IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 2, April 2025, pp. 1345~1354
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i2.pp1345-1354

Deep learning-based classifier for geometric dimensioning and tolerancing symbols

Laxmi Bewoor[1], Anand Bewoor[2], Pravin P. Hujare[3], Praveen Rathod[3], Vedant Yetekar[3], Shrish Dollin[3]
[1]Department of Computer Engineering, Vishwakarma Institute of Information Technology, Pune, India
[2]Department of Mechanical Engineering, Cummins College of Engineeringfor Women, Pune, India
[3]Department of Mechanical Engineering, Vishwakarma Institute of Information Technology, Pune, India

Article Info

Article history:

Received Feb 22, 2024
Revised Oct 25, 2024
Accepted Nov 14, 2024

Keywords:

Deep learning
Engineering applications
Geometric dimensioning and tolerancing symbols
Object detection
YOLOv8

ABSTRACT

This research investigates the recognition of geometric dimensioning and tolerancing (GD&T) symbols using a deep learning model for object detection. GD&T, playing a pivotal role in engineering and manufacturing, provides essential specifications for product design and production. Manual processes for GD&T are often time-consuming and error prone. The study demonstrates outstanding accuracy in automating GD&T symbol recognition in engineering applications using YOLOv8. A carefully curated dataset, encompassing a wide range of GD&T symbols, was employed for training and evaluating the model. The YOLOv8 architecture, renowned for its robust performance, was meticulously fine-tuned to cater to the specific requirements of GD&T symbol detection. This research not only addresses the challenges in manual GD&T processes but also showcases practical implications for improved quality control and streamlined engineering workflows. By automating GD&T symbol recognition, this study contributes to the efficiency and precision crucial in the engineering and manufacturing domains.

Corresponding Author:

Laxmi Bewoor
Department of Computer Engineering, Vishwakarma Institute of Information Technology
2/3/4, Kondhwa (Bk.), Pune, India
Email: laxmi.bewoor@viit.ac.in

1. INTRODUCTION

Geometric dimensioning and tolerancing (GD&T) is a standardized language of symbols and notations used to communicate the design intent for the shape, size, and form of manufactured parts. It plays a pivotal role in engineering and manufacturing by establishing clear and unambiguous specifications for product quality and functionality [1]. GD&T symbols, embedded in engineering drawings, convey crucial information about the permissible variations in a product's dimensions, ensuring that manufactured components meet the design requirements and function as intended. The symbols and notations of GD&T are defined in the American Society of Mechanical Engineers (ASME) Y14.5 standard. ASME Y14.5 is the most widely used GD&T standard in the world [2]. The purpose of GD&T is to ensure that manufactured parts meet the design intent of the engineer. GD&T does this by defining the permissible variations in the geometry of a part. These variations are called tolerances. Tolerances are specified on engineering drawings using a combination of symbols and notations. GD&T symbols are used to specify the form, dimension, and tolerance of a feature. The form of a feature is its shape. The dimension of a feature is its size. The tolerance of a feature is the amount of variation that is allowed in the form, dimension, or both of a feature. There are 14 basic GD&T symbols. These symbols are used to specify the form, dimension, and tolerance of features. The 14 basic GD&T symbols are show in Table 1.

Despite its significance, manual detection and classification of GD&T symbols pose several challenges. The process is often time-consuming, labour-intensive, and prone to human error. Manual inspection can lead to inconsistencies in interpretation, inconsistencies in the application of GD&T principles, and ultimately, product defects [3]. Moreover, the increasing complexity of engineering designs and the growing volume of technical documentation further exacerbate the limitations of manual GD&T symbol recognition.

Table 1. Standard ASME GD&T symbols

Symbol	Characteristics	Category
⏤	Straightness	Form
▱	Flatness	
○	Circulality	
⌭	Cylindrycity	
⌒	Profile of a line	Profile
⌓	Profile of surface	
∠	Angularity	Orientation
⊥	Perpendycularity	
//	Parallelism	
⌖	Position	Location
◎	Concentricity	
⌯	Symmetry	
↗	Circular runout	Runout
⌰	Total runout	

In light of these challenges, this research aims to explore the application of deep learning techniques, specifically focusing on YOLOv8, for the automated detection and classification of GD&T symbols. Deep learning algorithms have demonstrated remarkable capabilities in pattern recognition, image classification, and object detection, making them well-suited for automating the task of GD&T symbol recognition [4]. By leveraging deep learning models, we can achieve accurate, efficient, and scalable GD&T symbol identification, thereby enhancing quality control and streamlining engineering workflows.

