AI Based Blueprint Error Detection

Assist Prof.Kavita Reddy[1], Deepak T[2], Ganesh Y V[3], Felix xavier[4]
Department of Computer Science-Artificial Intelligence and Machine Learning,AMC engineering college, Bengaluru, India

***Abstract*—Persistent errors in technical documentation, which result in expensive rework, safety risks, and project delays, pose serious operational and financial risks to the Architecture, Engineering, and Construction (AEC) sector. [1] This paper suggests an advanced, integrated deep learning framework for automated blueprint quality control as a solution to this problem. The system uses a hybrid Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) based Optical Character Recognition (OCR) path for the accurate extraction of semantic annotations and dimensions, in conjunction with an optimized single-stage object detection architecture (YOLO) for reliable geometric entity localization. [2] A spatial-semantic Late Multimodal Fusion technique that aligns disparate visual and textual features is the main methodological advancement. A deterministic Rule-Based Constraint Validation Engine then uses this aligned data to transparently verifyA deterministic Rule-Based Constraint Validation Engine then uses this aligned data to transparently check for geometric conflicts, dimensional inconsistencies, and design code compliance. [3] With an overall Macro Average F1Score of 95.5%, a thorough evaluation shows how effective the system is. This balanced performance highlights the model's capacity to reduce both time-consuming false positives and extremely harmful false negatives (missed critical errors), providing a clear and dependable route toward digital transformation in documentation review. [4,5]**

***Index Terms*—YOLO, Multimodal Fusion, Optical Character Recognition (OCR), Computer Vision, Document Image Analysis, Architectural Engineering, Deep Learning, Design Constraint Checking, and Anomaly Detection.**

I. INTRODUCTION

A. Context and Motivation in the AEC Sector
The accuracy and consistency of technical documentation, also known as blueprints or construction documents, are crucial to the Architecture, Engineering, and Construction (AEC) industry's ability to meet strict deadlines and navigate intricate regulatory frameworks. One of the main sources of vulnerability for these documents is the conventional manual review process, which frequently leads to systemic errors that spread throughout the project lifecycle. [1] These documentation errors are more than just administrative annoyances; they set off a disastrous chain reaction that results in significant project setbacks like delays, significant rework, budget overruns, and increased safety risks. [1, 6]
Several typical error typologies that are present in manual drafting and review processes are identified through analysis of construction failures. These include, but are not limited to, using out-of-date or "one-size-fits-all" drawings that have been copied from earlier projects without being adjusted to current specifications; missing or incomplete critical information as a result of insufficient quality control; inconsistent data between different drawing views, schedules, or specifications; and overall ambiguity resulting from ambiguous drawing standards. [6] These problems, which stem from inadequate coordination and gaps in documentation, underscore the necessity of systemic process enhancements. [1] Adopting tactics and resources that put accuracy, consistency, and effective teamwork first throughout the whole documentation process is necessary to mitigate these high-impact errors. [1]
The intricacy of AEC documentation necessitates AI capabilities that can identify minute but significant discrepancies. Errors often fall into three categories: Coordination/Consistency (e.g., contradicting information), Annotation/Specification (e.g., inadequate detailing), and Geometric/Structural (e.g., clashes). [6] All three must be successfully addressed by an automated system. In particular, coordination errors require a solution that compares textual metadata (dimensions, specifications) with visual data (geometry). Because of this requirement,

straightforward unimodal computer vision detection is insufficient; in order to identify systemic conflicts, the solution must simultaneously capture design elements and the properties that go along with them..

