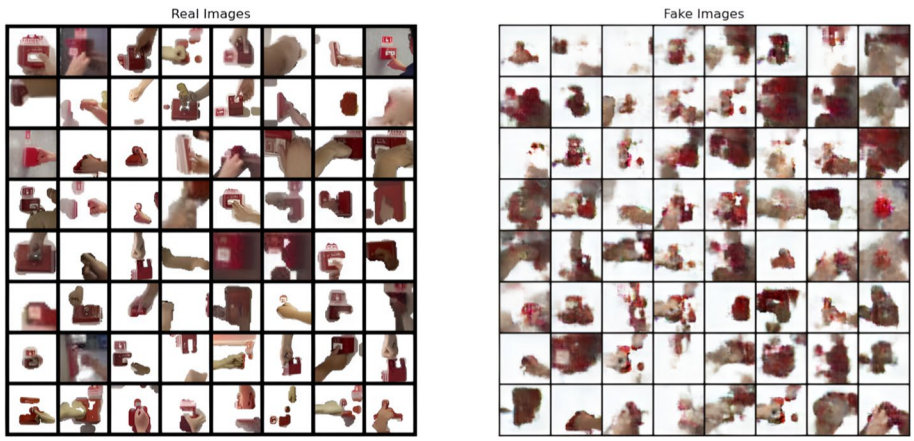


Fig. 8 The dataset generated by the DCGAN network

positioning, merging of skin color and red positioning, and finally, the extraction of the region of interest. This series of operations effectively eliminates a substantial number of non-button areas, enhancing feature perception in the processed image. The highlighted region of interest after these changes likely isolates the target button area, facilitating more accurate and efficient detection of touch button behavior in subsequent model training and testing processes.

As can be seen from the Fig. 8, the method adopted by DCGAN can generate some effective data sets, laying the foundation for the expansion of subsequent data sets.

Experimental result

Experimental setting and dataset

Hardware configurations for our experiments include Python-3.9.13 torch-2.0.1 + cu118 CUDA:(NVIDIA A100, GPU, 40 GB),i7CPU, 32 GB RAM, The software environment includes Linux, During the experiments, To ensure uniformity, all input images were resized to dimensions of 512×512 pixels, The hyperparameters settings for the different models are shown in Appendix 4.

Accommodating variations in sample sizes. These configurations provide the computational foundation for our experiments, allowing for efficient training and evaluation of our model for touch button behavior detection.

We enlisted volunteers for artificial simulation, mimicking the touch button behavior by placing their hands on the subway emergency button without actually pressing it. Subsequently, we gathered action photos as experimental samples. Figure 9 illustrates a subset of these sample cases, focusing on diverse touch scenarios at different stations, locations, and scenes. The buttons include the emergency stop button and the automatic fire alarm button, representing various forms of touch button abnormal behavior.

Fig. 9 The sample case of the button touching

We address these challenges by constructing a dataset comprising 10,947 images, with 5381 allocated as training samples and 5566 as test samples. Leveraging the YOLOv8n as the benchmark network, our research introduces innovative methodologies.

Evaluation indicators

In this paper, we employ a range of evaluation metrics to comprehensively assess the performance of our model. The classified comprehensive evaluation index (F1) and mean Average Precision (mAP) are used to measure the accuracy of the model in classification and object detection tasks, while the number of parameters (Params) and Giga floating point operations per second (GFLOPs) reflect the complexity and computational efficiency of the model. Additionally, Frames Per Second (FPS) evaluates the model's capability for real-time processing. Together, these metrics ensure a thorough and multi-faceted understanding of the model's performance.

Ablation experiments and contrast experiments

Choosing to improve the different networks based on YOLOv8n yields the following results as follows in Table 4.

Upon examination of performance curves in Fig. 10, including the F1-confidence curve, Precision-Recall (P-R) curve, and the mean Average Precision (mAP) curve, it is evident that the enhanced YOLOv8 outperforms the original YOLOv8. Notably, the improved model exhibits clear advantages in various performance indicators. Noteworthy improvements are observed even at 100 iterations, surpassing the performance achieved by the original YOLOv8 at 120 iterations. This demonstrates the effectiveness of our algorithm in achieving superior performance metrics within a shorter timeframe.

In summary, our model exhibits substantial improvements over the baseline network YOLOv8. The Precision, recall rate, and mAP values have seen significant increases of 6.5%, 5%, and 5.8%, respectively, with a 1.3 percentage point improvement in mAP50:95. These enhancements demonstrate a notable and effective improvement in the model's performance, making it particularly well-suited for the button touch scenario explored in this study.

Table 4 Comparison table of the different algorithms

	YOLOv8n (baseline)	YOLOv8-DETR	YOLOv8-DETR-P2	YOLOv8-DETR-P2-DCNv2	YOLOv8-DETR-P2-DCNv2-Dynamic	YOLOv8-DETR-P2-DCNv2-Dynamic NWD	YOLOv8-DETR-P2-DCNv2-Dynamic NWD-DA(ours)	YOLOv8-DETR-P2-Ortho	YOLOv8-DETR-P2-C2f_Faster_Rep_EMA	YOLOv8-DETR-P2 -Ortho-InnerIoU
GFLOPs	8.1	11.7	26.4	26.3	26.3	26.3	26.3	26.4	24.6	26.4
P	0.773	0.833	0.837	0.835	0.834	0.838	0.838	0.836	0.823	0.827
R	0.756	0.777	0.783	0.784	0.782	0.794	0.806	0.781	0.754	0.784
mAP50	0.771	0.809	0.813	0.813	0.814	0.82	0.829	0.811	0.795	0.806
mAP50-95	0.304	0.302	0.307	0.313	0.308	0.314	0.317	0.307	0.297	0.298
Weight size	5.9M	11.8M	12M	12.1M	12.1M	12.1M	12.1M	12.2M	63.7M	12.2M
FPS	179.9	98.6	85.6	44.6	43.5	78.3	76.4	42.0	55.5	68.8
layers	168	232	262	263	275	275	275	346	418	346
parameters	3005843	6091124	6185364	6216751	6220477	6220477	6220477	6191988	5482056	6191988

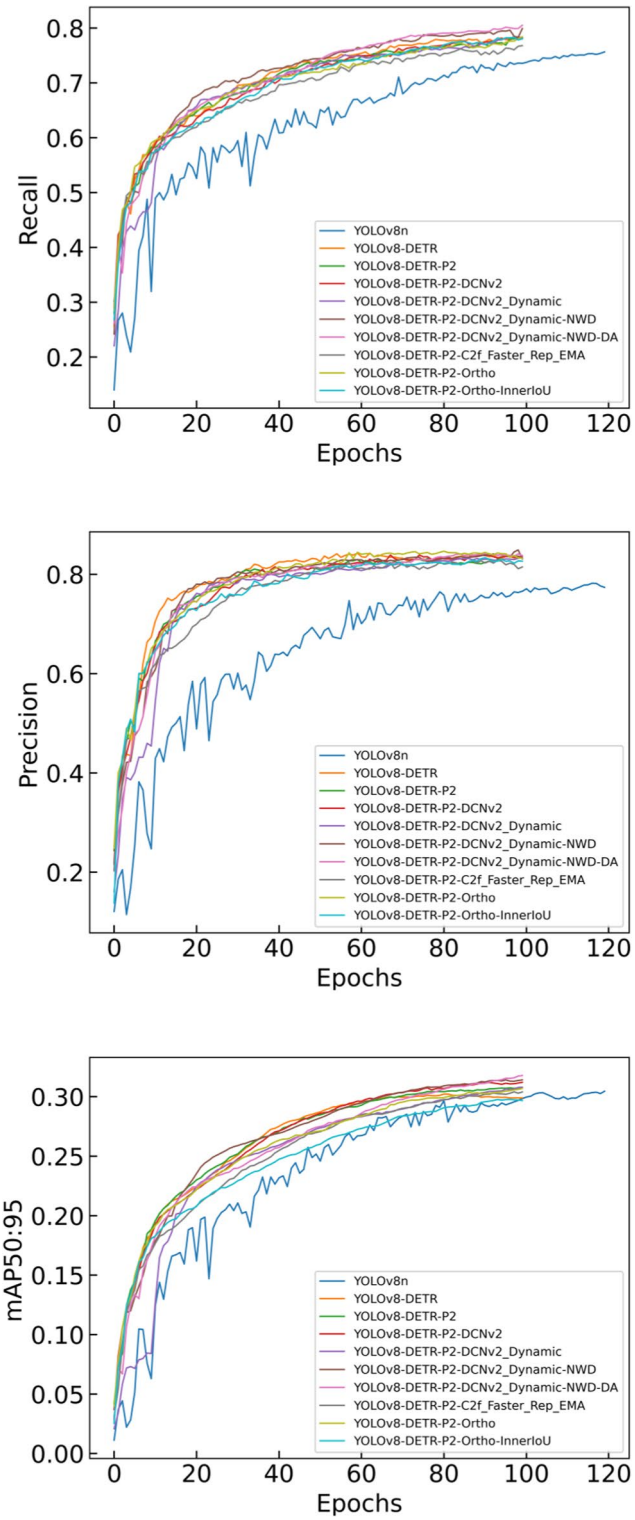


Fig. 10 Results of the different algorithms

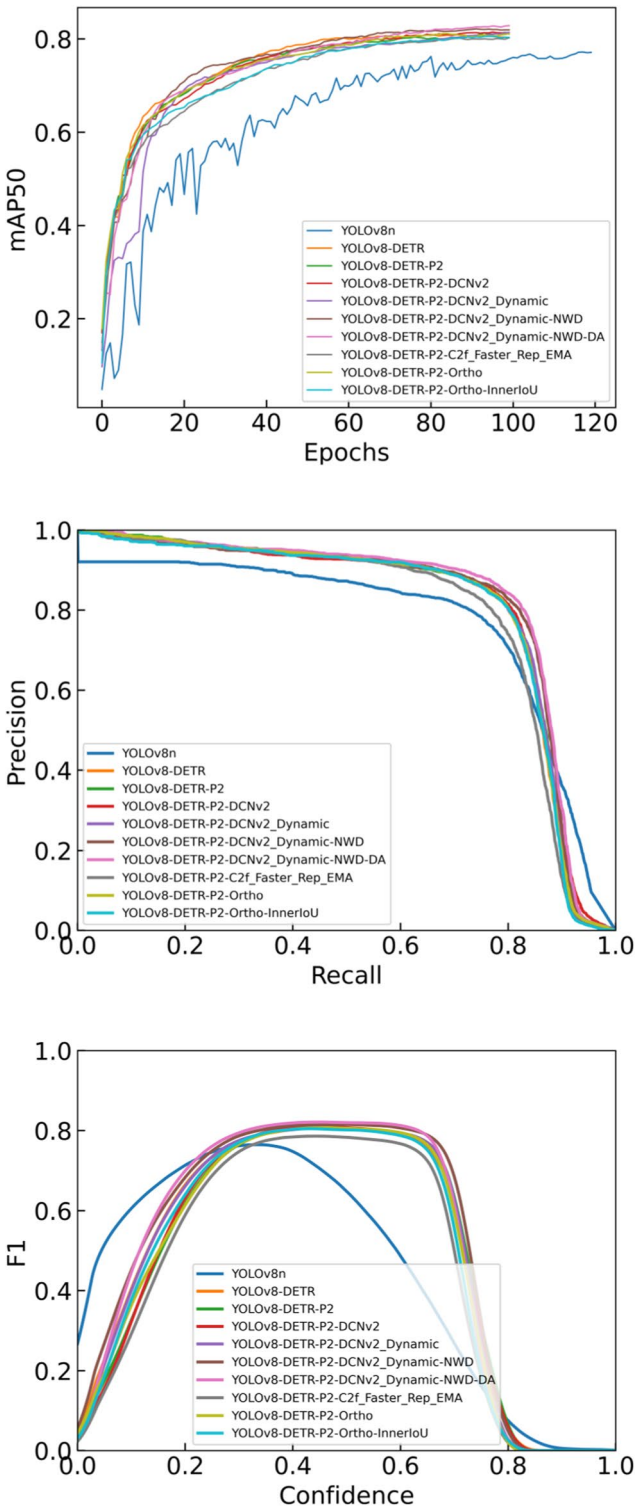


Fig. 10 continued

In the course of our analysis, the effectiveness of each module was assessed as follows. Starting with the baseline YOLOv8, the introduction of the decoding detection head of

RT-DETER alone resulted in notable improvements. Precision, recall, and mAP50 values increased by 6%, 2.1%, and 3.8%, respectively. Upon incorporating the P2 test head, precision, recall, mAP50, and mAP50:95 values saw increments of 0.4%, 0.6%, 0.4%, and 0.5%, respectively. Reintroducing the DCNv2 module increased the mAP50:95 value by 0.6%. The introduction of the Dynamic module led to a 0.1% increase in mAP50. The inclusion of the NWD loss function resulted in improvements of 0.4%, 1.2%, 0.6%, and 0.6% in precision, recall, mAP50, and mAP50:95, respectively. Finally, employing the DA dataset enhancement strategy increased recall, mAP50, and mAP50:95 by 1.2%, 0.9%, and 0.3%, respectively. In-depth analysis revealed that the decoding detection head of RT-DETER, with its ability to extract features effectively, led to substantial improvements. The Dynamic module further optimized the DCNv2 module, resulting in significant enhancements in mAP values and better performance on the distribution of test data, making it more applicable for practical scenarios. The NWD loss function demonstrated clear effectiveness, particularly in measuring similarity between bounding boxes, proving to be more suitable for small object detection. The subsequent use of the DA dataset augmentation strategy demonstrated a strong fit to the linear relationship between the training set and the test set, resulting in a significant increase in test set indices.