The paper is organized as follows: section 2 describes related work related to various deep learning models in engineering drawing context. Section 3 describes the proposed methodology. Section 4 describes implementation details. Section 5 elaborates results and discussions. Section 6 concludes the paper.

2. RELATED WORK

The automated processing of engineering drawings are challenging as it relies on the precise detection and classification of symbols embedded within these documents. Conventional symbol recognition approaches primarily focus on detecting symbols but fail to infer their orientation. This limitation can lead to inaccuracies in interpreting design intent and translating drawings into manufacturing instructions. Despite this fact very little contribution has been witnessed for developing deep learning models for engineering drawing processing [5]. This section is providing a brief review of work in use of deep learning models for engineering drawing processing; however very little contribution is witnessed for the GD&T symbol detection and classification. Lin *et al.* [6] address challenges in interpreting diverse engineering drawings by introducing a recognition system using GD&T. Leveraging PyTorch, OpenCV, and YOLO, the system achieves notable accuracy rates, including 85% for view detection and 80% for text and symbol recognition.

The integration of GD&T and deep learning offers a practical solution, storing results directly in a database for efficient verification and error prevention. To address this challenge, researchers have explored keypoint-based deep learning approaches for symbol detection and orientation estimation. Faltin *et al.* [7] compared three keypoint-based models: keypoint region-based convolution neural network (R-CNN), YOLOv7-Pose, and a custom two-stage approach. Their experimental results demonstrated that keypoint-based networks outperformed traditional detection-only methods, achieving improved accuracy in both symbol detection and orientation estimation. In addition to symbol recognition, deep learning has also demonstrated its potential in recognizing line objects and flow arrows in piping and instrumentation diagrams (P&IDs). Moon *et al*. [8] proposed a deep learning-based method for recognizing lines and flow arrows in image-format P&IDs. Their method consists of three steps: preprocessing, detection, and post-processing. The preprocessing step removes the outer border and title box from the diagram. The detection step employs a deep learning model to identify continuous lines, line signs, and flow arrows. The post-processing step adjusts line types based on the detected line signs and merges recognized lines with flow arrows. Experiments demonstrated that the proposed method achieved high recognition performance, with an average precision of 96.14% and an average recall of 89.59%. Toro *et al.* [9] introduce the "eDOCr" tool, emphasizing optical character recognition (OCR) for automation in production quality control. The tool demonstrates impressive precision, recall, and character error rate, providing an effective solution for seamless integration between engineering drawings and quality control processes. Specifically, the tool achieves 90% precision and recall in detection, an F1-score of 94% in recognition, and a character error rate of 8%. Bickel *et al*. [10] address the automated recognition of symbols in principle sketches, employing deep learning networks for detection. The innovative methodology involves generating diverse training data, allowing for effective recognition in early phases of product development, offering potential cost and time savings. Unfortunately, specific numerical results for the recognition accuracy are not explicitly provided in the paper.

Elyan *et al.* [11] contribute by presenting advanced methods for symbol detection and classification, incorporating bounding-box detection and deep generative adversarial neural networks (GANs). Their approach demonstrates high accuracy in symbol recognition and classification, promising advancements in diverse applications across industries, including oil and gas, construction, and engineering. The reported results include a bounding-box detection accuracy of more than 94% and improved symbol classification with limited training examples using GAN-based methods. Collectively, these papers underscore the growing importance of deep learning in automating the interpretation and analysis of engineering drawings, paving the way for improved efficiency and accuracy in various industrial processes [12]–[17]. These advancements in deep learning have opened up new avenues for automated symbol recognition in engineering drawings. Keypoint-based deep learning approaches offer promising solutions for accurate symbol detection and orientation estimation, while deep learning-based methods for line and flow arrow recognition have the potential to streamline the automation of P&IDs.

3. METHODOLOGY

GD&T symbols have connectivity information (lines) and some form of annotation (text). However, no public dataset is available for evaluation purposes. Subsection 3.1 introduces a proposed approach for end-to-end symbols recognition from GD&T drawings. The subsequent subsection will discuss in detail the dataset acquisition and its preprocessing followed by methodology adopted for fine tuning YOLO model.

