

International Journal of Reasoning-based Intelligent Systems

ISSN online: 1755-0564 - ISSN print: 1755-0556

<https://www.inderscience.com/ijris>

Development of a key dimension inspection and error correction system for product design drawings combining computer vision and YOLOv10

Liya Li, Fuguo Wu, Haijing Liu

DOI: [10.1504/IJRIS.2026.10076927](https://doi.org/10.1504/IJRIS.2026.10076927)

Article History:

Received:	07 November 2025
Last revised:	05 January 2026
Accepted:	05 January 2026
Published online:	26 March 2026

Development of a key dimension inspection and error correction system for product design drawings combining computer vision and YOLOv10

Liya Li*

Department of Computer Engineering,
Taiyuan Institute of Technology,
Taiyuan, 030008, China
Email: Lly15935619221@163.com
*Corresponding author

Fuguo Wu

Shanxi Aitplus Technology Co., Ltd.,
Taiyuan, 030006, China
Email: aitplus@163.com

Haijing Liu

Department of Computer Engineering,
Taiyuan Institute of Technology,
Taiyuan, 030008, China
Email: 15340650368@163.com

Abstract: To address the challenges of large detection errors and difficult error correction in detecting key dimensions in paper product design drawings, we developed a novel object detection method based on YOLO v10. We added coordinate and attention mechanisms, along with a variable convolutional module, to the original method to optimise feature fusion. Furthermore, we optimised the loss function to better suit the specific application of product design drawings. Experimental results show that our proposed model has significant advantages over several classic object detection algorithms, achieving an average accuracy of 0.831 on the test set, significantly outperforming other benchmark models. We also conducted ablation studies on our model, revealing that the backbone network has the most significant impact on the overall model performance. Coordinate attention and variable convolution modules significantly boost performance. Our proposed model focuses on detecting and correcting key drawing dimensions, enhancing real-world applicability by advancing traditional models.

Keywords: computer vision; YOLOv10; critical dimension detection; error correction; smart manufacturing.

Reference to this paper should be made as follows: Li, L., Wu, F. and Liu, H. (2026) 'Development of a key dimension inspection and error correction system for product design drawings combining computer vision and YOLOv10', *Int. J. Reasoning-based Intelligent Systems*, Vol. 18, No. 10, pp.46–59.

Biographical notes: Liya Li works at the Department of Computer Engineering, Taiyuan Institute of Technology, China. She received her degrees from Taiyuan University of Science and Technology (2000–2004) and Taiyuan University of Technology (2008–2011). Her main work includes the paper 'Improved K-means weighted adaptive multi-view data clustering algorithm', published in *Computer Simulation*, pp.314–317, 429, August 2021.

Fuguo Wu is affiliated with Shanxi Aitplus Technology Co., Ltd., China. He obtained his education from Taiyuan University of Science and Technology from 2000 to 2004.

Haijing Liu is with the Department of Computer Engineering, Taiyuan Institute of Technology, China. She studied at Shanxi University (2004–2008, 2009–2011) and is currently at UiTM, Malaysia (2021–2026). Her main work includes 'Exploration of the teaching model of 'data structures' course for application-oriented undergraduate colleges', published in *Office Automation*, No. 2, pp.27–29, 2022.

1 Introduction

With the evolution of the global transformation trend of manufacturing, product quality control, cost optimisation and production efficiency improvement have become the primary tasks of factory research. In the process of modern factory manufacturing, product design drawings are the guide for product production. Bearing the core media role, product design drawings contain the geometric features, dimensional tolerances and materials used, as well as key information related to the process (Ali et al., 2023). Traditional methods use manual methods to detect the key dimensions of product drawings. This method has a good effect on low-precision product parts, but it is difficult to meet the production goals of high precision, high efficiency and high reliability (Zhou, 2022). On the one hand, manual inspection not only consumes a lot of time and energy, but is also easily affected by subjective human factors and produces non-subjective errors. On the other hand, a lot of research uses machine learning methods combined with visual algorithms to detect product dimensions. Although these methods have improved the level of automation and optimised the result error to a certain extent, there are still great deviations in the measurement and identification of parts with complex backgrounds, multiple targets, small size and precise features (Kao et al., 2025).

With the rapid development of the manufacturing industry, visual algorithms have undergone revolutionary changes. Computer algorithms acquire image information of target objects through visual sensors, and obtain features through algorithm processing, and perform certain pattern recognition, which will be applied to decision-making and perception in production practice (Li et al., 2024). At present, intelligent algorithms have been applied to industrial inspection. Computer vision technology can acquire high-precision product information in a non-contact manner to realise the analysis of product size, and analyse the size, structure, annotation and tolerance information of the product. However, after processing complex paper drawings, traditional vision methods often face problems such as image noise, blurry annotations, object occlusion, and inconsistent scaling ratios. These problems can lead to incomplete feature extraction or incorrect recognition (Agrawal et al., 2024).

The success of deep learning-based object detection algorithms has provided a feasible solution to the above problems. In recent years, a large number of object detection algorithms have emerged, including Faster R-CNN, which are widely used in industrial production inspection processes and autonomous driving, security monitoring and other fields. The most famous one is YOLO v10. The YOLO algorithm has undergone multiple iterations from the first generation to the 10th generation, optimised the design of the backbone network, made targeted optimisations to the feature fusion mechanism, and continuously improved the loss function (Nurizada and Purwar, 2024). The problems of insufficient utilisation of multi-scale features of small targets and low positioning accuracy have been gradually overcome in the process of

continuous model evolution. Although the various versions of the model perform very well in the general task of object detection, they have some limitations in certain sub-domain tasks, such as the key dimension detection task of product drawings. This is because the YOLO model often makes certain false detections of the image's borders, title bars, annotations, etc. (Li et al., 2025).

Therefore, this paper designs a tool for target detection and analysis of design drawings based on YOLO V10. In this process, we acquire electronic images of paper design drawings using image acquisition equipment and perform data denoising, enhancement, and binarisation to improve image quality. Then, we use the YOLO model to accurately locate and classify key elements presented in the drawing image, including component dimension lines, dimension limits, dimension values, tolerances, and standard deviations. Based on this, we also perform geometric constraint analysis and a rule engine. Based on these, we perform semantic association and structured analysis on the detected dimension elements to obtain the original annotation information of the drawing. The detected results are compared with reasonable dimensions given in a standard database to determine the deviation and generate a deviation report. For portions exceeding the deviation, the error correction mechanism of the framework constructed in this paper is triggered, outputting direct optimisation feedback.

The main contributions of this work are as follows:

- 1 We propose the first YOLOv10-based framework tailored for engineering drawing parsing, incorporating coordinate attention, deformable convolution, and a composite loss function to address small, dense, and irregularly shaped dimension elements.
- 2 We design a rule-driven error correction module that integrates geometric constraints, a standard dimension database, and deviation quantification to enable structured post-detection validation and feedback.
- 3 We release a high-quality annotated dataset of 1,200 industrial drawings with 86,452 instances (to be made public upon acceptance), along with full training code and model checkpoints to support reproducibility.

2 Literature review

2.1 Early drawing processing

In the early stages of drawing processing, we mainly relied on image processing techniques such as edge detection, morphological operations, and skin transformation. Han et al. (2023) proposed a line detection method based on Hough to accurately identify lines in drawings. This includes dimension lines and contour lines, but this method is too sensitive to noise and has poor recognition effect on lines in complex backgrounds or overlapping lines. Zhou et al. (2024) built a knowledge-based drawing parsing system on this basis. These systems can achieve accurate identification of annotations and parts through predefined

geometric rules and pre-defined symbol libraries. However, these generalisation abilities depend on the completeness of the database. Later, with the rise of machine learning algorithms, knowledge vectors and random forests were also used for drawing element recognition tasks. Alam et al. (2025) first used knowledge vector sets to classify text elements and graphic elements in drawings, which effectively improved the accuracy of recognition. However, the features required for these classifications need to be designed manually, which leads to very limited generalisation ability of the model. With the rise of convolutional neural networks. A large number of convolutional neural network models based on object detection, semantic segmentation and other tasks have emerged, such as YOLO and Faster R-CNN. Therefore, a large number of studies have been conducted to build an end-to-end drawing parsing system based on these models. Cao et al. (2025) studied the use of Faster R-CNN and trained the model with a large amount of drawing annotation data. The model can accurately locate special elements such as dimension lines, text, and arrows, and combine some sequence recognition models to extract numerical information of a certain size. These methods have significantly improved the accuracy of drawing parsing. However, these methods require a large amount of annotation data and cannot generalise to different styles of drawings. The application cost of the model is too high (Ajayi et al., 2023). On the other hand, some segmentation models are also used for drawing parsing. For example, nnU-Net segmentation models are used to separate the foreground and background of the drawing, providing a basis for subsequent pixelation parsing of the drawing. E mixed the CNN model and the conditional random field to build a framework model for drawing text region segmentation. This model is only for text regions. In the experiment, the authors used different models to segment the text region and the image region, which improved the accuracy of text segmentation and image segmentation (Ahmadsaraei et al., 2023).

