© 2025 IJRTI | Volume 10, Issue 4 April 2025 | ISSN: 2456-3315

Automated Error Detection in Technical Drawings Using Machine Learning Algorithms: A Comprehensive Review

[1]Urmila Nagargoje, [2]Amol Ghule, [3]Sidhant Tarte, [4]Khushi Varma, [5]Mayur Todmal
[1]*Assistant Professor*,[2]*Assistant Professor*, [3,4,5]*Student*
[1]Mechanical Engineering Department,
[1]*S. G. V. V. S Adsul's Technical Campus, Chas, Ahilyanagar*, India
[1]Urmila.nagargoje@gmail.com

***Abstract*— Technical drawings play a pivotal role in the fields of engineering design, architecture, and manufacturing, where precision and accuracy are paramount. These drawings serve as the primary medium for communicating intricate design specifications, which must be error-free to ensure the integrity and functionality of the final product. However, manual error detection in technical drawings is a labor-intensive process that is susceptible to human oversight, particularly as the complexity and size of these drawings increase. In response to this challenge, recent advancements in machine learning (ML) have paved the way for automating error detection, offering the potential to significantly improve accuracy, efficiency, and consistency.**

This paper presents a comprehensive review of the application of machine learning algorithms for automated error detection in technical drawings. It explores a wide range of techniques, including supervised and unsupervised learning models, deep learning methods, and anomaly detection approaches, all of which have shown promising results in identifying various types of errors such as missing dimensions, incorrect scaling, and annotation discrepancies. The review also delves into key aspects of data preparation, such as the use of labeled datasets, feature extraction methods, and the integration of domain-specific knowledge for enhanced model performance. Furthermore, the paper evaluates the performance metrics commonly used to assess the efficacy of ML models in this domain, including accuracy, precision, recall, and F1 score, and discusses the challenges associated with these metrics in the context of technical drawings.

Despite the promising progress, several challenges remain, such as improving the robustness of ML models to handle complex, real-world drawing errors and enhancing their interpretability to ensure that detected errors can be easily understood and corrected by engineers. Additionally, the generalization of these models across diverse industries and drawing styles remains an ongoing area of research. This study underscores the growing importance of integrating artificial intelligence (AI) technologies into engineering workflows, with the goal of streamlining design processes, enhancing quality control, and fostering innovation in the development of technical systems and products. The paper concludes by highlighting potential future directions for research, including the exploration of hybrid AI approaches, the development of more comprehensive and diverse datasets, and the use of explainable AI techniques to increase model transparency and trust.

***Index Terms*— Automated Error Detection, Technical Drawings, Machine Learning, Image Processing, Anomaly Detection, Feature Extraction, Deep Learning, Error Identification, Engineering Design, CAD (Computer-Aided Design), Supervised Learning, Unsupervised Learning, Neural Networks, Model Evaluation Metrics, Accuracy in Technical Drawings, AI in Engineering, Predictive Models, Model Interpretability, Quality Assurance.**

I. Introduction

Technical drawings act as the blueprint for manufacturing, construction, and design processes. These drawings communicate geometric and functional intent, detailing dimensions, materials, tolerances, and assembly instructions. Errors in technical drawings—whether they are geometric inconsistencies or missing annotations—can lead to significant downstream consequences including increased costs, delays, or even product failure. As engineering systems become more complex, so too does the potential for undetected errors.

In traditional workflows, experienced engineers manually review technical drawings to ensure that specifications align with standards and functional requirements. However, human limitations such as fatigue, cognitive overload, and bias can lead to missed errors, especially in large-scale or intricate design projects. Even minor mistakes in annotations or symbols can compromise design interpretation and hinder downstream processes like simulation, prototyping, and fabrication.

With the advent of advanced machine learning techniques, especially deep learning, new opportunities have emerged to enhance or fully automate the error detection process in technical drawings. Unlike rule-based systems, ML algorithms can learn from data, identify subtle patterns, and adapt to diverse drawing styles and standards. This review aims to map out the landscape of ML-based automated error detection systems, summarize the key developments, and propose future avenues of research. The adoption of ML in this domain is aligned with broader trends in digital transformation, enabling smarter and more resilient engineering practices.

II. Background And Motivation

Manual inspection of drawings is not only labor-intensive but also relies heavily on the expertise and attentiveness of reviewers. Errors such as misaligned parts, conflicting dimensions, and missing annotations can easily be overlooked. CAD systems, although increasingly sophisticated, often provide only syntax-level checking (e.g., checking line connectivity) and lack semantic understanding.

