Deep Learning-Based Multi-Task Recognition of Objects and Text in Engineering Drawings

Jyun-Yu You[1] and Ying-Hua Huang [2]

[1] Graduate School of Engineering Science and Technology, National Yunlin University of Science and Technology, Taiwan
[2] Department of Civil and Construction Engineering, National Yunlin University of Science and Technology, Taiwan
D11310206@yuntech.edu.tw, huangyh@yuntech.edu.tw

Abstract –

Engineering drawings remain a critical medium for conveying design intent in architectural and construction projects. Despite the increasing adoption of Building Information Modeling (BIM), two-dimensional drawings are still widely used in practice, particularly during early design stages and interdisciplinary coordination. The interpretation of such drawings largely relies on manual processes, which are time-consuming and prone to human error, especially when drawings contain dense graphical elements and complex textual annotations.

This study proposes an automated framework that integrates deep learning–based object detection and optical character recognition (OCR) to extract graphical symbols and textual information from architectural floor plans. A tile-based detection strategy combined with global non-maximum suppression (NMS) is employed to address large-format drawings and multi-scale objects. Detected text regions are further processed using an OCR module to extract text for subsequent analysis. Experimental results demonstrate the feasibility of the proposed pipeline and highlight key challenges related to complex layouts, rotated text, and domain-specific terminology. Overall, the proposed framework establishes a structured foundation for future semantic interpretation and intelligent engineering applications.

Keywords –

Engineering drawings, Floor plan recognition, Object detection, OCR, Semantic interpretation

1 Introduction

Engineering drawings remain a primary medium for conveying design intent in building projects. Despite the increasing adoption of Building Information Modeling (BIM), a substantial portion of design information is still generated and exchanged in two-dimensional floor plans. These drawings typically contain dense geometric components and domain-specific textual annotations, making automated interpretation particularly challenging.

Existing studies have explored object detection or text recognition in architectural drawings using deep learning techniques. However, most approaches focus on single-task recognition and lack an integrated framework capable of jointly extracting geometric objects and textual information from complex floor plans. This limitation restricts their applicability to downstream tasks such as information structuring and BIM model generation.

To address this gap, this study proposes a deep learning-based multi-task recognition framework for architectural floor plans. While the core components leverage established architectures like YOLO and OCR, the contribution of this work lies in integrating these components into a unified pipeline tailored to the characteristics of floor plans, including large-format images, dense layouts, mixed graphical-textual content, and Chinese annotations. This integrated design provides a practical basis for structured information extraction and subsequent engineering-oriented interpretation.

2 Literature Review

2.1 Automated Drawing Analysis

Early studies on automated engineering drawing analysis primarily relied on rule-based and traditional image processing techniques, including heuristic rules, line tracing, and template matching [1]. Due to significant variability in drawing styles, symbol conventions, and annotation density, these methods generally exhibited limited robustness and poor generalization [2], [3].

With advances in deep learning, convolutional neural networks (CNNs) have become the dominant approach for the analysis of engineering drawings. Architectures such as U-Net, Mask R-CNN, and YOLO have demonstrated promising performance in detecting symbols and structural components in complex line-based drawings [4]-[6]. However, engineering drawings differ fundamentally from natural images, as they are characterized by dense geometric structures, high-contrast linework, scale variation, and the close interleaving of graphical and textual information, which continue to pose challenges for reliable automated recognition.

Within this category, compared with P&ID diagrams, architectural floor plans exhibit less standardized symbol libraries and higher layout variability.

2.2 Research Focus and Limitations of Existing Studies

A large body of existing research has focused on piping and instrumentation diagrams (P&ID), in which diagram structures and symbol libraries are relatively standardized, enabling high detection accuracy using deep learning–based methods [7]. In contrast, architectural floor plans have received comparatively limited attention.

Most current studies on architectural drawings address isolated tasks, such as object detection, room boundary extraction, or room type recognition [3], [5], [8]. Recent surveys classify automated floor plan analysis into room boundary detection, room type recognition, object detection, and text detection and recognition [9].

Nevertheless, most approaches remain single-task oriented, and few studies attempt to integrate multiple recognition tasks within a unified framework. As a result, many systems provide only low-level outputs, such as bounding boxes or masks, without cross-task integration or semantic interpretation.

2.3 Text Recognition in Engineering Drawings

Textual annotations in engineering drawings convey critical information, including dimensions, material specifications, and construction notes. Although recent studies have explored text detection and OCR in engineering diagrams [6], [10], recognition performance remains sensitive to text orientation, font variation, background noise, and image resolution.