3.1. Data acquisition and pre-processing

A comprehensive dataset of engineering drawings was collected from various industrial partners. The dataset consisted of a diverse range of drawings, encompassing various engineering disciplines and symbol types. To ensure the robustness of the model, the dataset was carefully curated to include a wide range of symbol variations, including different sizes, orientations, and lighting conditions. Before feeding the data into the deep learning models, it was essential to perform pre-processing steps to enhance its quality and consistency. This involved:

- Image normalization: normalizing the images to a standard size and color space ensured uniformity and facilitated better feature extraction. The images were resized to a resolution of 1024×768 pixels for optimal performance.
- Noise reduction: applying image filtering techniques removed unwanted noise and artifacts from the drawings, improving symbol visibility and reducing potential errors during recognition [18].

3.2. Challenges in earlier approaches

In the initial stages of our research, we explored the use of the sliding window approach for object detection [19]. However, this approach proved to be extremely slow and inefficient, especially when dealing with large and complex images containing multiple objects. We then transitioned to a CNN-based model, hoping to improve detection accuracy. However, CNNs are primarily designed for single object classification tasks, and we encountered limitations in detecting multiple objects simultaneously within an image. This was particularly challenging for our task, as engineering drawings often contain multiple objects of interest, including feature control frames, symbols, tolerance values, and datums.

3.3. Addressing challenges with YOLOv8

To overcome the limitations of previous approaches, we opted for the YOLOv8 object detection model. YOLOv8 is specifically designed for real-time object detection [20] and can handle multiple objects simultaneously within an image [21]. Its ability to predict bounding boxes and class probabilities for multiple objects made it well-suited for our task. Despite the effectiveness of YOLOv8, we still encountered challenges in accurately classifying symbols. This was particularly evident for symbols with similar appearances, such as the GD&T symbol 'position'. The model tended to misclassify similar symbols, leading to inaccurate detection results.

3.4. Overcoming symbol classification challenges with specialized models

To address the symbol classification challenges, we employed a multi-stage approach using three separate YOLOv8 models, each dedicated to a specific category of objects:

- Feature control frame detection: the first YOLOv8 model was trained to identify and localize feature control frames (FCFs) within the engineering drawings. This model was trained over 300 instances of feature control frames.
- Symbol detection inside FCFs: the second YOLOv8 model was tasked with detecting symbols, tolerance values, and datums within the FCFs identified by the first model. This model was trained over 350 instances of symbols.
- Symbol type classification: the third YOLOv8 model was responsible for classifying the symbols detected by the second model. This model was trained on a dataset of images containing various types of GD&T symbols. The model was trained in over 7 symbol types (perpendicularity, surface profile, position, flatness, maximum material condition, run out, and parallelism) of 300 instances.

By utilizing specialized models for each object category, we were able to significantly improve the accuracy of symbol classification. This multi-stage approach allowed for focused processing of each object type, leading to reduced misclassifications, and enhanced overall detection performance.

3.5. Addressing challenges with low-resolution images

Due to the relatively low resolution of some of the FCFs, we found that the second and third YOLOv8 models had difficulty extracting features from the cropped images. This is because the low resolution of the FCFs resulted in a loss of fine-grained details that are crucial for accurate symbol detection and classification. To address this issue, we implemented a pre-processing step where the cropped FCF images were pasted onto a white 640×640 pixels image before being passed to the respective models. This simple modification significantly improved the feature extraction capabilities of the models and ultimately enhanced the overall accuracy of symbol recognition. The white background provided a more uniform canvas for the models to analyze, allowing them to better differentiate between the symbols and their surroundings. Additionally, the larger image size provided more context for the models to make informed decisions about symbol detection and classification. This straightforward technique effectively addressed the challenges posed by low-resolution FCFs and contributed to the overall success of our multi-stage YOLOv8 approach for symbol recognition in engineering drawings. Figure 1 illustrates FCF for enhanced feature recognition whereas Figure 2 illustrates symbol extraction from enhanced feature.

Figure 1. Preprocessing feature control frames for enhanced feature recognition
(Bounding box enclosed inside an empty canvas of 640×640)

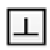

Figure 2. Preprocessing symbols for enhanced symbol classification
(Bounding box enclosed inside an empty canvas of 640×640)

3.6. Tolerance value extraction using easy optical character recognition

To accurately extract tolerance values from the detected symbols, we employed the OCR [22], [23] with EasyOCR open-source OCR library. EasyOCR is a lightweight and versatile OCR tool that can handle a wide range of font styles, sizes, and orientations. We utilized the pre-trained model provided by EasyOCR for our task. This approach eliminated the need for training a custom OCR model, saving time and computational resources. EasyOCR's ability to handle diverse font styles and orientations ensured that tolerance values were accurately extracted from a variety of engineering drawings.