Traditional CAD error-checking functions are limited by their dependence on static rules that may not capture context-sensitive or design-specific nuances. They often fail to detect logical inconsistencies or interpret custom annotations and symbols. Furthermore, in distributed engineering environments where multiple teams collaborate across geographic locations, inconsistencies in interpretation of standards may further increase the risk of errors.

Machine learning models can be trained to detect both low-level and high-level errors by learning from labeled data. This approach allows for continuous improvement and adaptation to specific industries or drawing conventions. The motivation for applying ML includes:

- Reducing human workload and errors
- Ensuring compliance with standards
- Enabling real-time validation during the design process
- Enhancing the quality of manufacturing documentation
- Accelerating design iteration cycles
- Improving overall design reliability and safety

The shift towards smart factories, digital twins, and cyber-physical systems necessitates the integration of intelligent tools capable of self-learning and adaptive behavior. ML-based error detection systems represent a step toward autonomous engineering validation, contributing to more agile, efficient, and error-resilient design ecosystems.

III. CLASSIFICATION OF ERRORS IN TECHNICAL DRAWING

To design effective ML-based detection systems, it is essential to classify the types of errors that occur in technical drawings:

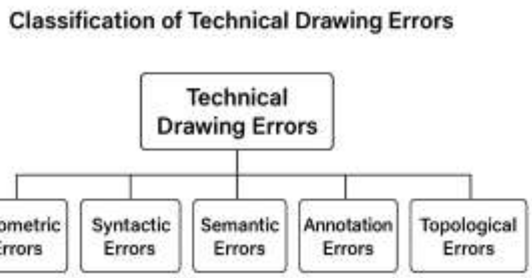

Figure 1: Classification of Technical Drawing Errors

- **Geometric Errors**: These involve inconsistencies in dimensions, overlapping lines, misaligned entities, missing projections, or incorrect scale factors. Geometric errors can misrepresent the physical structure of the object and lead to incompatible parts or assembly failures.
- **Syntactic Errors**: These include violations of formal drafting standards (e.g., ISO 128, ANSI Y14.5). Examples include incorrect line types for hidden edges, improper section lines, or misused dimensioning styles. Such errors impact the readability and interpretability of the drawings.
- **Semantic Errors**: These errors arise when the intent of the drawing is compromised, even though the syntax and geometry may be valid. For instance, specifying an incorrect tolerance or material can result in performance failures. Semantic errors are particularly challenging to detect because they require contextual and domain-specific understanding.
- **Annotation Errors**: Annotations such as dimensions, surface finishes, welding symbols, and material notes must be precise and complete. Missing, duplicate, or ambiguous annotations fall under this category. Incorrect text positioning or formatting can also lead to misinterpretation.
- **Topological Errors**: These involve the structural relationships between drawing elements. For instance, a drawing may show disconnected components that should be joined, or contain open loops in what should be closed profiles. These errors often cause problems in downstream processes like meshing, 3D modeling, or CNC programming.

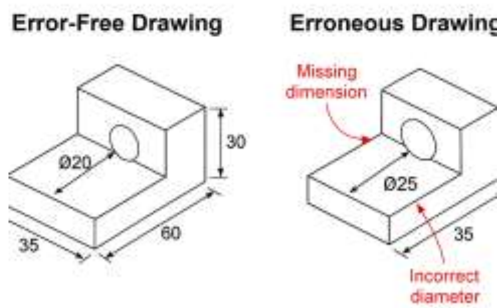

Figure 2: Example of an error-free vs. erroneous drawing

Identifying and categorizing these errors forms the foundation for building tailored machine learning pipelines capable of intelligent and context-aware detection.

Table 1: Common Error Types and Detection Challenges

Error Type	Description	Detection Challenges
Missing Dimensions	Critical dimension values are absent or omitted	Difficult to detect without clear reference standards
Incorrect Scaling	Dimensions do not match intended size ratios	Can go unnoticed in unscaled technical drawings
Overlapping Components	Components overlap in a way that may cause confusion	Hard to identify without advanced geometric analysis
Misplaced or Missing Annotations	Annotations that are incorrectly placed or missing	Common in hand-drawn or poorly digitized drawings
Incorrect Proportions	Ratios between different parts are incorrect	Requires comparison with a valid geometric model
Inaccurate Angles	Angles are misrepresented, leading to distorted shapes	Hard to spot unless compared with known angle templates
Inconsistent Layering	Drawing layers are not correctly prioritized	Difficult to distinguish without 3D analysis

IV. MACHINE LEARNING TECHNIQUES FOR ERROR DETECTION

4.1 Supervised Learning

In supervised learning, labeled datasets are used to train algorithms to classify or predict the presence of specific errors. This approach is highly effective when historical data is available with annotated ground truth.