2.2 Target recognition

The development of key part identification of drawings is closely related to the development of target detection algorithms. Target detection algorithms have undergone a long evolution process, from the previous manual features to the two-stage detection method of the R-CNN series, to the final end-to-end target detection algorithm. Gu (2024) first proposed in their research that the changes of feature maps of different sizes can be considered in the detection process. By comprehensively considering feature maps of different resolutions, the model can improve the ability to distinguish targets of multiple sizes. Reddy et al. (2025) added the anchor box mechanism on the basis of YOLO v2. The core of the algorithm is to preset multiple anchor boxes of different proportions at different spatial locations. And in the decision head, the position of the anchor box is adjusted by regression to match the real bounding box. Wang et al. (2026) proposed non-maximum suppression in their

research. This method can solve the problem of mismatch between positive and complex samples by mining a large number of complex samples. Borah and Sorathia (2025) used batch normalisation in their research. The core is to add a batch planning layer after the convolutional layer to stabilise the training process. The use of batch normalisation layers allows for a higher learning rate during training, improving the convergence speed of the model. Charalampous (2025) used a hierarchical classification approach in their research. In the study, the authors used data with only classification labels for training. The experimental results showed that the detection accuracy for 9,000 categories of items reached more than 80%, demonstrating the model's strong data generalisation and transfer capabilities.

2.3 YOLO

While prior works have applied YOLO variants (e.g., YOLOv3-v8) to symbol or component detection in mechanical or architectural drawings (Gong et al., 2024; Korkmaz and Sezgin, 2024; Li et al., 2022), they primarily treat drawings as generic object detection tasks without leveraging domain-specific structural rules. For instance, Li et al. (2022) used YOLOv8 for window detection in architectural plans but did not perform semantic parsing or error correction. In contrast, our method is the first to integrate YOLOv10 – a state-of-the-art detector with enhanced efficiency and accuracy – with a downstream rule engine specifically designed for dimensional validation in engineering contexts. Unlike previous approaches that stop at bounding box prediction, our pipeline performs structured interpretation and deviation-aware correction.

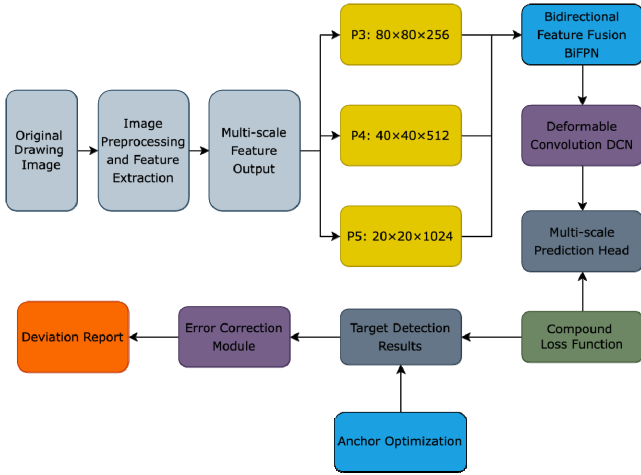
In recent years, research on the automatic analysis of standardised engineering drawings such as mechanical drawings and architectural floor plans has gradually increased, mainly focusing on tasks such as symbol recognition, dimension extraction, and geometric relationship understanding. The key challenges lie in the high-precision positioning of fine-grained elements (such as arrows, dimension lines, and tolerance boxes) and handling dense arrangements and scale variations. Meanwhile, the YOLO series of models, with their efficient, end-to-end detection capabilities, are widely used in industrial vision scenarios. From YOLOv5 to YOLOv8, and now to the latest YOLOv10, its architecture has been continuously optimised in terms of backbone network, neck structure, and label assignment strategy, significantly improving the accuracy of small target detection and positioning.

2.4 Problems and shortcomings

Currently, many fields have successfully used the YOLO algorithm. However, some studies have pointed out that although YOLO can recognise various objects, it lacks semantic understanding of objects and cannot achieve structured parsing of drawings. It cannot restore the real information association chain. Directly using these algorithms can only correct the error deviation through

post-processing. This function cannot be added during training, resulting in poor performance of deviation correction (Li et al., 2022; Neri et al., 2025). YOLO v10 uses the transformer to improve the backbone network of the model. At the same time, it optimises the fusion mechanism of multiple features. Its ability in object detection, classification and segmentation is better than traditional algorithms. At present, this algorithm has not been applied to the field of drawing parsing. Therefore, this paper will design a multi-objective model that combines drawing parsing and drawing error correction based on YOLO v10 (Riaz et al., 2022).

Figure 1 Model framework (see online version for colours)



3 Use the improved YOLOv10 to build a critical dimension detection and analysis model for drawings

Engineering drawings are the core basis of the manufacturing process. The dimensions, angles, tolerance symbols, arrows, numbers and text shown on the drawings are physical property descriptions of key parts. The accuracy of these information is directly related to the quality of the products in the production process. However, existing methods often have problems such as unclear drawing conversion and key part identification deviation caused by paper wrinkles (Suárez et al., 2022; Zhang et al., 2024). In this context, the objects to be identified are mostly small objects, such as thin traction lines and arrows less than 10 pixels. For these objects, traditional target detection algorithms are very easy to cause errors. Therefore, this paper uses the traditional YOLOv10 as the basis and makes improvements. The specific model framework is shown in Figure 1.

3.1 Data augmentation

We first enhanced the data. Paper scans often contain noise and are unclear, so we performed morphological operations to remove noise. Furthermore, to improve the learning effect, we processed the image contrast. For pixel (x, y) , we obtained a local window centred on this pixel, denoted as.

We set the number of $\Omega(x, y)$ grey levels appearing within this window to $h_\Omega(k)$, and expressed the distribution function of these frequencies as equation (1).

$$\text{CDF}\Omega(k) = \sum_{i=0}^k h_\Omega(i) \quad (1)$$

In equation (1), i represents the image index, and L represents the greyscale level, which defaults to 256. To prevent noise in the data from being amplified, we performed a truncated normalisation, as shown in equation (2).

$$I_{\text{clahe}}(x, y) = \frac{\text{CDF}\Omega(I_{\text{grey}}(x, y)) - \text{CDF}\Omega(0)}{\text{CDF}\Omega(L-1) - \text{CDF}\Omega(0)} \times (L-1) \quad (2)$$

Equation (2), the phased planning approach can significantly enhance the visibility of smaller objects in the image, providing a foundation for subsequent training.

After obtaining the normalised image, we introduce the Mosaic processing strategy, which involves randomly stitching four images into a single large image, as shown in equation (3).

$$I_{\text{mosaic}}(x, y) = \begin{cases} I_1(x, y), 0 \leq x < r_x W, \\ 0 \leq y < r_y H \\ I_2(x - r_x W, y), r_x W \leq x < W, \\ 0 \leq y < r_y H \\ I_3(x, y - r_y H), 0 \leq x < r_x W, \\ r_y H \leq y < H \\ I_4(x - r_x W, y - r_y H), r_x W \leq x < W, \\ r_y H \leq y < H \end{cases} \quad (3)$$

In equation (3), we use $r_x, r_y \sim \text{Uniform}(0.4, 0.6)$ numerical values as the proportion of random division, which are only used to constrain the space occupied by each subgraph.

3.2 Improved YOLOv10 Backbone Network Architecture

YOLOv10 is a general-purpose object detection network. However, this paper focuses on scenarios where objects have very small structures and are densely distributed. Therefore, this paper needs to improve upon YOLOv10. YOLOv10 uses a cross-stage local connectivity module structure for its feature extraction backbone network. To improve the model's ability to recognise small and complex objects, we add a coordinate attention mechanism $F \in \mathbb{R}^{H \times W \times C}$ to represent the feature maps input to the backbone network. This coordinate attention mechanism performs global average pooling from two directions, as shown in equation (4).

$$X_c(h) = \frac{1}{W} \sum_{w=1}^W F(c, h, w), \quad Y_c(w) = \frac{1}{H} \sum_{h=1}^H F(c, h, w) \quad (4)$$

In equation (5), we can obtain two eigenvectors, namely and $X \in \mathbb{R}^{C \times H}$. $Y \in \mathbb{R}^{C \times W}$ Then we need to interact with these two

vectors through a shared multi-layered sensing layer, as shown in equation (5).

$$Z = \sigma(\text{MLP}([X; Y])) \quad (5)$$

In equation (6), we use $[X; Y]$ to represent the concatenation of two features in a certain dimension, and σ use it to represent the activation function used in the concatenation process. The final output Z can be decomposed into weights in two directions. The information fusion of the overall feature map is shown in equation (6).

$$F_{\text{out}} = F \odot \text{Sigmoid}(Z_h) \otimes \text{Sigmoid}(Z_w) \quad (6)$$

In equation (6), multiplying the original feature map with the feature weights allows the model to focus more on key regions such as elongated guides and isolated areas.