Simultaneously, control experiments were conducted using YOLOv8-DETR-P2-Ortho, YOLOv8-DETR-P2-C2f_Faster_Rep_EMA, and YOLOv8-DETR-P2-Ortho-InnerIoU. In these experiments, Ortho represented a classical algorithm, while InnerIoU served as an alternative GIOU parameter. YOLOv8-DETR-P2-C2f_Faster_Rep_EMA involved replacing the original C2f module with Faster_Rep_EMA, where Faster_Rep is based on convolution.

A comprehensive analysis across Precision rate, recall, mAP50, and mAP50:95 metrics revealed that the control experiments' algorithms exhibited lower measurements compared to the DCNv2-Dynamic module. This further substantiates the effectiveness of the improved algorithm in outperforming classical and alternative configurations.

Effect of the NWD weights: As shown in Eq. 9, a represents the weight of the NWD. In order to verify the influence of NWD weight on the network and the weight of NWD loss function a, five parameters of 0.1,0.3,0.5,0.7 and 0.9 were selected to record mAP and other parameters to form the radar mAP shown below in Fig. 11. It can be seen that at the fit of 0.5, the value of mAP50 is the largest, so the weight is 0.5.

The mechanism of attention methods on DCNv2 Dynamic: Just as shown in Fig. 3 Architecture diagram of M_CA_Attention, we proposed an attention mechanism, named M_CA_Attention, which can help DCNv2 to achieve more effective feature extraction. Of course, we can replace the attention mechanism. In order to verify the effectiveness of M_CA_Attention, a comparative experiment was done, and the results are as follows.

To prove the effectiveness of attention proposed in this paper, several methods are compared in the following forms 'CoordAtt [35]'; 'SegNext-Attention [37]'; 'deformable-LKA [38]'; 'BAMBlock [39]'; 'CPCA [34]'; 'LSKA [40]'; 'LSKBlock [41]'; 'SE [42]'; 'SpatialGroupEnhance [43]'; 'M-CA-Attention'.

As can be seen from Fig. 12, our proposed scheme, which yields the largest max value and outperformed the SE and CPCA attention mechanism, can extract more reasonable feature data and make target detection more accurate.

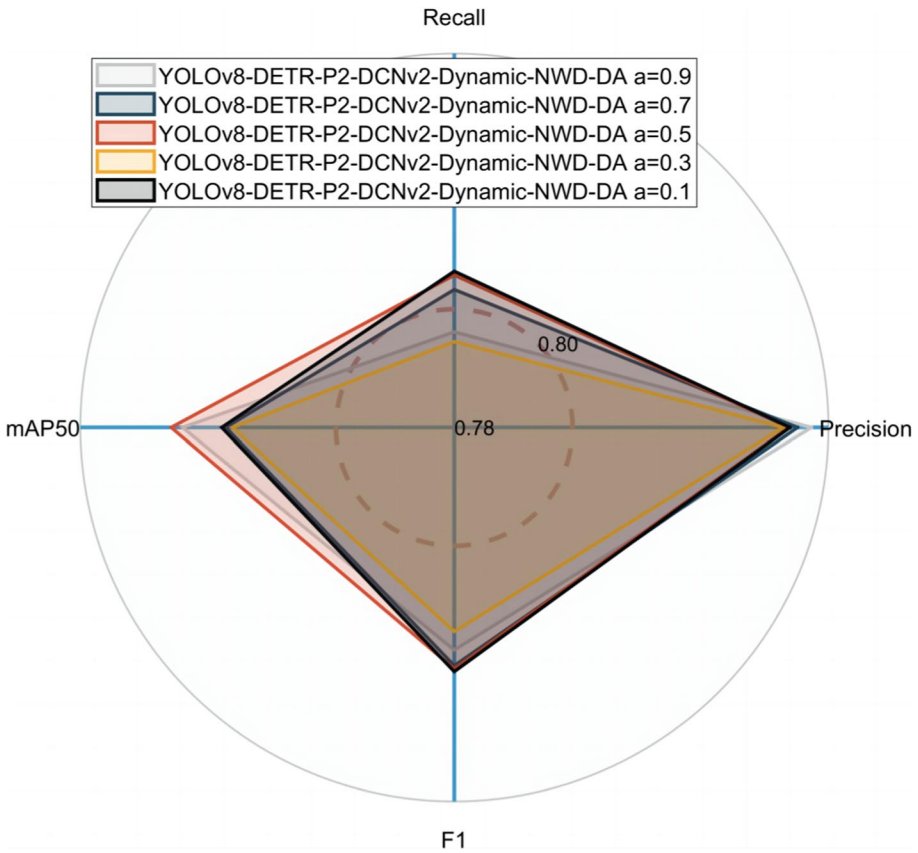


Fig. 11 Effect of NWD weights at different values

Comparison with different versions of the YOLO and RT-DETR algorithms

In this article, we conducted a comprehensive comparison of the proposed algorithm metrics with several other algorithms, including YOLOv3, YOLOv5, YOLOv6, RTDETR versions, and relevant improvements. The comparison encompasses various model sizes denoted by labels such as YOLOv3-tiny, YOLOv8-C2f-SCConv, and different versions represented by letters like l, x, m, s, etc. Additionally, RTDETR-lsknet involves embedding the lsknet into the network, where r50 and r101 denote the backbone networks of ResNet.

Throughout this comparative analysis, we recorded the precision, recall, mAP50, and F1 for each algorithm and variant. The summarized results are presented below in Table 5 for reference and further examination.

As evident from the Table 5 and Fig. 13, our proposed algorithm outperforms others in terms of mAP and F1 indices. The mAP is 3.5% higher than RTDETR-L, and the F1 index is 1.3% higher than RTDETR-L. Although the precision is only 0.4% higher compared to the highest RTDETR-L, and the recall is 0.1% lower than YOLOv8x, the substantial advantage in the mAP value suggests that our algorithm excels in capturing more detailed features related to button touch.

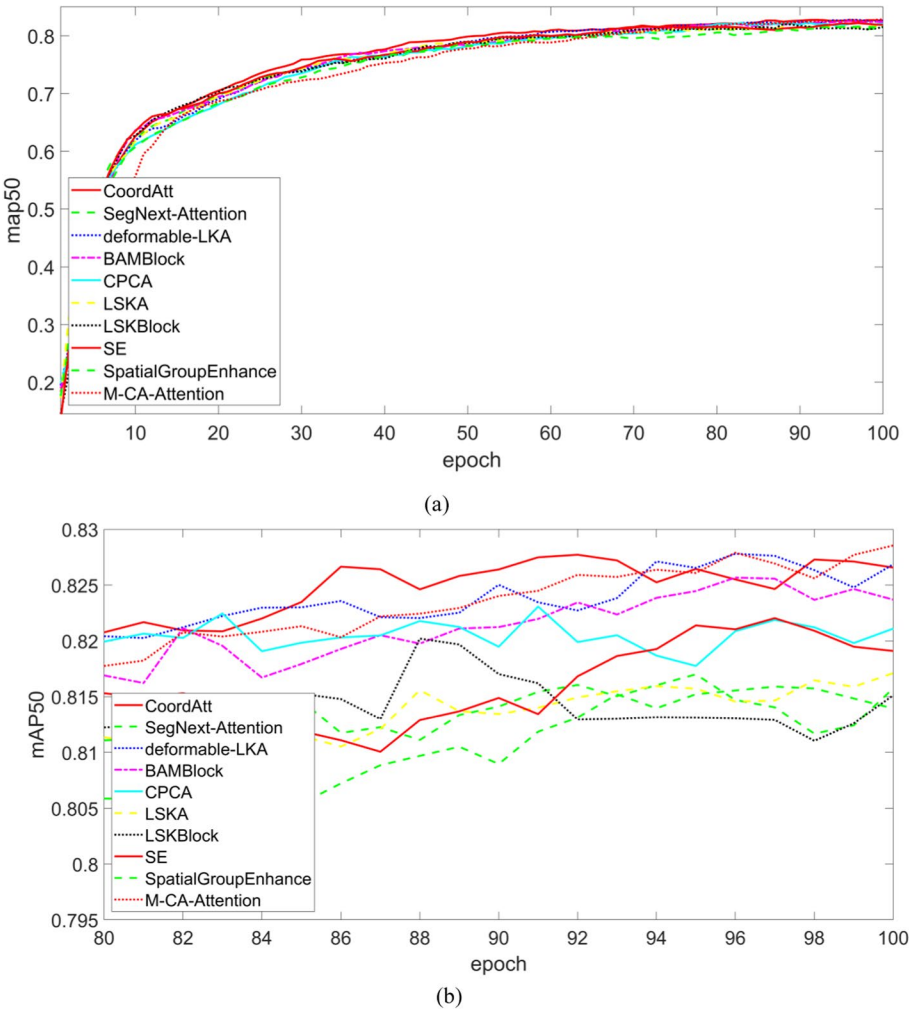


Fig. 12 mAP indicators of different attention mechanisms **a** epoch: 1–100 **b** epoch: 80–100

Experimental discussion of datasets

We designed a training process for a Generative Adversarial Network (GAN). The generator aims to produce images that resemble real data, while the discriminator attempts to distinguish between real and generated images. During training, the generator attempts to deceive the discriminator, and the discriminator tries to identify the generated images. This competitive relationship drives the generator to continuously improve its generation capabilities, resulting in increasingly realistic images.

The generator and discriminator are two key components of a Deep Convolutional Generative Adversarial Network (DCGAN).

Generator: It receives random noise as input and outputs a realistic image. In DCGAN, the generator typically includes multiple transposed convolutional layers, each layer containing convolution, batch normalization, and ReLU activation functions. These layers work together to generate increasingly realistic images.

Discriminator: The discriminator's task is to distinguish between real and generated images. In DCGAN, the discriminator typically includes multiple convolutional layers,

Table 5 Results of YOLO and RT-DETR algorithms

MODEL	Precision	Recall	mAP50	F1
RTDETR-X	0.82787	0.78441	0.78174	0.805554254
RTDETR-l	0.83356	0.78441	0.79339	0.808238471
RTDETR-lsknet	0.81924	0.77578	0.78526	0.796917916
RTDETR-r101	0.8241	0.79447	0.79958	0.809013792
RTDETR-r50	0.82904	0.79483	0.79991	0.81157465
RTDETR-repvit	0.82292	0.78567	0.77756	0.803863702
YOLOv3	0.80975	0.80141	0.79897	0.805558414
YOLOv3-spp	0.8132	0.80273	0.80352	0.807931081
YOLOv3-tiny	0.81693	0.78458	0.80137	0.80042827
YOLOv5	0.77547	0.74596	0.766	0.760428809
YOLOv5-C3-ODConv	0.72668	0.68163	0.71857	0.703434455
YOLOv5-LSKNet	0.71271	0.712	0.71529	0.712354823
YOLOv5-BiFPN	0.76515	0.72997	0.75271	0.747146109
YOLOv5-P6	0.76225	0.7341	0.75705	0.747910215
YOLOv5l	0.81837	0.79914	0.81495	0.808640691
YOLOv5m	0.82168	0.80466	0.81827	0.813080941
YOLOv5x	0.81002	0.79439	0.80389	0.802128867
YOLOv6	0.77426	0.73536	0.75779	0.754308811
YOLOv6l	0.79677	0.76949	0.79259	0.782892428
YOLOv8-C2f-SCConv	0.76971	0.73021	0.75535	0.749439889
YOLOv8-DETR-DWR	0.82687	0.77021	0.79716	0.79753493
YOLOv8-DETR-fasternet	0.74018	0.64804	0.67917	0.691052207
YOLOv8l	0.81939	0.80075	0.80728	0.809962772
YOLOv8m	0.82057	0.80165	0.81451	0.810999668
YOLOv8s	0.81333	0.7977	0.81456	0.80543918
YOLOv8x	0.81044	0.80668	0.80484	0.808555629
Faster-rcnn	–	–	0.661	–
ssd	–	–	0.598	–
ours	0.83793	0.80525	0.82855	0.821265026

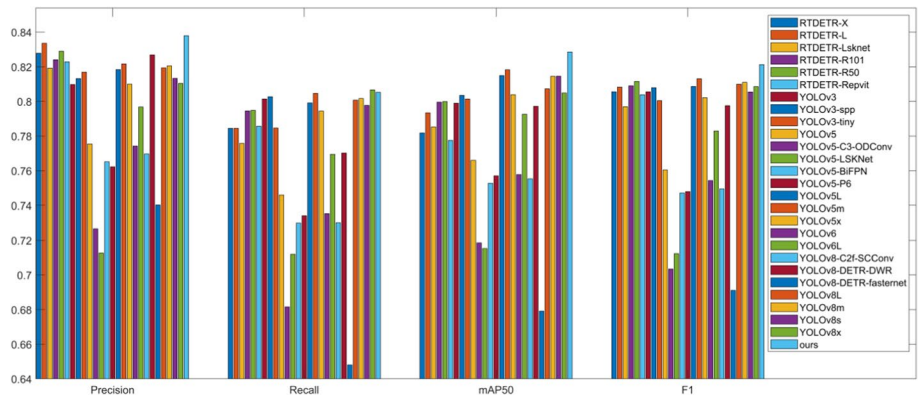


Fig. 13 The index comparison diagram of multiple algorithms

each layer containing convolution, batch normalization, and Leaky ReLU activation functions. These layers work together to accurately determine the authenticity of the image.