These challenges are further amplified in drawings containing Chinese annotations. Chinese characters exhibit high visual complexity and lack explicit word boundaries, while engineering texts often involve domain-specific terminology and mixed-language expressions. Consequently, even when character-level recognition is achieved, the absence of contextual understanding frequently limits the interpretability of extracted text [11], [12] , rendering text recognition alone insufficient for downstream engineering applications.

2.4 Research Gap and Motivation

The reviewed literature reveals several limitations. Architectural floor plans remain underrepresented compared to P&ID diagrams, and most existing studies focus on single-task recognition with limited integration of graphical and textual information. Moreover, text recognition for Chinese engineering annotations remains insufficiently addressed, and few approaches link recognition outputs to higher-level engineering semantics or application-oriented representations.

These gaps motivate the development of an integrated framework that jointly considers object detection and text recognition. Accordingly, this study proposes a scalable detection–recognition pipeline for architectural floor plans, aiming to bridge the gap between visual extraction and intelligent engineering applications.

3 Proposed Framework

The proposed framework consists of four main stages: image tiling, object detection, result merging, and text recognition. The overall workflow is designed to accommodate high-resolution drawings and diverse annotation characteristics commonly found in engineering practice. The complete processing pipeline and the interactions between individual modules are illustrated in Figure 1.

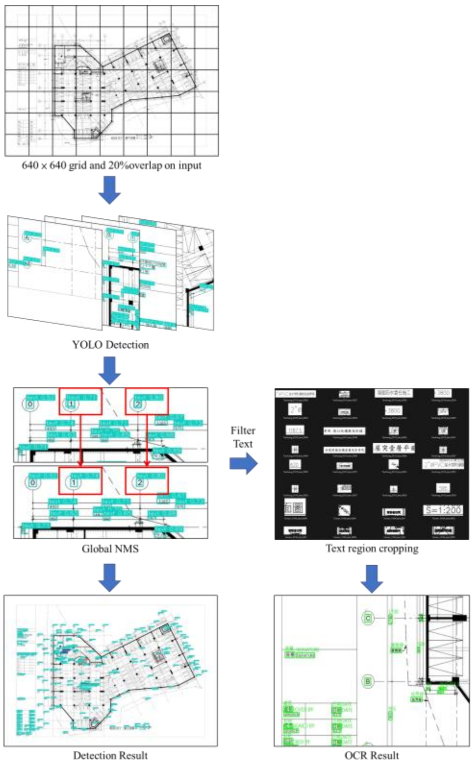

Figure 1. Overall framework

3.1 Image Tiling Strategy

Architectural floor plans are typically provided at high resolutions, which exceed the input size constraints of common deep learning models. To address this issue, a tile-based strategy is adopted, in which each input drawing is divided into fixed-size image patches (640 × 640 pixels). An overlap is introduced between adjacent tiles to mitigate boundary truncation effects and preserve contextual information for objects located near tile edges.

The selection of the tile size and overlap ratio was guided by a trade-off between detection accuracy and computational efficiency. Larger tiles preserve more contextual information but increase memory consumption, whereas smaller tiles may lead to object truncation and fragmented detections. An overlap between adjacent tiles was therefore introduced to mitigate boundary effects and reduce missed detections near tile edges.

3.2 Object Detection Using YOLO

A YOLO-based object detection model is employed to identify graphical symbols and text regions within each image tile. Detection is performed independently on each tile, producing bounding boxes, class labels, and confidence scores. The detected bounding boxes are then mapped back to the original image's coordinate system.

To eliminate redundant detections arising from overlapping tiles, a global non-maximum suppression (NMS) process is applied. This step retains the most representative bounding boxes based on confidence scores while removing highly overlapping predictions, thereby ensuring robust object localization across large-format drawings.

3.3 Text Region Filtering and Cropping

Among all detected objects, bounding boxes classified as text are selected for subsequent processing. Each text region is cropped from the original drawing, and padding is applied when necessary to ensure sufficient context. This selective filtering allows the OCR module to focus exclusively on candidate text regions, reducing computational cost and false recognition.

3.4 OCR-Based Text Recognition

The cropped text regions are processed by an OCR module to recognize text. The recognized text, confidence scores, and spatial information are recorded and associated with their corresponding bounding boxes. These outputs serve as intermediate results for future semantic interpretation and structured information extraction.