B. Conventional Blueprint Review's Drawbacks Document review's inherent reliance on human expertise and manual execution is the main obstacle. [7] Traditionally, technical drawings must be manually analyzed by technicians in order to extract crucial measurements and transcribe these dimensional data points into production or quality control management systems. [8] Significant difficulties are brought about by this manual reliance, such as a high risk of transcription errors, scalability issues when managing large volumes of complex documents, which lead to significant cost overruns, and coordination gaps when information is dispersed among stakeholders. [7]

This reliance has obvious consequences: manual interpretation is costly, time-consuming, and inherently prone to error. The fast iteration cycles required by contemporary construction schedules are incompatible with the slow pace. [8] To speed up information entry and enhance overall production management, it is essential to implement systems that automate the extraction of technical data directly from the file, removing transcription errors. [8]

C. Synopsis of AI Technical Document Analysis Solutions

In the field of document image analysis, artificial intelligence (AI) and deep learning have become revolutionary technologies. AI systems can automate the historically difficult tasks involved in reading technical drawings by combining Computer Vision (CV), Machine Learning (ML), Optical Character Recognition (OCR), and Natural Language Processing (NLP). [7] The system can examine blueprints to precisely identify structural elements like walls, doors, windows, and columns thanks to AI image recognition capabilities. It can even help generate cost estimates for these elements. [9] This automation centralizes data organization, significantly reduces human error, and speeds up turnaround times while offering project managers useful insights. [7]

1)For applications ranging from internal catalog checks to quicker turn-ins and cost control, artificial intelligence (AI) offers the potential for complete automation, achieving better-than-human accuracy in data detection and extraction. [8, 10] Combining these tools speeds up design review and offers predictive analytics that can foresee problems before they result in expensive setbacks. [9] D. The Proposed System's Structure and Contribution In order to achieve superior automated blueprint error detection, this paper describes the design, implementation, and rigorous evaluation of an integrated deep learning framework. This study makes three main contributions:

2) Robust Multimodal Feature Extraction: The creation of parallel processing streams using a specialized, post-processed OCR model for high-fidelity extraction of semantic annotations and an optimized CNN (YOLO) for geometric entity detection.

3) Spatial-Semantic Late Fusion: This innovative, reliable late-fusion method aims to detect conflicting information errors by spatially aligning and reconciling the extracted textual and visual features.

1) Explainable Constraint Validation: This approach satisfies the nonnegotiable requirement for regulatory transparency in engineering applications by integrating a transparent Rule-Based Constraint Validation Engine that offers auditable detection logic for geometric conflicts and design code compliance. [3]

The paper is organized as follows: The pertinent literature on both historical and contemporary document analysis methods is reviewed in Section 2. The system implementation is explained in detail in Section 3, including the data paths, fusion mechanism, and validation logic. The experimental setup, performance metrics, and quantitative findings are shown in Section 4. Section 5 concludes by summarizing the contributions and outlining potential directions for future research.

II. RELATED WORKS / LITERATURE REVIEW

A. Initial Techniques for Geometric Recognition and Raster-to-Vector Conversion Early attempts to transform digitized analog raster images into structured vector formats appropriate for Computer-Aided Design (CAD) systems laid the groundwork for automated technical drawing analysis. [11] In order to extract increasing semantic meaning from the raw pixel data, these conventional systems used a multi-level recognition hierarchy. [11] In order to convert the digital image into a CAD document made up of lines and shapes, the first level concentrated on the

