Automatic Semantic Parsing of Structural Drawings and Rebar Schedules Based on a Hybrid Deep Learning Framework

Rong-Lu Hong[1], Jin-Bin Im[1], Lijing Xu[1], Wooshin Shim[1], Seunghyeon Wang[1] and Ju-Hyung Kim[1]

[1] Department of Architectural Engineering, Hanyang University, Seoul, Republic of Korea
abraham0814@hanyang.ac.kr, verylyzeng@hanyang.ac.kr, jbim0202@hanyang.ac.kr, js_math@naver.com, wsh1019@hanyang.ac.kr , kcr97jhk@hanyang.ac.kr

Abstract

Reinforcement detailing quality in reinforced concrete structures is directly associated with structural safety and project costs. However, manually cross-checking structural construction drawings against rebar schedules remains inefficient and highly susceptible to human error. Although demand for digital transformation in construction has become increasingly urgent, achieving high-precision automated information extraction continues to be challenging owing to the ultra-high-resolution of engineering drawings and the semantic complexity of domain-specific annotations. To address these issues, this paper proposes a staged visual semantic parsing framework tailored to structural construction drawings and rebar schedules. The proposed approach incorporates an overlapping tiling strategy to preserve local contextual information within large-scale drawings and constructs an end-to-end pipeline that integrates YOLOv11-based text detection with TrOCR-based sequence recognition, enabling accurate dense rebar annotation and tabular data extraction. Experimental results from real-world engineering datasets demonstrate that the proposed method exhibits strong robustness under high-density text conditions and complex background interference, achieving a Character Accuracy of 90.93% (CER = 0.0907) on the held-out test set, outperforming the best baseline by approximately 37.2 percentage points. These findings validate the feasibility and accuracy of leveraging deep-learning techniques for automatic structured information reconstruction from two-dimensional engineering documents, establishing a technical foundation for advancing automated quantity take off, intelligent code-compliance assessment, and digital quality management in the construction industry.

Keywords –

Structural Drawings; Rebar Schedule; Optical Character Recognition; Text Recognition

1 Introduction

The construction quality of reinforced concrete (RC) structures is a fundamental determinant of the lifecycle safety performance and service durability of infrastructure systems [1]. As a core element in ensuring structural load-bearing capacity and long-term durability, reinforcement detailing accuracy not only forms the cornerstone of quality control but also represents a critical variable in project cost management and quantity surveying [2]. Previous statistical studies have indicated that reinforcement costs account for approximately 30% of the total project cost in conventional RC structures [1]. Reinforcement deviations and the resulting rework pose substantial risks to effective cost control [3]. Consequently, in modern engineering projects characterized by a large number of components and highly complex specification systems, project managers are required to rigorously verify and document key parameters, such as reinforcement quantity, diameter, spacing, and layout patterns, as presented in structural construction drawings and rebar schedules, to ensure compliance between construction deliverables and design specifications [2,4].

However, in the digital transformation of the construction industry, automated structural data acquisition poses fundamental challenges. Although building information modeling (BIM) and computer-aided design (CAD) technologies have been widely adopted, 2D rasterized drawings, such as PDF files or scanned documents, remain the legally authoritative single source of truth in critical processes, including multi-party coordination, construction approval, and contractual delivery [4]. While CAD-to-BIM conversion theoretically provides an alternative pathway for data acquisition, its applicability is severely constrained by its strong dependence on strict drafting standards and layer conventions [5]. In complex engineering practice, heterogeneity in design habits, software version

incompatibility, and non-standardized layer management frequently result in significant semantic ambiguity when directly parsing geometric primitives [4,5]. By contrast, computer-vision-based approaches aim to emulate the cognitive processes of human engineers by interpreting information through visual features rather than file attributes [6]. Therefore, developing a generic parsing method that does not rely on specific design software or layer standards and that can deconstruct drawing content directly from the visual domain is essential for achieving true automation and robustness [7].

Although computer-vision techniques have made notable progress in general domains such as document analysis and legal contract review [8,9], their application to structural engineering drawings remains hindered by unresolved technical barriers. Existing general-purpose optical character recognition (OCR) systems are primarily designed for documents with well-aligned layouts and homogeneous backgrounds [7], whereas structural drawings exhibit distinctive characteristics, including high information density, large spatial scale, and weak semantic explicitness [10]. First, structural drawings typically possess extremely high resolutions, often exceeding 4K, and direct input into generic detection models frequently leads to graphic memory overflow or loss of small object features [11]. Second, reinforcement annotations commonly employ highly compressed domain specific syntax, such as "Φ8@200," and are often subject to severe visual overlap with dimension lines, grid lines, and background noise [1,10]. This complex spatial topology and specialized symbolic semantics combination renders generic document-processing techniques insufficient for practical accuracy in engineering scenarios [7].