Peak Signal-to-Noise Ratio (PSNR): PSNR is an objective indicator for evaluating image reconstruction quality, commonly used to measure the difference between compressed and original images. It is calculated based on the mean squared error (MSE) of pixel differences. A higher PSNR value indicates better image quality.

Structural Similarity Index (SSIM): SSIM is an index for measuring the similarity between two images, considering changes in brightness, contrast, and structure. The SSIM value ranges from -1 to 1, with 1 indicating identical images.

Fréchet Inception Distance (FID): FID is a commonly used method for evaluating the quality of generated images. It assesses image quality by calculating the distance between generated and real images in the feature space of the Inception network. A lower FID value indicates higher quality in the generated images.

We employed PSNR, SSIM, and FID as quality assessment metrics. The method involved selecting original and generated images after each training iteration, calculating the metrics between them, and saving the values. These metrics were then plotted as curves. From the Fig. 14, it can be observed that the best performance occurs around iteration 175. We selected the weights from iteration 175 as the baseline condition for generating data in the later stages.

Post DCGAN, the diversity of samples is enhanced, leading to improvements in the mAP and other indicators. To validate the effectiveness of the data enhancement algorithm proposed in this paper, multiple dataset enhancement methods are employed for verification based on YOLOv8n and the advancements introduced in this method. The comparison aims to evaluate performance differences under various dataset strategies. The primary strategies for dataset enhancement include:

A. Utilize the background map as the training sample data, overlay the added target map through image processing onto the background map. For each selected background map, superimpose 5 target maps to augment the training sample to 6977.

B. Employ the background map as the training sample data and incorporate the expanded target map through image processing, overlaying it onto the background map. Stack 3 target maps on each selected background map to augment the training sample to 9000.

C. Take the background map as the training sample data and integrate the augmented target map through image processing onto the background map. Stack 3 target maps on each selected background map to expand the training sample to 7381.

D. Employ the enhancement scheme proposed in this paper, using the test set sample as the background map. Overlay the added target map through image processing

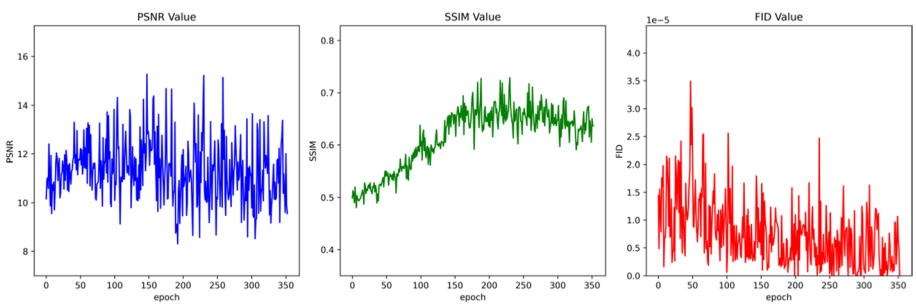


Fig. 14 Quality assessment metrics curve

onto the background map and stack 3 target maps on each selected background map to increase the training sample to 6381. The results obtained are shown below in Table 6.

In addition, the original methods without number enhancement are: 'YOLOv8-CBAM', 'YOLOv8-CoordAtt', 'YOLOv8-EA', 'YOLOv8-act', 'YOLOv8-BiFPN', 'YOLOv8-efficientViT', 'YOLOv8-EMA', 'YOLOv8-P2', 'YOLOv8-P6', 'YOLOv8-repvit', 'YOLOv8-SimAM', 'YOLOv8n'. The experimental methods include:

A: 'YOLOv8-CBAM', 'YOLOv8-CoordAtt', 'YOLOv8-SimAM', 'YOLOv8-BiFPN', 'YOLOv8-efficient', 'YOLOv8-P2', 'YOLOv8-P6', 'YOLOv8n'.

B: 'YOLOv8-CBAM', 'YOLOv8-P6', 'YOLOv8-SE_Attention', 'YOLOv8-SimAM', 'YOLOv8-act', 'YOLOv8-efficientvit', 'YOLOv8-P2', 'YOLOv8'.

C: 'YOLOv8-BiFPN', 'YOLOv8act', 'YOLOv8efficient-vit', 'YOLOv8n', 'YOLOv8l'.

D: 'YOLOv8-CBAM', 'YOLOv8-CoordAtt', 'YOLOv8-EMA', 'YOLOv8-SimAM', 'YOLOv8-BiFPN', 'YOLOv8-efficientViT', 'YOLOv8-P2', 'YOLOv8-P6', 'YOLOv8-repvit', 'YOLOv8n', 'YOLOv8-act'.

Due to the length, instead of listing the index value of each algorithm, the data is presented according to the enhancement method, and the following results are obtained in Fig. 15 and Table 6.

It is evident in Fig. 15 that the proposed data enhancement method in this paper effectively leads to improvements in performance indicators. In contrast, other schemes fail to achieve significant enhancements, and, in some cases, the indicators show a worsening effect. Therefore, the data enhancement scheme put forth in this paper is deemed effective.

Abnormal touch-button detection result

In the provided images in Fig. 16, let's analyze the performance of different algorithms. In image A, when detecting the occluded emergency button, both YOLOv5l and YOLOv8n fail to identify it. However, both RTDETR-L and our model are able to effectively detect the small occluded target.

Moving on to image B, RTDETR-L incorrectly frames the human body, and YOLOv8n fails to detect the touch button behavior. On the other hand, both YOLOv5l and our model successfully detect the touch button behavior. It is worth noting that in our model, the detected probability is 0.62, which is higher than the 0.34 achieved by YOLOv5l.

Table 6 Results under different data-augmentation experiments

MODEL		Precision	Recall	mAP50	mAP50-95
No using DA	YOLOv8n	0.77381	0.75663	0.7711	0.30456
A	YOLOv8n	0.7783	0.76141	0.76552	0.30228
A	YOLOv8n-CBAM	0.78428	0.7629	0.77612	0.3081
B	YOLOv8n	0.74537	0.72925	0.73457	0.28646
B	YOLOv8n-CBAM	0.76819	0.73468	0.74562	0.29373
C	YOLOv8n	0.77024	0.72395	0.76407	0.30266
C	YOLOv8n-BiFPN	0.77256	0.72889	0.76114	0.30201
D	YOLOv8n	0.78271	0.76015	0.78312	0.31604
D	YOLOv8n-CBAM	0.79392	0.76276	0.79203	0.32304

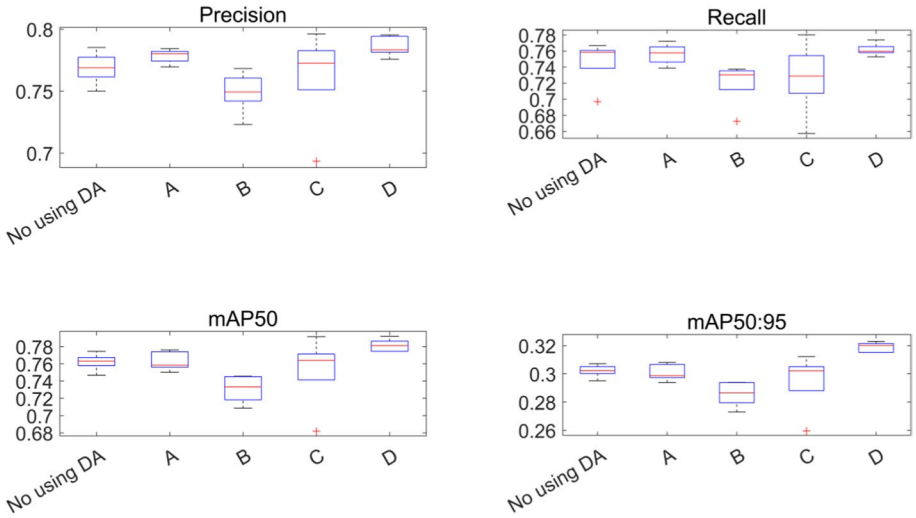


Fig. 15 Different indicator data for different methods

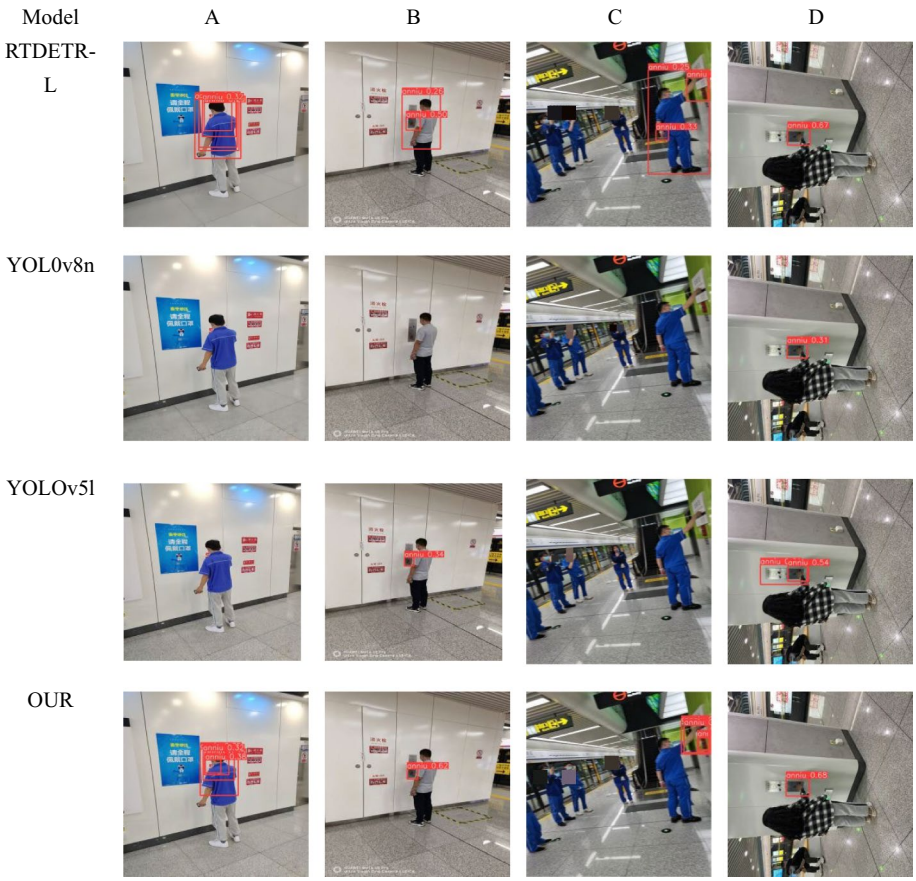


Fig. 16 Detection results for the different algorithms

In the case of image C, YOLOv5l, YOLOv8n, RTDETR-L, and our model can all effectively identify small targets. However, RTDETR-L also incorrectly detects the human body, resulting in an apparent error.

Finally, in image D, YOLOv5l misses the detection, but the remaining three models can detect abnormalities. Among them, our model outputs the highest probabilities.

From these performance comparisons, it is evident that our algorithmic model improves target confidence, enhances the detection rate of small targets and occluded cases, eliminates error detections of the human body, and demonstrates high robustness.

