



Intelligent GD&T symbol detection in mechanical drawings: a comparative study of YOLOv11, Faster R-CNN, and RetinaNet for quality assurance

Tadigotla Narendra Reddy¹ · Nitesh Kumar³  · Nachappa Pemmanda Ponnappa¹ · Nagasiri Mohana¹ ·
Prakash Vinod¹ · Mervin A. Herbert² · Shrikantha S. Rao²

Received: 25 February 2025 / Accepted: 5 August 2025

© The Author(s) 2025

Abstract

Geometric dimensioning and tolerancing (GD&T) symbols play a vital role in engineering drawings by specifying allowable variations in part geometry to ensure manufacturing precision and functional performance. Manual identification and extraction of these symbols is labour-intensive, prone to human error, and increasingly unsuitable for fast-paced production environments, as it significantly increases quality inspection time and indirectly delays overall product delivery. This research is specifically conducted to support the development of intelligent quality management systems by integrating machine learning algorithms capable of detecting GD&T symbols directly from CAD-generated mechanical drawings. Such capability is essential for automating inspection processes and enabling reliable data extraction from design files, which are foundational to digital manufacturing workflows. Additionally, with many commercial quality automation tools being prohibitively expensive for small and medium-sized enterprises (SMEs) and micro, small, and medium enterprises (MSMEs), there is a pressing need for cost-effective, indigenous solutions. This study addresses that gap by evaluating three state-of-the-art deep learning-based object detection models—YOLOv11, Faster R-CNN, and RetinaNet—for GD&T symbol recognition. Each model was trained on a custom dataset annotated with diverse GD&T symbols, and performance was assessed using standard evaluation metrics: accuracy, recall, F1 score, and inference speed. The results show that while all three models demonstrate robust performance, YOLOv11 strikes the best balance between detection accuracy and real-time execution. This comparative study not only guides R&D teams in selecting the most suitable model for quality automation tasks but also contributes to the broader goal of enabling affordable, scalable, and intelligent visual inspection systems for SMEs and MSMEs.

Keywords GD&T symbol recognition · YOLOv11 object detection · Automated visual inspection · Manufacturing quality assurance · CAD drawing analysis · Automated quality control

Abbreviations

CAD	Computer-aided design
GD&T	Geometric dimensioning and tolerance
QMS	Quality management software
SMEs	Small and medium-sized enterprises
F1 score	F1 measure or score
OCR	Optical character recognition
MHSA	Multi-head self-attention
CUDA	Compute unified device architecture
GPU	Graphics processing unit
RAM	Random access memory
IDE	Integrated development environment
SGD	Stochastic gradient descent

✉ Nitesh Kumar
nitesh.naik@manipal.edu

¹ Centre for Smart Manufacturing Cell, CMTI, Bengaluru, India

² Department Mechanical Engineering, National Institute of Technology, Surathkal, Karnataka, India

³ Department of Mechanical and Industrial Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India

OS	Operating system
CNC	Computer numerical control
ERP	Enterprise resource planning
TPU	Tensor processing unit
IoT	Internet of Things
ML	Machine learning

Introduction

In the era of Industry 4.0, automated quality control and inspection systems have become essential for ensuring manufacturing precision, reducing rework, and minimizing rejection rates—factors that significantly influence product reliability and directly impact how industries maintain and deliver quality to their target market segments (Kashevnik et al., 2023a). The increasing complexity of production processes, coupled with the growing demand for high-precision components—particularly in sectors such as automotive, aerospace, and defense where Class A materials require comprehensive component inspection—has, alongside pressure to reduce lead times and costs, accelerated the shift toward intelligent automation in manufacturing (Katyayani et al., 2023). To understand what is specifically inspected and emphasized during quality check, it is recognized that a cornerstone of quality assurance lies in the accurate interpretation of geometric dimensioning and tolerancing (GD&T) symbols in engineering drawings. These symbols define the allowable limits for form, orientation, location, and size, thereby ensuring the functionality, interchangeability, and compliance of manufactured components with design intent (Chaturvedi & Ghose, 2023; Yasmine et al., 2023). Widely governed by standards such as ASME Y14.5 and ISO 1101, GD&T plays a vital role across the design, production, and inspection stages (Jiang, 2023; Sui et al., 2024). Traditionally, recognizing and interpreting GD&T symbols has been a manual task in most manufacturing setups, particularly those supplying critical components to industries such as aerospace, defense, and precision engineering, where the inspection process demands significant human expertise and domain-specific knowledge (Bengamra et al., 2024; Liu & Xi, 2023). However, manual extraction is not only time-consuming and error-prone but also increases quality check time, indirectly extending product delivery cycles. In fast-paced manufacturing environments, this is increasingly unsustainable. The demand for rapid, accurate, and scalable symbol detection has become a necessity, particularly as companies aim to reduce inspection bottlenecks and improve throughput (Kashevnik et al., 2023b). An intelligent system capable of automatically identifying and extracting GD&T data from CAD drawings could accelerate inspection processes by up to 50%,

significantly enhancing efficiency (Jiang, 2023; Yang et al., 2023). However, commercial solutions for such automation are often expensive and inaccessible to small and medium-sized enterprises (SMEs) and micro, small, and medium enterprises (MSMEs), where budget constraints hinder adoption (Aggarwal et al., 2023; Ali et al., 2024; Tornincasa, 2024). Deep learning, particularly convolutional neural networks (CNNs), has shown immense promise in image analysis tasks such as object detection (Kocharian et al., 2024; Mohamed, 2023). Among these, the YOLO (You Only Look Once) framework has emerged as one of the fastest and most accurate solutions for real-time detection. YOLOv11, its latest version, improves on both accuracy and inference speed, making it a strong candidate for industrial applications involving symbol recognition from CAD-generated engineering drawings (Baciu et al., 2023; Peng et al., 2024; Petruccioli et al., 2022; Zhou & Hartman, 2023). Its ability to operate in real time without compromising precision makes it ideal for integration into modern quality management systems (QMS) (Lin et al., 2023; Tandler, 2014). In addition to YOLOv11, other state-of-the-art models such as Faster R-CNN (Souza, 2024) and RetinaNet are widely used in object detection tasks. Faster R-CNN is known for its high detection accuracy using Region Proposal Networks (RPN), while RetinaNet addresses class imbalance using focal loss (Schlagenhauf et al., 2022; Zupan & Kunc, 2024). However, these models are generally slower than YOLO, limiting their effectiveness in real-time applications where speed is critical (Helle, 2023). This study presents a comparative analysis of YOLOv11, Faster R-CNN, and RetinaNet for the task of automated GD&T symbol detection in CAD engineering drawings. The models were trained on a custom dataset representing diverse GD&T symbols and tested using standard evaluation metrics—accuracy, recall, F1-score, and inference time. Our findings indicate that YOLOv11 provides the most optimal balance of detection accuracy and real-time performance, making it the most suitable for industrial deployment (Ajiga et al., 2024; Azamfirei et al., 2023). The integration of object detection models into QMS can substantially reduce manual effort, improve inspection consistency, and provide actionable feedback in real time. This is particularly valuable for SMEs/MSMEs, where scalable, cost-effective automation solutions are critical to competing in increasingly digitized supply chains (Mohamed, 2023; Psarommatis et al., 2024; Saihi et al., 2023).

Furthermore, symbol detection challenges vary significantly across industries due to differences in drawing complexity, annotation style, and precision requirements. For instance, automotive engineering drawings often feature densely packed annotations to accommodate numerous components within constrained spaces. Aerospace applications demand extreme precision and highly detailed

tolerancing, as even minor deviations can impact safety and performance. In contrast, consumer electronics involve miniaturized symbols embedded within intricate and compact layouts, often presenting low-contrast and high-density visual conditions. These domain-specific complexities necessitate detection models that are not only highly accurate but also robust to variations in symbol density, annotation layout, scale, and drawing resolution.

Automating the detection of GD&T symbols from CAD drawings remains a technical and operational challenge, especially for industries where inspection speed and precision are non-negotiable. Manual approaches continue to dominate, despite being inefficient and inconsistent, and commercially available automation tools are often cost-prohibitive for SMEs and MSMEs. Although deep learning offers a viable alternative, a lack of comparative insights into which models best handle symbol-dense, varied, and cluttered technical drawings limits practical adoption. This study confronts this problem by benchmarking three leading object detection models—YOLOv11, Faster R-CNN, and RetinaNet—specifically for GD&T symbol recognition. By doing so, it aims to bridge the gap between theoretical potential and real-world deployment of AI-driven quality control in scalable, resource-constrained manufacturing environments.

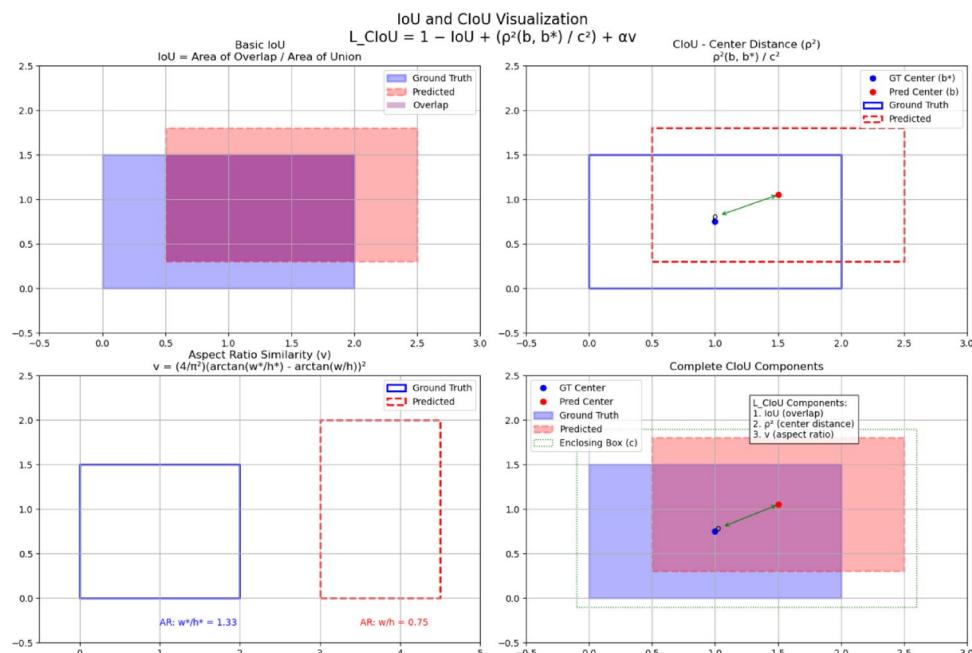


Fig. 1 IoU and CIoU visualization. This figure illustrates the components and computation of Intersection over Union (IoU) and Complete IoU (CIoU) metrics in object detection. The visualization comprises four subplots: **a** Basic IoU computation showing the overlap between ground truth (blue) and predicted (red) bounding boxes, with the intersection region highlighted in purple; **b** Centre distance measurement (p) between ground truth and predicted box centers, demonstrating the

Mathematical formulations and model expressions

In this study, to evaluate the performance of deep learning models for detecting Geometric Dimensioning and Tolerancing (GD&T) symbols, considering the underlining mechanism followed by the mathematical foundation for each model: YOLOv11, Faster R-CNN, and RetinaNet. These expressions define each model's unique approach to achieving high accuracy in complex symbol detection tasks and include deeper mathematical aspects specific to object localization and bounding box refinement.

YOLOv11 formulations

YOLOv11 utilizes an end-to-end convolutional approach with specialized anchor-free mechanisms. The advanced elements involve the Objectness, IoU Loss, and the CIoU (Complete IoU) Loss, which further refine the bounding box predictions and improve robustness. Complete IoU (CIoU) Loss CIoU considers not only the overlapping area between the predicted and ground-truth boxes but also adds aspects of aspect ratio, center distance, significantly improving YOLOv11's performance for symbols with varying shapes and orientations (Fig. 1).