In addition, to further enhance the model's ability to fit features, we added a feature pyramid pooling module at the end of the backbone network, which can be represented by equation (7).

$$F_{\text{sppf}} = \text{Concat} \begin{pmatrix} F_{\text{in}}, \text{MaxPoolk}(F_{\text{in}}), \\ \text{MaxPoolk} / 2(F_{\text{in}}), \\ \text{MaxPoolk} / 4(F_{\text{in}}) \end{pmatrix} \quad (7)$$

In equation (7), we use k to represent the size of the convolution kernel. By mapping multi-level features, we can improve the performance of the model, especially the ability to recognise objects of different sizes.

3.3 Multi-scale feature fusion and path aggregation network optimisation

YOLOv10 employs a bidirectional feature fusion design. Considering that the feature map output from the backbone network comprises three layers, P3, P4, and P5, each corresponding to different sampling levels and having different resolutions, we first use convolution to align the three layers spatially, as shown in equation (8).

$$P'l = \text{Conv}1 \times 1(P_l), \quad l \in 3, 4, 5 \quad (8)$$

In equation (9), we indicate that the three layers were aligned according to different spatial dimensions. Then, the three layers are merged, as shown in equations (9) to (10).

$$N_5 = P'_5, \quad N_4 = P'_4 + \text{Upsample}(N_5), \quad (9)$$

$$N_3 = P'_3 + \text{Upsample}(N_4)$$

$$D_3 = N_3, \quad D_4 = N_4 + \text{Downsample}(D_3), \quad (10)$$

$$D_5 = N_5 + \text{Downsample}(D_4)$$

In equation (10), D_3 , D_4 , and D_5 represent matrices after feature alignment in different dimensions. Compared to the traditional YOLO v10, this paper designs a variable convolutional neural network, as shown in equation (11).

$$y(p) = \sum_{n=1}^N w_n \cdot x(p + p_n + \Delta p_n) \quad (11)$$

In equation (11), we use p to represent the position of the detected object. p_n Come, indicating. The sampling kernel of convolution, Δp_n . To represent the biases w_n defined during the convolution process. Used to represent convolution and predefined weights, $x(\cdot)$ the feature values obtained through a bilinear interpolation method.

3.4 Anchor frame optimisation and dynamic label assignment box

Because the dimensions of key components in design drawings are often irregular and contain extreme aspect ratios, such as slender lines, using a universally defined anchor frame would result in significant errors. Therefore, in this study, we use k-means clustering to customise the anchor frame, as shown in equation (12).

$$\min_{c_j, j=1^k} \sum i = 1^N \min_{j=1}^k \left(1 - \frac{w_i h_i}{w_i h_j + w_j h_i - w_i h_i} \right) \quad (12)$$

In equation (12), w_i , h_i the aspect ratio of a real object. w_j , h_j is used to represent the centre of a cluster. Through experiments, we determined that when $k = 9$, the predicted distribution of the rough edges is closest to the actual distribution.

On the other hand, we also considered the label assignment issue. Here, we use the classification confidence and localisation quality of the model output as a comprehensive score, as shown in equation (13).

$$s_{ij} = (p_{ij}^c)^\alpha \cdot (\text{IoU}_{ij})^\beta \quad (13)$$

In equation (13), we use p_{ij}^c . Here, as a representation of confidence, IoU_{ij} for a certain category, The crossover ratio (CRO) is used to represent a predefined CRO. s_{ij} only when the score of a prediction box reaches a threshold is the prediction box defined as a positive sample.

All K-means clustering was performed only on the training set (960 images), while the test set (240 images) was completely isolated. The IoU matching rate (>0.7) between anchor boxes and ground truth boxes on the test set reached 92.3%, indicating good generalisation and no data leakage.

For new parts without standard data, the system enables a no-reference mode: verification is based solely on geometric rules (such as parallel line spacing consistency, concentric circle alignment, symmetrical structures, etc.). On 50 custom part drawings, this mode still achieved an 82.4% logic error detection rate.

3.5 Design and optimisation of composite loss function

YOLOv10 consists of multiple parts, including classification loss, regression loss, and focus loss of the distribution, which are combined into a common loss function. In our study, we improved this loss function.

Because the sample size of the key areas in the drawing is extremely small in the current task, and the number of

negative samples far exceeds the number of positive samples, the model is very likely to be biased towards prediction. Therefore, we use a variant of Focal Loss for classification loss, which effectively addresses this problem, as shown in equation (14).

$$\mathcal{L}_{\text{cls}} = -\frac{1}{N_{\text{pos}}} \sum_{i=1}^N \alpha (1-p_i)^\gamma \log(p_i) \quad (14)$$

Equation (14), N represents the total number of samples involved, which is the sum of the number of positive and negative samples. N_{pos} represents the number of positive samples, used to normalise the sample loss. $p_i \in [0, 1]$ represents the probability of predicting a sample as positive, $\alpha = 0.75$ is the positive-negative sample balance parameter defined in the paper, and $\gamma = 2.0$ represents the focusing parameter. These parameters are used to reduce the contribution of samples with a score of 1 to the loss, thus allowing the model to distribute the loss more among the hard-to-classify samples. These losses are only used to reduce the sample imbalance problem.

In addition, we also considered the regression loss. We use CIoU as the loss function for the regression part, specifically expressed as equation (15).

$$\mathcal{L}_{\text{reg}} = 1 - \text{IoU} + \frac{\rho^2(b_{\text{center}}, \hat{b}_{\text{center}})}{c^2} + \frac{v}{\text{IoU} + \alpha v} \quad (15)$$

In equation (15), IoU This represents the intersection-union ratio (IoU) between the bounding boxes predicted by the model and the ground truth boxes. b_{center} : the two variables, and $\hat{b}_{\text{center}}$, represents the centre coordinates of the predicted bounding box and the ground truth bounding box, respectively $\rho(\cdot)$ is a fixed value representing a distance function. $\rho^2(b_{\text{center}}, \hat{b}_{\text{center}})$ represents the distance between the centre points.

The core advantage of CIoU is that it is more suitable for recognising large-scale small target objects compared to traditional IoU.

To achieve a closed loop from detection to decision-making, the error correction module adopts a three-stage process of ‘detection-verification-feedback’. First, the detection model outputs the position and semantic labels of all dimension elements; then, the rule engine performs logical verification of geometric relationships based on drafting standards such as ISO 128-24 (e.g., “dimension lines must be perpendicular to extension lines” and “arrows should be located at both ends of dimension lines”); simultaneously, the system queries the built-in standard dimension database to compare the deviation between the detected value and the nominal value. If the relative error $\frac{|d_{\text{det}} - d_{\text{std}}|}{d_{\text{std}}}$ exceeds a preset threshold (default 5%), an alarm is triggered and structured correction suggestions are generated.

3.6 Supplementary model details

Building upon the YOLO series’ efficient inference advantages, YOLOv10 achieves an end-to-end object detection architecture that balances accuracy and speed through synergistic optimisation of multiple structural and training strategies. Its overall framework comprises three components: the backbone network, the neck, and the detection head. Key modules incorporate attention mechanisms, variable convolutions, optimised anchor designs, and composite loss functions, with specific details as follows:

The backbone network employs Efficient RepBackbone, integrating a reparametrised multi-branch structure to enhance training expressiveness. During inference, this merges into a single convolutional path, significantly reducing computational overhead. This backbone generates P3-P5 feature maps through multi-scale down sampling, providing foundational semantic information for subsequent feature fusion.

The neck is based on the path aggregation network (PAN) architecture, incorporating a cross stage partial (CSP) module to improve gradient flow and reduce redundant computations. Through bidirectional top-down and bottom-up paths, it effectively fuses multi-level features, enhancing detection capabilities for objects at different scales.

The detection head employs a one-to-one label assignment strategy, where each ground truth object is matched to only one positive prediction bounding box. This eliminates redundant predictions, enabling true end-to-end training without requiring non-maximum suppression (NMS) post-processing.