4. EXPERIMENTAL RESULTS

This work is the first attempt for classifying GD&T symbols, so the experiments were carried out with different programming environments viz. Jupieter and Spyder for python 3.7 version and i7 machine. The use of Jupiter and Spyder required lot of time for training the neurl network hence the work was carried out on cloud-based infrastructure environment which is Google Colab. The details of the same are given in following section.

4.1. Training specification

All three YOLOv8 models were trained on Google Colab T4 GPUs, each equipped with an NVIDIA Tesla T4 GPU with 2,560 CUDA cores and 16 GB GDDR6 memory. The training process for each model utilized a batch size of 16 and an Adam optimizer with a learning rate of 0.001. The models were trained for 1,000, 500, and 500 epochs.

4.2. Model performance

The YOLOv8 models demonstrate state-of-the-art performance in real-time object detection, achieving high accuracy and fast inference speeds. The performance of the YOLOv8 models of the proposed work was evaluated using standard object detection metrics, including precision (P), recall (R), mean average precision (mAP)-50, and mAP50-95 [24]. These metrics were calculated (Table 2) on a separate test dataset that was not used during the training process, ensuring an unbiased evaluation of the models' generalization ability.

4.2.1. Model 1: feature control frame detection

To effectively detect FCFs within engineering drawings, we trained the initial YOLOv8 model on a dataset of 300 FCF instances. Utilizing an 80-20% split for training and validation data, we trained the model for 1,000 epochs. This resulted in a training time of approximately 3.78 hours. Figure 3 illustreates feature control frame detection by YOLO model for the test data. Figure 4 provides different curves providing details of training and validation loss. It also provides the metric values of precision and recall graphically.

Table 2. Feature control frame detection performance metrics for FCF detection

Metric	Value
Precision (P)	0.895
Recall (R)	0.945
mAP50	0.913
mAP50-95	0.856

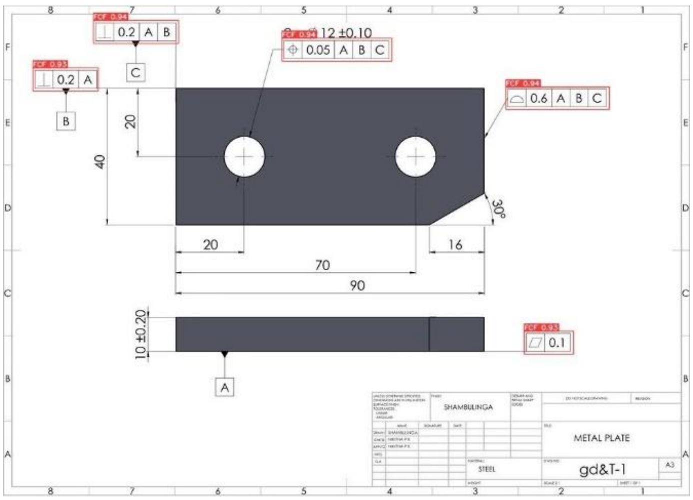

Figure 3. Feature control frames detection results by YOLO

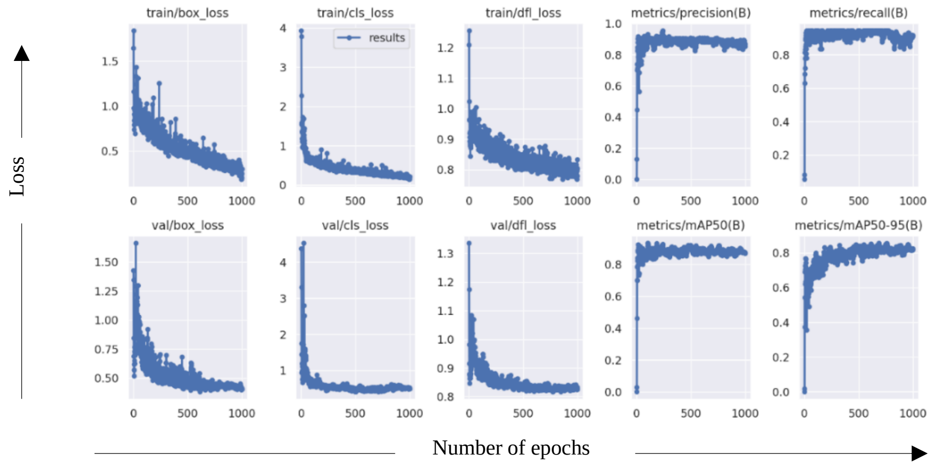

Figure 4. Feature control frames training curves

4.2.2. Model 2: feature detection within feature control frames

To detect features (symbols, tolerances, and datums) within the detected FCFs, we trained the second YOLOv8 model on a dataset of 350 instances. Utilizing an 80-20% split for training and validation data, we trained the model for 500 epochs (Table 3). This resulted in a training time of approximately 1,357 hours. Figure 5 illustreates feature viz. symbol, tolerance, and datum detection by YOLO model for the test data. Figure 6 provides different curves providing details of training and validation loss. It also provides the metric values of precision and recall graphically.