The CIoU loss is defined as:

normalized distance term in CIoU; **c** Aspect ratio similarity visualization comparing different box shapes and their impact on the v term; and **d** Complete CIoU visualization integrating all components. The figure demonstrates how CIoU extends traditional IoU by incorporating centre point distance and aspect ratio consistency, leading to more accurate bounding box regression

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^*)}{c^2} + \alpha v$$

where IoU is Intersection over Union between predicted box and ground-truth box. Measures how much two boxes overlap. $\rho^2(b, b^*)$ is Squared Euclidean distance between centers of predicted box, $b = (x, y)$ and ground-truth box, $b^* = (x^*, y^*) : (x - x^*)^2 + (y - y^*)^2$. c is the diagonal length of the smallest enclosing box in the bounding area. $v = \frac{4}{\pi^2} (\arctan \frac{w^*}{h^*} - \arctan \frac{w}{h})$ measures the consistency in aspect ratios, and α is a scaling factor to balance the loss components based on IoU . $\alpha = \frac{v}{(1 - IoU) + v}$ is a scaling factor to reduce the effect of v when IoU is high.

- (1) Dynamic anchor-free prediction: YOLOv11's anchor-free mechanism uses center-based key point instead of predefined anchors. The loss for the center-based key point prediction (x_k, y_k) in the feature space is optimized by (Fig. 2):

$$L_{center} = -\log(\sigma(x_k - x_c)) + \sigma(y_k - y_c))$$

where x_c and y_c are the actual centers of the bounding box and σ is the sigmoid activation. This helps YOLOv11 achieve faster and more accurate localization without relying on predefined anchor scales.

- (2) YOLOv11 performs multi-scale detection at three feature pyramid levels. The fused prediction score S_{fused} across the pyramid is given by (Fig. 3):

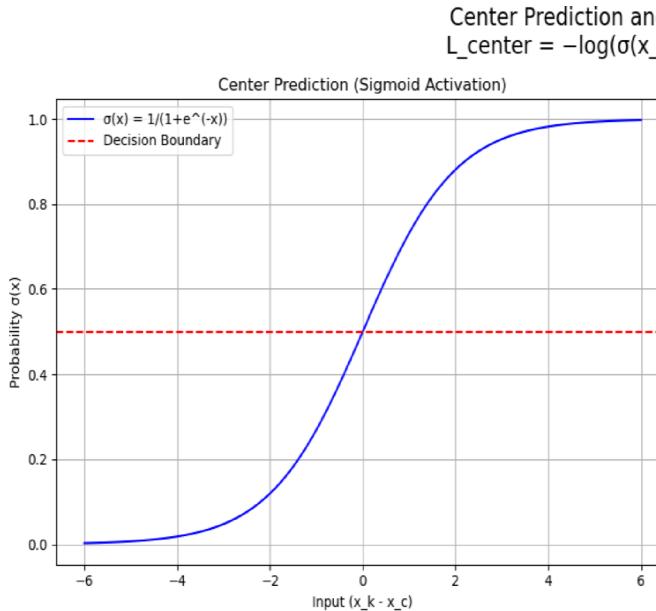


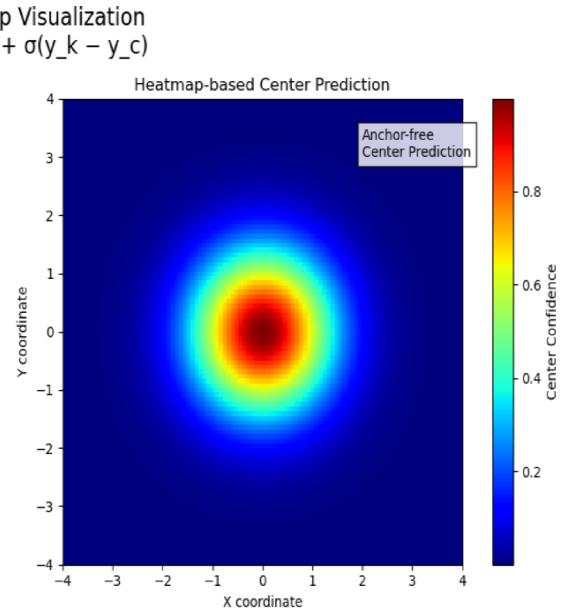
Fig. 2 Centre and heatmap visualization. The figure demonstrates the center point prediction approach in modern object detection frameworks. **a** Sigmoid activation function visualization showing the probability mapping for center point prediction, with the decision boundary at 0.5 marked by a red dashed line. **b** A 2D confidence heatmap

$$S_{fused} = \sum_{i=1}^N \alpha_i S_i$$

where each S_i is the score at feature level i , and α_i are learned weights that dynamically adjust the contribution of each scale, optimizing detection performance on symbols of various sizes. The YOLOv11 model extracts picture information using a convolutional neural network (CNN) backbone. It includes detection layers that predict bounding boxes and class probabilities for GD&T symbols. During training, YOLOv11 receives the annotated dataset and iteratively updates model weights using back propagation and optimization strategies such as stochastic gradient descent (SGD). The key formulae involved in this stage are as follows:

$$\begin{aligned} L = & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{conf} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} \left(C_i - \hat{C}_i \right)^2 + \lambda_{cla} \\ & + \lambda_{class} \sum_{j=0}^{S^2} 1_i^{obj} (p_i(c) - \hat{p}_i(c))^2 \end{aligned}$$

where S is number of grid cells per row/column. B is number of predicted boxes per cell. 1_{ij}^{obj} , 1 if object is present in box j of cell i , else 0. x_i, y_i is predicted box



illustrating the center point prediction intensity across the feature map, where brighter regions indicate higher confidence. The visualization emphasizes how the model predicts object centers using continuous probability distributions rather than discrete anchor points

center coordinates. C_i is predicted confidence score of object presence. $p_i(c)$ is predicted class probability for class c . λ_{coord} , λ_{conf} , λ_{class} are weighting terms for localization, confidence, and classification loss respectively.

Faster R-CNN formulations

Faster R-CNN utilizes a region proposal network (RPN) for object proposal generation, with a unique RoI (Region of Interest) Pooling and Bounding Box Refinement step that enhance its ability to detect symbols in complex layouts.

(1) Region proposal network (RPN)—Anchors and Bounding Box Refinement.

- The RPN generates anchors across multiple scales and aspect ratios, defined by:

$$A = \{(w_a, h_a) \mid w_a \in 8, 16, 32, h_a \in 8, 16, 32\}$$

where w_a and h_a denote the width and height of the anchor boxes. Each anchor is refined based on the predicted offsets (t_x, t_y, t_w, t_h) as (Fig. 4):

$$\begin{aligned} \{x' = x_a + t_x * w_a\} \\ \{y' = y_a + t_y * h_a\} \\ \{w' = w_a \exp(t_w)\} \\ \{h' = h_a \exp(t_h)\} \end{aligned}$$

where x_a, y_a are the anchor center coordinates. t_x, t_y, t_w, t_h are learned offsets from the region proposal network (RPN). x', y', w', h' are final predicted bounding box (center+size).

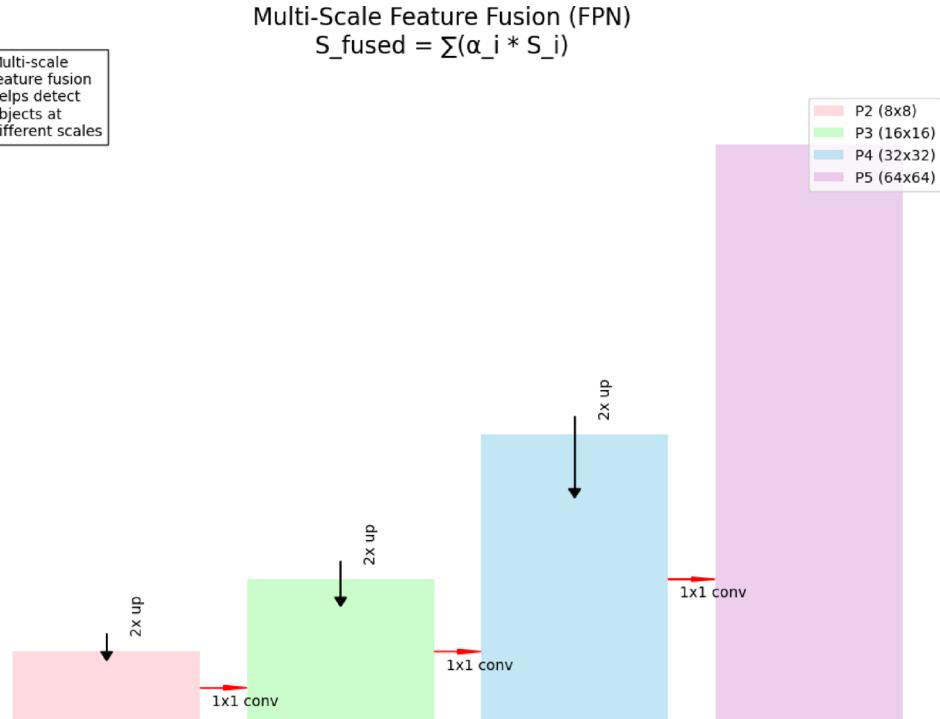
- RoI pooling: Faster R-CNN's RoI pooling layer extracts fixed-size feature maps from each proposal, facilitating efficient classification and localization. Given a proposal of size $H \times W$ and an output of $h \times w$ the pooling operation is defined as (Fig. 5):

$$f_{i,j} = \max_{(x,y) \in R_{i,j}} f(x,y)$$

where $R_{i,j}$ is the i th and j th bin region in the proposal, $f(x,y)$ is Feature map value at position (x,y) and $f_{i,j}$ is output value after max pooling in bin (i,j) . This pooling allows Faster R-CNN to handle variable-sized proposals and maintain consistent feature extraction.

- Faster R-CNN loss—multi-task loss function: The total loss function combines classification and regression losses across both the RPN and the detection network:

Fig. 3 Multi-scale feature fusion architecture. This figure presents the feature pyramid network (FPN) architecture for multi-scale feature fusion. The visualization shows four feature levels (P2–P5) with increasing receptive fields and decreasing spatial resolutions. Black arrows indicate the top-down pathway with $2\times$ upsampling operations, while red arrows represent lateral 1×1 convolution connections. Each level is color-coded and annotated with its spatial dimensions, demonstrating the hierarchical feature representation. The final prediction score is computed as a weighted sum across all scales: $S_{\text{fused}} = \sum(\alpha_i * S_i)$, where α_i represents learnable scale-specific weights. This architecture enables effective detection of objects at various scales while maintaining both semantic and spatial information



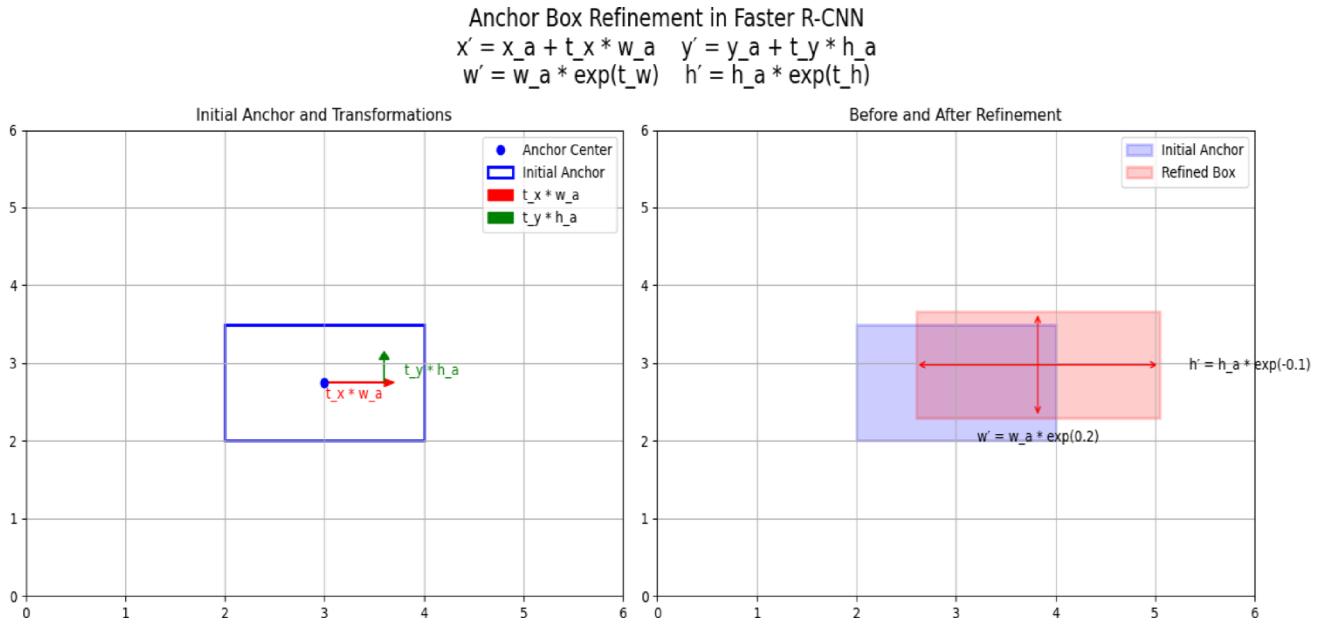


Fig. 4 Anchor box refinement process. This figure details the anchor box refinement process in Faster R-CNN through two complementary visualizations: **a** Component-wise transformation showing the initial

anchor box (blue) and the predicted transformations (t_x, t_y) scaled by anchor dimensions (w_a, h_a); and **b** Complete refinement demonstration comparing the initial anchor box with the refined prediction