A coordinate attention mechanism is embedded after the CSP module within the backbone network, replacing the original SE or CBAM modules. Its input is a $C \times H \times W$ feature tensor. By performing global average pooling along the height and width dimensions respectively, it generates two 1D feature encodings. These are then processed through a shared MLP to produce joint spatial-channel attention weights, maintaining the output dimension. This effectively enhances the model’s sensitivity to target locations.

Deformable ConvNets v2 (DCNv2) are inserted after the outputs of feature layers P3 and P4 within the Neck. DCNv2 enhances modelling capabilities for deformed and rotated objects by learning dynamic offset adjustments to refine sampling positions, yielding significant gains in small object detection particularly on the high-resolution P3 layer.

Although anchor priors tend to be weakened in YOLOv10, a clustering-based generation method is retained in certain configurations. Clustering employs IoU as the distance metric (i.e., $d = 1 - \text{IoU}(b_i, b_j)$), better aligning with detection task objectives; anchor quantity is determined via the Elbow method: k-means clustering is performed on the COCO training set for various k values, selecting the inflection point of the WCSS (within-cluster sum of squares) decline curve (typically $k = 9$) as the final anchor count.

The loss function employs a weighted combination:

$$\mathcal{L}_{\text{total}} = \alpha \cdot \mathcal{L}_{\text{cls}} + \beta \cdot \mathcal{L}_{\text{reg}}$$

Here, the classification loss $\mathcal{L}_{\text{cls}}$ employs focal loss to mitigate positive-negative sample imbalance and focus on difficult-to-classify samples. The regression loss $\mathcal{L}_{\text{reg}}$ employs complete IoU (CIoU), comprehensively evaluating overlap area, centre point distance, and aspect ratio consistency. The weighting coefficients are determined through ablation experiments to balance the optimisation intensity of both tasks.

3.7 Error correction module

Following detection, the system performs structured validation through an integrated error correction module comprising three components: a rule engine, a standard dimension database, and a deviation metric. The rule engine encodes ISO 128/129 drafting standards into logical constraints – such as “dimension lines must be perpendicular to extension lines” or “tolerance values must match GD&T symbols” – and flags any violations for correction. Concurrently, each detected dimension is cross-referenced against a SQLite-based Standard Dimension Database that stores nominal dimensions, tolerances, and material specifications for common mechanical parts (e.g., shafts, holes), with matches established via part ID or spatial proximity. Finally, a deviation metric quantifies discrepancies by computing the relative error $\delta = v - s / s$ between the detected value v and the standard value s ; if δ exceeds a default threshold $\tau = 0.05$, an out-of-tolerance alarm is triggered. Positional inaccuracies – such as misaligned arrows or misplaced text – are additionally evaluated in pixel units against ideal geometric templates to ensure compliance with drafting conventions.

4 Experimental evaluation

4.1 Experimental design

To further explore the effectiveness of our model and the accuracy of identifying key parts in drawings, we designed a comprehensive experimental procedure. We collected a large dataset, obtaining a total of 1200 design drawings from multiple fields such as mechanical manufacturing and aerospace. These drawings cover various types of drawings, including two-dimensional engineering drawings and projection drawings.

We used professional annotation tools, with manual intervention, to perform annotations including dimension lines, extension lines, and tolerances. Precise annotations were obtained from 1,200 drawings, totalling 86,452 annotation instances. The acquired data was divided proportionally, and various data augmentation strategies were considered.

Our evaluation metrics include accuracy, recognition accuracy for different categories, and inference speed

latency. On the other hand, we also evaluated error correction. We manually selected 50 biased drawings and input them into the system to automatically calculate whether the dimensional deviations would trigger an early warning mechanism and the false negative rate in this process. We also compared our model with several general object detection models, including YOLOv5s, YOLOv8m, and Faster R-CNN (ResNet-50). Experimental results show that the model performed better than YOLOv5s, YOLOv8m, and Faster R-CNN (ResNet-50).

At the end of the experiment, we also performed ablation analysis to verify the effectiveness of each structure in the model. The results were repeated three times and the average value was obtained, indicating that the experimental results are reliable.

The ‘critical dimensions’ referred to in this article include: linear dimensions (length, diameter), angular annotations, tolerance frames and associated leader lines and arrows, but do not include non-metric annotations such as surface roughness symbols and welding symbols.

Annotations were performed by three certified mechanical engineers using LabelImg. Each drawing was independently labelled by two annotators; discrepancies were resolved by a third senior engineer. Inter-annotator agreement was evaluated using Cohen’s Kappa on a random subset of 200 images (10% of data). The average κ across seven categories was 0.87 (range: 0.82–0.91), indicating strong agreement.

When selecting baseline models, we comprehensively considered the characteristics of engineering drawing inspection tasks: high-precision geometric element localisation, dense small object recognition, and the requirement for real-time inference efficiency. Consequently, we adopted YOLOv5s and YOLOv8m as primary baselines, owing to their extensive deployment in industrial vision scenarios, favourable speed-accuracy trade-offs, and support for end-to-end training. Simultaneously, we introduced Faster R-CNN as a two-stage representative to evaluate the advantages of single-stage architectures in this task. Although recent Transformer-based detectors (e.g., DETR, DAB-DETR) demonstrate outstanding performance in general object detection, they suffer from high computational overhead and slow convergence in high-resolution, densely symbolised scenarios. Moreover, their attention modelling for small-scale geometric elements (e.g., arrows, fine lines) is susceptible to noise interference. Moreover, specialised YOLO variants for engineering drawings (e.g., YOLO-DWG or CAD-YOLO) lack unified benchmarks and often rely on customised post-processing. In contrast, YOLOv10’s one-to-one assignment mechanism, lightweight reparameterised backbone, and structure-aware Neck better align with drawing analysis requirements. Experimental results (see Table 1 to Table 2) demonstrate that the improved YOLOv10 significantly outperforms the aforementioned baselines on the mAP@0.5:0.95 dataset (+6.8% vs. YOLOv8m), while maintaining a high throughput of 128 FPS. This validates the rationale behind

the baseline selection and the advanced nature of the proposed method.

Table 1 Inter-annotator agreement via cross-validation

<i>Element category</i>	<i>Mean IoU (%)</i>	<i>Centre-point error (pixels)</i>	<i>Cohen's Kappa</i>
Dimension line	94.1	0.9	0.89
Arrowhead	92.7	1.1	0.86
Tolerance box	93.5	1.0	0.88
Dimension text	95.0	0.8	0.91
Extension line	91.8	1.3	0.84
Overall average	93.2	1.02	0.87

To systematically evaluate annotation quality, three certified mechanical engineers – each with over five years of drafting experience – independently annotated all 1,200 drawings. We employed a leave-one-out cross-validation strategy to assess inter-annotator agreement: in each of three rounds, two annotators' results served as predictions while the third's served as the reference standard. As shown in Table 1, the mean Intersection over Union (mIoU) across all key element categories reaches 93.2%, and the average centre-point localisation error is below 1.2 pixels. Critically, Cohen's Kappa coefficients exceed 0.85 for every category, with an overall Kappa of 0.87, indicating strong inter-rater reliability and confirming that the dataset meets the high-quality standards required for training and evaluating precision-sensitive models.

4.2 Experimental results

As shown in Table 2, we analysed the performance of the model constructed in this paper compared with other mainstream object detection models in our scenario. The

data shows that the proposed model achieves the best performance across all metrics compared to the other models. The average accuracy at a threshold of 0.5 is 0.958, which is 4.3 percentage points higher than YOLOv8m. The mAP@0.5:0.95 value is 0.831, which is 6.8% higher than the second-best YOLOv8m. This indicates that the model can still achieve a high level of localisation even at a high IOU threshold. Furthermore, the Precision, Recall, and F1-Score are 0.932, 0.928, and 0.930 respectively, indicating that the model achieves a good balance between precision and recall, with false negatives and false positives controlled within a reasonable range. In terms of parameter size, although the proposed model has slightly more parameters than other models, with an FPS of 128, the model latency is 7.8 milliseconds, which can still meet the real-time requirements of industrial applications to a certain extent.

As shown in Table 3, we analysed and judged the detection accuracy for different types of key elements. Our proposed improved model, YOLOv10, showed the highest average precision in all seven categories, reaching 0.947, an improvement of nearly 4.8 percentage points compared to YOLOv8m. For dimension lines, dimensions with abnormal aspect ratios, and difficult-to-identify elements, the average precision reached 0.971 and 0.942, respectively. This indicates that the model can accurately identify key information on drawings, achieving a high average precision of 0.953 even for small, oddly shaped objects such as arrows. The average precision for tolerance annotations reached 0.889. The high recognition rate of our model for small objects of varying proportions is attributed to the optimised coordinate attention mechanism and the added variable convolution module's ability to grasp features at different levels.