- **Convolutional Neural Networks (CNNs)**: CNNs are widely used for image-based tasks and are well-suited for analyzing raster or vector representations of technical drawings. They can identify anomalies in spacing, layout, component arrangements, and symmetry. By learning visual patterns, CNNs can detect irregularities such as missing elements or distorted symbols.

- **Random Forests & SVMs**: These classical algorithms are effective for structured data derived from feature extraction processes. They can analyze tabular datasets representing features like number of annotations, line orientations, dimension ranges, and text frequency. Their interpretability and fast training times make them suitable for prototyping and comparison.

- **Multi-Class Classifiers**: These models are capable of distinguishing among multiple types of errors simultaneously, making them useful in scenarios where drawings may contain overlapping or compound errors.

4.2 Unsupervised Learning

This approach is used when labeled data is scarce or unavailable. Unsupervised learning focuses on discovering patterns or anomalies without predefined categories.

- **Autoencoders**: These neural networks learn to compress and reconstruct input data. When trained on error-free drawings, they can flag reconstruction anomalies as potential errors. This method is effective for detecting subtle deviations in layout, geometry, or annotation density.

- **Clustering Algorithms**: Methods such as K-Means, DBSCAN, or Gaussian Mixture Models can be used to identify outlier patterns that deviate from the majority. For instance, an unusually high or low dimension-to-annotation ratio may signal an error.

4.3 Deep Learning Architectures Advanced deep learning models enable more complex and scalable solutions:

- **YOLO/Faster R-CNN**: These object detection frameworks can detect and localize components like holes, fasteners, slots, or annotation symbols. They are beneficial for identifying missing or misaligned graphical elements.

- **Graph Neural Networks (GNNs)**: These models are especially suited for relational data. In technical drawings, entities such as lines, circles, and annotations can be represented as nodes, with edges denoting topological or spatial relationships. GNNs can learn the structural context and detect disruptions in expected configurations.

- **Transformers**: Originally developed for NLP, transformers are useful in parsing text annotations and understanding their context within the drawing. They can handle sequences of labels, dimension chains, and hierarchies of notes, enabling detection of misplaced or inconsistent annotations.

Table 2: Comparison of Supervised vs. Unsupervised Models

Aspect	Supervised Learning	Unsupervised Learning
Data Labeling	Requires labeled data for training	Does not require labeled data
Output	Predicts specific outputs (e.g., class labels, values)	Identifies patterns or groups within the data
Common Algorithms	SVM, Random Forest, Logistic Regression, Neural Networks	K-Means, Hierarchical Clustering, PCA, Autoencoders
Use Cases	Classification, Regression	Clustering, Anomaly Detection, Dimensionality Reduction
Complexity	Often more complex due to the need for labeled data	Can uncover hidden patterns but harder to evaluate
Performance Metrics	Accuracy, Precision, Recall, F1 Score	Silhouette Score, Clustering Performance

After Emerging technologies in Engineering Graphics are revolutionizing the way designs, models, and simulations are created, analyzed, and shared. Some of the key advancements include Augmented Reality (AR) and Virtual Reality (VR), Artificial Intelligence (AI) and 3D Printing and Additive Manufacturing

V. Data Collection And Preprocessing

5.1 Data Sources

Creating a high-quality dataset is a prerequisite for effective model training and evaluation:

- **Proprietary industrial datasets** from companies using CAD tools like AutoCAD, SolidWorks, and CATIA, though often protected by confidentiality agreements.
- **Public academic datasets** from engineering institutions or online CAD repositories.
- **Synthetic data** generated using parametric modeling tools to simulate various error scenarios in a controlled manner.

5.2 Preprocessing Techniques Data preprocessing transforms raw drawings into formats suitable for machine learning:

- **Image Processing**: Includes noise reduction, binarization, line detection, morphological transformations, and resizing. Enhances clarity and reduces data artifacts.
- **Vectorization**: Raster images are converted to vector representations (e.g., SVG, DXF), facilitating analysis of geometrical properties.
- **Text Extraction**: Optical Character Recognition (OCR) tools like Tesseract can identify and digitize textual annotations. These can be further analyzed using NLP techniques.
- **Feature Engineering**: Numerical features such as symmetry scores, annotation density, line segment orientation histograms, and bounding box overlaps are derived to improve model interpretability and performance.