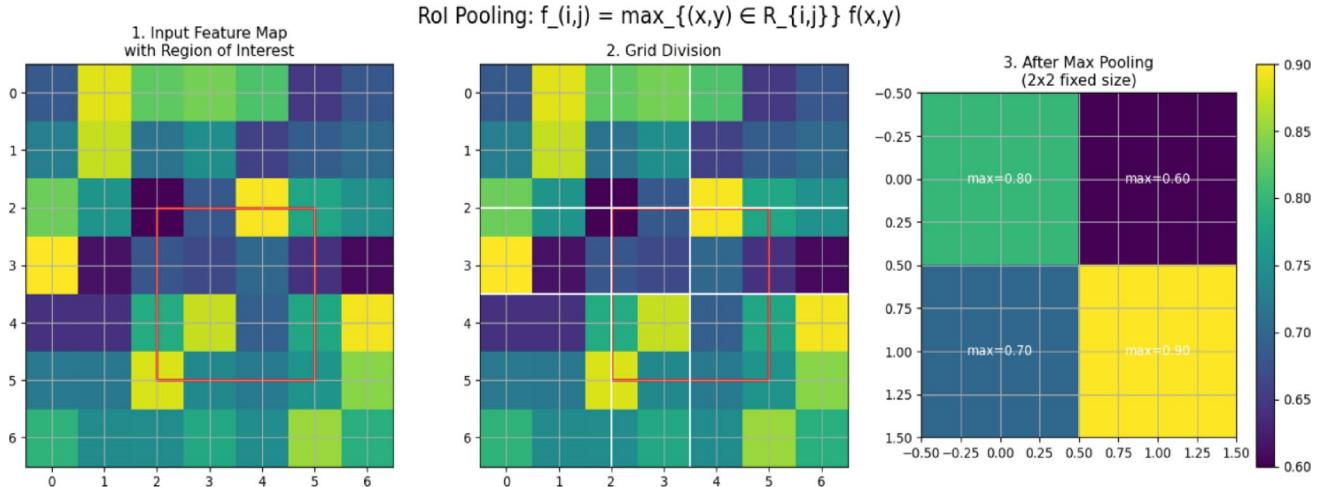


Fig. 5 ROI pooling operation process. A three-stage visualization of the region of interest (ROI) Pooling operation: **a** Input feature map (7×7) with the selected region of interest highlighted in red; **b** Grid division phase showing how the ROI is divided into a 2×2 grid with uniform cells; and **c** Final pooled output displaying the max-pooled values for

each grid cell. The figure illustrates how ROI Pooling transforms variable-sized regions into fixed-size feature representations (2×2 in this example). This operation is crucial for maintaining fixed-dimensional features for subsequent classification and regression tasks

$$\begin{aligned} \mathbf{L} &= \frac{1}{N_{cls}} \sum L_{cls} (p_i, p_i^*) \\ &+ \frac{\lambda}{N_{reg}} \sum L_{smooth-L1} (t_i - t_i^*) \end{aligned}$$

where L_{cls} is cross-entropy loss for classification. p_i, p_i^* are predicted versus true class labels. t_i, t_i^* are predicted versus true box offsets. λ is a balancing factor between classification and localization.

RetinaNet formulations

RetinaNet employs a single-stage architecture but addresses class imbalance using a Focal Loss. It also applies anchor-based detection with unique pyramid feature scaling to achieve fine-grained localization (Fig. 6).

- (1) Focal loss: Focal loss down-weights the contribution of easy-to-classify samples and focuses the model on hard examples. The formulation of focal loss is:

$$L_{focal} = -\alpha_t (1 - p_t)^\gamma \log (p_t)$$

where p_t is predicted probability for the ground-truth class, α_t is a scaling factor to balance positive/negative classes, γ is focusing parameter (commonly set to 2) that down-weights easy examples and $(1 - p_t)^\gamma$ is the feature pyramid with scales $P_3, P_4, P_5, \dots, P_7$ or multi-scale detection. Each level P_i is computed as:

$$P_i = \{Conv(C_i), \text{ if } i = 3 | \{Conv(P_i - 1), \text{ otherwise}\}$$

where C_i is the feature map from backbone layer iii. This pyramid design allows RetinaNet to detect symbols at

various scales, optimizing detection performance across small to large GD&T symbols.

- (2) Bounding box regression with L1 smooth loss: The bounding box regression in RetinaNet employs Smooth L1 loss for bounding box refinement, defined as:

$$L_{smooth-L1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| \geq 1 \\ |x| - 0.5, & \text{otherwise} \end{cases}$$

where $x = t_i - t_i^*$ is the residual between predicted and ground-truth coordinates.

These mathematical formulations underpin each model's design and working procedure. YOLOv11's CIOU loss, Faster R-CNN's RoI Pooling with multi-task loss, and RetinaNet's Focal Loss and feature pyramid scaling are all tailored for different aspects of accuracy, localization, and robustness in real-time quality control applications.

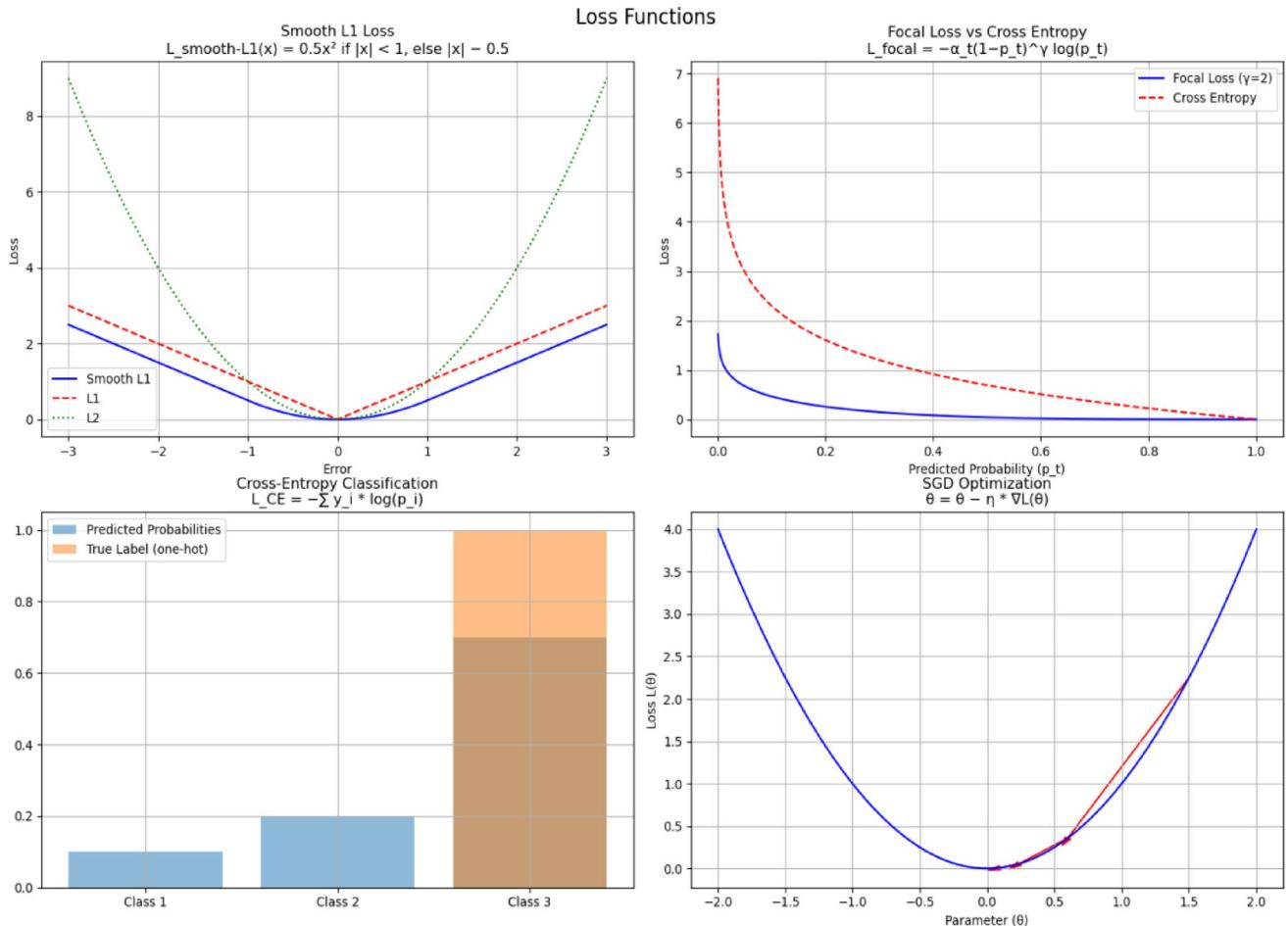


Fig. 6 Loss functions. A comprehensive visualization of four critical loss functions used in object detection: **a** Smooth L1 Loss compared against standard L1 and L2 losses, highlighting its robustness to outliers while maintaining smooth gradients near zero; **b** Focal loss versus standard cross-entropy **c** Multi-class cross-entropy visualization

showing probability distributions and true labels **d** Stochastic gradient descent (SGD) optimization process with gradient steps on a quadratic loss surface. The figure illustrates how these loss functions contribute to stable training and improved detection accuracy