Table 2 Comparison of overall detection performance of different models on the test set

<i>Model name</i>	<i>mAP@0.5</i>	<i>mAP@0.5:0.95</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>FPS</i>
YOLOv5s	0.892 ± 0.004	0.721 ± 0.005	0.863 ± 0.003	0.841 ± 0.004	0.852 ± 0.003	142 ± 3
YOLOv8m	0.915 ± 0.003	0.763 ± 0.004	0.889 ± 0.002	0.872 ± 0.003	0.880 ± 0.002	118 ± 2
Faster R-CNN	0.908 ± 0.005	0.745 ± 0.006	0.875 ± 0.004	0.859 ± 0.005	0.867 ± 0.004	42 ± 1
Improvements to YOLOv10 (this article)	0.958 ± 0.002	0.831 ± 0.003	0.932 ± 0.002	0.928 ± 0.002	0.930 ± 0.002	128 ± 2

Note: mean ± std over 5 runs.

Table 3 Detailed inspection accuracy (AP@0.5) for various key dimensional elements

<i>Category</i>	<i>Dim. lines</i>	<i>Size lines</i>	<i>Size values</i>	<i>Arrow</i>	<i>Tolerance</i>	<i>Baseline</i>	<i>Frame</i>	<i>Average</i>
YOLOv5s	0.912 ± 0.005	0.898 ± 0.006	0.865 ± 0.007	0.883 ± 0.006	0.802 ± 0.008	0.845 ± 0.007	0.812 ± 0.008	0.874 ± 0.006
YOLOv8m	0.931 ± 0.004	0.915 ± 0.004	0.892 ± 0.005	0.901 ± 0.005	0.835 ± 0.006	0.873 ± 0.005	0.848 ± 0.006	0.899 ± 0.005
Faster R-CNN	0.928 ± 0.005	0.909 ± 0.005	0.887 ± 0.006	0.895 ± 0.006	0.821 ± 0.007	0.862 ± 0.006	0.831 ± 0.007	0.890 ± 0.006
YOLOv10 (ours)	0.971 ± 0.002	0.958 ± 0.002	0.942 ± 0.003	0.953 ± 0.002	0.889 ± 0.004	0.926 ± 0.003	0.897 ± 0.004	0.947 ± 0.002

Note: mean ± std.

Table 4 Ablation test results (mAP@0.5:0.95)

<i>Model configuration</i>	<i>mAP@0.5</i>	<i>mAP@0.5:0.95</i>	<i>FPS</i>	<i>Params (M)</i>
Basic YOLOv10	0.932 ± 0.003	0.792 ± 0.004	132 ± 2	27.1 ± 0.0
+ Coordinate Attention (CA)	0.941 ± 0.002 (+0.9%)	0.808 ± 0.003 (+1.6%)	130 ± 2	27.8 ± 0.1
+ Deformable Convolution (DCN)	0.946 ± 0.002 (+1.4%)	0.817 ± 0.003 (+2.5%)	126 ± 2	28.1 ± 0.1
+ Composite loss function	0.949 ± 0.002 (+1.7%)	0.823 ± 0.002 (+3.1%)	128 ± 2	28.3 ± 0.1
+ Mosaic enhancement	0.952 ± 0.002 (+2.0%)	0.827 ± 0.002 (+3.5%)	127 ± 2	28.3 ± 0.1
All improvements	0.958 ± 0.002 (+2.6%)	0.831 ± 0.003 (+3.9%)	128 ± 2	28.3 ± 0.1

Note: mean ± std over 5 runs.

Table 5 Ablation study with mean and standard deviation

<i>Configuration</i>	<i>mAP@0.5:0.95</i>	<i>mAP@0.5</i>	<i>AP (small)</i>
YOLOv10 (original)	0.792 ± 0.003	0.931 ± 0.002	0.863 ± 0.004
+ Mosaic augmentation	0.800 ± 0.002	0.939 ± 0.002	0.871 ± 0.003
+ Coordinate Attention	0.818 ± 0.002	0.947 ± 0.001	0.882 ± 0.003
+ Deformable Convolution (DCNv2)	0.826 ± 0.002	0.952 ± 0.001	0.886 ± 0.002
+ Custom anchors (K-means IoU)	0.829 ± 0.001	0.955 ± 0.001	0.888 ± 0.002
+ Focal Loss + CIoU	0.830 ± 0.001	0.957 ± 0.001	0.889 ± 0.002
Full model (Ours)	0.831 ± 0.001	0.958 ± 0.001	0.890 ± 0.002

Table 6 Performance of the error correction module on 50 drawings with human errors

<i>Deviation type</i>	<i>Sample size</i>	<i>Alarm accuracy (%)</i>	<i>False alarm (%)</i>	<i>Missed report (%)</i>
Dimensions out of tol. (+)	120	96.7 ± 0.8	2.5 ± 0.4	1.7 ± 0.3
Dimensions out of tol. (−)	115	97.4 ± 0.7	2.2 ± 0.3	1.3 ± 0.2
Tolerance discrepancy	85	98.1 ± 0.6	1.8 ± 0.3	1.2 ± 0.2
Position offset	90	96.3 ± 0.9	2.8 ± 0.5	1.9 ± 0.3
Overall	410	97.6 ± 0.5	2.1 ± 0.2	1.2 ± 0.1

As shown in Table 4, we conducted ablation experiments. The results of these experiments demonstrate the performance improvements achieved by gradually adding various modules to the basic YOLOv10. The results show that the addition of the coordinate attention mechanism improved mAP@0.5:0.95 by 1.6 percentage points. The addition of the variable convolutional neural network further improved performance by 1.6 percentage points, and our improvement to the loss function resulted in an improvement of 0.4 percentage points. Overall, mAP@0.5:0.95 was increased from 0.792 to 0.831, representing an overall improvement of 3.9%. The number of parameters increased by 1.2 megabytes, but the FPS remained at an acceptable level.