To address the challenges posed by high-resolution, dense annotations, and compressed semantics in structural construction drawings, this paper proposes a staged visual semantic parsing approach that decomposes the task into text region detection and semantic recognition. In the detection stage, an adaptive overlapping tiling strategy is introduced to preserve local contextual information and reduce misdetections caused by small-scale text fragmentation [12]. Based on this strategy, a YOLOv11-based detection network was employed to achieve stable and accurate text region localization. For semantic recognition, a Transformer-based sequence recognition model was adopted to enhance intercharacter dependency modeling and improve robustness under complex background conditions [13]. By integrating YOLOv11 and TrOCR, an end-to-end parsing framework was constructed, enabling coordinated detection and recognition optimization. Experimental results confirmed the robustness and generalization capability of the proposed method in complex structural drawing scenarios.

2 Related Work

Early studies on text parsing in engineering drawings primarily relied on traditional image-processing techniques and handcrafted feature design methods such as morphological operations and connected component analysis[14]. Using this class of approaches, Ahmed et al. [15] achieved text and symbol extraction from engineering drawings, demonstrating relatively high accuracy under conditions in which drawing formats were fixed and background noise was minimal. Moreno-García et al. [14] combined rule-based reasoning with clustering algorithms to classify symbols and connection relationships in piping and instrumentation diagrams, thereby validating the feasibility of such methods in controlled environments [14,15]. However, these approaches are highly sensitive to environmental conditions, as variations in the scanning resolution, noise interference, symbol overlap, and background clutter can lead to substantial performance degradation. In practical engineering archives, in which drawing quality varies significantly, traditional methods relying on handcrafted features struggle to maintain a stable and robust parsing performance [7].

In recent years, deep-learning approaches, particularly those based on convolutional neural networks (CNNs), have been widely recognized and validated for visual analysis tasks in construction engineering [16]. Compared with traditional image-processing techniques, these methods are more effective in handling complex scenes while simultaneously improving accuracy and robustness, thereby providing strong technical support for construction informatization and intelligent decision-making [2,4]. In the context of the automatic engineering drawing parsing, previous studies commonly modeled text extraction as a two-stage pipeline consisting of text detection followed by text recognition. For example, Ravagli et al. [17] reformulated drawing text extraction as a two-stage task and employed the CTPN along with Tesseract to extract information from architectural floor plans. However, this approach exhibits limited effectiveness when dealing with non-horizontal text. To improve practical performance, Saba et al. [18] integrated an EAST detector with a CRNN-based EasyOCR module and achieved improved performance in terms of handling rotated and inclined text. Similarly, Schönfelder et al. [7] developed a recognition pipeline combining YOLOv7 and PARSeq, covering multiple types of drawing annotations and providing valuable methodological references for subsequent research.

To achieve deeper feature representation and stronger global context modeling, Zhang and He [8] introduced a method based on YOLOv10 and LORE for parsing complex engineering tables, demonstrating that deep-learning models significantly outperform traditional

approaches in terms of handling structured data. Khallouli et al. [9] further demonstrated the advantages of Transformer-based architectures, specifically TrOCR, in processing engineering drawings containing multi-lingual content and complex fonts, with a performance clearly surpassing that of conventional CNN-based baselines. However, within the specific context of RC structural drawings, previous studies have largely been limited to relatively simple tasks, such as text classification or geometric element extraction. For instance, Ma et al. [1] attempted to separate geometric and non-geometric content to extract reinforcement shapes; however, but approach remains insufficient for handling highly compressed composite semantic annotations such as "Φ8@200." Furthermore, existing general-purpose models often overlook the information loss caused by tiling large-format drawings during pre-processing and lack systematic investigation tailored to the joint interpretation of reinforcement schedules and structural drawings.