Methods

To set up the training environment for YOLOv11, Faster R-CNN, and RetinaNet models on GD&T symbol detection, a dedicated hardware and software configuration is essential. For hardware, high-performance GPUs, such as NVIDIA A100, RTX 4090 are recommended for better accuracy, as these models, especially YOLOv11 and Faster R-CNN, are highly GPU-intensive and require substantial computational power. For software, Linux (e.g., Ubuntu 20.04) is typically preferred due to its compatibility with GPU drivers and machine learning libraries. Key machine learning libraries include PyTorch, the primary framework used here, customization for all three models. Additionally, CUDA and cuDNN are necessary to leverage GPU acceleration fully. Development tools such as Jupyter Notebook or PyCharm are ideal for their support in debugging, visualization, and experimentation during the training process. Once the environment is set up, the training and evaluation of YOLOv11, Faster R-CNN, and RetinaNet models were conducted on a custom dataset consisting of approximately 500 high-resolution CAD engineering drawings annotated for GD&T symbols. Annotation was performed using the Roboflow annotation suite, ensuring precise bounding boxes in compliance with the ASME Y14.5 standard. The dataset contains approximately 1800 labelled instances across 14 GD&T classes, with a notably imbalanced class distribution. To mitigate these imbalances, stratified sampling ensured fair representation across training and validation sets. Additionally, data augmentation strategies—including random rotation, Gaussian noise injection, flipping, and scale jittering—were used to synthetically balance rare classes. To evaluate model generalization, a hold-out validation approach was employed by partitioning the dataset into training, validation, and test subsets, each comprising distinct engineering drawings. This methodology ensured that validation and testing occurred on unseen samples, thereby mitigating the risk of data leakage. While conventional k -fold cross-validation is advantageous for providing more reliable variance estimates, the hold-out strategy was chosen to reduce computational complexity. Given the nature of the dataset—comprising structurally diverse CAD drawings—the hold-out design preserved inter-sample variability across sets. Despite the lack of repeated sampling inherent in k -fold methods, this approach offered a practical balance between evaluation rigor and resource constraints. The dataset also reveals domain-specific biases: (1) overrepresentation of mechanical components, (2) variable symbol sizes (bounding boxes range from 12×12 px to 90×90 px), and (3) heavy visual clutter due to overlapping annotations and dense dimension lines. These complexities necessitated robust modelling strategies including focal loss weighting,

anchor box optimization via k-means clustering, and curriculum learning to enhance detection accuracy across both frequent and underrepresented symbol classes. Each model was initialized with weights pretrained on generic object detection tasks, which helped speed up convergence. The training process was conducted in three stages: a general training stage where models learned broad symbol features, a fine-tuning stage focused on complex symbol recognition with advanced augmentations, and a hyperparameter-tuning stage where parameters like learning rate, batch size, and class weights were adjusted to address class imbalances and maximize performance. For YOLOv11, hyperparameter tuning was conducted using the built-in evolutionary search algorithm in the Ultralytics framework. Initial values for hyperparameters such as lr0, momentum, weight_decay, cls_loss_gain, box_loss_gain, and anchor_t were set based on Ultralytics's default values, which are empirically robust for object detection tasks. The evolutionary algorithm then performed automatic mutations and crossovers over 50 generations, each time training a variant of the model and evaluating it using mAP@0.5 on the validation set. This approach enabled efficient exploration of the hyperparameter space and helped identify an optimal balance between generalization and symbol-specific precision—especially crucial given the long-tailed class distribution of GD&T symbols. For Faster R-CNN and RetinaNet, we initialized from COCO-pretrained models (Torch Vision model zoo), and conducted manual grid search tuning. Initial values for learning rate, batch size, and weight decay were selected based on best practices in the literature and adjusted iteratively. In each run, performance was monitored using classification loss, box regression loss, and validation mAP. The best configurations were chosen based on stable convergence and high accuracy across both frequent and rare symbol classes. To achieve good accuracy some of the best practices in training strategies were applied. Such as, Anchor box adjustments, particularly for YOLOv11 and RetinaNet, were made by applying k-means clustering to better match the dimensions of GD&T symbols in the dataset, improving detection accuracy for symbols of varying sizes. For object detection in YOLOv11, Faster R-CNN, and RetinaNet models, certain mathematical formulas are integral to training and evaluation:

- (1) *Mean average precision (mAP)* mAP measures model accuracy by averaging the precision at different recall levels across all classes. For each class c , the average precision (AP) is calculated, and the mean is taken over all classes:

$$mAP = \frac{1}{|C|} \sum_{c \in C} AP(c)$$

where C represents the set of classes, and $AP(c)$ is the average precision for class c . mAP is a widely used metric for overall model performance.

- (2) *Focal loss* Used in RetinaNet to handle class imbalance by down-weighting method for easy-to-classify examples. It is formulated as:

$$FL(p_t) = -\alpha(1 - pt)^\gamma \log(p_t)$$

where pt is the probability of the correct class, α is a balancing factor, and γ is a focusing parameter that reduces the loss for well-classified examples. Loss function tuning involved using weighted focal loss for Faster R-CNN and RetinaNet, which specifically helped the models handle class imbalances and improved detection accuracy for less common symbols. Multi-GPU parallelization further optimized the training time by enabling larger batch sizes, which contributed to training stability. Additionally, curriculum learning, where models were first trained on simpler symbols and then gradually exposed to more complex features, helped each model adapt to the nuances of the GD&T symbols with increased resilience. After training, evaluating each model's performance is crucial for understanding their effectiveness. Metrics such as mean average precision (mAP), recall, and precision were used to measure the detection and classification accuracy of each model on the GD&T symbols. Examining loss curves, particularly box loss, classification loss, and distribution focal loss, provided insights into model convergence and helped identify areas where improvements were needed. To evaluate the performance of the model, based on the training dataset classification loss curve, distribution focal loss curve and box loss curve across different frequency of the different values of bounding box dimensions.

The classification loss curve reflects how effectively the model learns to distinguish between classes (i.e., different GD&T symbols) throughout training. A steady decline in classification loss indicates the model is successfully learning to classify symbols, while erratic behavior or a plateau can suggest issues such as class imbalance or insufficient training. The distribution focal loss curve specifically helps address class imbalances by prioritizing hard-to-classify symbols; a smooth, decreasing focal loss implies the model is becoming better at detecting less common or more challenging symbols. Both curves, when analyzed together, provide a deeper understanding of the model's convergence. A gradual and stable reduction in these loss curves suggests that the model is improving in both general accuracy and robustness, particularly on underrepresented symbols,

which is crucial for real-world deployment where balanced accuracy across classes is often required. If the loss curves show early convergence or do not drop substantially, it may indicate that the model architecture, data, or hyperparameters need refinement for better overall performance and resilience. Post-training evaluations included visualizations like pair plots and confusion matrices to understand symbol distribution, bounding box overlap, and model accuracy on specific classes, offering a clear perspective on the detection strengths and limitations of each model.

Results and discussion

- (1) *YOLOv11* With the highest mean average precision (mAP) of 0.936 with processing speed of 22 ms per image, with single-stage architecture, observed high precision and recall.

Initially, YOLOv11 model's performance was evaluated using several metrics, as shown in Fig. 7a–d, highlighting its ability to detect geometric tolerancing symbols. The F1-confidence curve (Fig. 7a) reveals a strong correlation between model confidence and F1 scores, with an F1 score of 0.92 at a confidence threshold of 0.309. Features like circularity, symmetry, and surface profile maintain excellent F1 scores, while position and perpendicularity show more variability. The accuracy-confidence curve (Fig. 7b) shows ideal accuracy (1.00) at a confidence level of 0.877, indicating strong performance with higher confidence. The precision–recall curve (Fig. 7c) achieves a mean average precision (mAP@0.5) of 0.936, with top performances for Straightness (0.995), circularity, concentricity, surface profile, and symmetry (0.994), and Cylindricity (0.993). Lower performances are seen in position (0.792), perpendicularity (0.823), total runout (0.851), and circular runout (0.863). The recall-confidence curve (Fig. 7d) shows high recall (0.95) even at low confidence thresholds, but position, perpendicularity, and total runout exhibit reduced recall as confidence increases. These results highlight the model's strong performance in detecting basic geometric features, though further work is needed to improve spatial relationship detection. Lower detection accuracy for position, perpendicularity, and total runout symbols can be attributed to multiple factors inherent to the dataset and symbol characteristics. These symbols frequently appear in densely annotated regions, often near other GD&T features or dimension lines, leading to overlapping bounding boxes and visual ambiguity. Additionally, position and perpendicularity symbols are structurally similar to other geometric features, increasing the likelihood of misclassification. Variations in annotation

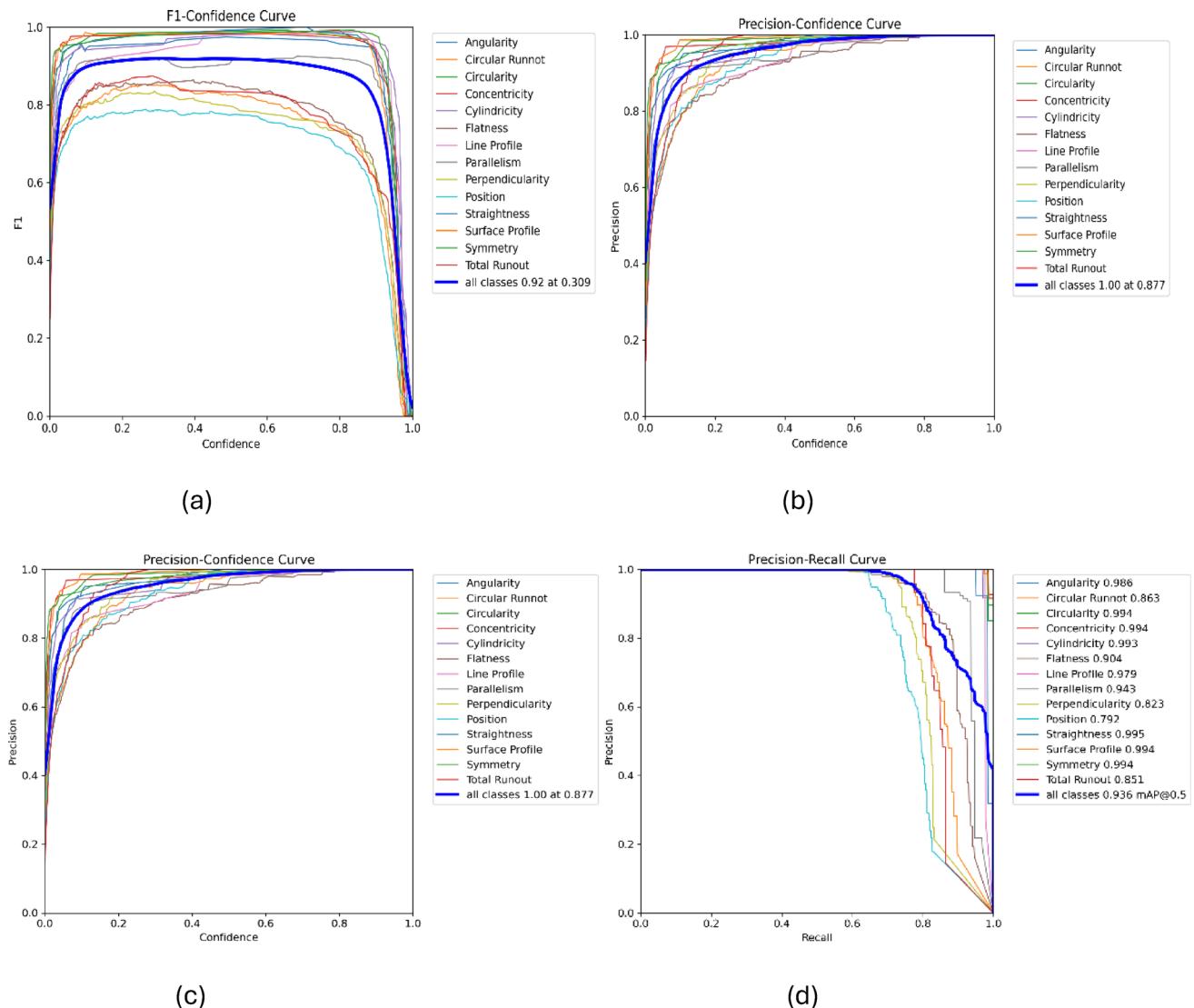


Fig. 7 F1-Confidence curve and precision-confidence curve across the symbols with their confidence values and value maps for each symbol matrix. **a** F1-confidence curve across the maximum value prediction

clarity across drawings—particularly due to differing CAD standards or resolution inconsistencies—further degrade detection performance. Addressing these limitations may involve post-processing techniques such as non-maximum suppression (NMS) refinement, or leveraging hybrid architectures that combine object detection with context-aware modules. Incorporating attention-based mechanisms or integrating spatial relationship constraints could also enhance symbol differentiation and improve model robustness in high-clutter scenarios.

The model's capacity to balance recall, accuracy is illustrated by the evaluation of confidence level shown in Fig. 7, which provided valuable information on the model's performance at various confidence levels for all trained symbols

1.00; 100%, **b** Precision–confidence curve, **c** Recall–confidence curve, **d** Precision–recall curve

simultaneously. The model's performance was validated using the F1 score, precision (P), and recall (R).

Figures 8 and 9 illustrate the outcomes of testing that was conducted directly on the engineering drawing to evaluate the prediction accuracy for the trained GD&T symbols.

(2) Faster R-CNN With the mean average precision (mAP) of 0.930 with processing speed of 75 ms per image, it excels in detecting geometric features, at a higher computational cost compared to YOLOv11.