Table 3. Feature detection performance metrics within FCF

Metric	Value
Precision (P)	0.905
Recall (R)	0.916
mAP50	0.91
mAP50-95	0.706

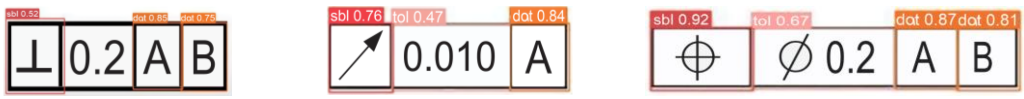

Figure 5. Feature (symbol, tolerance, datum) detection results

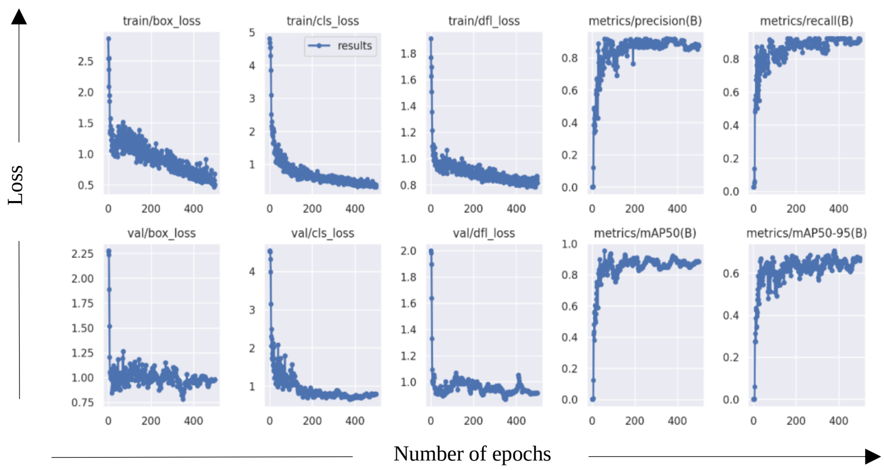

Figure 6. Feature (symbol, tolerance, datum) training curves

4.2.3. Model 3: symbol type classification

To classify the detected symbols into specific types (e.g., perpendicularity, surface profile, position, flatness, maximum material condition, run out, and parallelism), we trained the third YOLOv8 model on a dataset of 300 instances. Employing an 80-20% split for training and validation data, the model was trained for 500 epochs (Table 4). This resulted in a training time of approximately 1.329 hours. Figure 7 illustrates symbol classification by YOLO model for the test data. Figure 8 provides different curves providing details of training and validation loss. It also provides the metric values of precision and recall graphically. Figure 9 shows the results of text extraction using EasyOCR.