3 Material and Methods

3.1 Data collection

The dataset used in this study was collected from 31 real engineering projects, all involving the aboveground structural systems of apartment buildings across multiple provinces and cities in China. Each project comprises one structural construction drawing and one corresponding rebar schedule, yielding a total of 62 high-resolution engineering images (31 matched pairs). These projects originate from multiple design firms, encompassing a range of drafting styles and drawing conventions, thereby ensuring a degree of diversity in page layout, font style, and annotation practice within the dataset. The structural component types covered include beams, columns, walls, and foundations. The image resolution ranged from 7560 × 5174 to 15000 × 5914 pixels, reflecting the large-scale and high-information-density characteristics of structural engineering drawings.

As illustrated in Fig. 1(a), structural construction drawings are primarily used to represent the spatial arrangement of structural components and their corresponding identification labels, such as component names denoted by codes such as GBZ1. Fig. 1(b) shows the corresponding rebar schedules, which systematically describe the internal rebar configuration of each component, including the component name, elevation information, quantity and specification of longitudinal reinforcement (e.g., number of bars and diameter), type and spacing of stirrups or ties, and, in certain cases, cross-sectional details illustrating the spatial arrangement of reinforcement.

In addition to rebar-related content, the original structural drawings contained auxiliary textual information, including contractor details, client information, and general project notes. To protect project privacy and prevent irrelevant information from affecting model learning, the auxiliary regions were anonymized using mosaic processing prior to annotation. Text regions were manually annotated using the PPOCRLabel tool. A category-based annotation strategy was adopted: rebar-related semantic elements (component name, elevation, longitudinal reinforcement quantity and specification, stirrup or tie type and spacing) were annotated using green bounding boxes and served as exclusive targets for model training; auxiliary textual information was annotated using blue bounding boxes and selectively retained or excluded according to task requirements.

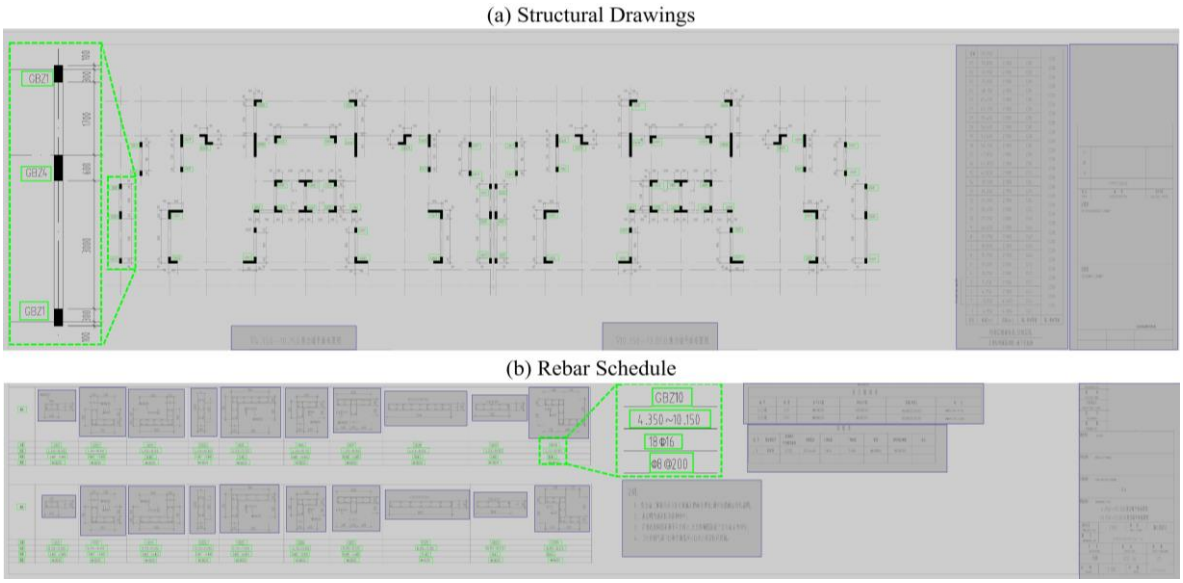

Figure 1. Example of structural engineering drawings: (a) structural construction drawing; (b) rebar detail schedule

Although cross-sectional detail drawings are graphical rather than textual in nature, their positions were also annotated and retained separately, given their potential value for downstream structural information association. To ensure consistency between the annotation data and recognition output, symbols representing different reinforcement grades were normalized to Φ during annotation, thereby reducing recognition noise caused by symbol variation and scanning inconsistencies.