Visual comparison of heatmap

Figure 17 shows a comparison of the thermal heatmap visualization results of the YOLOv8 et al. with and without incorporating the proposed method. By adding our method to the model, we can improve the attention toward button touch behavior while reducing attention to the background. This enhances the credibility and accuracy of the detection.

Comparing figures (a) and (b), it is clear that after using DCNv2-Dynamic, the attention is more focused on the button position, increasing attention towards the correct area. This demonstrates the effectiveness of the module after its incorporation. Furthermore, from the other visual diagram, it can be observed that the model's attention position is accurate with minimal errors. Specifically, the YOLOv8n has low probability of correctly detecting touch button areas as its attention is not focused on them, while the detection effect of YOLOv5l is poor. This proves that in the presence of occlusion, the algorithm proposed in this paper exhibits superior performance and robustness.

Non-parametric validation experiments

Wilcoxon Rank-Sum Test: In the results of the Wilcoxon rank-sum test, the null hypothesis can be rejected if the p-value is less than the significance level (e.g., 0.05), indicating a significant difference between the two samples. To scientifically validate the effectiveness and robustness of the proposed method, this paper conducted 10 random selections of test samples to verify the outcomes of YOLOv8-det-tr-P2-DCNv2-Dynamic, YOLOv8, ours, and RTDETR-L, using mAP50 as the evaluation index. The results of each experiment, along with the corresponding test statistics and p-values, were recorded and analyzed to ascertain the satisfaction of the test results. The outcomes of each experiment are depicted in Fig. 18.

As evident from the figure, our algorithm exhibits exceptional robustness, surpassing other methods in terms of index data. Box plots are employed to compare

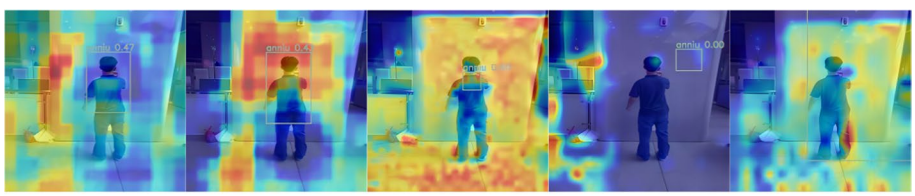


Fig. 17 **a** YOLOv8-DETR-P2 **b** our **c** RTDETR-L **d** YOLOv8n **e** YOLOv5l

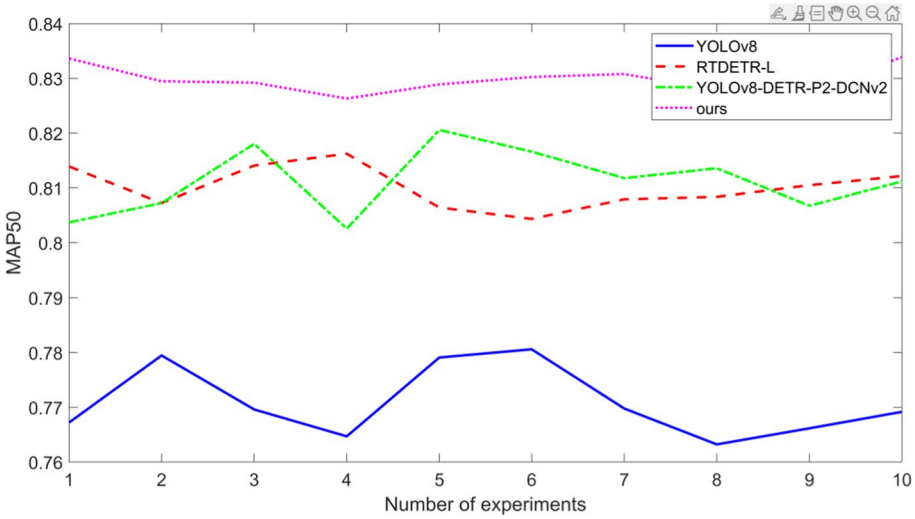


Fig. 18 Results diagram of indicator data for different algorithms

the median and distribution of two samples, while QQ plots are utilized to compare quantiles. The box plot and QQ plot of the data provide comprehensive insights into the data characteristics, as illustrated in Fig. 19.

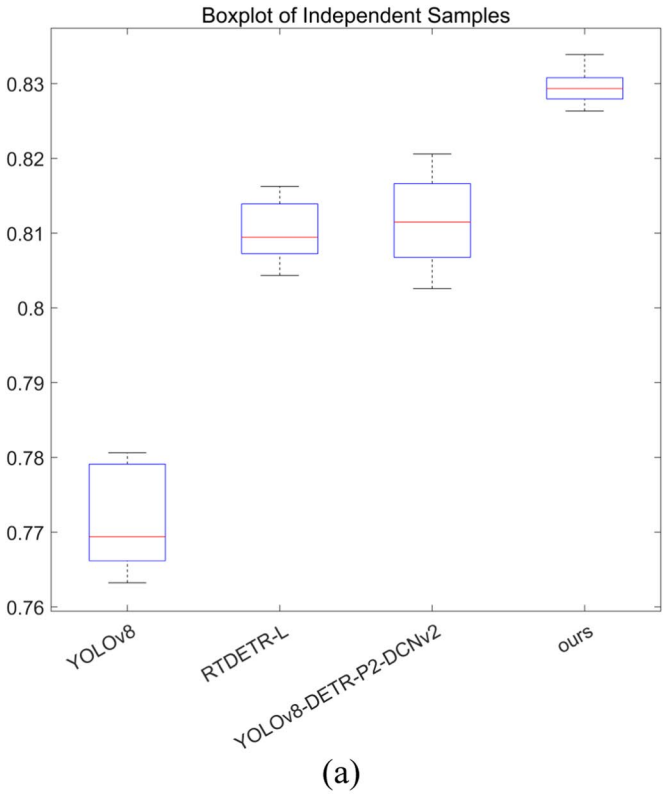
The CDF (Cumulative Distribution Function) describes the cumulative probability distribution, indicating the probability that a random variable is less than or equal to a specific value. On the x-axis, we have the values of the random variables, while the y-axis represents the cumulative probability of being less than or equal to the corresponding x-value. Figure 20 displays the scatter plot and CDF, revealing that our algorithm's data is relatively concentrated with a discernible trend, demonstrating a pronounced advantage in the mAP 0.5 index.

In the experiment, the significance level was set to 0.05. The data from our algorithm and three other algorithms were analyzed. As presented in Table 7, the Wilcoxon Rank-Sum Test Statistic data was 155, yielding a P-value of 0.0002, which is less than the significance level. The results indicate the rejection of the null hypothesis, confirming significant differences in performance between each pair of algorithms. Therefore, it can be concluded that noteworthy distinctions exist in the performance of the compared algorithms.

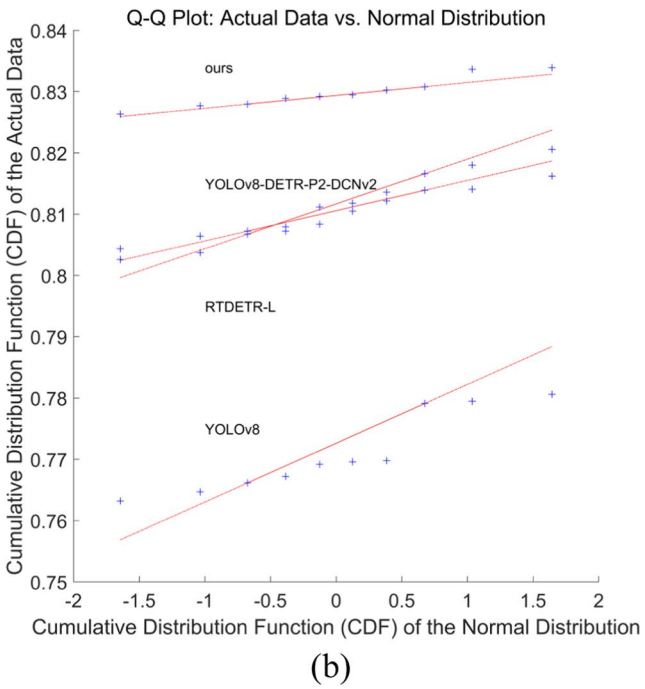
In contrast to the SOTA

We selected state-of-the-art (SOTA) algorithms in the field of target detection for comparative experiments (Table 8), including the YOLOV6V3 model [44], YOLO-NAS model [7], and DINO model (v1 version) [45]. The YOLOv6-S achieved 484 FPS on the COCO dataset, attaining a 45.0% AP. The YOLO-NAS L version exhibited a 52.22 mAP value, showcasing superior performance. The DINO model demonstrated excellent performance with a 51.3 AP under the standard setting of ResNet-50.

Leveraging these SOTA algorithms, we conducted comparative experiments using the button dataset as raw data without data augmentation. We adjusted the network

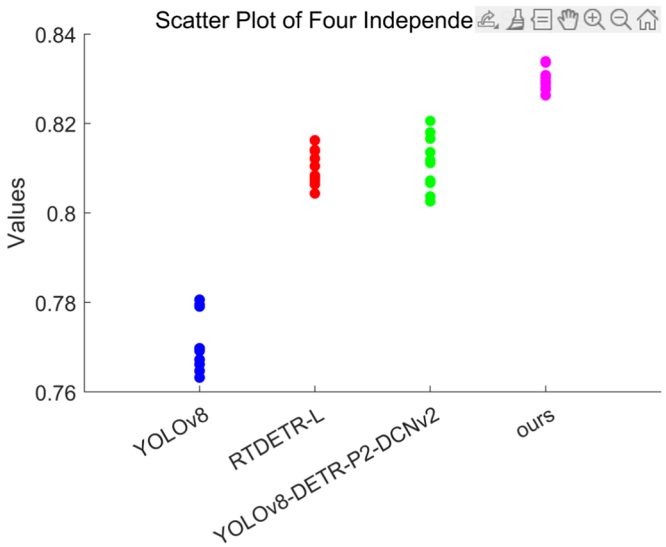


(a)

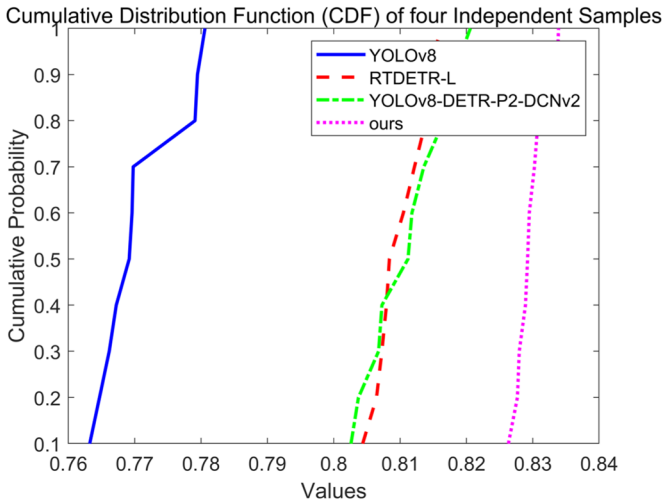


(b)

Fig. 19 Experimental data visualization **a** boxplot **b** QQ plot **c** Scatter plot, **d** CDF



(c)



(d)

Fig. 19 continued

Table 7 Box plots and QQ plots

	Ours-YOLOV8	Ours-RTDETR-L	Ours-YOLOv8-DETR-P2-DCNv2	Results
Wilcoxon Rank-Sum Test Statistic	155	155	155	
P-value	0.0002	0.0002	0.0002	< 0.05

model configuration for optimization and calculated the target detection indices as the comparison metrics.It can be seen that our algorithm has obvious advantages, and it has obvious advantages in button touch detection.

Table 8 Results comparing our model with SOTA

MODEL		Precision	Recall	mAP50	mAP50-95
YOLOv6-v3	V6l6, EPOCH = 150	–	–	0.706	0.274
YOLO-NAS	EPOCH = 100	–	–	0.746	–
DINO	DINO_5scale, EPOCH = 80	–	–	0.746	0.256
Ours	–	0.838	0.806	0.829	0.317

Other performance comparisons and model advantages are discussed

(1) As evident from the above analysis, our model exhibits a substantial improvement compared to YOLOV8n, with notable enhancements in mAP, recall, precision, and other indicators. In comparison to RT-DETR, our model outperforms RTDETR-L with a 3.5% higher mAP and a 1.3% higher F1 index, showcasing excellent performance.

The selection of parameters for comparison includes training time, the number of model parameters, and GFLOPs. Our model demonstrates a reduction in training time compared to the RT-DETR model, with the training time reduced from the original 13 h to 9 h. This improvement in training efficiency is a noteworthy advantage of our model.