4 Dataset Construction

4.1 Data Sources

The dataset used in this study consists of 118 architectural floor plans collected from nine real-world building projects, including residential buildings, townhouses, educational facilities, and public infrastructure. The drawings exhibit substantial variation in scale, layout, annotation density, and visual style, reflecting realistic engineering conditions. Detailed statistics of the dataset, including project distribution, drawing characteristics, and data splits, are summarized in Table 1.

Table 1. Overview of the dataset

Item	Description
Number of projects	9
Number of floor plans	118
Drawing types	Residential and public buildings
Image resolution	3500–5000 px (short edge)
Annotation types	Symbols + text

As shown in Table 1, the dataset covers multiple building types and exhibits substantial variation in resolution and annotation density, which reflects realistic engineering drawing conditions rather than controlled benchmark datasets.

4.2 Annotation Strategy

Textual annotations are labeled on a line-by-line basis using horizontal rectangular bounding boxes. Blurred or partially occluded text is annotated as long as its textual nature can be reasonably identified. Graphical symbols are annotated separately to support multi-class object detection.

4.3 Dataset Generation Workflow

Annotated full-size drawings are divided into overlapping tiles following the image tiling strategy. Bounding boxes fully contained within a tile are retained, while partially truncated boxes are selectively preserved for specific classes, such as text. Tiles without annotations are discarded to reduce dataset imbalance. The resulting dataset is randomly split into training, validation, and testing sets using a 7:2:1 ratio.

5 Experimental Results

Preliminary experiments were conducted using multiple YOLO variants, including models from the YOLOv8 and YOLOv11 families. Performance was evaluated using standard object detection metrics, including mean Average Precision (mAP) at IoU thresholds of 0.50 and 0.50–0.95, as well as bounding box precision. Quantitative detection performance for all evaluated models is reported in Table 2. Representative qualitative detection results under different drawing conditions are presented in Figure 2.

Table 2. Detection performance for all classes

Model	mAP@50	mAP@50–95
YOLOv8n	0.624	0.469
YOLOv8s	0.942	0.752
YOLOv8m	0.957	0.793
YOLOv8l	0.957	0.802
YOLOv8x	0.961	0.781
YOLOv11n	0.629	0.473
YOLOv11s	0.947	0.767
YOLOv11m	0.966	0.791
YOLOv11l	0.972	0.815
YOLOv11x	0.945	0.787

The results in Table 2 indicate an overall performance improvement with increasing model capacity up to a certain scale, particularly under the mAP@50–95 metric, suggesting enhanced robustness for multi-scale and densely distributed objects. This trend is especially relevant for architectural drawings, where object sizes and spatial distributions vary significantly within a single layout.

Compared with mAP@50, the mAP@50–95 metric provides a stricter evaluation of localization robustness, which is particularly relevant for architectural floor plans with dense and closely spaced elements. The observed performance gains of mid- to large-scale models suggest an enhanced ability to capture multi-scale contextual cues, albeit with increased computational cost, indicating that model selection should balance accuracy and practical deployment constraints.

Figure 2. Qualitative detection results

Figure 2 illustrates representative detection results under different drawing conditions, including dense annotation regions, overlapping symbols, and varying line thicknesses. While most objects are correctly localized, failure cases are mainly observed in regions with rotated text and severe overlap.

The results indicate that the tile-based detection strategy combined with global non-maximum NMS effectively improves detection coverage for large-format drawings. As summarized in Table 2 and illustrated in Figure 2, both textual regions and graphical symbols can be detected with acceptable accuracy across diverse drawing layouts. Nevertheless, several challenges remain, particularly in handling complex layouts and non-standard text orientations.

The OCR results demonstrate the feasibility of extracting textual content from detected regions using an off-the-shelf OCR model, although recognition accuracy varies with font style, text orientation, and background complexity. In practice, the OCR module performs reliably for short, horizontally aligned annotations with relatively clean backgrounds, while errors are more frequently observed for rotated or densely packed text. These observations suggest that OCR reliability is strongly influenced by detection quality, highlighting the importance of tighter integration between detection and recognition modules. Representative examples of Chinese text recognition results, illustrating typical successful recognition and failure cases, are shown in Figure 3.