The Faster R-CNN model was trained on the same dataset as YOLOv11, comprising engineering drawings featuring geometric tolerancing symbols. The dataset was originally in YOLO format but was transformed to COCO format to

conform to the Faster R-CNN architecture. A ResNet-50 backbone, ImageNet, was employed for feature extraction, and the model was fine-tuned to enhance the detection of small and geometrically complex symbols by modifying anchor ratios and scales. During the training phase, Faster R-CNN exhibited a notable enhancement in classification accuracy, especially in differentiating between the foreground (GD&T symbols) and the background (non-symbolic areas). At the conclusion of training, the foreground classification accuracy above 90%, demonstrating the model's efficacy in symbol identification (Fig. 10a). The initial training phases demonstrated an elevated incidence of false negatives owing to the model's challenges in identifying less conspicuous symbols. As training advanced, the incidence of false negatives markedly diminished, indicating enhanced memory and a reduction in the omission of symbols (Fig. 10b). Moreover, the bounding box regression loss constantly decreased, signifying improved accuracy

in localizing detected symbols, which is essential for precisely defining symbol borders, even in intricate engineering drawings (Fig. 10c).

This improvement led to decrease in background sampling errors (Fig. 10d) and an augmentation of positive anchor regions, hence enhancing symbol localization and diminishing noise (Fig. 10e). The model's detection efficacy was assessed utilizing criteria including mean average precision (mAP), precision, recall, and confidence thresholds. The Faster R-CNN model attained an overall mAP of 0.930, marginally inferior to YOLOv11's mAP of 0.936. Faster R-CNN surpassed YOLOv11 in the detection of intricate symbols, including Position and Perpendicularity, achieving average precision values of 0.820 and 0.850, respectively, whereas YOLOv11 attained values of 0.792 and 0.823 (Fig. 10f). Furthermore, Faster R-CNN's two-stage architecture facilitates more accurate detection on further more training samples. Its superior handling of fine-grained

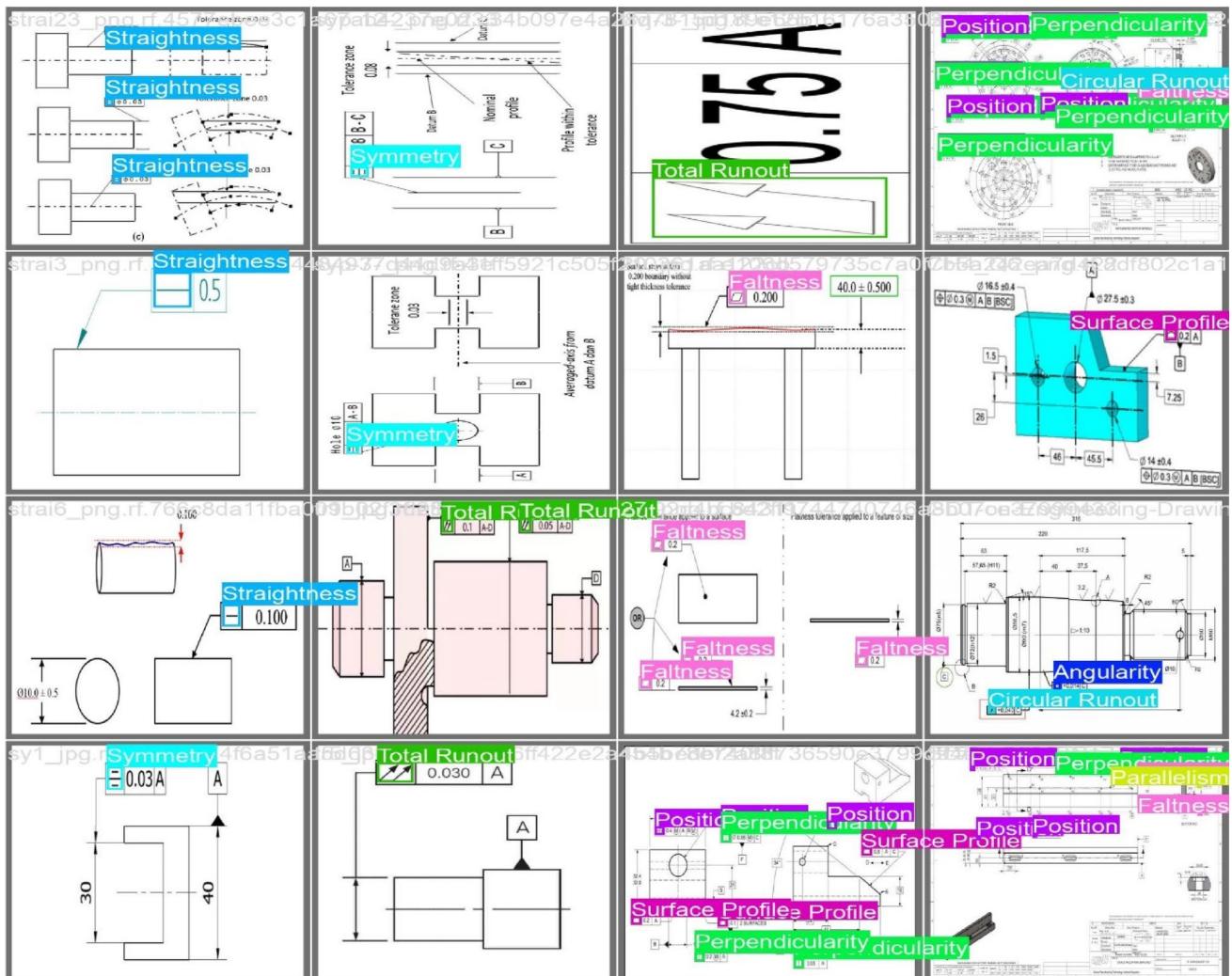


Fig. 8 Detection of GD&T symbols in an engineering drawing. The highlighted symbols represent those identified by the model during the recognition process, ability to accurately detect and localize geometric tolerancing symbols

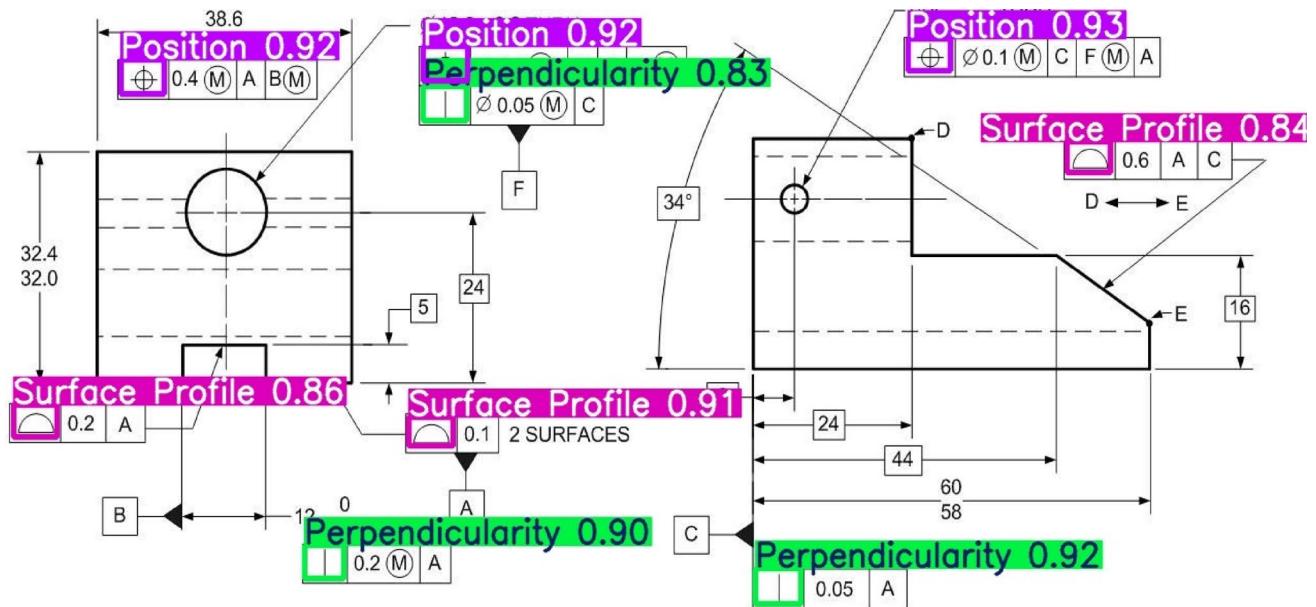


Fig. 9 Identification of GD&T symbols with corresponding prediction scores. The highlighted symbols show the model's successful detection, prediction scores reflecting the model's confidence in accurate identifying

details makes it ideal for applications in precision-driven industries, such as automotive and aerospace, where accurate symbol recognition is essential for quality control, design verification and few other applications.

The model's recall at low confidence thresholds remained elevated (exceeding 95%), indicating its strength in identifying symbols with differing degrees of prominence (Fig. 10f). The detection results on engineering drawings indicated that the Faster R-CNN model proficiently detected and localized symbols with high accuracy, which is crucial for automating GD&T analysis (Fig. 10a, f). The model's prolonged training led to significant decreases in misclassification and improved detection confidence for difficult symbols. Compared to YOLOv11, Faster R-CNN demonstrated superior localization accuracy and recall, particularly for geometrically intricate symbols, while YOLOv11 excelled in real-time detection, rendering it ideal for applications necessitating rapid processing. The investigation indicated that YOLOv11 excels in swift evaluations, but Faster R-CNN is preferable for meticulous analytical jobs requiring high precision. The results indicate that a hybrid methodology, combining the YOLOv11 with the precision of Faster R-CNN, may offer an optimal option for GD&T analysis processes as the results depict the efficiency to detect. As shown in Fig. 11 presents the detection results of the Faster R-CNN model applied to an engineering drawing, specifically focusing on the identification of GD&T (geometric dimensioning and tolerancing) symbols. The figure visually demonstrates how the model, after undergoing training and evaluation, successfully detects and localizes.

As illustrated in Fig. 12, the results indicate a significant reduction in misclassification errors and improved localization accuracy, particularly for challenging and less prominent symbols. This improvement is attributed to the model's refined feature extraction capabilities and optimized region proposal network, which effectively distinguishes GD&T symbols from complex backgrounds.

(3) *RetinaNet* Achieving a mean average precision (mAP) of 0.928 with processing speed of 45 ms per image, RetinaNet effectively balances detection accuracy and speed. Its use of focal loss allows to perform well in class imbalance conditions, making it particularly effective for detecting less frequent geometric symbols, though at a higher inference time compared to YOLOv11.