The reduction in the number of parameters is substantial when comparing with RT-DETR-L, which has 667 layers, 29,428,460 parameters, and 105.5 GFLOPs. In contrast, our model has 275 layers, 6,220,477 parameters, and 26.3 GFLOPs. It is evident that there is a significant reduction in the number of layers, parameters, and GFLOPs, while achieving improved performance.

(2) Our proposed model achieves excellent performance on the button touch dataset, and the analysis for this success includes the following factors: (1) The button touch dataset consists of many small targets, recognizing touches that involve sensing the part where the hand and button make contact. However, numerous images in the dataset exhibit occlusion. The method employed in this paper, DCNv2-Dynamic, along with the M_CA attention mechanism, grants the network the ability to extract deformable features. The application of attention mechanisms enhances feature perception, enabling the network to capture more semantic information, resulting in a significant performance improvement compared to YOLOV8; (2) The button touch dataset in this paper mainly contains single-object scenarios within single images. In RT-DETR, there is a significant issue of repetitive labeling in the vicinity of button touches. However, the P2 layer in this paper refines contextual semantic extraction more accurately, and the adopted NWD loss function strengthens the capability to extract small target regions. Consequently, our algorithm mitigates the problem of repetitive labeling observed in RT-DETR, leading to significant improvements in performance metrics; (3) In our dataset, there is a notable discrepancy between the training and testing sets, resulting in a clear weakness of misaligned features. The introduction of data augmentation aligns the features between training and testing samples, contributing to improved metric performance.

Validation in the other datasets

(1) Effectiveness Experiments of the Model with Different Test Sets

To validate the algorithm's effectiveness, multiple datasets were chosen for experimental comparison, and the results are presented below in Table 9. The football dataset can be obtained via the reference https://universe.roboflow.com/bronkscottema/football-players-zm06l, while other datasets are available for download at https://universe.roboflow.com.

We took three data sets for experiments, in which our method is better than YOLOv8, where the Traffic Camera Object Detection (car) data set is 8 percentage points higher than YOLOv8. Although it is less than RTDETR-r101, it has an obvious advantage over YOLOv8n.

(2) Effectiveness Experiment of Module and Attention Mechanism

To assess the proposed model and attention mechanism's effectiveness, separate comparison experiments were conducted. The experiments aimed to demonstrate the efficacy of the DCNv2-Dynamic module and the M_CA attention mechanism. Various datasets were used for these experiments, with mAP50 as the evaluation index. Table 10 results indicate that, on the NEU-DET dataset, adopting DCNv2-Dynamic led to an almost 6 percentage points increase in mAP data. For the coco128_person dataset, our module increased by 4 percentage points, and the attention mechanism increased by 1 percentage point. However, in the BCCD data, RBC and WBC performed well, but the overall performance did not match YOLOV8n. After analyzing the reasons, it was found that the Platelets class had an impact on performance, particularly due to size considerations, affecting our algorithm's overall effectiveness.

Ablation experiments for DCNv2-Dynamic and M_CA_Attention. We did the experiments with 3 datasets, containing coco128_person (Table 11), Traffic Camera Object Detection (Table 12), and electrictransafety (Table 13). datasets are available for download at https://universe.roboflow.com. We also record the index data of YOLOv8n + DETR + P2, YOLOv8n + DETR + P2 + DCNv2, YOLOv8n + DETR + P2 + DCNv2-Dynamic, and conclude that the module and attention mechanism proposed in this paper are effective.

Analysis of the table reveals that after the incorporation of the DCNv2-Dynamic module, there is a marked improvement in precision, recall, and mean average precision (mAP) values, indicating the effectiveness of the DCNv2-Dynamic module. Furthermore, the enhancement in precision and mAP when DCNv2 is employed alone validates the efficacy of the DCNv2 module.

Table 9 The mAP 0.5 results for the different datasets

Data	YOLOv8	RTDETR-r101	RTDETR-L	YOLOv8-DETR-P2-DCNv2	Ours
Football	0.953	–	0.984	0.963	0.97
Traffic Camera Object Detection(car)	0.692	0.791	0.795	–	0.778
NEU-DET	0.737 [46]	0.822	0.742	0.756	0.775

Table 10 Results for the different datasets

Dataset	Model	mAP50						mAP
		Crazing	Inclusion	Patches	pitted_surface	rolled-in_ scale	Scratches	
NEU-DET	Faster-rcnn [46]	0.35	0.685	0.927	0.84	0.647	0.898	0.725
	Ssd [46]	0.302	0.641	0.866	0.741	0.575	0.71	0.639
	YOLOv8n [46]	0.333	0.827	0.941	0.825	0.566	0.930	0.737
	YOLOv8n + DCNv2-Dynamic module	0.428	0.723	0.977	0.995	0.706	0.889	0.786
	YOLOv8n + M_CA attention	0.554	0.748	0.937	0.995	0.674	0.869	0.796
BCCD	Model	mAP50			mAP			
		Platelets',	'RBC',	WBC'				
	Faster-rcnn [46]	0.976	0.849	0.868	0.897			
	Ssd [46]	0.974	0.846	0.869	0.896			
	YOLOv8n47	0.976	0.883	0.942	0.934			
	YOLOv8n + DCNv2-Dynamic module	0.908	0.896	0.987	0.931			
	YOLOv8n + M_CA attention	0.912	0.889	0.971	0.924			
coco128_ person	model	Precision	Recall	mAP50	mAP50-95			
	YOLOv8n	0.698	0.379	0.54	0.317			
	YOLOv8n + M_CA attention	0.534	0.592	0.551	0.318			
	YOLOv8n + DCNv2-Dynamic module	0.587	0.668	0.582	0.349			

Table 11 Ablation experiments of coco128_person

Model	Precision	Recall	mAP50	mAP50-95
YOLOv8n + DETR + P2	0.466	0.407	0.37	0.18
YOLOv8n + DETR + P2 + DCNv2	0.472	0.387	0.357	0.182
YOLOv8n + DETR + P2 + DCNv2-Dynamic	0.626	0.42	0.394	0.199

After adding P2-DCNv2-Dynamic, the precision rate increased by 0.4 percentage points.

As can be seen from the table, when the method is added, the index improvement effect is obvious.

Field application experiment

Installation of the system has been completed in a subway system in China, featuring an online control center platform installed in the rail control center. The monitoring videos from stations are transmitted to the central control via a network channel, where the platform is configured with algorithms to diagnose potential anomalies. Additionally, mobile terminals are set up at stations to receive alerts regarding button touches.

As depicted in the Fig. 20, the system platform is initially configured with AI rules, including model names, camera locations, server IP addresses, and task durations. This configuration equips the system with the capability to diagnose button touches.

Table 12 Traffic camera object detection

Model	Precision	Recall	mAP50	mAP50-95
YOLO-DETR-P2	0.819	0.735	0.788	0.403
YOLO-DETR-P2-DCNv2	0.821	0.74	0.786	0.401
YOLO-DETR-P2-DCNv2-Dynamic	0.825	0.723	0.778	0.402

Table 13 Electrictransafety

Model	Precision	Recall	mAP50	mAP50-95
YOLO-DETR-P2	0.7	0.715	0.753	0.555
YOLO-DETR-P2-DCNv2	0.721	0.747	0.79	0.572
YOLO-DETR-P2-DCNv2-Dynamic	0.725	0.759	0.784	0.578

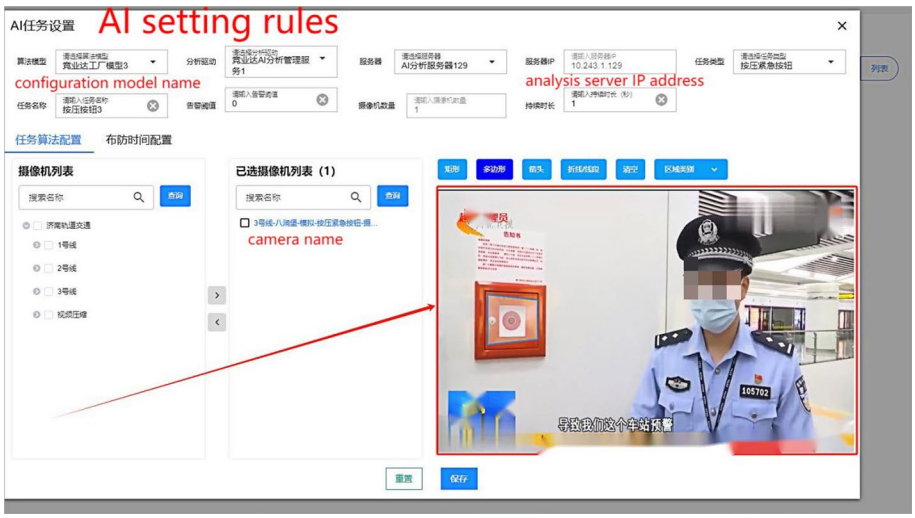
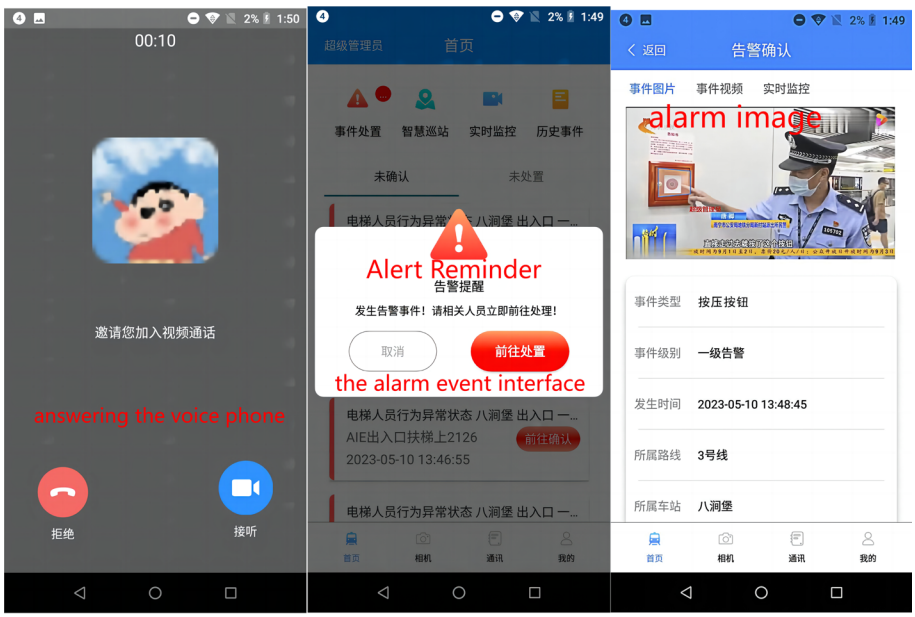


Fig. 20 AI setting rules interface (including configuration model name, camera name, analysis server IP address, etc.)

As depicted in the Figs. 21, 22, when an abnormality is detected, the platform triggers an alarm notification, displaying details such as event location, occurrence time, rail line number, BIM map, and event type. Simultaneously, it establishes a language connection with the mobile terminal, providing spoken alerts about the button touch event. Upon answering the call, the system presents essential information about the event, including alarm content and recommended handling procedures.

As depicted in the Fig. 23. The system further tracks the event process and automatically generates a comprehensive report. This entire process achieves the detection of button touches, automating the confirmation of personnel identity, on-site situations, and the documentation of handling processes. The system further tracks the event process and automatically generates a comprehensive report. This entire process achieves the detection of button touches, automating the confirmation of personnel identity, on-site situations, and the documentation of handling processes.

Fig. 21 The alarm prompt of the platform can display the description of the alarm event, so as to facilitate the operation personnel to grasp what abnormal events occur in where



(a) (b) (c)

Fig. 22 The interface of the mobile terminal shows **a** answering the voice phone **b** the alarm event interface **c** the alarm image

Conclusion and outlook

This paper introduces an abnormal touch button detection model built upon an enhanced YOLOv8 architecture. Our model exhibits the capability to autonomously identify instances of abnormal button touches, consequently alleviating the workload associated with abnormal detection. This timely recognition of button touches allows for a prompt understanding of the actual situation, facilitating the formulation of pertinent operational plans and the implementation of emergency procedures as needed.

(a)

(b)

Fig. 23 Generation of event reports **a** System platform generation template **b** Report generation results

To address the challenges posed by the detection of small targets, particularly in the context of touch buttons, we implemented a crucial improvement: the YOLOv8-DETR-P2-DCNv2-Dynamic-NWD-DA method. This approach aims to enhance the model's capability to detect small targets effectively. The key modifications include replacing the last layer of YOLOv8 with RTDETRDecoder and introducing the P2 small target detection layer. Additionally, to address the diversity and complex background of abnormal images, we introduced the DCNv2-Dynamic algorithm along with the NWD loss function for multiscale feature extraction. During dataset analysis, we identified a disparity in feature distribution between the training and test sets. To mitigate this, we proposed dataset augmentation methods using image processing and the GAN algorithm to enhance detection accuracy. The improved model demonstrated significant enhancements. In comparison to YOLOv8n, the precision, recall

rate, and mAP50 values increased by 6.5%, 5%, and 5.8%, respectively, showcasing a notable improvement in overall performance.