Recognition of Geometric Elements. [11] This successfully recovered basic graphic elements in orthographic projections by identifying line segments and grouping them into two-dimensional shapes using methods like Edge Following and Chain Coding. [11]
The recognition of building elements was the focus of the second level, which abstracted geometric primitives into semantic design entities, such as recognizing a collection of lines as a "wall" or "column." [11] The Recognition of Spatial Articulation level was the third and hardest. The goal of this level was to identify the "void" spaces and how they were arranged, bringing to light the basic topological elements of a design that are only hinted at in the visual representation. [11] A lot of preprocessing was needed for this function, which is essential for complex constraint checking. [11] In order to prepare the image, reduce line segments to unity thickness, fill in any tiny gaps, and normalize unintentional perturbations, techniques like thinning or skeletonization were essential. [11, 12] By eliminating unnecessary details, such as a particular wall
thickness, the system could concentrate on making inferences. In order to convert graphical data into a topological representation, recognition was frequently based on template matching for particular junction types and examining the connectivity of these junctions. [11] The Rule-Based Logic (RBL) component is functionally required to formalize the topological and dimensional checks that were previously handled by explicit geometric and edge-following algorithms; modern systems must reach this same level of semantic abstraction. B. Document Layout Analysis (DLA) Using Deep Learning Paradigms Document Layout Analysis (DLA), a basic task involving the detection, segmentation, and relational analysis of physical blocks (text, figures, tables) within document images, has been transformed by deep learning. [13]
The sequential, two-stage approach frequently employed in previous methods is a major challenge in DLA, especially for complex documents like blueprints. [14] These methods perform document understanding by combining the text results with image features after first detecting and recognizing text using commercial OCR systems. [14] The main drawback is that errors in text reading build up early in the pipeline, which spreads and amplifies errors downstream. [14, 15]
Current research increasingly supports multimodal approaches that combine textual and visual elements in a synergistic way to overcome this. Aligning features at the pixel and block levels greatly improves performance on complex layout analysis datasets, frequently surpassing unimodal detectors, as models such as M2Doc show. [13] This demonstrates that in order to capture complex relationships and achieve superior layout detection accuracy for architectural documents, it is necessary to analyze textual (semantic) and visual (geometric) data separately before fusion.

C.A Comparative Analysis of Object Detection Systems

When choosing an object detection architecture in computer vision, there is a crucial trade-off between inference time and detection accuracy.[16] Faster R-CNN and other two-stage detectors use selective region proposals to achieve high accuracy, but they have complicated architectures and long inference times. Inference times from single-stage detectors, such as YOLO (You Only Look Once), can be up to 300 times faster than those from two-stage detectors because they process entire images in a single shot.[16]
Inference speed is crucial for high-throughput industrial quality control in AEC. In comparison to two-stage detectors, early YOLO versions frequently lagged slightly in detection accuracy; however, recent architectural developments have greatly reduced this gap and occasionally achieved superior accuracy.[16]
Combining intricate spatial-temporal By strengthening the model's capacity to understand object relationships, relationship modeling—such as that accomplished by applying Dynamic Graph Neural Networks (DGNN) to YOLO—further improves performance metrics (mAP@0.5 and precision), which is crucial for identifying geometric inconsistencies in drawings. [17] Because of its speed and contemporary architectural improvements that guarantee high detection fidelity, the single-stage architecture is still the best option.

D. Document Intelligence and Multimodal Fusion Techniques

For complex document analysis, where context depends on both visual structure (geometry) and semantic content (text), multimodal learning—which seeks to leverage information from various data sources—is essential. [18, 19] The main objective of feature fusion is to handle incomplete or noisy data, where the robustness of one modality (such as highly accurate visual object detection) can offset the reliability of another (such as noisy OCR output). [18] Fusion strategies are typically divided into three categories: Early Fusion, Late Fusion, and Hybrid methods, depending on where in the processing pipeline they take place. [19] Prior to being processed by modality-specific models, Early Fusion merges raw or low-level features. It is extremely sensitive to input noise and runs the risk of overcomplicating the input, even though it captures fine-grained interactions. [18] The outputs or high-level processed representations from separate, specialized models are combined to perform late fusion. [18, 19] This method reduces the risks of error propagation that are typical in sequential pipelines while maintaining modality-specific processing integrity. [14]

For instance, the M2Doc model effectively aligns textual and visual features using fusion modules at both the pixel and block levels, showing that this integrated approach greatly enhances layout detection performance across complex datasets. [13] Late Fusion is the best approach for blueprint analysis, where geometric and dimensional data are extracted by essentially different algorithms. It aligns the final, high-fidelity embeddings for final decision-making while enabling specialized pre-processing for both vision and OCR paths.