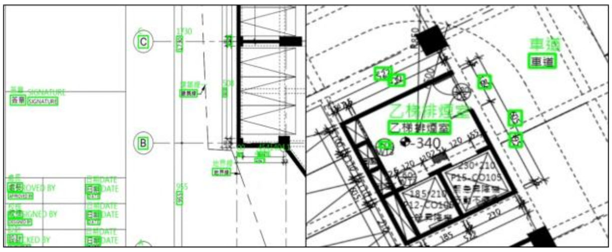

Figure 3. Examples of Traditional Chinese text recognition results on floor plans

To provide a more in-depth evaluation of the proposed framework, two supplementary analyses were conducted. First, an ablation study was performed to examine the effect of the tile-based strategy and overlap settings on object detection performance. Second, a quantitative OCR comparison was conducted to assess how different text extraction workflows influence the number of recognized text regions and the average OCR confidence. The corresponding results are summarized in Table 3 and Table 4, respectively.

Table 3. Ablation study on the effect of tiling and overlap ratio for YOLOv11l

Method	mAP@50	mAP@50–95
YOLOv11l without tiling	0.045	0.013
YOLOv11l tiling (no overlap)	0.882	0.699
YOLOv11l tiling (10% overlap)	0.972	0.815
YOLOv11l tiling (15% overlap)	0.992	0.858

As shown in Table 3, the tile-based strategy substantially improves detection performance compared with direct inference on the full-size drawing. Without tiling, YOLOv11l achieves only 0.045 mAP@50 and 0.013 mAP@50–95, indicating that large-format floor plans cannot be reliably processed in a single pass under the current model setting. Introducing tiling without overlap already leads to a significant improvement, while adding overlap further enhances detection robustness. Among the tested settings, a 15% overlap achieves the best performance, suggesting that overlap helps preserve contextual information near tile boundaries and reduces missed detections for objects located across adjacent patches.

Table 4. Quantitative comparison of text extraction workflows for OCR pre-processing

Method	Direct OCR on full drawings	OCR on tiled drawings	YOLO-based text detection + OCR
Number of detected boxes	153/241	173/241	233/241
Average OCR confidence	0.785	0.835	0.823

Table 4 presents a quantitative comparison of OCR performance under three text extraction workflows. Direct OCR applied to full drawings detects 153 out of 241 text instances, whereas OCR on tiled drawings improves the number of detected text regions to 173. The proposed YOLO-based text detection followed by OCR achieves the highest text coverage, with 233 out of 241 text regions successfully identified. Although the average OCR confidence of the tiled OCR approach is slightly higher than that of the YOLO-based pipeline, the latter provides substantially better text localization coverage. This result indicates that the main advantage of the proposed framework lies in improving text region recall before recognition, which is particularly important for engineering drawings containing dense and spatially distributed annotations.

6 Discussion

The experimental findings suggest that integrating object detection and OCR provides a viable approach for automated information extraction from engineering drawings. The tile-based strategy effectively addresses input size limitations while preserving detection performance across different drawing scales.

Nevertheless, the results also highlight inherent challenges associated with engineering drawings. Complex layouts, heterogeneous annotation styles, and domain-specific terminology pose difficulties for both detection and recognition models. These observations underscore the importance of domain-adaptive training strategies and robust post-processing mechanisms.

At the current stage, the proposed framework is not intended to achieve full automation. Instead, it aims to provide a structured basis upon which higher-level semantic interpretation and intelligent engineering applications can be developed.

From an application perspective, the current framework can serve as a front-end information extraction module for engineering workflows that require rapid access to drawing content. In particular, detected symbols and recognized annotations can be converted into structured records to support space inventory, drawing review, preliminary quantity takeoff, and BIM-assisted data entry. For example, text labels associated with rooms or components can be extracted together with their spatial locations, allowing practitioners to reduce the manual effort required for locating, transcribing, and organizing drawing information. While the full integration of these downstream applications remains outside the immediate scope of this paper, the current results demonstrate that the proposed pipeline can provide an operational basis for construction-specific decision support and further semantic interpretation. However, these use cases currently remain at the structured information extraction stage and have not yet been validated through deployment in a full production workflow.

7 Future Work

Future research will focus on several key directions:

1. OCR Model Fine-Tuning: Constructing a domain-specific Chinese annotation dataset with diverse fonts, sizes, orientations, and background noise to improve recognition robustness.

2. Hyperparameter Optimization: Systematically analysing the influence of detection-related hyperparameters, including input resolution, overlap ratio, confidence thresholds, and NMS strategies.
3. Extended Ablation and Sensitivity Analysis: Further examining the influence of model architectures, overlap ratios, confidence thresholds, and post-processing settings on detection and recognition performance across more diverse drawing conditions.
4. Semantic Interpretation: Integrating large language models to transform recognized graphical and textual information into structured engineering semantics for downstream applications such as quantity takeoff and cost estimation.