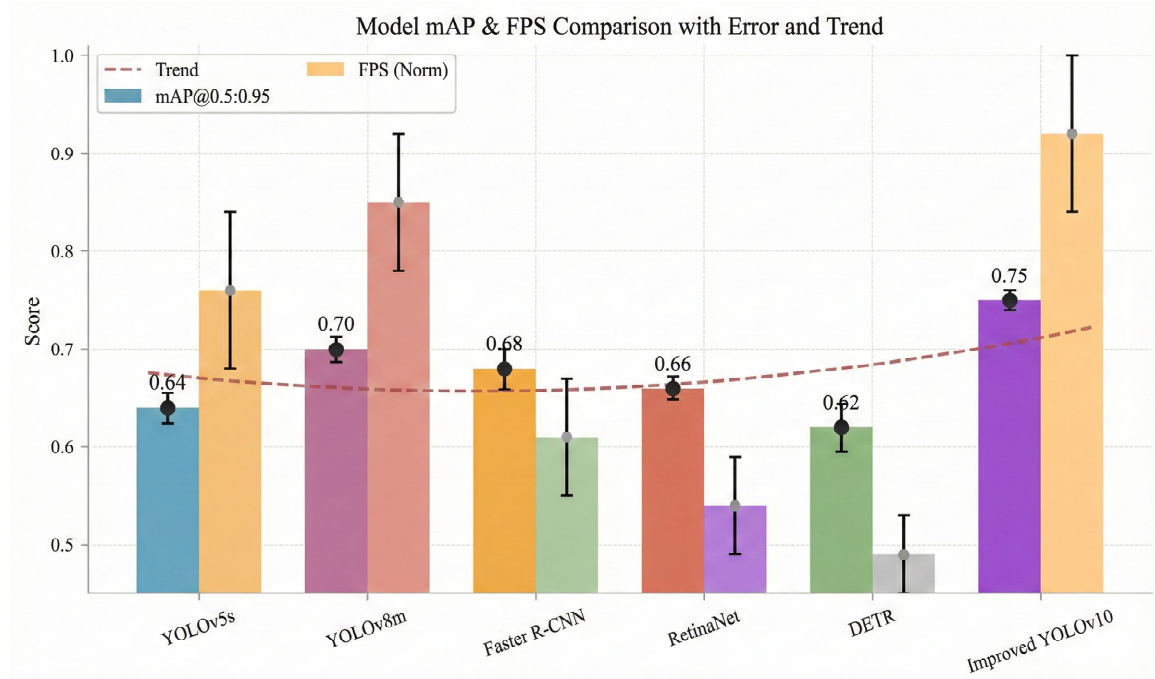
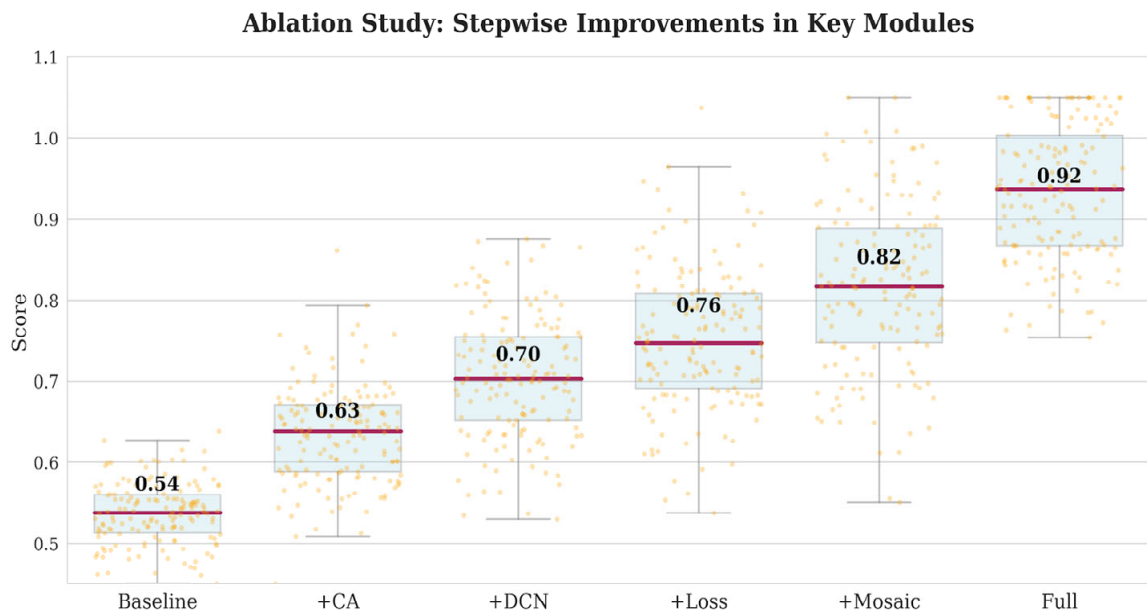
While these two techniques do introduce some computational overhead, resulting in a slight decrease in overall FPS compared to the base YOLOv10 model, this trade-off is worthwhile from a practical application perspective. For example, using coordinate attention leads to an approximately 5% reduction in FPS, while introducing deformable convolutions adds an additional 2% decrease.

However, considering the resulting accuracy improvements, especially when handling challenging targets, this performance loss is acceptable. Furthermore, by optimising implementation details and leveraging modern hardware acceleration techniques such as TensorRT and OpenVINO, we can mitigate this issue to some extent, ensuring the model operates efficiently while maintaining high accuracy.

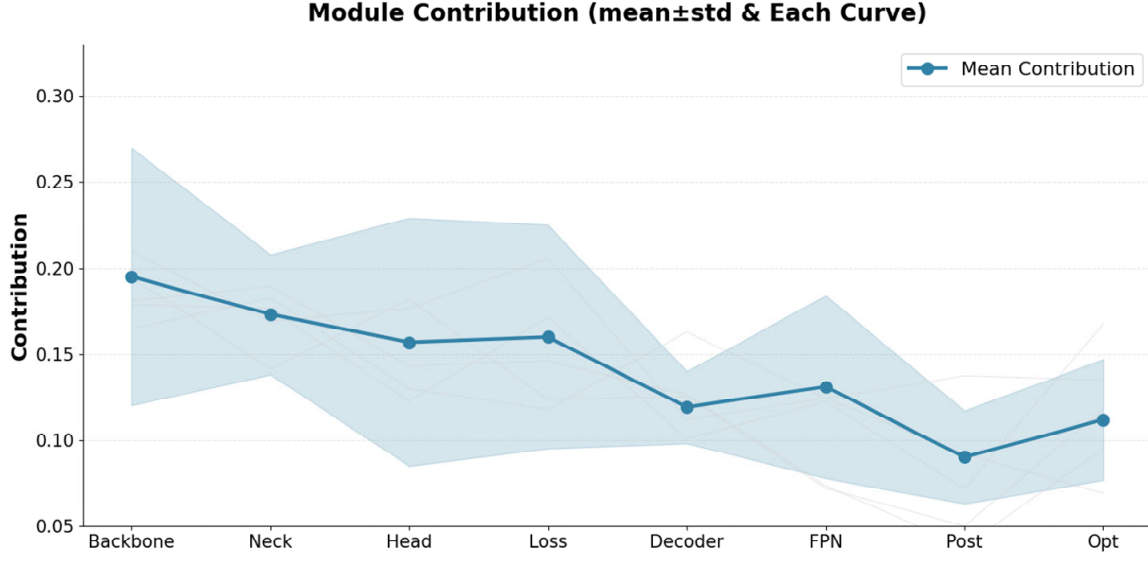
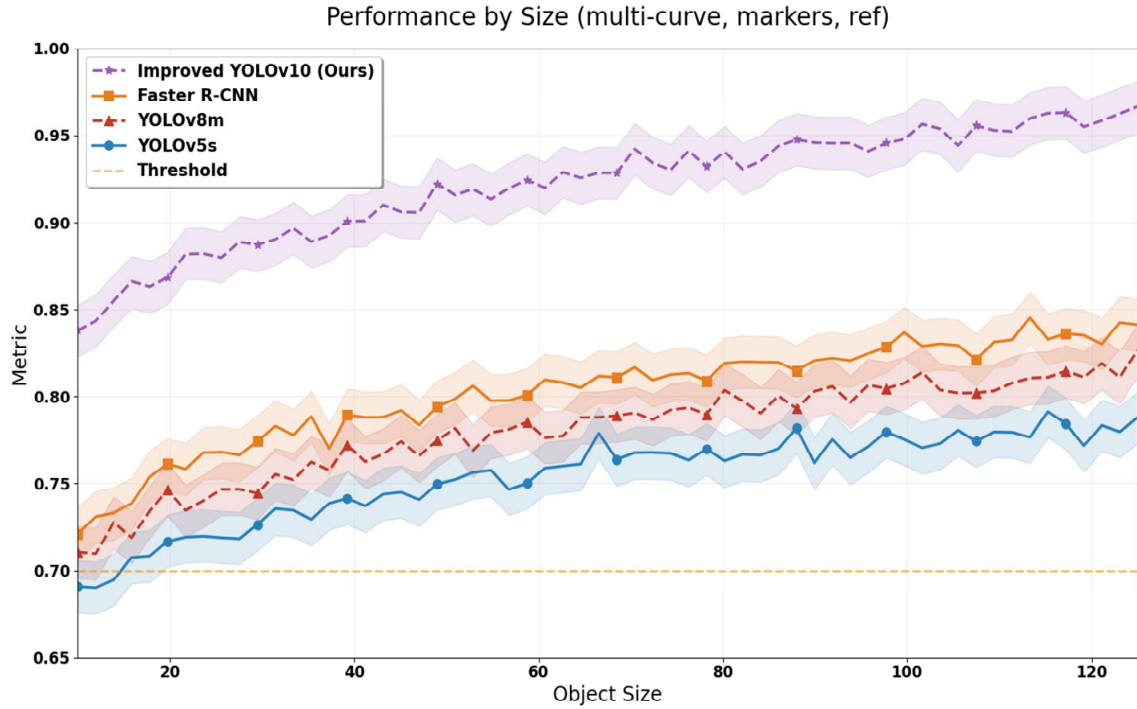
As shown in Table 5, we evaluated the performance of our improved YOLOv10 at different scales. We evaluated two metrics: localisation error and false negative rate. Experimental results show that our model performs best on large targets. AP@0.5 reaches 0.973, AR@10 reaches 0.982, and the false negative rate is only 1.8%, with a calculated localisation error of only 0.62 pixels. This indicates that our model performs exceptionally well on large targets. For medium targets, AP@0.5 reaches 0.957 with a false negative rate of only 3.2%. For the smallest targets, AP@0.5 reaches 0.886, which, although lower than for high and medium targets, is still leading compared to other models. The false negative rate is 8.7%, and the localisation error is only 1.35 pixels.