3.2 Data Preprocessing Strategy

Owing to the ultra-high-resolution characteristics of the original structural construction drawings, directly feeding the entire image into a deep-learning model often results in excessive GPU memory consumption and degrades the recognition performance for dense small-scale text targets. To address this issue, as illustrated in Fig. 2, an overlap tiling strategy was introduced during the data pre-processing stage to decompose large-scale engineering drawings into manageable units [12].

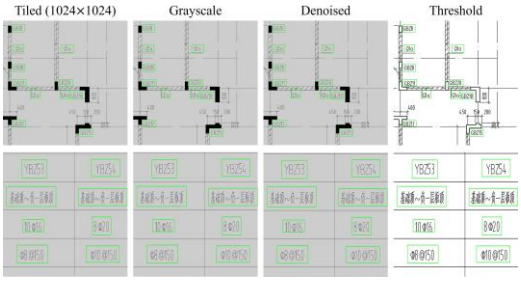

Figure 2. Example of data preprocessing.

Specifically, each original image was cropped into sub-images with a fixed size of 1024 × 1024 pixels. An overlap region of 200–300 pixels was applied between adjacent tiles to prevent text elements from being fragmented, while simultaneously reducing the risk of disrupting tabular structures or splitting component identifiers across tiles. For regions located at image boundaries where the cropped area was smaller than 1024 × 1024 pixels, white background padding was applied to achieve a uniform input size, thereby ensuring consistency during the model inference stage. Through this tiling procedure, 3510 valid cropped samples were generated from original 62 drawings.

Following spatial tiling, additional pre-processing operations tailored for text recognition tasks were applied to the cropped images, including grayscale conversion, noise suppression, and threshold-based segmentation. These steps enhance the contrast of the text regions while attenuating background interference. The described pre-processing pipeline preserves the integrity of the rebar-related semantic information and provides stable and standardized input data for subsequent text detection and Transformer-based sequence recognition models.

3.3 Hybrid Pipeline Architecture

This study proposes a staged hybrid OCR parsing architecture for structural construction drawings in which reinforcement semantic parsing is decomposed into two sequential stages: text-region detection and character-sequence recognition. This separation reduces learning complexity and enables independent optimization of each stage [17].

As shown in Fig. 3, after tiling and pre-processing, the drawings were first processed during the detection stage to localize reinforcement-related text regions and output bounding boxes. The detected regions were then cropped and forwarded to the recognition stage to generate the corresponding character sequences. This staged design effectively addresses the coexistence of dense text distributions and highly compressed semantic expressions typical of structural construction drawings.

3.3.1 Detection Stage

The detection stage employs the YOLOv11 Small model to localize text regions in structural drawings [19]. Model parameters were initialized using yolo11s.pt weights pre-trained on the COCO dataset. As a single-stage detection architecture, YOLOv11 maintains

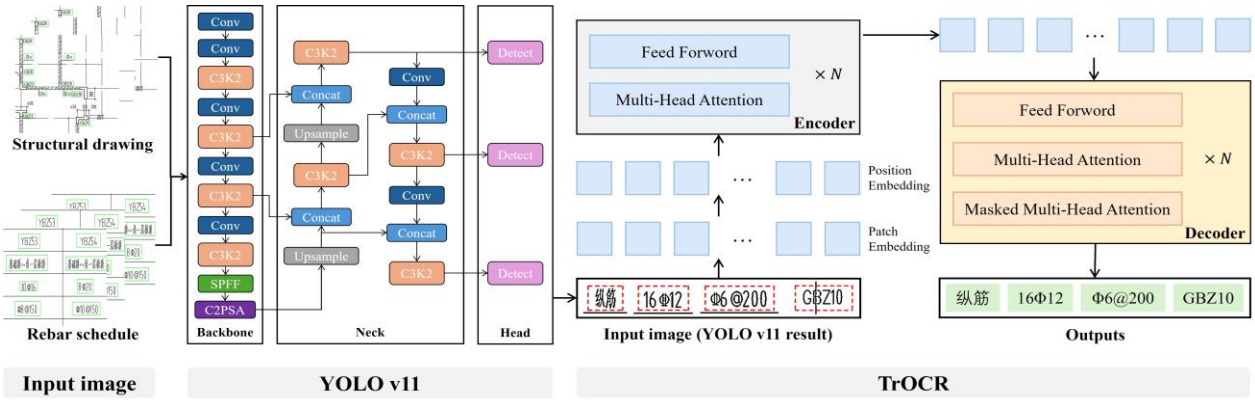

Figure 3. Hybrid Pipeline Architecture for Drawing OCR Analysis

reliable localization performance for small-scale and densely distributed text targets, making it suitable for engineering drawing scenarios. Text detection was formulated as a single-class object-detection task, allowing training to focus on spatial localization accuracy without category-level ambiguity.