E. Optical Character Recognition for Technical Drawings: Obstacles and Solutions Conventional optical character recognition (OCR) systems frequently struggle with the particular complexity of technical drawings because they are primarily made for natural scenes or straightforward document layouts. [2] Blueprints offer particular

challenges include a variety of fonts and text scales, non-standard symbols, dense and overlapping text placements, and poor image quality resulting from scanning or low-resolution analog copies. [2,14]

The reliance on a two-stage sequential approach—first, text detection and recognition through external OCR, and second, document understanding based on the fusion of these potentially error-ridden text results with image features—was a common flaw in earlier document understanding systems. [14] During the downstream conversion and analysis process, errors that have accumulated during the initial text reading stage will unavoidably spread and get worse. [15]

Recent research suggests hybrid architectures, like the CNN-LSTM framework designed for robust text extraction, to address the high error rate. [2] While LSTMs offer sequential modeling capabilities, which are essential for deciphering word context and reading order in intricate layouts, CNNs are superior at feature extraction from image patches (character recognition). Additionally, specialized post-processing is essential. To find and highlight discrepancies in the OCR output, error detection techniques like pattern matching (regular expressions for structural output), statistical analysis, and linguistic analysis are used. [15, 1] To consistently convert extracted dimensional strings (like "12'-6") into structured variables (like "wall length = 3810mm") appropriate for machine-readable constraint validation, this last correction step is required.

1. F. Constraint Validation and Anomaly Detection Using Rules The detection logic needs to be auditable and explainable for high-stakes applications like code compliance and structural integrity. [3] Deep learning models are excellent at extracting features and identifying patterns (Anomaly Detection blueprints like Isolation Forest can handle high-dimensional data [20]), but they usually lack the transparency needed for regulatory compliance, which creates the "black box" issue. Consequently, a Rule-Based Constraint Validation Engine (RBCVE) must be included in the last step of error detection. [3] Deterministic logic and code rules that match human-understandable specifications, like design codes or geometric constraint specifications, form the foundation of the RBCVE. [3, 21] These guidelines look for:

2) Geometric Conflicts: For instance, a door positioned inside a load-bearing column or two structural elements occupying the same spatial coordinates.

3) Dimensional Inconsistencies: For instance, a wall segment's total dimensions as determined by the OCR do not match the sum of its partial dimensions.

4) Code Compliance: For example, confirming that the minimum fire separation requirements are met by the specified wall thickness (extracted using OCR and entity classification). Similar to manually-written rules, the explainable nature of these synthesized code rules offers transparent detection logic and enables

effective, lightweight online detection execution without the computational overhead of continuous deep learning inference for the final check. [3] This hybrid method combines the unavoidable transparency of formalized validation logic with the speed and precision of deep feature extraction.

III. IMPLEMENTATION

Computer vision, multimodal fusion, and deterministic constraint validation are all integrated into the robust, distributed deep learning architecture of the suggested AI-Based Blueprint Error Detection System. The Visual Stream, which detects geometric elements, and the Semantic Stream, which extracts textual and dimensional annotations, are the two parallel, interdependent streams that make up the system workflow. A spatially-aware Late Fusion mechanism that feeds a Rule-Based Constraint Validation Engine makes the final decision. Fig. 3.1 shows the overall architecture.

A. Preprocessing and Augmenting Data
Careful data preparation is necessary for technical drawing analysis in order to standardize the highly variable input documents (e.g., different scan qualities, line weights, and resolutions).[22] Initial Image Preparation: Noise reduction and contrast adjustment are the main goals of the initial image processing applied to the input raster image (PDF or high-resolution TIFF). [12, 22] In order to create a cleaner document surface that greatly increases the interpretation accuracy for both the vision and OCR components, noise removal is essential because it removes spurious pixel artifacts. [12]

Geometric Normalization (Skeletonization): Thinning, also known as Skeletonization, is a particular preprocessing method used for the Visual Stream. [11] Through this process, lines and contours are reduced to unity thickness, or a width of one pixel. [22] For later geometric analysis, this abstraction is crucial because it standardizes components like walls anddetail lines, eliminating the need for particular line weights or drafting