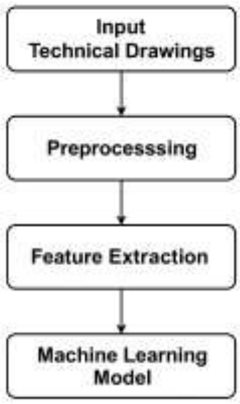

Figure 3: ML model pipeline for error detection

VI. Model Evaluation And Performance Metrics

To ensure the effectiveness of ML models, appropriate evaluation metrics must be used:

- **Accuracy**: Measures overall correctness but may be misleading in imbalanced datasets.
- **Precision & Recall**: Particularly important when false positives or false negatives carry high risk. Precision indicates correctness of positive predictions, while recall measures completeness.

- **F1-Score**: Combines precision and recall, useful for balanced assessment.
- **IoU (Intersection-over-Union)**: Common in object detection to assess how well predicted bounding boxes match actual ones.
- **Confusion Matrix**: Helps diagnose misclassification by visualizing prediction results across multiple error types.

Additionally, **ROC-AUC**, **PR curves**, and **mean average precision (mAP)** may be used depending on the model type and application context.

VII. Challenges And Limitation

Despite significant advancements, several challenges hinder the widespread adoption of ML in technical drawing error detection:

- **Data Scarcity**: Most real-world technical drawings are proprietary, limiting access for model training and benchmarking.
- **Domain Variability**: Engineering domains vary significantly in symbols, standards, and annotation styles, making generalization difficult.
- **Annotation Ambiguity**: Text in technical drawings may be context-specific or use abbreviations, complicating semantic interpretation.
- **Integration Barriers**: Incorporating ML models into commercial CAD tools requires compatibility with various file formats and APIs.
- **Real-Time Requirements**: In industrial applications, error detection must be fast enough to provide feedback during live design sessions.
- **Model Interpretability**: Deep models often function as black boxes, raising concerns in high-stakes applications where explanation and traceability are crucial.

VIII. Future Research Direction

To overcome current limitations and enhance the applicability of ML models in error detection, future work should focus on:

- **Hybrid Approaches**: Combining deterministic rule-based checks with probabilistic ML models for more accurate and explainable results.
- **Few-Shot and Transfer Learning**: Using pre-trained models to reduce data dependence and adapt quickly to new domains.
- **Explainable AI (XAI)**: Developing tools that provide human-readable justifications for model outputs to build user trust.
- **Benchmark Datasets and Competitions**: Establishing public datasets and challenge platforms to encourage innovation and standardization.
- **Contextual and Functional Reasoning**: Enhancing models to infer design intent and detect errors that violate functional requirements rather than just geometric rules.
- **Active Learning**: Engaging human experts in the loop to iteratively improve model performance and label new data efficiently.

IX. Conclusion

Automated error detection in technical drawings is a promising application area for machine learning, with potential benefits across manufacturing, construction, and design sectors. This paper reviewed the key types of errors, ML methodologies, data preprocessing strategies, and evaluation criteria used in the field. While current solutions show strong potential, several technical and operational challenges must be addressed.

Future advancements in deep learning, explainability, and standardization are expected to significantly enhance the capabilities and adoption of ML-based error detection systems. The integration of these intelligent tools within design environments represents a critical step toward smarter, more efficient, and error-resilient engineering workflows, reinforcing the importance of AI-driven automation in modern engineering design.

REFERENCES

[1] A. Kusiak, "Smart manufacturing," *Int. J. Prod. Res.*, vol. 55, no. 8, pp. 2468–2479, 2017.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.

[3] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in *Proc. Machine Learning for Signal Processing (MLSDA)*, 2014, pp. 1–6.

[4] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, 2017.

[5] W. Samek, T. Wiegand, and K. R. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," *arXiv preprint arXiv:1708.08296*, 2017.

[6] Z. Wang and H. Liu, "A survey of deep learning techniques for technical drawing recognition and classification," *Eng. Appl. Artif. Intell.*, vol. 92, pp. 103636, 2020.

[7] C. C. Aggarwal, *Data Mining: The Textbook*, 1st ed. Berlin, Germany: Springer, 2015.

[8] Y. Zhang and F. Tsui, "Error detection and correction in technical drawings using machine learning," *J. Manuf. Process.*, vol. 29, pp. 109–118, 2017.

[9] S. Jain and P. Soni, "A comprehensive survey on machine learning-based approaches in CAD models for error detection," *Int. J. Eng. Technol.*, vol. 11, no. 2, pp. 158–168, 2020.

[10] V. Ramakrishnan and S. Chowdhury, "A hybrid approach for error detection in technical drawings: A comparative study," *J. Comput. Des. Eng.*, vol. 6, no. 1, pp. 88–96, 2019