The RetinaNet model was trained on the same dataset as YOLOv11 and Faster R-CNN and validated, by employing engineering drawings featuring geometric tolerancing symbols. The model design utilized a ResNet-50 backbone pre-trained on ImageNet, augmented by a Feature Pyramid Network technique (FPN) for improved multi-scale feature detection in real-time. The classification loss shown a significant early decrease from 0.45 to roughly 0.25 during the first 200 epochs, then stabilizing around 0.18 (Fig. 13a). This pattern shows swift initial acquisition of distinguishing characteristics for GD&T symbols, followed by meticulous adjustment of categorization boundaries. The regression loss exhibited analogous convergence patterns, commencing at 0.15 and settling at 0.13 after 800 epochs, with a

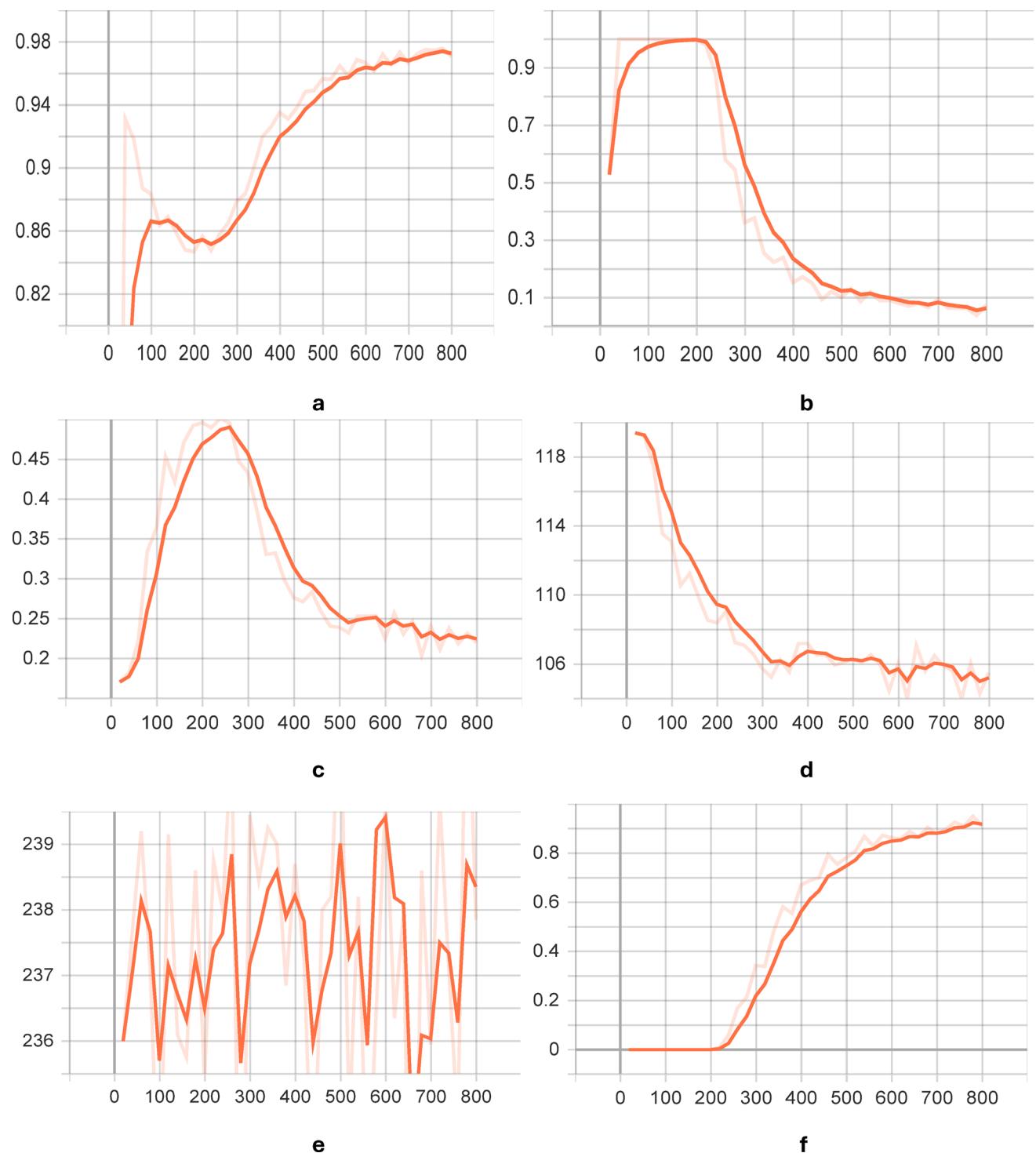


Fig. 10 Training and evaluation performance of Faster R-CNN on the detection of GD&T symbols. **a** Foreground classification accuracy—classification accuracy of 90% on foreground classes, highlighting its strong capability in recognizing individual symbols, **b** False negatives reduction—false negatives reduced significantly after 500 epochs, suggesting an improvement in recall over time, **c** Bounding box regression loss—bounding box regression loss decreased steadily, indicating enhanced localization of detected symbols, **d** Background sampling errors—decrease in background sampling errors, resulting from optimi-

zed region proposal sampling, led to improved localization and reduced noise. **e** Positive anchor region augmentation—Increased positive anchor regions contributed to better symbol localization and improved detection performance, **f** Precision of complex symbol detection—Faster R-CNN model outperformed YOLOv11 in detecting intricate symbols, such as Position and Perpendicularity, with higher average precision values, highlighting its enhanced ability to handle complex spatial relationships

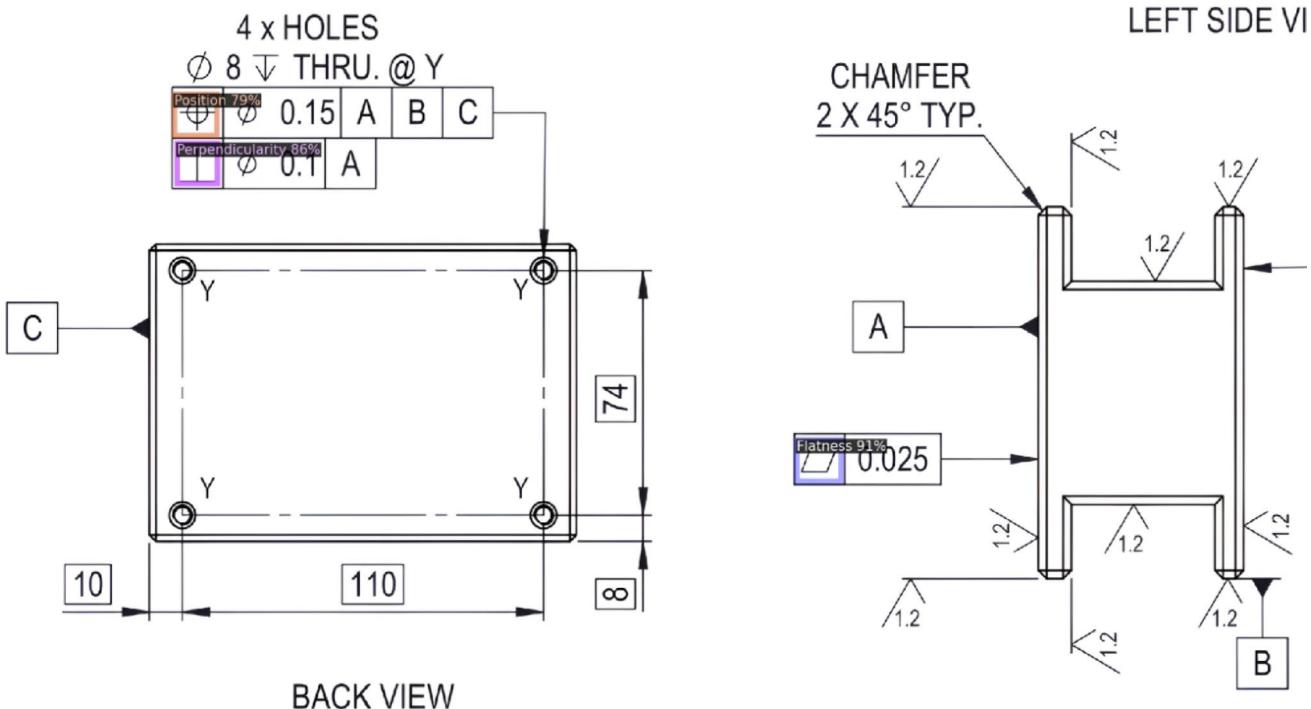
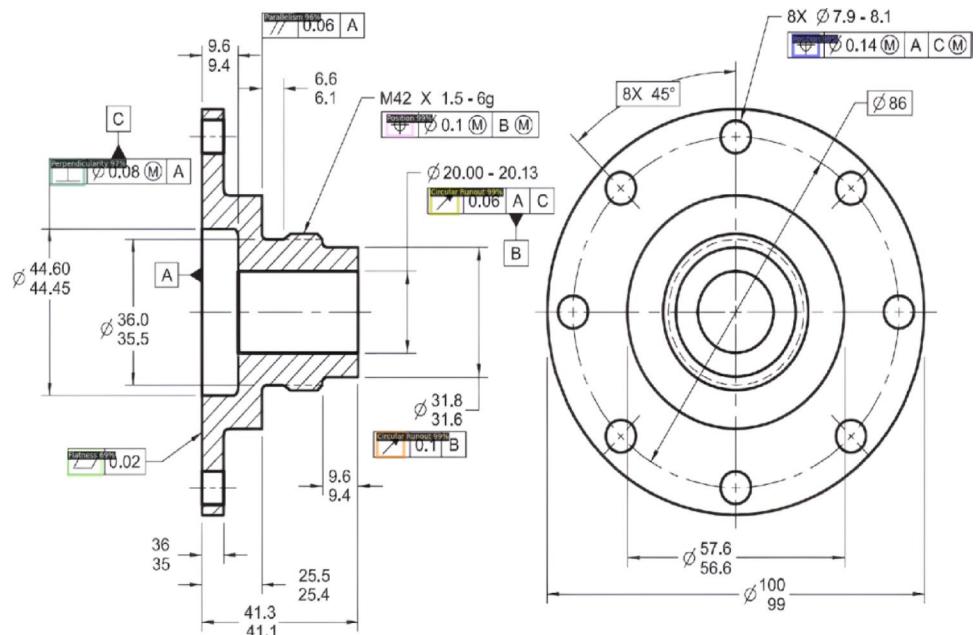


Fig. 11 The detection results of the Faster R-CNN model applied to an engineering drawing, where GD&T symbols are accurately localized and identified post training process and accounting for the evaluation of precision curve and background sampling error

Fig. 12 An additional engineering drawing with GD&T symbols detected by the trained Faster R-CNN model. This image highlights the model's enhanced detection performance, with significant reductions in misclassification and improved localization accuracy, particularly for challenging and less prominent symbols



significant spike towards the conclusion of training indicating possible learning rate modifications (Fig. 13b). The model's performance measures indicated notable dynamics regarding feature learning and detection ability. The anchor optimization process, assessed via positive anchor ratios, exhibited significant variability between 41 and 49% during training (Fig. 13c), reflecting adaptive modifications to differing symbol scales and aspect ratios. The total loss

trajectory exhibited steady improvement, decreasing from an initial value of 1.3 to stabilize at approximately 0.3 after 600 epochs (Fig. 13d), indicating effective convergence of both classification and regression elements. RetinaNet attained stable outcomes with an overall mAP of 0.928, in comparison to both YOLOv11 and Faster R-CNN. The focal loss technique shown particular efficacy in preserving elevated detection accuracy for infrequent symbols, while