3.3.2 Recognition Stage

In the recognition stage, a Transformer-based TrOCR model is adopted to perform character sequence modeling of the text regions produced by the detection stage. By constraining the recognition task to pre-detected text regions, the proposed framework substantially reduces the interference of background structures on character modelling.

The TrOCR model is initialized using a hybrid strategy [13]. The visual encoder is loaded from pre-trained Microsoft/trocr-base-printed weights, providing visual feature representations with a strong generalization capability. By contrast, the text decoder is initialized from scratch using a lightweight transformer architecture specifically adapted to the highly regularized and syntactically compact reinforcement annotations in structural drawings. This design leverages pre-trained visual representations while avoiding the modelling redundancy of general-purpose language model decoders in domain-specific scenarios.

3.4 Experimental Strategy

All experiments were conducted on a Windows 11 Pro platform using Python 3.9 and PyTorch 2.8.0. The hardware configuration consisted of an NVIDIA GeForce RTX 5080 GPU and an AMD Ryzen 9 9950X processor. Separate datasets and training strategies were designed for the detection and recognition stages to reflect their distinct task characteristics.

3.4.1 Detection Model Training

The text-detection model was trained using a dataset generated through the tiling procedure, which contains 3510 cropped engineering drawing images. The dataset was divided into training, validation, and test sets in a ratio of 8:1:1 to ensure sufficient training data while enabling independent performance monitoring and final evaluation. The model was trained for up to 100 epochs using an early stopping strategy based on the validation performance. Training was terminated if no improvement was observed for 20 consecutive epochs to mitigate overfitting under limited-sample conditions. Owing to the GPU memory constraints associated with the 1024 × 1024-pixel inputs, the batch size was set to eight. The default YOLO optimization configuration was adopted, including built-in learning-rate warm-up and decay mechanisms, to facilitate stable transfer-learning adaptation to the text-localization task [19].

The detection performance was evaluated using the mean average precision (mAP), including mAP50 and mAP50:95. [6]mAP50 corresponds to an intersection over union (IoU) threshold of 0.50, whereas mAP50:95 represents the average precision computed over IoU thresholds ranging from 0.50 to 0.95 with a step size of 0.05 [10]. IoU is defined as

$$IoU = \frac{Area\left(B_{pred} \cap B_{gt}\right)}{Area\left(B_{pred} \cup B_{gt}\right)} \tag{1}$$

where B_{pred} and B_{gt} denote the predicted and ground-truth bounding boxes, respectively. The detection was considered correct when the IoU exceeded a given threshold. AP is defined as the area under the precision–recall curve, as follows:

$$AP = \int_0^1 p(r)dr \tag{2}$$

As text detection was formulated as a single-class detection task in this study, the mAP was equivalent to the AP of this class under the corresponding IoU threshold.

3.4.2 Recognition Model Training

The training data for the text-recognition model were generated by cropping the text region output from the detection stage, yielding 11427 character-level annotated samples. The dataset was divided into training, validation, and test sets in the same 8:1:1 ratio. The recognition model was trained for 50 epochs by using a staged training strategy. During the first 10 epochs, the visual encoder was frozen and only the decoder was trained. Subsequently, the higher encoder layers were partially unfrozen, and the learning rate was reduced to one-fifth of its initial value to enhance domain adaptation while preserving the pre-trained visual features.

The training employed the AdamW optimizer with an initial learning rate of 1×10^{-4} and a batch size of 32. Weight decay and label smoothing were applied, and early stopping based on the validation set character error rate (CER) was used, with training terminated after 10 consecutive epochs without improvement.

Recognition performance was evaluated using the CER and character accuracy (CA), respectively defined as

$$CER = \frac{S + D + I}{N} \tag{3}$$

$$CA = 1 - \frac{S + D}{N} \tag{4}$$

where *S*, *D*, and *I* denote the number of substitutions, deletions, and insertions, respectively, and *N* is the total number of characters in the ground-truth text.