Table 4. Feature detection performance metrics for symbol type classification

Metric	Value
Precision (P)	0.945
Recall (R)	0.999
mAP50	0.995
mAP50-95	0.838

Figure 7. Symbol type classification results

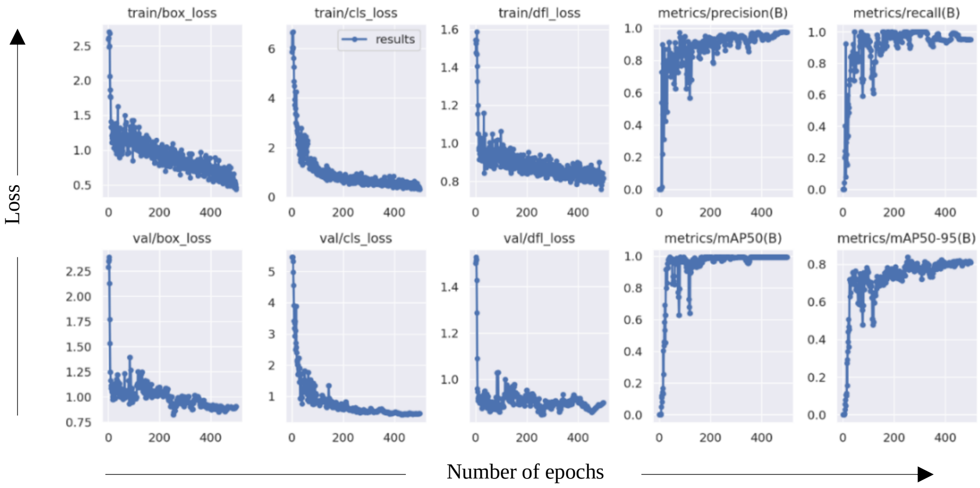

Figure 8. Symbol type training curves

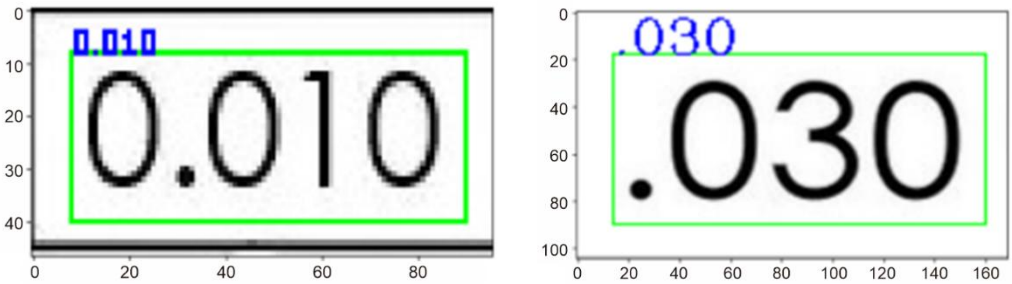

Figure 9. Results of text extraction using EasyOCR

5. DISCUSSION

In this study, YOLOV8 was employed to explore its potential use in important applications of mechanical industry like identification of GD&T symbols on complex part drawings. The work reported in this paper was focused on seven (out of 14) ASME standard GD&T symbols [25]. The dataset of engineering part grawing was carefully curated through image normalization and noise reduction. A multi-stage approach using three separate YOLOv8 models was executed whwrein 1st stage consisting of feature control frame detection, then symbol detection inside FCFs and in 3rd stage classification of symbol type. Further, pre-trained model provided by EasyOCR was deployed for extract tolerance values accurately from the detected GD&T symbols from a variety of engineering drawings. It is important to note that, the pre-processing task is quite important in order to enhance the model's performance.

Three YOLOv8 models were trained. The initial YOLOv8 model could detect FCFs within engineering drawings and the 2nd YOLOv8 model could detect features (symbols, tolerances, and datums) within the detected FCFs. 3rd trained YOLOv8 model could classify the detected symbols into specific category. The experimental results demonstrate that the YOLOv8 models are effective and accurate for symbol recognition in engineering drawings. The models achieved high precision, recall, and mAP values across all symbol types. The multi-stage approach, utilizing specialized models for each object category, proved to be effective in improving the overall accuracy of symbol recognition (Table 5). The performance metric validates the potential of YOLOv8 as a reliable tool for automating the symbol identification process in engineering workflows, reducing manual effort and improving efficiency.

Table 5. Precision, recall, and mAP values all symbol types

Symbol type	No. of training symbols	No. of testing symbols	Precision (P)	Recall (R)	mAP50	mAP50-95
Perpendicularity	90	18	0.945	0.999	0.995	0.838
Surface profile	60	12	0.912	0.961	0.947	0.802
Position	75	15	0.972	0.953	0.975	0.739
Flatness	45	9	0.938	0.942	0.935	0.781
Maximum material condition	12	2	0.964	0.937	0.937	0.769
Run out	5	1	0.901	0.952	0.924	0.718
Parallelism	15	3	0.982	0.964	0.987	0.824

6. CONCLUSION

In conclusion, the multi-stage YOLOv8 object detection approach presented in this study demonstrates a promising approach for symbol recognition in engineering drawings. The models achieved high accuracy, outperformed previous methods, and proved robust to symbol variations. Further research should focus on refining symbol classification, improving tolerance value extraction, and expanding the training dataset using domain-specific data augmentation techniques to further enhance the performance of symbol recognition in engineering drawings. Future research directions in automated symbol recognition may focus on enhancing the robustness of deep learning models to handle variations in symbol appearance, lighting conditions, and image quality. Additionally, developing algorithms capable of accurately detecting and classifying complex combinations of symbols, including overlapping symbols and symbols with intricate geometries, would further advance the field. Finally, investigating techniques to achieve real-time symbol recognition on large-scale datasets would address the computational challenges associated with processing extensive volumes of engineering drawings. Future work could explore transfer learning and domain-specific augmentations to further enhance performance, as well as integrating the model into broader systems for end-to-end automation in engineering design.