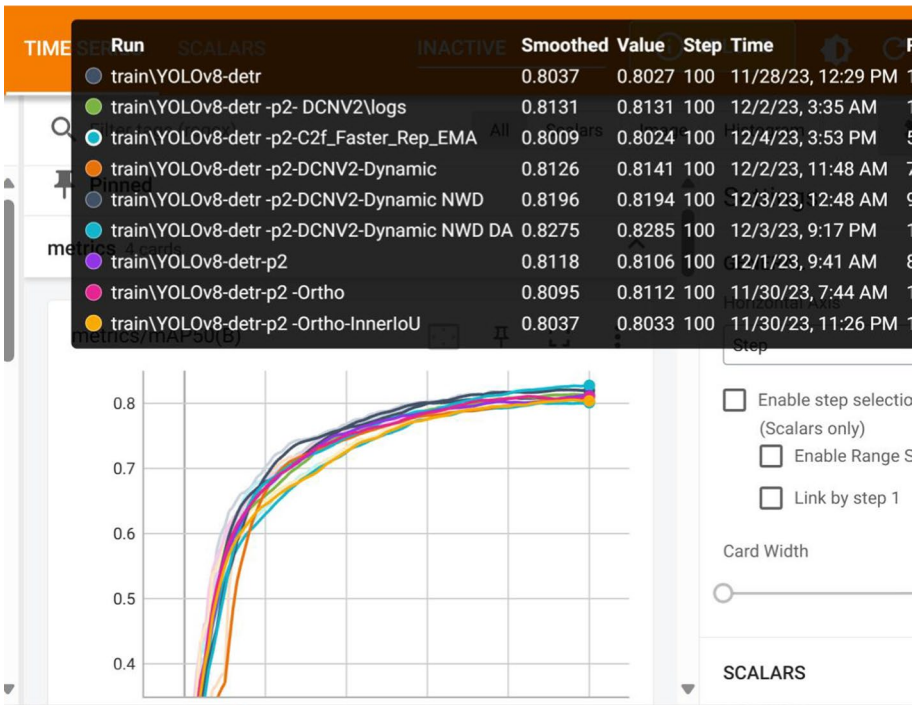
In the existing literature, there is a lack of publicly available datasets specifically designed for subway touch buttons, posing a challenge for experiments in anomaly detection. To address this limitation, our study utilizes a dataset obtained from field pendulum photos, containing a substantial number of touch button images. Following proper permissions and deductions, we intend to make this dataset publicly accessible, aiming to support future research in related fields.

While the upgraded model exhibits potential for practical detection scenarios, it is acknowledged that the model's size and detection speed are currently constrained. Future efforts will concentrate on the implementation of lighter models, such as pruning and distillation models, to enhance the model's detection speed while preserving its accuracy.

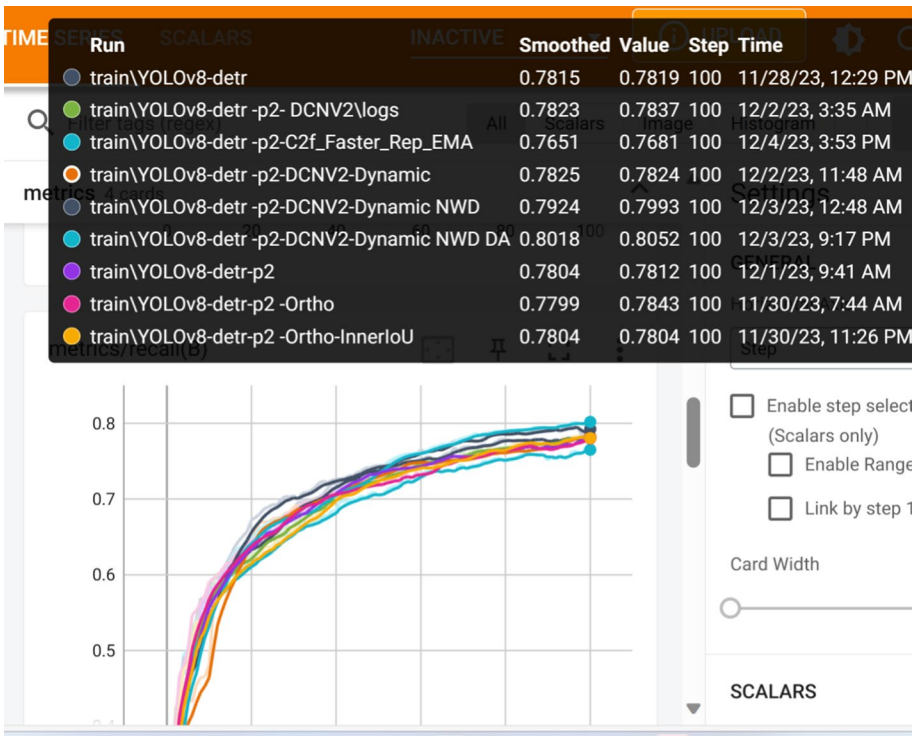
In future research, we plan to investigate video collection methods for touch buttons, identify the causes of touches, and explore solutions for automatic pedestrian tracking. Additionally, we aim to establish connections between the existing ticketing system and the public security system. When the system detects an illegal touch button, it will facilitate seamless information transfer to the public security system with a simple one-click process.

Appendices

Appendix 1: Ablation experiments and contrast experiments in tensorboard (4.3)



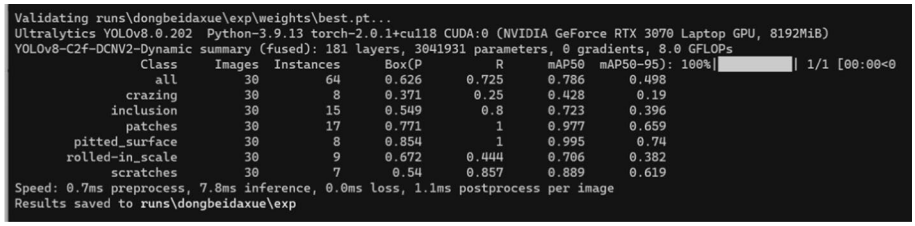
Results under different data-augmentation experiments



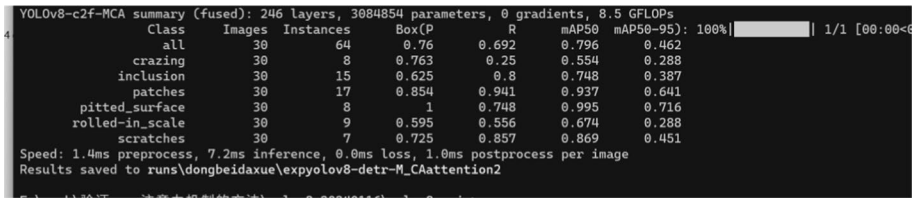
Appendix 2: 4.5 Experimental discussion of datasets

Method	MODEL	Precision	Recall	mAP50	mAP50-95	F1
B	YOLOv8-CBAM	0.78428	0.76294	0.77612	0.3081	0.773462834
	YOLOv8-CoordAtt	0.78134	0.77219	0.77329	0.30616	0.776738054
	YOLOv8-SimAM	0.77587	0.74881	0.75796	0.29868	0.76209987
	YOLOv8-BiFPN	0.76953	0.73877	0.75013	0.29379	0.753836343
	YOLOv8-efficienet	0.7802	0.75782	0.75838	0.29844	0.768847172
	YOLOv8-P2	0.77395	0.75548	0.76752	0.30504	0.764603474
	YOLOv8-P6	0.76234	0.75871	0.7554	0.29967	0.760520668
	YOLO8n	0.7783	0.76141	0.76552	0.30228	0.769762362
A	YOLOv8-CBAM	0.78428	0.76294	0.77612	0.3081	0.773462834
	YOLOv8-CoordAtt	0.78134	0.77219	0.77329	0.30616	0.776738054
	YOLOv8-SimAM	0.77587	0.74881	0.75796	0.29868	0.76209987
	YOLOv8-BiFPN	0.76953	0.73877	0.75013	0.29379	0.753836343
	YOLOv8-efficienet	0.7802	0.75782	0.75838	0.29844	0.768847172
	YOLOv8-P2	0.77395	0.75548	0.76752	0.30504	0.764603474
	YOLOv8-P6	0.76234	0.75871	0.7554	0.29967	0.760520668
	YOLO8n	0.7783	0.76141	0.76552	0.30228	0.769762362
C	YOLOv8-BiFPN	0.77256	0.72889	0.76114	0.30201	0.750089924
	YOLOv8act	0.69342	0.65749	0.68194	0.25952	0.674977187
	YOLOv8efficient-vit'	0.77821	0.74576	0.76466	0.29745	0.76163952
	YOLOv8	0.77024	0.72395	0.76407	0.30266	0.746377968
	YOLOv8l'	0.79627	0.78013	0.79151	0.31225	0.788117375
D	YOLOv8-CBAM	0.79392	0.76276	0.79203	0.32304	0.778028136
	YOLOv8-CoordAtt	0.77573	0.77381	0.78122	0.32016	0.77476881
	YOLOv8-EMA	0.78337	0.75979	0.77453	0.31521	0.771399845
	YOLOv8-SimAM	0.78337	0.75979	0.77453	0.31521	0.771399845
	YOLOv8-BiFPN	0.79535	0.75278	0.78452	0.32104	0.773479712
	YOLOv8-efficientViT'	0.79094	0.76333	0.79215	0.31968	0.776889768
	YOLOv8-P2	0.77841	0.75943	0.77664	0.31686	0.768802875
	YOLOv8-P6	0.78081	0.75422	0.78401	0.31914	0.767284702
	YOLOv8-repvit	0.79508	0.76788	0.78291	0.31367	0.781243321
	YOLOv8	0.78271	0.76015	0.78312	0.31604	0.771265062
	YOLOv8-act	0.7571	0.70571	0.74731	0.2971	0.730502309
NO using DA	YOLOv8-CBAM	0.76885	0.75836	0.76496	0.30439	0.763568973
	YOLOv8-CoordAtt	0.76539	0.75871	0.76166	0.30195	0.762035361
	YOLOv8-EA	0.78526	0.7668	0.77448	0.30714	0.77592022
	YOLOv8-act	0.74991	0.69691	0.74681	0.29507	0.72243925
	YOLOv8-BiFPN	0.77484	0.75244	0.76325	0.30211	0.763475734
	YOLOv8-efficientViT'	0.78062	0.76033	0.76355	0.30195	0.770341419
	YOLOv8-ema	0.78777	0.75359	0.77316	0.30625	0.770301025
	YOLOv8-P2	0.77596	0.75225	0.76364	0.30381	0.763921071
	YOLOv8-P6	0.7831	0.75907	0.7678	0.30194	0.770897783
	YOLOv8-repvit	0.77554	0.7553	0.76326	0.29737	0.765286198
	YOLOv8-simam	0.79299	0.75746	0.77097	0.30207	0.774817899
	YOLOv8n	0.77381	0.75663	0.7711	0.30456	0.765123573

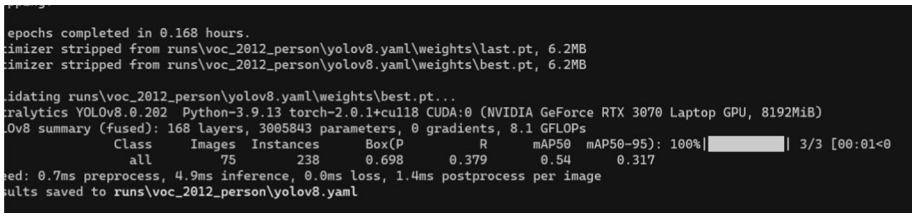
Appendix 3 The resulting data of the network