4 Results

4.1 Text Detection Results

The training process was systematically analyzed to evaluate the convergence behavior and localization performance of the text-detection model. Training was terminated early at epoch 61 on the basis of validation performance. As illustrated in Fig. 4, the box loss and DFL loss for both the training and validation sets decreased steadily throughout the optimization process, with no observable divergence between the two curves, confirming the stability of convergence. The detection model surpassed an mAP of 0.95 at IoU = 0.50 within the early training epochs, and the stricter mAP50:95 metric reached approximately 0.636 toward the final stage, indicating that the model attains reliable performance in both coarse localization and precise boundary regression.

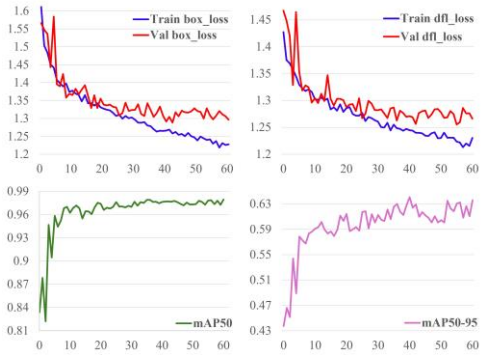

Figure 4. Training curves of the detection model

Fig. 5 presents representative detection results on the test set. The predicted bounding boxes exhibit a high degree of spatial overlap with the ground truth annotations, with no notable false positives or missed detections observed, confirming the effectiveness of the adopted detection strategy in the context of engineering drawings and providing a reliable input foundation for the subsequent text recognition stage.

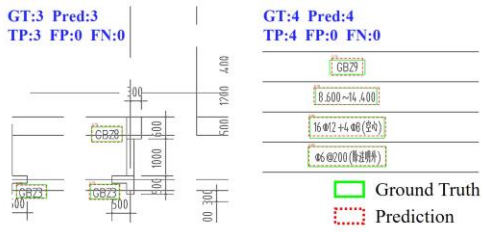

Figure 5. Typical examples of detection test results

4.2 Text Recognition Results

The performance of the text-recognition module is analyzed following the introduction of enhanced regularization strategies. As shown in Fig. 6, the training process converged stably, with the training loss and validation loss decreasing in parallel while maintaining a narrow gap. The validation CER reached its optimal value of 0.0281 (CA = 97.19%) at epoch 7. After unfreezing the top two encoder layers at epoch 11, no appreciable performance degradation was observed. Training was terminated early at epoch 17 based on validation performance, indicating that the model achieves favorable convergence and generalization under the combined regularization and staged unfreezing strategy.

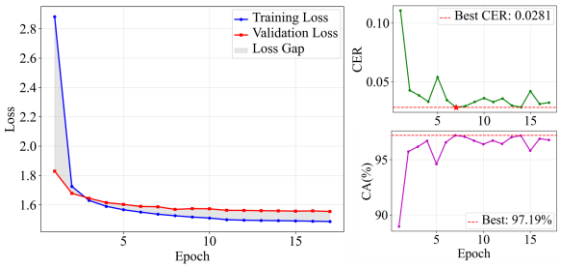

Figure 6. Training curves of the text recognition

To evaluate the effectiveness of the proposed pipeline, three representative open-source OCR methods (docTR, EasyOCR, and Tesseract) were employed as baselines for comparative experiments. None of the baseline methods incorporated a dedicated object detection module; instead, they applied the same sliding window strategy to segment the original engineering images into patches for direct recognition. The results are summarized in Table 1. Among the three baselines, EasyOCR achieved the best performance, yet its CER remained as high as 0.4622 (CA = 53.78%), with a precision of only 0.196, indicating that patch-based recognition using fixed windows can improve text recall but still introduces substantial background noise and redundant detections in the absence of semantic-level text localization, falling short of the high-accuracy recognition demands of engineering drawings. The proposed YOLOv11 + TrOCR pipeline, through a staged detect-then-recognize strategy, achieved a CA of 90.93% (CER = 0.0907) on the held-out test set, representing an improvement of approximately 37.2 percentage points over the best baseline EasyOCR (CA = 53.78%), fully demonstrating the necessity and effectiveness of incorporating a semantic-level dedicated detection module for structural drawing recognition.

Table 1. Comparison with Baseline OCR Model

Model	Tesseract	docTR	EasyOCR	**Ours**
CER	0.9341	0.6791	0.4622	**0.0907**
CA	6.59	32.09	53.78	**90.93**

Table 2 presents representative recognition results on the test set along with their corresponding error types. For standard structural annotation texts that appear frequently in engineering drawings, such as rebar diameter notations 18Φ14, component identifiers GBZ4, and stirrup spacing expressions Φ8@200, the model achieved zero-character error rate, indicating stable and reliable recognition capability for conventional annotation text.