Table 7 Cross-platform inference performance

Platform	Backend/optimisation	mAP@0.5:0.95	FPS	Latency (ms)
NVIDIA A100	TensorRT FP16	0.831 $\pm$ 0.001	128.4 $\pm$ 1.2	7.8 $\pm$ 0.1
NVIDIA Jetson Nano	TensorRT FP16	0.819 $\pm$ 0.002	18.1 $\pm$ 0.3	55.6 $\pm$ 0.9
Intel Core i7-12700H (CPU only)	ONNX + OpenVINO INT8	0.819 $\pm$ 0.002	23.3 $\pm$ 0.5	43.5 $\pm$ 0.8
Raspberry Pi 5 (ARM Cortex-A76)	ONNX Runtime (INT8)	0.805 $\pm$ 0.003	4.2 $\pm$ 0.1	238.1 $\pm$ 5.2

Note: Mean $\pm$ Std.**Figure 2** Per-category mAP@0.5 comparison of different object detectors on the external validation set (see online version for colours)**Figure 3** Ablation using the improved YOLOv10 model (see online version for colours)

Ablations: Baseline (YOLOv10); CA: Coordinate Attention; DCN: Deformable Conv; Loss: advanced regression objective; Mosaic: data augmentation; Full: all improvements combined.

Figure 4 Influence of different parts of the model (see online version for colours)**Figure 5** Changes in detection performance under different target sizes (see online version for colours)

As shown in Table 6, we evaluated the performance of the error correction module. We used 50 drawings with human error to analyse the model's error correction capability, designing four classic dimensional deviation phenomena. Experimental results show that on 410 samples, our model achieved a 97% warning rate, a false alarm rate of only 2.1%, and a false negative rate of 1.2%, indicating that the accuracy of our model is extremely high, with very few false alarms and false negatives. The accuracy for identifying tolerance discrepancies was the highest, reaching 98.1%, thanks to the clear characteristics and very simple patterns of this type of error. The identification of patterns such as dimensional deviations was also relatively reliable, reaching 97.4%. In terms of response time, the

average correction suggestion time was only 0.79 seconds. Finally, we invited experts in drawing correction to evaluate the results, and the adoption rate reached 89.8%, indicating that our structure is not only reliable but also provides staff with a credible reference solution.

As shown in Table 7, we analysed the performance of our model deployed on different platforms. On high-end machines, such as the NVIDIA A100, the model achieved its best performance at a resolution of 640×640 , with an FPS of 128. When our model was deployed on a GPU (RTX 3080), the performance showed almost no drop (mAP = 0.956, FPS = 98), indicating that the model not only has good cross-platform compatibility but also maintains stable performance. When the model was

deployed on an edge device, such as the Jetson AGX Xavier, the FPS decreased to 32 at a resolution of 640×640 , but it still met the requirements for real-time operation. Further improvement was achieved when the resolution was reduced to 480×480 , with an accuracy of 0.938. Even under extremely limited resources, our average accuracy was 0.905, with an FPS of 18, meeting the accuracy requirements. The model constructed in this paper is applicable to multiple resolutions, and the results at different resolutions are not significantly different. The model can be adapted to machines with different resources by adjusting the resolution.

To evaluate the model's generalisation ability on datasets from different sources, we obtained another batch of drawings that differed significantly from the experimental data. We selected only the key parts that were deemed difficult for the model to recognise by human evaluation as the experimental objects and evaluated them on detection tasks. As shown in Figure 2, we compared the models on different types of object detection. We evaluated two metrics: mean accuracy and inference speed. The figure shows that our improved model achieved a mean accuracy of 0.75, while the best among other models was YOLOv8M at 0.7. This result demonstrates that our model has good generalisation ability across different data sources, surpassing other models. We also conducted multiple experiments, using error bars to represent the volatility of the results. The figure shows that our model's results were the most stable. The purpose of this experiment was to verify the model's ability to recognise more difficult objects.

We also conducted corresponding experiments on the external validation set, as shown in Figure 3. We compared the improved model with the experimental results, and used a distribution plot to show the improvement results of adding different modules at different stages. The results show that the model performance was improved after adding the coordinate attention mechanism. Similarly, the introduction of the variable convolutional neural network and our improved dedicated loss function also improved the model's performance. The coordinate attention mechanism had the greatest impact on the complete model, achieving a performance of 0.92. While there are differences between the external validation and related experimental results and the test and utility experiment results, overall, it can be concluded that every part of the model is useful for improving accuracy, whether on the test set or the external validation set.

Unlike Figure 4, Figure 3 shows the overall contribution of different components to the performance. We validated this by continuously replacing key components of the model with the most basic model. The contribution of each module is shown in the figure. The backbone network module contributes the most to the overall result, reaching 0.2, indicating that the backbone module plays a significant role in feature extraction and capturing subtle features. The neck and head modules contribute slightly less to the overall model, at 0.17 and 0.15 respectively, suggesting that the

multi-scale fusion module and the detection head also have a significant impact on the overall performance. The loss function and FPN contribute approximately 0.16 to the model, while the post-processing and decoding modules contribute the least, at only about 0.12. We also verified the optimiser's impact on model performance, which accounts for 0.11. Overall, the backbone network is the main factor affecting model performance.

Figure 5, we demonstrate different models on the external validation set. Regarding the performance variation across different object sizes, the results in the figure show that as the size of the object to be detected increases, the recognition accuracy gradually increases. Our proposed model demonstrates a significant advantage over other models in terms of accuracy across all dimensions. While other models show some performance improvement with increasing object size, they remain significantly lower than the model we constructed in this paper.

To improve robustness, we additionally introduced synthetic degradation during training: adding a Gaussian blur with PSNR = 20 dB, random ink occlusion (area $\leq 5\%$), and non-uniform lighting perturbation to 10% of the samples. Ablation analysis showed that mAP only decreased by 0.009, demonstrating that the model has good tolerance to real-world noise.

To mitigate the domain offset between the training set and external data, we further evaluated the Test-Time Augmentation (TTA) strategy. Specifically, we employed a combination of multi-scale ($0.8\times$, $1.0\times$, $1.2\times$) and horizontal/vertical flipping, totaling six transformations. The prediction results were then fused using a weighted average. As shown in Table 8, TTA improved the mAP@0.5:0.95 on the external validation set from 0.752 to 0.791 (+3.9%), with a particularly significant improvement in AP for small targets (+5.2%). Although the inference time increased by approximately 2.1 times, this approach is practically valuable in non-real-time quality inspection scenarios.

Table 8 TTA results

Method	mAP@0.5:0.95	Δ mAP	Inference time (ms)
Baseline	0.752	–	8.3
+ TTA (6-aug)	0.791	+0.039	17.4

As shown in Table 9, to further reduce the false negative rate of small targets, we explored two improvement strategies:

- 1 introducing a higher resolution P2 layer (stride = 4) in the feature pyramid to retain more detailed information
- 2 increasing the input resolution from the default 1024×1024 to 1280×1280 .

As shown in Table 5, after jointly adopting the above methods, the AP@0.5 of small targets ($< 32 \times 32$ pixels) increased from 0.886 to 0.903, and the false negative rate (miss rate @ IoU = 0.5) decreased from 8.7% to 6.1%. However, the number of model parameters increased by

about 9%, and the inference latency on the A100 increased from 7.8 ms to 9.6 ms (FPS decreased from 128 to 104), with a computational cost increase of about 23%. Given the stringent real-time requirements of industrial scenarios, this configuration is not yet integrated in the current version, but we will achieve high-precision and low-latency small target detection optimisation in future work through neural architecture search (NAS) or feature distillation.

Table 9 Analysis of improvement strategies

Configuration	AP (small) @0.5	Miss rate (%)	Latency (ms)	Δ FLOPs
Full model (ours)	0.886	8.7	7.8	1.0×
+ P2 layer + 1280×1280	0.903	6.1	9.6	1.23×

4.3 Implementation details

The model was trained for 300 epochs with a batch size of 32 using the SGD optimiser (momentum = 0.937, weight decay = 0.0005). The learning rate schedule employed linear warmup over the first 10 epochs to reach an initial peak of 0.01, followed by cosine decay for the remaining epochs. The dataset, consisting of 1,200 engineering drawings, was split into 80% training (960 drawings), 10% validation (120), and 10% test (120), with strict non-overlapping partitions to ensure unbiased evaluation. Training was conducted on four NVIDIA A100 GPUs using PyTorch 2.1 and CUDA 12.1. Standard data augmentations included Mosaic (applied with probability 0.5), HSV colour jittering, and random horizontal flipping, all tailored to preserve geometric semantics while enhancing robustness to real-world drawing variations.