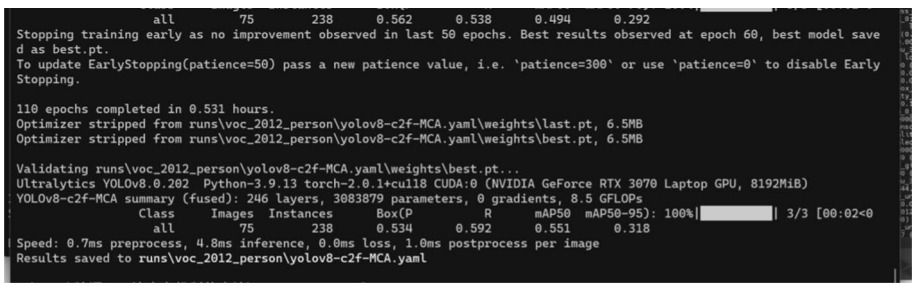
YOLOv8-DCNv2-dynamic on DEU-DET



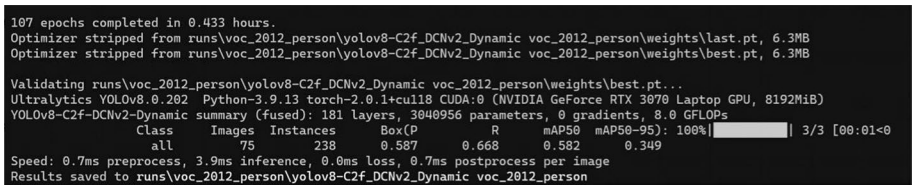
YOLOv8-M_CA on DEU-DET



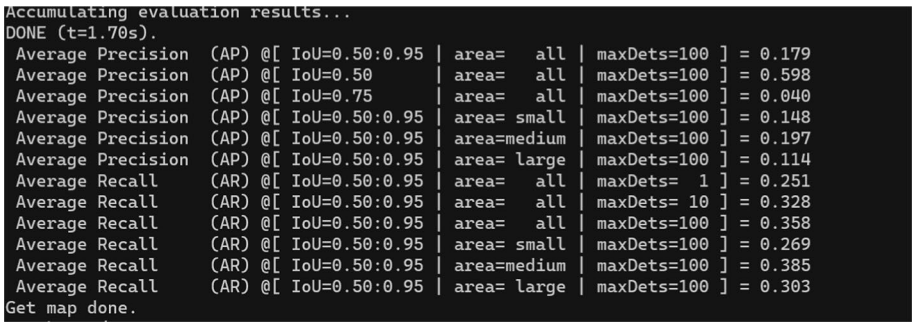
YOLOv8 on coco128_person



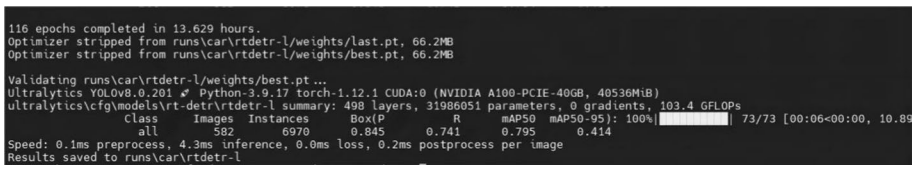
YOLOv8-MCA on coco128_person



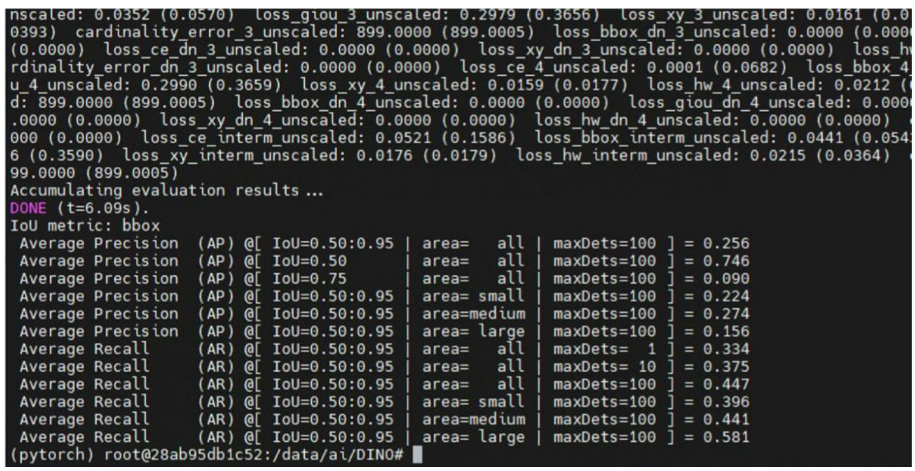
YOLOv8_DCNV2dynamic on coco128_person



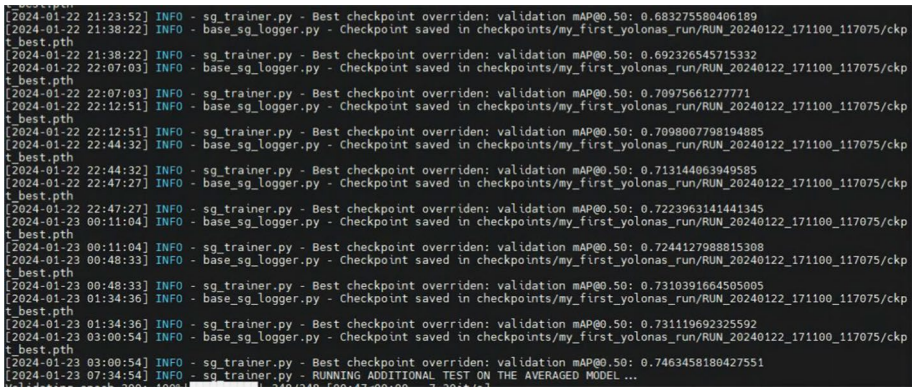
SSD on our dataset(button)



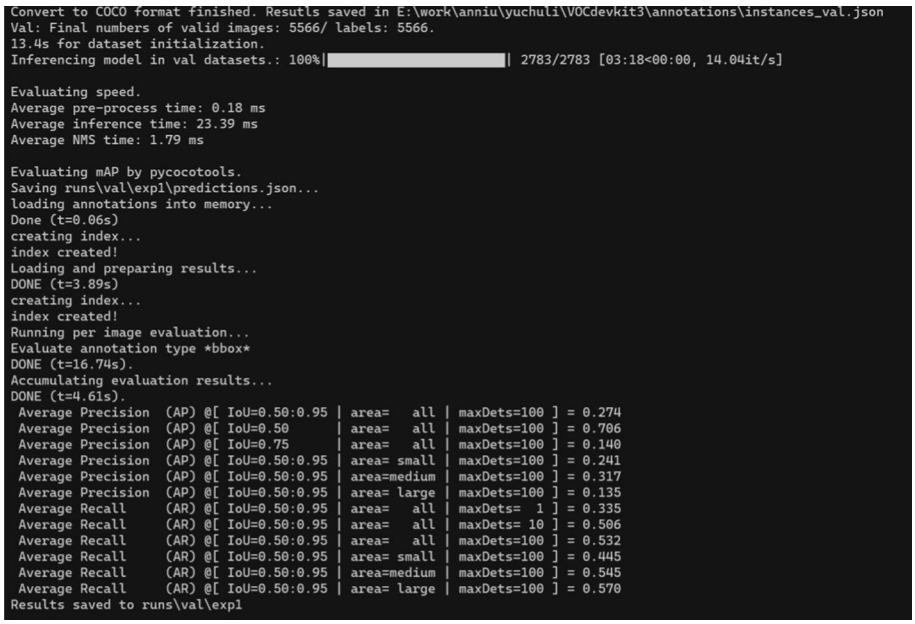
RT-DETR-l on Traffic Camera Object Detection



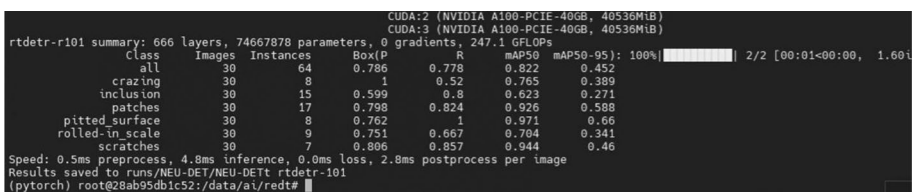
DINO on our dataset(button)



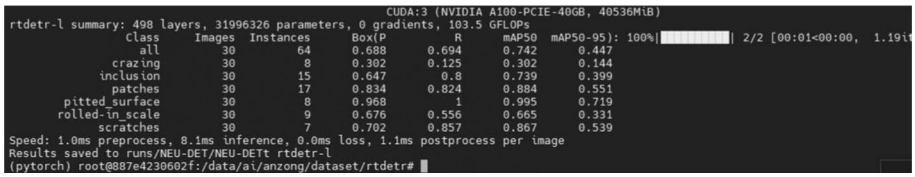
YOLONAS on our dataset(button)



YOLOv6v3(6 l)



RTDETR-101 on DEU-DET



RTDETR-l on DEU-DET

Appendix 4 The profiles for the individual networks

1. DINO

Options = {'dn_scalar': 100, 'embed_init_tgt': True, 'dn_label_coef': 1.0, 'dn_bbox_coef': 1.0, 'use_ema': False, 'dn_box_noise_scale': 1.0}, dataset_file = 'coco', coco_path = './coco_path', coco_panoptic_path = None, remove_difficult = False, fix_size = False, output_dir = 'logs/DINO/R50_custom_finetune', note = '', device = 'cuda', seed = 42, resume = '', pretrain_model_path = None, finetune_ignore = None, start_epoch = 0, eval = False, num_workers = 10, test = False, debug = False, find_unused_params = False, save_results = False, save_log = False, world_size = 1, dist_url = 'env://', rank = 0, local_rank = 0, amp = False, distributed = False, data_aug_scales = [480, 512, 544, 576, 608, 640, 672, 704, 736, 768, 800], data_aug_max_size = 1333, data_aug_scales2_resize = [400, 500, 600], data_aug_scales2_crop = [384, 600], data_aug_scale_overlap = None, num_classes = 2, lr = 0.0001, param_dict_type = 'default', lr_backbone = 1e-05, lr_backbone_names = ['backbone.0'], lr_linear_proj_names = ['reference_points', 'sampling_offsets'], lr_linear_proj_mult = 0.1, dDETR_lr_param = False, batch_size = 1, weight_decay = 0.0001, epochs = 100, lr_drop = 11, save_checkpoint_interval = 1, clip_max_norm = 0.1, onecyclelr = False, multi_step_lr = False, lr_drop_list = [33, 45], modelname = 'dino', frozen_weights = None, backbone = 'resnet50', use_checkpoint = False, dilation = False, position_embedding = 'sine', pe_temperatureH = 20, pe_temperatureW = 20, return_interm_indices = [0, 1, 2, 3], backbone_freeze_keywords = None, enc_layers = 6, dec_layers = 6, unic_layers = 0, pre_norm = False, dim_feedforward = 2048, hidden_dim = 256, dropout = 0.0, nheads = 8, num_queries = 900, query_dim = 4, num_patterns = 0, pDETR3_bbox_embed_diff_each_layer = False, pDETR3_refHW = -1, random_refpoints_xy = False, fix_refpoints_hw = -1, dabDETR_YOLO_like_anchor_update = False, dabDETR_deformable_encoder = False, dabDETR_deformable_decoder = False, use_deformable_box_attn = False, box_attn_type = 'roi_align', dec_layer_number = None, num_feature_levels = 5, enc_n_points = 4, dec_n_points = 4, decoder_layer_noise = False, dln_xy_noise = 0.2, dln_hw_noise = 0.2, add_chann.