Table 2. Typical examples of text recognition test results

Image	GT	Pred	CER	Error Type
	18Φ14	18Φ14	0	-
	编号	编号	0	-
	GBZ4	GBZ4	0	-
	Φ8@200	Φ8@200	0	-
	8.700~ 52.200	8.700~ 2.200	0.154	Digit Missing
	2Φ12+ 6 Φ14 (实心)	2Φ12+ 6 Φ14 (实必)	0.188	Char Confusion
	-2.520	-25.20	0.333	Decimal Shift

To further identify directions for model optimization, a character-level error decomposition was conducted on the 131 non-exact-match samples. It should be noted that the majority of these samples contained only one-to-two-character deviations, and the overall CER remained at a low level. Among a total of 304 edit operations, Digit Missing accounted for the highest proportion (148 occurrences, 48.7%), primarily manifesting as the omission of individual digits by the model and affecting 15.9% of all test samples. Decimal Shift ranked second (52 occurrences, 17.1%, affecting 8.7%), mainly attributable to the high visual similarity between elevation values and dimensional annotations, which makes it difficult for the model to distinguish between the two based solely on local image features. Char Confusion and special symbol errors together accounted for only 10.5% of all edit operations, occurring infrequently yet carrying relatively notable impact at the semantic level. Other errors (72 occurrences, 23.7%) were predominantly caused by missing or extraneous spaces and did not affect practical semantic parsing. The above analysis indicates that the residual recognition deviations of the current model are primarily concentrated in numerically dense short texts, and performance can be further improved through the introduction of structured numerical constraints or domain-adaptive post-processing strategies.

5 Conclusions

This study proposed a staged visual semantic parsing framework for 2D rasterized structural construction drawings, targeting the automated extraction of reinforcement information from structural construction drawings and rebar schedules. A high-resolution engineering drawing dataset comprising 62 drawings, was constructed, and an overlap tiling strategy was employed to process large-format images, effectively alleviating GPU memory constraints and text fragmentation.

From a methodological perspective, a hybrid OCR pipeline integrating YOLOv11 and TrOCR was developed in which reinforcement semantic parsing was decomposed into text-detection and -recognition stages. Experimental results demonstrate that the text-detection model can stably localize dense text regions under complex structural drawing backgrounds, thereby providing reliable inputs for the recognition stage. The text recognition model achieves a CA of 90.93% (CER = 0.0907) on the held-out test set, representing an improvement of approximately 37.2 percentage points over the best baseline method EasyOCR (CA = 53.78%) under the same sliding window strategy. Analysis of representative samples showed that the model accurately recognized high-frequency structural annotations, including reinforcement diameters, component identifiers, and spacing expressions, whereas recognition errors were primarily concentrated in complex text instances characterized by dense numerical content or highly compressed semantics.

Despite achieving stable recognition performance, several limitations remain. This study normalized different reinforcement grades using the symbol Φ to ensure recognition consistency, without performing fine-grained differentiation among different grade symbols. Furthermore, the current scope is confined to character-level text detection and recognition, and does not yet encompass the reconstruction of structured engineering semantics or the cross-drawing information association between structural construction drawings and rebar schedules. To bridge this gap, future research will proceed along four stages: (1) structured information reconstruction, parsing raw OCR output into structured fields (component name, rebar diameter, spacing, and quantity) based on domain-specific grammatical rules (e.g., the KS D 3504 reinforcement notation standard) and regular expressions; (2) component-level entity matching, using recognized component identifiers (e.g., GBZ1) as indices to establish one-to-one or one-to-many cross-drawing mapping relationships with corresponding rebar schedule entries; (3) compliance verification, automatically comparing matched structural information against design codes to verify mandatory requirements such as minimum rebar spacing and concrete cover

thickness; and (4) automated quantity extraction, calculating the reinforcement quantity for each component to support quantity surveying and cost management. Future work will also expand the fine-grained recognition of reinforcement-grade symbols and validate the generalization capability of the model across a broader range of engineering project types and drafting standards.

Acknowledgement

This research was funded by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (Ministry of Science and ICT) (grant number RS202400356697).

References

[1] Z. Ma, Q. Zhao, Y. Zhu, T. Cang, and X. Hei. An Automatic Extraction Method of Rebar Processing Information Based on Digital Image. Mathematics, 10:2974, 2022.