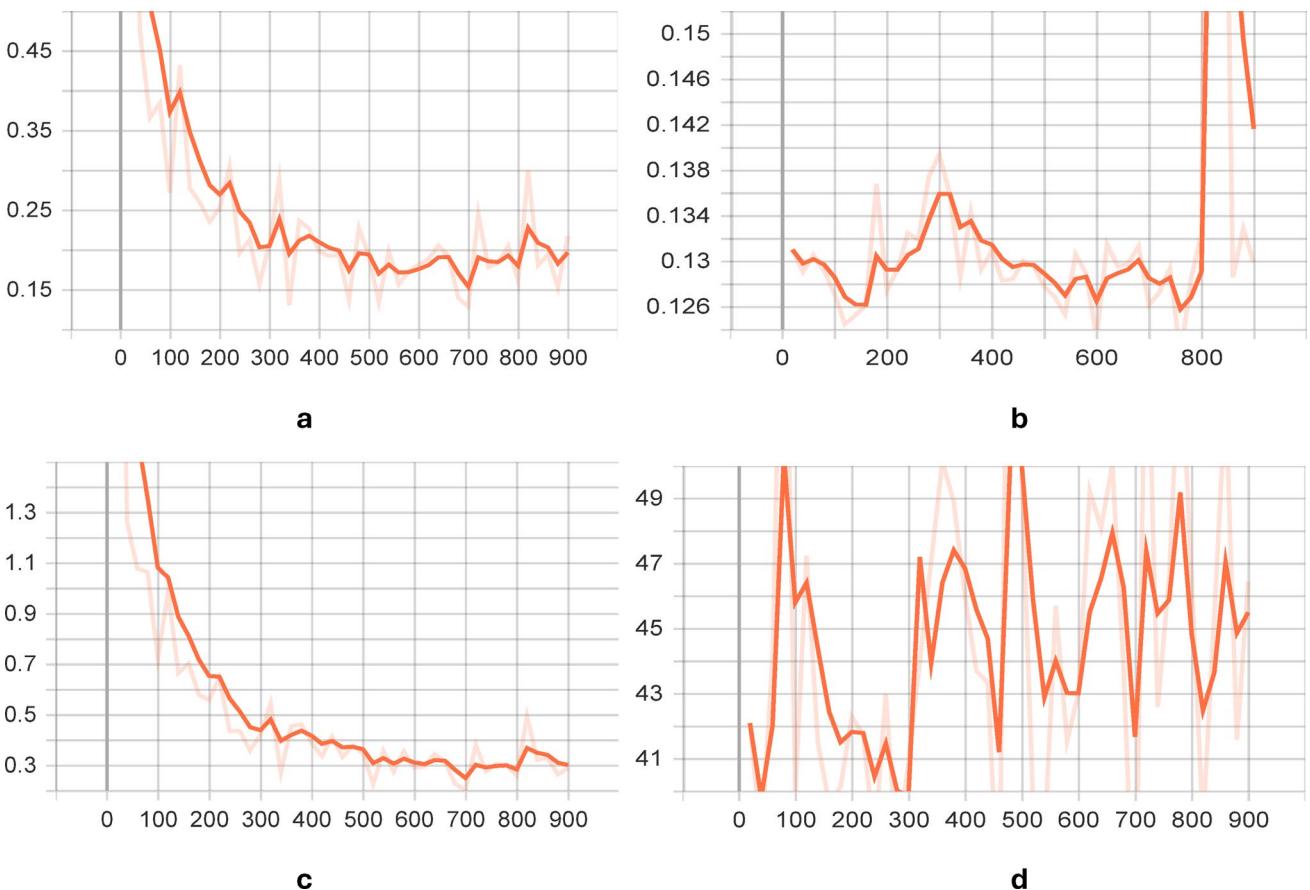


Fig. 13 Training and Evaluation Performance of RetinaNet. **a**Classification loss versus epochs: Shows rapid initial learning with a steep drop from ~0.45 to ~0.25 within the first 200 epochs, then stabilizing around 0.18, indicating efficient acquisition of distinguishing GD&T features. **b**Regression loss versus epochs: Illustrates a steady decline from 0.15 to ~0.13, showing convergence with a minor spike near the

end, likely due to learning rate adjustments. **c**Positive anchor ratio versus epochs: Fluctuates between 41 and 49%, demonstrating adaptive scaling of anchors based on symbol characteristics and aspect ratios. **d**Total loss versus epochs: Declines from 1.3 to ~0.3 over 600+ epochs, reflecting convergence of the combined classification and regression objectives

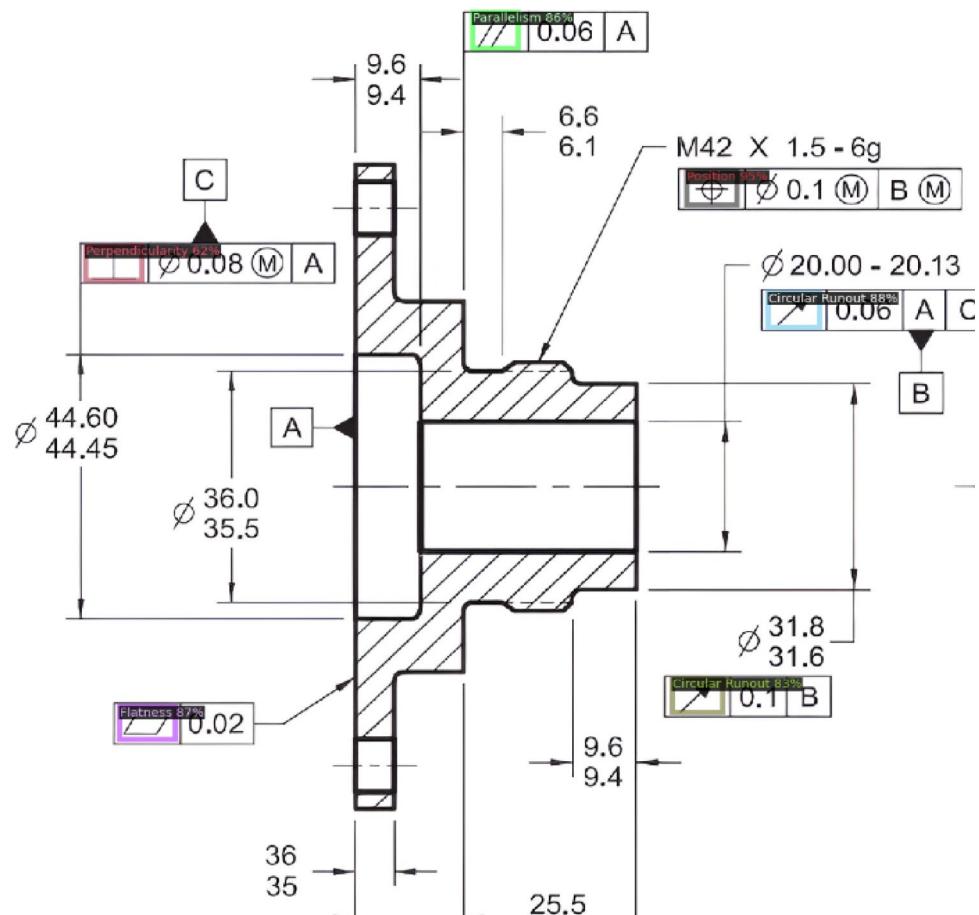
inhibiting the predominant background class from overshadowing the training process.

RetinaNet's performance attributes are in between the evaluation of YOLOv11 and the accuracy of Faster R-CNN. Although its inference time (45 ms per image) was slower than YOLOv11 (22 ms), it was considerably faster than Faster R-CNN (75 ms), while achieving similar accuracy levels. This renders RetinaNet especially appropriate for applications necessitating both adequate speed and elevated accuracy in GD&T symbol identification.

In conclusion, this paper provides a comprehensive evaluation of three advanced object detection architectures—YOLOv11, Faster R-CNN, and RetinaNet—for the identification and classification of GD&T symbols in engineering drawings. As shown in Fig. 14, RetinaNet effectively detects GD&T symbols in an engineering drawing, demonstrating its ability to maintain high accuracy across varying symbol scales and orientations. The model's performance is largely attributed to its focal loss mechanism,

which enhances detection confidence by addressing class imbalance and reducing false positives. This capability makes RetinaNet particularly suitable for GD&T analysis, where symbols may appear in diverse sizes and orientations as indicated in Fig. 14, within complex engineering drawings. The comparative analysis includes detection accuracy, computational efficiency, and practical application across diverse drawing contexts. All three models demonstrated exceptional capabilities, with YOLOv11 achieving the highest mean average precision (mAP) of 0.936, followed by Faster R-CNN at 0.930 and RetinaNet at 0.928. While the mAP values are similar, each model exhibits distinct strengths. YOLOv11 excels in processing speed, making it ideal for real-time applications and high-throughput analysis, though it may struggle with detecting smaller or more complex symbols. Faster R-CNN, on the other hand, delivers superior precision in detecting intricate symbols like Position and Perpendicularity, thanks to its two-stage architecture. This enables more comprehensive

Fig. 14 Detection results demonstrating RetinaNet's performance on an engineering drawing with GD&T symbols. The detection results highlight RetinaNet's particular strength in maintaining high accuracy across different symbol scales and orientations, while effectively managing detection confidence through its focal loss mechanism



feature analysis, particularly for geometrically complex symbols. RetinaNet, with its Feature Pyramid Network (FPN), achieves similar performance for complex symbols while ensuring more uniform detection across different symbol scales. Faster R-CNN's computational cost is higher, but its superior localization accuracy and ability to detect complex symbol relationships make it the best choice for tasks requiring high precision. RetinaNet offers a balanced approach, with good detection performance and moderate computational requirements. Its use of focal loss helps address class imbalance, making it especially effective for handling drawings with varying symbol densities. The training dynamics of each model revealed unique patterns. YOLOv11 converged quickly, making it efficient for training iterations, while Faster R-CNN showed gradual improvements in classification accuracy and bounding box regression, leading to better localization. RetinaNet demonstrated stable convergence, maintaining consistent performance across different symbol frequencies and scales. From an application perspective, YOLOv11 is optimal for real-time analysis and automated quality control, where speed is critical. Faster R-CNN is better suited for tasks that require precise detection and localization of complex symbols, such as design verification. RetinaNet is a versatile option

for general-purpose GD&T analysis, providing a good balance of accuracy and speed across a range of drawing complexities. In summary, the ideal model depends on the specific requirements of the application. In high-throughput production settings, YOLOv11's speed may outweigh the slight accuracy advantages of Faster R-CNN. For precision-critical applications, such as component design verification, Faster R-CNN's superior accuracy justifies its higher computational cost. RetinaNet's balanced performance makes it a strong candidate for more general applications, offering reliable detection across a variety of symbol types and drawing complexities. While mean average precision (mAP) and inference time provide a broad understanding of model performance, real-world feasibility also depends on hardware constraints such as memory usage, GPU requirements, and model size. Table 1 summarizes the computational footprint of each model, including average GPU memory consumption and peak VRAM usage measured during inference using an NVIDIA RTX 4090.

This comparison highlights the computational trade-offs decision-makers must consider. While YOLOv11 is the fastest and consumes the least memory, it slightly compromises on symbol-level precision, especially in complex drawings. Faster R-CNN offers the best accuracy in detecting intricate

Table 1 Comparison of detection performance and computational cost for YOLOv11, Faster R-CNN, and RetinaNet models on the GD&T symbol detection task

Model	mAP@0.5	Avg inference time (ms/img)	Model size (MB)	Peak GPU memory (GB)
YOLOv11	0.936	12.8	128	5.1
Faster R-CNN	0.930	45.2	171	7.8
RetinaNet	0.928	23.5	156	6.5

Metrics include mean Average Precision (mAP@0.5), average inference time per image (in milliseconds), model size (in megabytes), and peak GPU memory usage during inference (in gigabytes). These values were recorded on an NVIDIA RTX 4090 GPU. The results highlight the trade-off between accuracy, speed, and hardware requirements, which is critical for selecting models in real-time or resource-constrained industrial applications

GD&T features, but at the expense of higher memory usage and longer inference times. RetinaNet strikes a middle ground, balancing detection quality with moderate GPU demands, making it well-suited for deployments with limited hardware availability.

Throughout testing, each model demonstrated a degree of robustness to real-world noise and variations in CAD drawings. Data augmentation techniques such as Gaussian noise injection, symbol rotation, and scale jittering helped simulate challenging drawing conditions and contributed to generalization performance. Notably, YOLOv11 maintained high accuracy on symbols embedded within cluttered regions or overlapping dimension lines, while Faster R-CNN was more resilient to occlusions and low contrast. However, performance may vary across datasets with different symbol styles, resolutions, or annotation standards. Thus, future work should explore domain adaptation or transfer learning techniques to improve model generalization on diverse datasets and enhance their reliability for industrial deployment. A detailed error analysis revealed model-specific limitations. YOLOv11 showed occasional false positives in visually dense areas due to its one-stage architecture, favoring recall over precision. Faster R-CNN, though highly accurate, had a higher false-negative rate for small symbols and required more inference time. RetinaNet, while balanced overall, sometimes confused visually similar symbols due to limited intra-class differentiation. These insights are essential for real-world integration, especially when selecting models based on production goals—speed, precision, or versatility. Practical deployment in Industry 4.0 environments, such as integration into Quality Management Systems (QMS), will benefit from model-specific tuning and hybrid approaches that combine the strengths of different detectors.