2. YOLOv8, ours

Atience = 50, batch = 16, imgsz = 512, save = True, save_period = -1, cache = False, device = 0, workers = 0, project = runs/voc_2012_person, name = YOLOv8-C2f-MCA.yaml, exist_ok = False, pretrained = True, optimizer = SGD, verbose = True, seed = 0, deterministic = True, single_cls = False, rect = False, cos_lr = False, close_mosaic = 10, resume = False, amp = True, fraction = 1.0, profile = False, freeze = None, overlap_mask = True, mask_ratio = 4, dropout = 0.0, val = True,

split = val, save_json = False, save_hybrid = False, conf = None, iou = 0.7, max_det = 300, half = False, dnn = False, plots = True, source = None, show = False, save_txt = False, save_conf = False, save_crop = False, show_labels = True, show_conf = True, vid_stride = 1, stream_buffer = False, line_width = None, visualize = False, augment = False, agnostic_nms = False, classes = None, retina_masks = False, boxes = True, format = torchscript, keras = False, optimize = False, int8 = False, dynamic = False, simplify = False, opset = None, workspace = 4, nms = False, lr0 = 0.01, lrf = 0.01, momentum = 0.937, weight_decay = 0.0005, warmup_epochs = 3.0, warmup_momentum = 0.8, warmup_bias_lr = 0.1, box = 7.5, cls = 0.5, dfl = 1.5, pose = 12.0, kobj = 1.0, label_smoothing = 0.0, nbs = 64, hsv_h = 0.015, hsv_s = 0.7, hsv_v = 0.4, degrees = 0.0, translate = 0.1, scale = 0.5, shear = 0.0, perspective = 0.0, flipud = 0.0, fliplr = 0.5, mosaic = 1.0, mixup = 0.0, copy_paste = 0.0,

3. RT-DETR

Epochs = 100, patience = 50, batch = 16, imgsz = 640, save = True, save_period = -1, cache = False, device = 0, workers = 0, project = runs/voc_2012_person, name = YOLOv8-DETR-l, exist_ok = False, pretrained = True, optimizer = SGD, verbose = True, seed = 0, deterministic = True, single_cls = False, rect = False, cos_lr = False, close_mosaic = 10, resume = False, amp = True, fraction = 1.0, profile = False, freeze = None, overlap_mask = True, mask_ratio = 4, dropout = 0.0, val = True, split = val, save_json = False, save_hybrid = False, conf = None, iou = 0.7, max_det = 300, half = False, dnn = False, plots = True, source = None, show = False, save_txt = False, save_conf = False, save_crop = False, show_labels = True, show_conf = True, vid_stride = 1, stream_buffer = False, line_width = None, visualize = False, augment = False, agnostic_nms = False, classes = None, retina_masks = False, boxes = True, format = torchscript, keras = False, optimize = False, int8 = False, dynamic = False, simplify = False, opset = None, workspace = 4, nms = False, lr0 = 0.01, lrf = 0.01, momentum = 0.937, weight_decay = 0.0005, warmup_epochs = 3.0, warmup_momentum = 0.8, warmup_bias_lr = 0.1, box = 7.5, cls = 0.5, dfl = 1.5, pose = 12.0, kobj = 1.0, label_smoothing = 0.0, nbs = 64, hsv_h = 0.015, hsv_s = 0.7, hsv_v = 0.4, degrees = 0.0, translate = 0.1, scale = 0.5, shear = 0.0, perspective = 0.0, flipud = 0.0, fliplr = 0.5, mosaic = 1.0, mixup = 0.0, copy_paste = 0.0,

Author contributions

An Junfeng, Liu Jiqiang: put forward the research ideas and designed the research scheme; An Junfeng: Conduct the experiment; An Junfeng, Li Gang, Liu Jiqiang, Lu Mengmeng: Collect, clean and analyze the data; An Junfeng, Lu Mengmeng, Wang Chongqing, Ligang, Liu Jiqiang: Draft the paper; An Junfeng, Lu Mengmeng: The final version of the paper is revised.

Funding

Basic application of ABP-IOT technology in subway vent valve (No. zr2020qe268), The Mount Tai Industry Leading Talent Project Special Fund Support (tscx202312018). China National Railway Group Co., Ltd. Science and Technology Research and Development Plan Project (Project No. N2023W012).

Data availability

Data will be made available on request.

Declarations

Competing interests
All of the authors declare that there is no conflict of interest relationship.

Received: 23 February 2024 Accepted: 31 May 2024
Published online: 29 August 2024

References

1. Li R. Research on the key technology of intelligent identification of passenger flow in high-speed railway stations based on deep learning. China Acad Railway Sci. 2023. https://doi.org/10.27369/d.cnki.gtdky.2022.000001.
2. Espinosa R, Ponce H, Gutiérrez S, Martínez-Villaseñor L, Brieva J, Moya-Albor E. A vision-based approach for fall detection using multiple cameras and convolutional neural networks: a case study using the UP-Fall detection dataset. Compu Biol Med. 2019;115: 103520. https://doi.org/10.1016/j.compbiomed.2019.103520.
3. Khraief C, Benzarti F, Amiri H. Elderly fall detection based on multi-stream deep convolutional networks. Multimed Tools Appl. 2020;79:19537–60. https://doi.org/10.1007/s11042-020-08812-x.
4. Cao Z, Qin Y, Xie Z, Liu Q, Zhang E, Wu Z, Yu Z. An effective railway intrusion detection method using dynamic intrusion region and lightweight neural network. Measurement. 2022;191: 110564. https://doi.org/10.1016/j.measurement.2021.110564.
5. Ling H-B, Huang D, Cui J, Wang C-D. HOLT-Net: detecting smokers via human–object interaction with lite transformer network. Eng Appl Artif Intell. 2023;126: 106919. https://doi.org/10.1016/j.engappai.2023.106919.
6. Lv W, Xu A, Zhao Y, Wang G, Wei J, Cui C, Du Y, Dang Q, Liu Y. DETRs Beat YOLOs on real-time object detection. 2023; arXiv:2304.08069.
7. Research Team. YOLO-NAS by Deci Achieves State-of-the-Art Performance on Object Detection Using Neural Architecture Search. 2023. https://deci.ai/blog/YOLO-nas-object-detection-foundation-model/. Accessed 12 May 2023.
8. Li E, Wang Q, Zhang J, Zhang W, Mo H, Wu Y. Fish detection under occlusion using modified you only look once v8 integrating real-time detection transformer features. Appl Sci. 2023;13:12645. https://doi.org/10.3390/app132312645.
9. Wang Z, Yuan G, Zhou H, Ma Y, Ma Y. Foreign-object detection in high-voltage transmission line based on improved YOLOv8m. Appl Sci. 2023;13:12775. https://doi.org/10.3390/app132312775.
10. Yaping D, Yingjiang L. Overview of the YOLO algorithm and its target detection studies in autonomous driving scenarios. Comput Appl. http://kns.cnki.net/kcms/detail/51.1307.TP.20230904.1321.006.html. Accessed 17 Sep 2023
11. Yang G, Wang J, Nie Z, Yang H, Yu S. A lightweight YOLOv8 tomato detection algorithm combining feature enhancement and attention. Agronomy. 1824;2023:13. https://doi.org/10.3390/agronomy13071824.
12. Yang W, Wu J, Zhang J, Gao K, Du R, Wu Z, Firkat E, Li D. Deformable convolution and coordinate attention for fast cattle detection. Comput Electron Agric. 2023;211: 108006. https://doi.org/10.1016/j.compag.2023.108006.
13. Song L, Tao S, Fang Ke J. Improving the road injury detection algorithm for YOLOv8. Comput Eng Appl. 2023.
14. Yuan H, Tao L. Detection and identification of fish in electronic monitoring data of commercial fishing vessels based on improved YOLOv8. J Dalian Ocean Univ. 2023;38:533–42.
15. Geng H, Liu Z, Jiang J, Fan Z, et al. Embedded road crack detection algorithm based on improved YOLOv8. Comput Appl. 2024;44(5):1613.
16. Zhou Y, Yan Y, Chen H, et al. Pv cell defect detection based on improved YOLOv8. Progress in laser and optoelectronics. http://kns.cnki.net/kcms/detail/31.1690.tn.20230821.1446.128.html. Accessed 17 Sep 2023.
17. Xiong E, Zhang R, Liu Y et al. Ghost-YOLOv8 detection algorithm. Comput Eng Appl. http://kns.cnki.net/kcms/detail/11.2127.TP.20230811.1059.002.html. Accessed 17 Sep 2023.
18. Chen Y, Zhang S, Ran X et al. Based on improved YOLOv8. Telecommun Technol. https://doi.org/10.20079/j.issn.1001-893x.230515007.
19. Liang G, Xingzhu, Chenxing X et al. A dense pedestrian detection algorithm for improving YOLOv8. J Atlas. http://kns.cnki.net/kcms/detail/10.1034.T.20230731.0913.002.html. Accessed 17 Sep 2023.
20. Wang G, Chen Y, An P, Hong H, Hu J, Huang T. UAV-YOLOv8: a small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios. Sensors. 2023;23:7190. https://doi.org/10.3390/s23167190.
21. Chen S, Li Y, Zhang Y, Yang Y, Zhang X. Soft X-ray image recognition and classification of maize seed cracks based on image enhancement and optimized YOLOv8 model. Comput Electron Agric. 2024;216: 108475. https://doi.org/10.1016/j.compag.2023.108475.
22. Wang Z, Liu Y, Duan S, Pan H. An efficient detection of non-standard miner behavior using improved YOLOv8. Comput Electr Eng. 2023;112(109021):10045–7906. https://doi.org/10.1016/j.compeleceng.2023.109021.
23. Yang S, Wang W, Gao S, Deng Z. Strawberry ripeness detection based on YOLOv8 algorithm fused with LW-Swin Transformer. Comput Electron Agric. 2023;215: 108360. https://doi.org/10.1016/j.compag.2023.108360.
24. Zhao H, Jin J, Liu Y, Guo Y, Shen S. FSDF: a high-performance fire detection framework. Expert Syst Appl. 2024;238: 121665. https://doi.org/10.1016/j.eswa.2023.121665.
25. Cao S, Wang Y, Wang Z, Guan R, Ding L, Lv Y. Empowering photovoltaic power generation with edge computing: a recognition and location approach for hot spot. Electron Lett. 2023;59: e13056. https://doi.org/10.1049/ell2.13056.
26. Fan J, Wang M, Li B, Liu M, Shen D. ACD-YOLO: improved YOLOv5-based method for steel surface defects detection. IET Image Process. 2023. https://doi.org/10.1049/ipr2.12983.
27. Ji Y, Di L. Textile defect detection based on multi-proportion spatial attention mechanism and channel memory feature fusion network. IET Image. 2023. https://doi.org/10.1049/ipr2.12957.
28. Xu W, Liu C, Wang G, Zhao Y, Yu J, Muhammad A, Li D. Behavioral response of fish under ammonia nitrogen stress based on machine vision. Eng Appl Artif Intell. 2024;128: 107442. https://doi.org/10.1016/j.engappai.2023.107442.

29. Liu Y, Zheng Y, Shao Z, Wei T, Cui T, Xu R. Defect detection of the surface of wind turbine blades combining attention mechanism. Adv Eng Inf. 2024;59: 102292. https://doi.org/10.1016/j.aei.2023.102292.
30. Liu Y, An S, Ren Y, Zhao J, Zhang C, Cheng J, Liu K, Wei Y. DP-FishNet: dual-path pyramid vision transformer-based underwater fish detection network. Expert Syst Appl. 2024;238: 122018. https://doi.org/10.1016/j.eswa.2023.122018.
31. Bai Y, Yu J, Yang S, Ning J. An improved YOLO algorithm for detecting flowers and fruits on strawberry seedlings. Biosyst Eng. 2024;237:1–12. https://doi.org/10.1016/j.biosystemseng.2023.11.008.
32. Hui Y, Wang J, Li B. STF-YOLO: a small target detection algorithm for UAV remote sensing images based on improved SwinTransformer and class weighted classification decoupling head. Measurement. 2024;224: 113936. https://doi.org/10.1016/j.measurement.2023.113936.
33. Zhu A, Hu H, Lin S, Dai J. Deformable ConvNets v2: more deformable, computer vision and pattern recognition. Better Results. 2018. https://arxiv.org/pdf/1811.11168.pdf
34. Huang H et al. Channel prior convolutional attention for medical image segmentation. 2023. arXiv preprint. arXiv:2306.05196
35. Hou Q et al. Coordinate Attention for Efficient Mobile Network Design. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2021:13708–13717.
36. Wang J, Xu C, Yang W, Yu L. A Normalized Gaussian Wasserstein Distance for Tiny Object Detection. Comput Vision Pattern Recognit. 2022.https://doi.org/10.48550/arXiv.2110.13389
37. Guo M-H, et al. Segnext: rethinking convolutional attention design for semantic segmentation. Adv Neural Inf Process Syst. 2022;35:1140–56.
38. Azad R et al. Beyond self-attention: deformable large kernel attention for medical image segmentation. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2024.
39. Park J et al. Bam: Bottleneck attention module. 2018. arXiv preprint arXiv:1807.06514.
40. Lau KW, Po L-M, AbbasUrRehman Y. Large separable kernel attention: rethinking the large kernel attention design in CNN. Expert Syst Appl. 2024;236: 121352.
41. Li, Y et al. Large Selective Kernel Network for Remote Sensing Object Detection. 2023; arXiv preprint arXiv:2303.09030.
42. Hu J, Shen L, Sun G. Squeeze-and-excitation networks. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018
43. Li X, Hu X, Yang J. Spatial group-wise enhance: Improving semantic feature learning in convolutional networks. 2019. arXiv preprint arXiv:1905.09646.
44. Li C et al. YOLOv6 v3. 0: a full-scale reloading. 2023. arXiv preprint arXiv:2301.05586.
45. Zhang H, Li F, Liu S, et al. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. 2022. arXiv preprint arXiv:2203.03605.
46. Su P, Han H, Liu M, Yang T, Liu S. MOD-YOLO: rethinking the YOLO architecture at the level of feature information and applying it to crack detection. Expert Syst Appl. 2024;237: 121346. https://doi.org/10.1016/j.eswa.2023.121346.

Publisher's Note