REFERENCES

[1] A. Petruccioli, F. Pini, and F. Leali, "Tolerance specification model for systematic application of GD&T in product design," in *Design, Systems, and Complexity*, 2021, doi: 10.1115/IMECE2021-70894.

[2] S.-A. Tahan and S. Levesque, "Exploiting the process capability of profile tolerance according GD&T ASME-Y14.5M," in *2009 International Conference on Computers & Industrial Engineering*, Troyes, France, 2009, pp. 1285–1290, doi: 10.1109/ICCIE.2009.5223816.

[3] A. Aidibe, S. A. Tahan, and M. K. Nejad, "Interlaboratory empirical reproducibility study based on a GD&T benchmark," *Applied Sciences*, vol. 10, no. 14, 2020, doi: 10.3390/app10144704.

[4] H. Bhanbhro, Y. K. Hooi, W. Kusakunniran, and Z. H. Amur, "A symbol recognition system for single-line diagrams developed using a deep-learning approach," *Applied Sciences*, vol. 13, no. 15, 2023, doi: 10.3390/app13158816.

[5] H. Kim *et al.*, "Deep-learning-based recognition of symbols and texts at an industrially applicable level from images of high-density piping and instrumentation diagrams," *Expert Systems with Applications*, vol. 183, 2021, doi: 10.1016/j.eswa.2021.115337.

[6] Y.-H. Lin, Y.-H. Ting, Y.-C. Huang, K.-L. Cheng, and W.-R. Jong, "Integration of deep learning for automatic recognition of 2D engineering drawings," *Machines*, vol. 11, no. 8, 2023, doi: 10.3390/machines11080802.
[7] B. Faltin, P. Schönfelder, and M. König, "Improving symbol detection on engineering drawings using a keypoint-based deep learning approach," in *30th EG-ICE: International Conference on Intelligent Computing in Engineering*, 2023, pp. 1–10.
[8] Y. Moon, J. Lee, D. Mun, and S. Lim, "Deep learning-based method to recognize line objects and flow arrows from image-format piping and instrumentation diagrams for digitization," *Applied Sciences*, vol. 11, no. 21, 2021, doi: 10.3390/app112110054.
[9] J. V. Toro, A. Wiberg, and M. Tarkian, "Optical character recognition on engineering drawings to achieve automation in production quality control," *Frontiers in Manufacturing Technology*, vol. 3, 2023, doi: 10.3389/fmtec.2023.1154132.
[10] S. Bickel, B. Schleich, and S. Wartzack, "Detection and classification of symbols in principle sketches using deep learning," *Proceedings of the Design Society*, vol. 1, pp. 1183–1192, Aug. 2021, doi: 10.1017/pds.2021.118.
[11] E. Elyan, L. Jamieson, and A. Ali-Gombe, "Deep learning for symbols detection and classification in engineering drawings," *Neural Networks*, vol. 129, pp. 91–102, 2020, doi: 10.1016/j.neunet.2020.05.025.
[12] D. Y. Yun, S. K. Seo, U. Zahid, and C. J. Lee, "Deep neural network for automatic image recognition of engineering diagrams," *Applied Sciences*, vol. 10, no. 11, 2020, doi: 10.3390/app10114005.
[13] H. Bhanbhro, Y. K. Hooi, Z. Hassan, and N. Sohu, "Modern deep learning approaches for symbol detection in complex engineering drawings," in *2022 International Conference on Digital Transformation and Intelligence (ICDI)*, 2022, pp. 121–126, doi: 10.1109/ICDI57181.2022.10007281.
[14] S. Sarkar, P. Pandey, and S. Kar, "Automatic detection and classification of symbols in engineering drawings," *arXiv-Computer Science*, pp. 1–7, 2022.
[15] S. Mani, M. A. Haddad, D. Constantini, W. Douhard, Q. Li, and L. Poirier, "Automatic digitization of engineering diagrams using deep learning and graph search," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2020, pp. 673–679, doi: 10.1109/CVPRW50498.2020.00096.
[16] S. M. Rahman, J. Bayer, and A. Dengel, "Graph-based object detection enhancement for symbolic engineering drawings," in *International Conference on Document Analysis and Recognition*, 2021, pp. 74–90, doi: 10.1007/978-3-030-86198-8_6.
[17] P. Schönfelder, F. Stebel, N. Andreou, and M. König, "Deep learning-based text detection and recognition on architectural floor plans," *Automation in Construction*, vol. 157, 2024, doi: 10.1016/j.autcon.2023.105156.
[18] M. Gada, "Object detection for P&ID images using various deep learning techniques," in *2021 International Conference on Computer Communication and Informatics (ICCCI)*, 2021, pp. 1–5, doi: 10.1109/ICCCI50826.2021.9402386.
[19] E.-S. Yu, J.-M. Cha, T. Lee, J. Kim, and D. Mun, "Features recognition from piping and instrumentation diagrams in image format using a deep learning network," *Energies*, vol. 12, no. 23, 2019, doi: 10.3390/en12234425.
[20] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-time flying object detection with YOLOv8," *arXiv-Computer Science*, 2023.
[21] Z. Afrin, F. Tabassum, H. B. Kibria, M. R. Imam, and M. R. Hasan, "YOLOv8 based object detection for self-driving cars," in *2023 26th International Conference on Computer and Information Technology (ICCIT)*, 2023, pp. 1–6, doi: 10.1109/ICCIT60459.2023.10441381.
[22] P. Schönfelder and M. König, "Deep learning-based text detection on architectural floor plan images," *IOP Conference Series: Earth and Environmental Science*, vol. 1101, no. 8, 2022, doi: 10.1088/1755-1315/1101/8/082017.
[23] Y. H. Lin, Y. H. Ting, Y. C. Huang, K. L. Cheng, and W. R. Jong, "Integration of deep learning for automatic recognition of 2D engineering drawings," *Machines*, vol. 11, no. 8, 2023, doi: 10.3390/machines11080802.
[24] P. Henderson and V. Ferrari, "End-to-end training of object class detectors for mean average precision," in *Asian Conference on Computer Vision*, 2017, pp. 198–213, doi: 10.1007/978-3-319-54193-8_13.
[25] ASME, "Y14.5 - Dimensioning and tolerancing (Y14.5 - 2018)," *American Society of Mechanical Engineers*. 2019. [Online]. Available: https://www.asme.org/codes-standards/find-codes-standards/y14-5-dimensioning-tolerancing