Conclusions

All three models—YOLOv11, Faster R-CNN, and RetinaNet—were trained on a custom GD&T symbol dataset to evaluate their effectiveness in identifying and classifying symbols in mechanical engineering drawings. YOLOv11, with its rapid inference capability, proved advantageous for real-time detection scenarios, making it suitable for applications where speed is critical. Faster R-CNN, despite being slower, delivered the highest accuracy, especially in identifying complex or intricate symbols, making it ideal for high-precision quality control applications where accuracy is paramount. RetinaNet achieved a balanced performance, providing both reasonable speed and reliable accuracy, making it a viable option for tasks that require a compromise between inference speed and detection precision. While these results demonstrate promising performance on the custom dataset, the generalizability of the models to other domains remains a potential limitation. Variations in drawing styles, symbol conventions, or noise levels across different datasets could impact model accuracy. Future work should explore evaluating these models on more diverse or publicly available datasets and investigate domain adaptation or transfer learning techniques to ensure robustness in real-world deployment. After evaluating the performance of all three models, YOLOv11 emerged as the best choice for GD&T symbol recognition. Its superior inference speed, coupled with its ability to maintain a high level of accuracy, makes it the most effective model for real-time detection and automated quality control systems. This model's performance, combined with the possibility of further fine-tuning with diverse datasets, ensures it can adapt to a range of real-world conditions, enhancing productivity, precision, and adherence to manufacturing standards. In real-world manufacturing environments, the trained models—particularly YOLOv11—can be integrated into Quality Management System (QMS) software to automate the detection and validation of GD&T symbols in engineering drawings. This integration can enhance inspection workflows by automatically flagging missing, incorrect, or non-standard symbols during the design review or production documentation stages. For instance, as part of a digital quality check pipeline, the model can verify compliance of technical drawings with GD&T standards before they reach the shop floor, reducing the risk of downstream production errors. However, implementing such systems presents challenges, including handling diverse drawing formats, variations in symbol quality, and ensuring accurate recognition under low-resolution scans or print distortions. The integration of YOLOv11 into any application to detect GD&T symbols can help quality inspections in Industry 4.0, paving the way

for future advancements in automated engineering design, production, and quality assurance.

Future scope

Further developments could focus on refining and upgrading to the newest versions of model to improve both speed and accuracy, potentially by exploring newer object detection techniques. Additionally, integrating the GD&T symbol detection system into a custom developed real-time quality check software throughout production. Future model updates, using more diverse datasets and enhanced training strategies, would further improve detection accuracy. This approach can also support integration with Quality Management Systems (QMS), promoting automation and ensuring higher standards of precision and efficiency in manufacturing.

In terms of real-world deployment, edge computing platforms such as NVIDIA Jetson Nano, Jetson Xavier NX, and Jetson Orin offer promising solutions for running YOLOv11 models efficiently in manufacturing environments. These devices support GPU-accelerated inference with low latency and are suitable for real-time inspection tasks directly on the shop floor. Running YOLOv11 on such embedded systems minimizes the need for external servers, reducing system complexity and enabling localized processing of CAD drawings. This is especially beneficial in resource-constrained or bandwidth-limited settings. A feasibility test using Jetson Xavier NX demonstrated real-time inference capabilities (~30 FPS with TensorRT optimization), validating the practicality of deploying the model in an on-device industrial setup. Future work will explore optimizing YOLOv11 deployment pipelines for embedded hardware to further reduce latency and power consumption, making this solution scalable across different manufacturing lines.

Building upon the strong performance of YOLOv11, future research can investigate newer object detection paradigms such as DETR (DEtection TRansformer) or RT-DETR models that have demonstrated superior capabilities in learning complex object relationships, which are essential for high-density technical drawings. Furthermore, integrating these models into real-time QMS software could allow not only symbol detection but also automatic extraction of dimensions, tolerances, and feature control frames from engineering drawings, improving the end-to-end automation of quality checks in smart manufacturing environments.

Acknowledgements The authors would like to thank the Centre of Smart Manufacturing Cell, CMTI, India, National Institute of Technology (NITK) Surathkal, for their support and resources provided for this study.

Author contributions T Narendra Reddy: Conceptualization, methodology, data curation, writing—original draft, and project administration. Nachappa P P: Model development, training, and validation. Nagasiri M: Dataset preparation and annotation, training, and validation. Prakash Vinod: Writing—review and editing, and validation. Nitesh Kumar: Writing—review and editing, and validation. Dr. Mervin A. Herbert: Supervision and project management. Dr. Shrikantha S. Rao: Methodological guidance and manuscript review.

Funding Open access funding provided by Manipal Academy of Higher Education, Manipal. This research was not supported by any funding body and had no role of external support in the design of the study, data collection, analysis, or interpretation of results.

Availability of data and material The data and material used in this study are available upon reasonable request. The dataset of engineering drawings containing GD&T symbols, as well as the trained models, can be provided for research purposes. Requests for access should be directed to the corresponding author.

Declarations

Conflict of interest The authors declare that they have no competing interests related to this study. There are no financial or personal conflicts of interest that could influence the research results or interpretations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Aggarwal, Kumar, V., & Gupta, R. (2023). Object detection based approaches in image classification: A brief overview. In *2023 IEEE Guwahati subsection conference (GCON), India* (pp. 1–5). <https://doi.org/10.1109/GCON58516.2023.10183609>
- Ajiga, D., Okeleke, P. A., Folorunsho, S. O., & Ezeigweneme, C. (2024). The role of software automation in improving industrial operations and efficiency. *International Journal of Engineering Research Updates*, 7(1), 022–035.
- Ali, A. S. H. R., Ali, A. M. S., Amin, Y. W. R., & Ali, S. H. R. (2024). The important role of GD&T in mechanical drawing, design and manufacturing for students of engineering institutes. In *SAE technical paper 2024-01-2052*. <https://doi.org/10.4271/2024-01-2052>
- Azamfirei, V., Psarommatis, F., & Lagrosen, Y. (2023). Application of automation for in-line quality inspection, a zero-defect manufacturing approach. *Journal of Manufacturing Systems*, 67, 1–22. <https://doi.org/10.1016/j.jmsy.2022.12.010>
- Baciu, I.-L., Bruzzo, C., Iazurlo, R., Rizzi, G., & Tarantino, A. (2023). A modern approach to dimensional inspection by implementing annotated 3D CAD model in a digital manufacturing environment.

- In 2023 IEEE international conference on metrology for extended reality, artificial intelligence and neural engineering (pp. 31). IEEE. <https://doi.org/10.1109/MetroXRAINE58569.2023.10405823>
- Bengamra, S., Mzoughi, O., Bigand, A., & Zagrouba, E. (2024). A comprehensive survey on object detection in visual art: Taxonomy and challenge. *Multimedia Tools and Applications*, 83(5), 14637–14670.
- Chaturvedi, R. P., & Ghose, U. (2023). Evaluation of small object detection in scarcity of data in the dataset using Yolov7. In 2023 International conference on disruptive technologies (ICDT) (pp. 336–339).
- Helle, R. (2023). *Intelligent symbol attributes extraction from engineering drawings*. Master's thesis, University of Turku, Department of Computing, Software Engineering, University of Turku.
- Jiang, J. (2023). Research and application of object recognition algorithm in architectural engineering drawing based on deep learning. In 2023 Asia-Pacific conference on image processing, electronics and computers (IPEC), Dalian (pp. 121–124). <https://doi.org/10.1109/IPEC57296.2023.00029>
- Kashevnik, A., Ali, A., & Mayatin, A. (2023a). AI-based method for frame detection in engineering drawings. In 2023 International Russian smart industry conference (SmartIndustryCon) (pp. 225–229).
- Kashevnik, A., Shilov, N., Teslya, N., Hasan, F., Kitenko, A., Dukareva, V., Abdurakhimov, M., Zingarevich, A., & Blokhin, D. (2023b). An approach to engineering drawing organization: Title block detection and processing. *IEEE Access*, 1, 1–1. <https://doi.org/10.1109/ACCESS.2023.3244603>
- Katyayani, Bhardwaj, K., & Poongodi, T. (2023). Deep learning approach for multi-object detection using YOLO algorithm. In 2023 6th international conference on contemporary computing and informatics (IC3I) (pp. 689–693).
- Kocharian, T., Burns, J., & Manoharan, S. (2024). Design and development of a geometric dimensioning and tolerancing course. In ASME international mechanical engineering congress and exposition, Paper No. IMECE2023-112112, V008T09A002 (p. 12). <https://doi.org/10.1115/IMECE2023-112112>
- Lin, Y.-H., Ting, Y.-H., Huang, Y.-C., Cheng, K.-L., & Jong, W.-R. (2023). Integration of deep learning for automatic recognition of 2D engineering drawings. *Machines*, 11(8), 802. <https://doi.org/10.3390/machines11080802>
- Liu, C., & Xi, C. (2023). A multi-strategy integrated optimized YOLOv11 algorithm for object detection. In 2023 IEEE 6th international conference on automation, electronics and electrical engineering (AUTEEE), Shenyang, China (pp. 1–5). <https://doi.org/10.1109/AUTEEE60196.2023.10407167>
- Mohamed, A. (2023). Enhancing geometrical dimensioning and tolerancing proficiency: A practical approach to training and learning. In The proceedings of the 6th international conference on future of education (Vol. 6, No. 1, Article 6108). Faculty of Technology and Applied Sciences, Open University Malaysia. <https://doi.org/10.17501/26307413.2023.6108>
- Peng, Y., Du, K., Hou, G., Li, S., & Huang, X. (2024). A novel assembly-oriented measurement datum transformation and tolerance reallocation method. *International Journal of Advanced Manufacturing Technology*, 131(9–12), 4281–4295. <https://doi.org/10.1007/s00170-024-13152-3>
- Petrucchioli, A., Pini, F., & Leali, F. (2022). Model-based approach for optimal allocation of GD&T. In C. Rizzi, F. Campana, M. Bici, F. Gherardini, T. Ingrassia, & P. Cicconi (Eds.), *Design tools and methods in industrial engineering II ADM 2021. Lecture notes in mechanical engineering*. Cham: Springer. https://doi.org/10.1007/978-3-030-91234-5_28
- Psaromatis, F., May, G., Azamfirei, V., & Konstantinidis, F. (2024). Optimizing efficiency and zero-defect manufacturing with in-process inspection: Challenges, benefits, and aerospace application. *Procedia CIRP*, 108, 278–283. <https://doi.org/10.1016/j.procs.2024.02.102>
- Saihi, A., Awad, M., & Ben-Daya, M. (2023). Quality 4.0: Leveraging industry 4.0 technologies to improve quality management practices—A systematic review. *International Journal of Quality and Reliability Management*, 40(2), 201–224. <https://doi.org/10.1080/IJQRM.06.2021.0276>
- Schlagenhauf, T., Netzer, M., & Hillinger, J. (2022). Text detection on technical drawings for the digitization of brown-field processes. In *Proceedings of the 16th CIRP conference on intelligent computation in manufacturing engineering*, CIRP ICME '22. CIRP.
- Souza, C. D. (2024). *Design for inspection: Optimizing medical device designs for manufacturing and inspection through case studies using geometric dimensioning and tolerancing (GD&T) in accordance with ASME Y14.5 2018 standards*. SSRN. <https://ssrn.com/abstract=4889073> or <https://doi.org/10.2139/ssrn.4889073>
- Sui, J., Chen, D., Zheng, X., & Wang, H. (2024). A new algorithm for small target detection from the perspective of unmanned aerial vehicles. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2024.365584>
- Tandler, W. (2014). Corporate GD&T implementation: When properly implemented, GD&T makes a big contribution to corporate profitability. *Quality*, 53(8), 22.
- Tornincasa, S. (2024). The GPS and GD&T language. Springer tracts in mechanical engineeringIn S. Tornincasa (Ed.), *Technical drawing for product design* (pp. 75–98). Cham: Springer. https://doi.org/10.1007/978-3-031-51187-5_4
- Yang, Y., He, S., Tang, L., Dong, C., Tian, W., & Li, R. (2023). Optimizing small target detection in engineering drawings: A scale-aware slicing approach. In 2023 5th International conference on ML, big data and business intelligence (pp. 1–6). <https://doi.org/10.1109/MLBDBI60823.2023.10482360>
- Yasmine, G., Maha, G., & Hicham, M. (2023). Overview of single-stage object detection models: From YOLOv1 to Yolov7. In 2023 International wireless communications and mobile computing (IWCMC) (pp. 1579–1584).
- Zhou, J., & Hartman, N. W. (2023). A framework for model-based visual inspection: A case study of bottle dimensional measurements in the plastics industry. In *Proceedings of the 2023 purdue conferences—CAD conference*. <https://doi.org/10.14733/cadconf.P.2023.74-79>
- Zupan, S., & Kunc, R. (2024). Overview of principles and rules of geometrical product specifications according to the current ISO standards. *Strojniški Vestnik: Journal of Mechanical Engineering*. <https://doi.org/10.5545/sv-jme.2023.753>