8 Conclusion

This study presents a compact and integrated deep learning framework for the joint recognition of geometric objects and textual information in architectural floor plans. By combining tile-based object detection with OCR-based text recognition, the proposed approach effectively addresses key challenges associated with large-format engineering drawings and heterogeneous annotations. While certain limitations remain, particularly in handling highly complex layouts and domain-specific textual variations, the proposed framework establishes a scalable and extensible foundation for future research on semantic interpretation and intelligent engineering drawing analysis.

Acknowledges

This research was supported by the National Science and Technology Council, Taiwan.

References

[1] F. C. A. Groen, A. C. Sanderson, and J. F. Schlag, "Symbol recognition in electrical diagrams using probabilistic graph matching," *Pattern Recognit. Lett.*, vol. 3, no. 5, pp. 343–350, Sep. 1985, doi: 10.1016/0167-8655(85)90066-2.

[2] C. F. Moreno-García, E. Elyan, and C. Jayne, "New trends on digitisation of complex engineering drawings," *Neural Comput. Appl.*, vol. 31, no. 6, pp. 1695–1712, Jun. 2019, doi: 10.1007/s00521-018-3583-1.

[3] Y. Zhao, X. Deng, and H. Lai, "A Deep Learning-Based Method to Detect Components from Scanned Structural Drawings for Reconstructing 3D Models," *Appl. Sci.*, vol. 10, no. 6, Art. no. 6, Jan. 2020, doi: 10.3390/app10062066.

[4] E. Elyan, L. Jamieson, and A. Ali-Gombe, "Deep learning for symbols detection and classification in engineering drawings," *Neural Netw.*, vol. 129, pp. 91–102, Sep. 2020, doi: 10.1016/j.neunet.2020.05.025.

[5] A. Rezvanifar, M. Cote, and A. B. Albu, "Symbol Spotting on Digital Architectural Floor Plans Using a Deep Learning-based Framework," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 2419–2428. doi: 10.1109/CVPRW50498.2020.00292.

[6] P. Schönfelder, F. Stebel, N. Andreou, and M. König, "Deep learning-based text detection and recognition on architectural floor plans," *Autom. Constr.*, vol. 157, p. 105156, Jan. 2024, doi: 10.1016/j.autcon.2023.105156.

[7] M. Gupta, C. Wei, and T. Czerniawski, "Semi-supervised symbol detection for piping and instrumentation drawings," *Autom. Constr.*, vol. 159, p. 105260, Mar. 2024, doi: 10.1016/j.autcon.2023.105260.

[8] P. N. Pizarro, N. Hitschfeld, I. Sipiran, and J. M. Saavedra, "Automatic floor plan analysis and recognition," *Autom. Constr.*, vol. 140, p. 104348, Aug. 2022, doi: 10.1016/j.autcon.2022.104348.

[9] Z. Xu, N. Jha, S. Mehadi, and M. Mandal, "Automatic floor plan analysis: Datasets, methods, and applications (2000–2025)," *Autom. Constr.*, vol. 178, p. 106378, Oct. 2025, doi: 10.1016/j.autcon.2025.106378.

[10] L. Jamieson, C. F. Moreno-Garcia, and E. Elyan, "Deep Learning for Text Detection and Recognition in Complex Engineering Diagrams," in *2020 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2020, pp. 1–7. doi: 10.1109/IJCNN48605.2020.9207127.

[11] M. T. Khan, Z. Yong, L. Chen, J. M. Tan, W. Feng, and S. K. Moon, "Automated Parsing of Engineering Drawings for Structured Information Extraction Using a Fine-tuned Document Understanding Transformer," Sep. 03, 2025, *arXiv*: arXiv:2505.01530. doi: 10.48550/arXiv.2505.01530.

[12] Z. Zhang *et al.*, "From Chaotic OCR Words to Coherent Document: A Fine-to-Coarse Zoom-Out Network for Complex-Layout Document Image Translation," in *Proceedings of the 31st International Conference on Computational Linguistics*, O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, and S. Schockaert, Eds., Abu Dhabi, UAE: Association for Computational Linguistics, Jan. 2025, pp. 10877–10890. Accessed: Dec. 24, 2025. [Online].

Available: https://aclanthology.org/2025.coling-main.723/