BIOGRAPHIES OF AUTHORS

Laxmi Bewoor holds Ph.D. in artificial intelligence as major and optimization as minor. Currently she is Associate Professor at Department of Computer Engineering, Vishwakarma Institute of Information Technology, Pune. Her research interest includes natural language processing, metaheuristics techniques, deep learning, artificial intelligence, and algorithms. She has successfully completed one funded research projects. 5 international patents are granted on her name. She has published more than 35 research articles in reputed journals and conference. She is reviewer of leading jounals of Springer, Elsevier, and IEEE. She can be contacted at email: laxmi.bewoor@viit.ac.in.

Anand Bewoor holds Ph.D. in mechanical engineering. Currently he is Professor of Department of Mechanical Engineering, Cummins College of Engineering for Women, Pune. His research interest includes manufacturing and quality engineering, condition monitoring and optimization. He has successfully completed 4 funded research projects. He has produced 7 Ph.D. and published more than 80 research articles in reputed journals and conferences. He is reviewer of leading jounals of Springer, Elsevier, and IEEE. He can be contacted at email: anand.bewoor@cumminscollege.in.

Dr. Pravin P. Hujare is working as an Associate Professor in Department of Mechanical Engineering, Vishwakarma Institute of Information Technology, Pune, India. He has done Ph.D. in noise & vibration control from COEP, Savitribai Phule Pune University, Pune. He has 21 patents and 35+ research publications in national and international conferences and journals. He has successfully completed one funded research projects. His research interests are machine design, mechanical vibration and noise control, machine learning, and data analytics. He can be contacted at email: pravin.hujare@viit.ac.in.

Praveen Rathod is pursuing Ph.D. degree in mechanical engineering. His research interest is in electric vehicles, ACD, CAM, and automotive technology. He can be contacted at email: praveen.rathod@viit.ac.in.

Vedant Yetekar is under graduate research scholar studying in Department of Mechanical Engineering, Vishwakarma Instittue of Information Technology, Pune. His keen interest is in deep learning applications for mechanical engineering. He can be contacted at email: vedant.22010546@viit.ac.in.

Shrish Dollin is currently graduate research scholar studying in Department of Mechanical Engineering, Vishwakarma Instittue of Information Technology, Pune. His major interest is in AI applications for mechanical engineering. He can be contacted at email: shrish.22010372@viit.ac.in.