[2] U. Woo, M. Lee, T. Lee, H. Choi, S.-M. Kang, and K.-K. Choi. Real-time rebar spacing measurement system for quality control in construction. Autom. Constr., 171:106012, 2025.

[3] X. Yuan, A. Smith, F. Moreu, R. Sarlo, C. D. Lippitt, M. Hojati, S. Alampalli, and S. Zhang. Automatic evaluation of rebar spacing and quality using LiDAR data: Field application for bridge structural assessment. *Autom. Constr.*, 146:104708, 2023.

[4] Y. Xiang, A.-M. Mahamadu, and L. Florez-Perez. Engineering information format utilisation across building design stages: An exploration of BIM applicability in China. *J. Build. Eng.*, 95:110030, 2024.

[5] C. Zhang, A.-X. Zhu, L. Zhou, M. Che, and T. Qiu. Constraints for Improving Information Integrity in Information Conversion From CAD Building Drawings to BIM Model. *IEEE Access*, 8:81190–81208, 2020.

[6] S. Wang, S. Park, J. Kim, and J. Kim. Safety Helmet Monitoring on Construction Sites Using YOLOv10 and Advanced Transformer Architectures with Surveillance and Body-Worn Cameras. *J. Constr. Eng. Manag.*, 151:04025186, 2025.

[7] P. Schönfelder, F. Stebel, N. Andreou, and M. König. Deep learning-based text detection and recognition on architectural floor plans. *Autom. Constr.*, 157:105156, 2024.

[8] Z. Zhang and Y. He. Research on Intelligent Verification of Equipment Information in Engineering Drawings Based on Deep Learning. *Electronics*, 14:814, 2025.

[9] W. Khallouli, M. S. Uddin, A. Sousa-Poza, J. Li, and S. Kovacic. Leveraging Transformer-Based OCR Model with Generative Data Augmentation for Engineering Document Recognition. *Electronics*, 14:5, 2025.

[10] Y. Man, Q. Zhang, Y. Cai, and Z. Chang. A Target Detection Algorithm for Marks Layout in Detail Drawings of Steel Structure. *Int. J. Steel Struct.*, 25:557–570, 2025.

[11] L. Jamieson, C. F. Moreno-Garcia, and E. Elyan. Towards fully automated processing and analysis of construction diagrams: AI-powered symbol detection. *Int. J. Doc. Anal. Recognit. IJDAR*, 28:71–84, 2025.

[12] A. Rezvanifar, M. Cote, and A. B. Albu. Symbol Spotting on Digital Architectural Floor Plans Using a Deep Learning-based Framework. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 2419–2428, Seattle, WA, USA, 2020.

[13] M. Li, T. Lv, J. Chen, L. Cui, Y. Lu, D. Florencio, C. Zhang, Z. Li, and F. Wei. TrOCR: Transformer-Based Optical Character Recognition with Pre-trained Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 37:13094–13102, Washington, DC, USA, 2023.

[14] C. F. Moreno-García, E. Elyan, and C. Jayne. New trends on digitisation of complex engineering drawings. *Neural Comput. Appl.*, 31:1695–1712, 2019.

[15] S. Ahmed, M. Liwicki, M. Weber, A. Dengel, Automatic Room Detection and Room Labeling from Architectural Floor Plans, in: 2012 10th IAPR Int. Workshop Doc. Anal. Syst., 2012: pp. 339–343.

[16] S. Wang, M. Kim, H. Hae, M. Cao, and J. Kim. The Development of a Rebar-Counting Model for Reinforced Concrete Columns: Using an Unmanned Aerial Vehicle and Deep-Learning Approach. *J. Constr. Eng. Manag.*, 149:04023111, 2023.

[17] J. Ravagli, Z. Ziran, and S. Marinai. Text Recognition and Classification in Floor Plan Images. In *Proceedings of the 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, pages 1–6, Sydney, Australia, 2019.

[18] A. Saba, R. Hantach, and M. Benslimane. Text Detection and Recognition from Piping and Instrumentation Diagrams. In *Proceedings of the 2023 8th International Conference on Image, Vision and Computing (ICIVC)*, pages 711–716, Dalian, China, 2023.

[19] R. Khanam and M. Hussain. YOLOv11: An Overview of the Key Architectural Enhancements. *arXiv preprint arXiv*:2410.17725, 2024.