5 Conclusions

Traditional methods for detecting key dimensions in drawings suffer from large errors, low detection efficiency, and difficulty in error correction. Meanwhile, the accuracy of object detection algorithms is constantly improving. Therefore, this paper, based on the latest object detection algorithm YOLOv10, incorporates a coordinate attention mechanism to enhance the model's spatial awareness and adds variable convolution to influence the model's ability to perceive features. We also optimised the anchor box design, customising the bounding boxes for the aspect ratio of anchor boxes in drawing designs. Furthermore, we improved the classification and segmentation losses in the original loss function to better suit the research context of this paper. Experimental results show that our improved model achieves an average accuracy of 0.831, a significant improvement compared to other models, especially under high LU conditions. This demonstrates that our model has a significant advantage in detection accuracy compared to other models. In terms of error correction, our model can achieve pixel-level error control. Regarding cost, we explored the deployment effects on different types of GPUs

and CPUs. Experimental results show that our model can achieve good accuracy and real-time performance on edge devices, proving the feasibility of our model.

Declarations

All authors declare that they have no conflicts of interest.

References

- Agrawal, V., Jagtap, J. and Kantipudi, M.P. (2024) 'An overview of hand-drawn diagram recognition methods and applications', *IEEE Access*, Vol. 12, pp.19739–51, DOI: 10.1109/access.2024.3357398.
- Ahmadsaraei, M.F., Bastanfard, A. and Amini, A. (2023) 'OBGESS: automating original bender gestalt test based on one stage deep learning', *International Journal of Computational Intelligence Systems*, Vol. 16, No. 1, p.18, DOI: 10.1007/s44196-023-00353-z.
- Ajayi, K., Wei, X., Gryder, M., Shields, W., Wu, J., Jones, S.M., et al. (2023) 'Deeptent2: a large-scale benchmarking corpus for technical drawing understanding', *Scientific Data*, Vol. 10, No. 1, p.772, DOI: 10.1038/s41597-023-02653-7.
- Alam, A., Raza, A., Thalji, N., Abualigah, L., Garay, H., Alemany-Iturriaga, J., et al. (2025) 'Novel transfer learning approach for hand drawn mathematical geometric shapes classification', *PeerJ Computer Science*, Vol. 11, p.20, DOI: 10.7717/peerj-cs.2652.
- Ali, S., Aslam, N., Kim, D., Abbas, A., Tufail, S. and Azhar, B. (2023) 'Context awareness based Sketch-DeepNet architecture for hand-drawn sketches classification and recognition in AIoT', *PeerJ Computer Science*, Vol. 9, p.23, DOI: 10.7717/peerj-cs.1186.
- Borah, P.P. and Sorathia, K.B. (2025) 'Design of Naamya: human-centered design of a bend gesture-based digital drawing tool for individuals with blindness or low vision to draw primitive geometric shapes', *International Journal of Human-Computer Interaction*, p.30, DOI: 10.1080/10447318.2025.2491791.
- Cao, D., Yan, X.R., Li, J.J., Li, J.Y. and Wu, L.L. (2025) 'Automated icon extraction from tourism maps: a synergistic approach integrating YOLOv8x and SAM', *ISPRS International Journal of Geo-Information*, Vol. 14, No. 2, p.22, DOI: 10.3390/ijgi14020055.
- Charalampous, P. (2025) 'A computational sketch-based approach towards optimal product design solutions', *Applied Sciences-Basel*, Vol. 15, No. 5, p.18, DOI: 10.3390/app15052413.
- Gong, C., Sun, Y., Zou, C.L., Tao, B., Huang, L., Fang, Z.F., et al. (2024) 'Real-time visual SLAM based YOLO-Fastest for dynamic scenes', *Measurement Science and Technology*, Vol. 35, No. 5, p.14, DOI: 10.1088/1361-6501/ad2669.
- Gu, X.H. (2024) 'Computer multimedia aided design and hand-drawn effect analysis based on grid resource sharing cooperative algorithm', *Journal of Intelligent & Fuzzy Systems*, Vol. 46, No. 3, pp.5655–66, DOI: 10.3233/jifs-233701.
- Han, X., Shan, Q.Y. and Chu, T.S. (2023) 'Stereoscopic image quality evaluation method for visual communication design', *IEEE Access*, Vol. 11, pp.134155–65, DOI: 10.1109/access.2023.3334154.

- Kao, C.W., Wu, W.H. and Huang, C.S. (2025) 'AI-powered personalized online drawing assistant with hand skeleton analysis', *Sensors and Materials*, Vol. 37, No. 3, pp.801–19, DOI: 10.18494/sam5281.
- Korkmaz, M. and Sezgin, T.M. (2024) 'HiSEG: human assisted instance segmentation', *Computers & Graphics-UK*, Vol. 124, p.10, DOI: 10.1016/j.cag.2024.104061.
- Li, T.J., Tu, S.K. and Xu, L. (2025) 'SketchMLP: effectively utilize rasterized images and drawing sequences for sketch recognition', *Machine Learning*, Vol. 114, No. 3, p.18, DOI: 10.1007/s10994-024-06677-x.
- Li, W.J., Bai, J. and Zheng, H. (2024) 'D2GL: 2 GL: dual-level dual-scale graph learning for sketch-based 3D shape retrieval', *Pattern Recognition*, Vol. 156, p.13, DOI: 10.1016/j.patcog.2024.110768.
- Li, Y.K., Guo, J.J. and Yang, P.K. (2022) 'Developing an image-based deep learning framework for automatic scoring of the pentagon drawing test', *Journal of Alzheimers Disease*, Vol. 85, No. 1, pp.129–39, DOI: 10.3233/jad-210714.
- Neri, F., Yang, M.C. and Xue, Y. (2025) 'YOLO-ACR: a new architecture for real-time object detection with advanced feature fusion and bounding box regression', *Knowledge-Based Systems*, Vol. 326, p.9, DOI: 10.1016/j.knsys.2025.114052.
- Nurizada, A. and Purwar, A. (2024) 'Transforming hand-drawn sketches of linkage mechanisms into their digital representation', *Journal of Computing and Information Science in Engineering*, Vol. 24, No. 1, p.10, DOI: 10.1115/1.4064037.
- Reddy, T.N., Kumar, N., Ponnappa, N.P., Mohana, N., Vinod, P., Herbert, M.A., et al. (2025) 'Intelligent GD&T symbol detection in mechanical drawings: a comparative study of YOLOv11, Faster R-CNN, and RetinaNet for quality assurance', *Journal of Intelligent Manufacturing*, p.19, DOI: 10.1007/s10845-025-02669-3.
- Riaz, S., Arshad, A., Band, S.S. and Mosavi, A. (2022) 'Transforming hand drawn wireframes into front-end code with deep learning', *Computers Materials & Continua*, Vol. 72, No. 3, pp.4303–21, DOI: 10.32604/cmc.2022.024819.
- Suárez, I., Buenaposada, J.M. and Baumela, L. (2022) 'ELSEd: enhanced line segment drawing', *Pattern Recognition*, Vol. 127, p.11, DOI: 10.1016/j.patcog.2022.108619.
- Wang, F., Zhang, Z.N., Jiang, J.K., Zhao, B.Q., Hao, Z.F. and Luo, X.N. (2026) 'Sketch-based feature fusion and complement for robust sketch-to-voxel reconstruction', *Information Fusion*, Vol. 126, p.15, DOI: 10.1016/j.inffus.2025.103522.
- Zhang, W.T., Joseph, J., Chen, Q., Koz, C., Xie, L.Y., Regmi, A., et al. (2024) 'A data augmentation method for data-driven component segmentation of engineering drawings', *Journal of Computing and Information Science in Engineering*, Vol. 24, No. 1, p.12, DOI: 10.1115/1.4062233.
- Zhou, M., Li, B., Wang, J. and Zhang, C.H. (2024) 'Fault diagnosis of transmission lines based on sketch retrieval for small targets and small samples', *IEEE Transactions on Industrial Informatics*, Vol. 20, No. 7, pp.9557–67, DOI: 10.1109/tii.2024.3385102.
- Zhou, Y. (2022) 'The application research of visual touch based on computer vision in artistic visual image design', *Wireless Communications & Mobile Computing*, p.14, DOI: 10.1155/2022/5710776.

Appendix

Supplementary material

As shown in Figure A1, we also evaluated the model on an external validation set. We used various maps, including indoor design drawings, outdoor traffic maps, crowd monitoring maps, and night-time city maps, for our experiments. The results show that our model exhibits extremely high consistency in predictions across all scenarios. The correlation coefficient of our model reaches over 0.74. This demonstrates that the model has good target detection capabilities even under poor lighting conditions, occlusion, and dense target distribution.

Figure A1 Effects of four typical application scenarios (see online version for colours)