Fig 3.1 SSystem Architecture Block Diagram

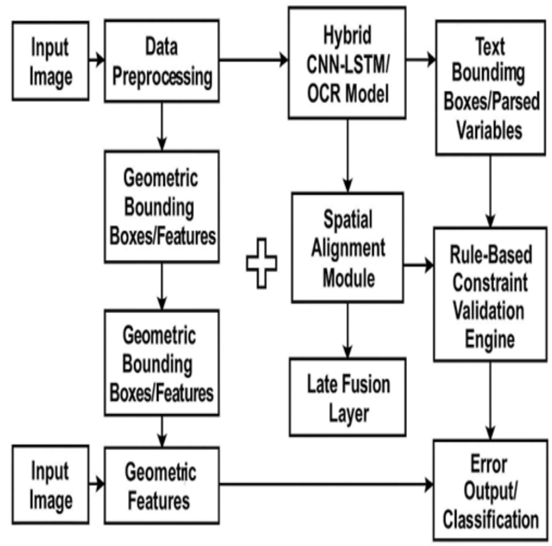


Fig. 3.1 System Architecture Block Diagram

(Placeholder: A block diagram showing the flow. Left side: Input Image → Data Preprocessing → CNN/YOLO Model
→ Geometric Bounding Boxes/Features. Right Side: Input Image → Data Preprocessing → Hybrid CNN-LSTM/OCR Model → Text Bounding Boxes/Parsed Variables. Center: Geometric Features + Parsed Variables → Spatial Alignment Module → Late Fusion Layer → Rule-Based Con-straint Validation Engine → Error Output/Classification.)

styles when identifying topological relationships like corners and intersections. [11] Thinning allows the system to concentrate on the basic spatial articulation of the design while eliminating secondary components (such as particular door types, wall thickness indicators) that are not necessary for topological recognition.[11]
Enhancement of Data: The training dataset is enhanced using common computer vision techniques, such as random rotation, horizontal/verticalflipping, noise injection (replicating low-quality scans), and controlled affine transformations. This procedure improves the model's capacity for out-of-distribution blueprint handling and generalization. B.Text Extraction from OCR A unique Hybrid CNN-LSTM framework designed for technical text extraction is used by the Semantic Stream. [2] Extraction of Features: To extract local image features, character shapes, spatial context, and font

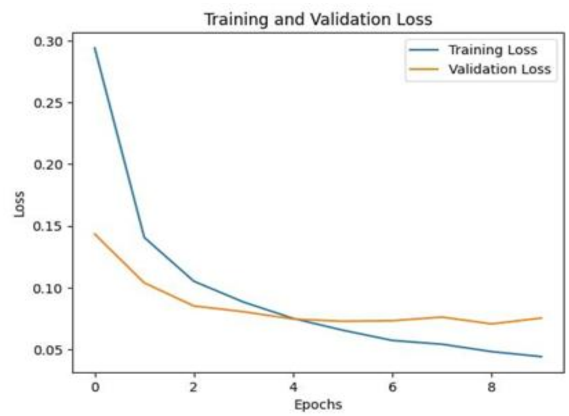
characteristics within text are captured using a sequence of Convolutional Layers (CNN).
areas recognized by the first stage of object detection (shared or dedicated).[23] Sequential Decoding: A Long Short-Term Memory (LSTM) network receives the CNN's extracted features. For sequential data modeling, the LSTM is very useful because it enables the system to interpret characters in the proper reading order and take advantage of contextual information across words and dimension strings (e.g., recognizing that "12'-6" is a single dimensional quantity). [2]

1) Post-Processing and Variable Translation: The OCR model's raw text output is extremely erratic and necessitates post-processing. [15, 1] For the Rule-Based Constraint Validation Engine, this critical step converts textual strings into quantitative variables that can be read by machines.
   1) Error Detection: To find common dimension or unit inconsistencies, pattern-matching techniques utilizing regular expressions are used (e.g., flagging '12' 6' as a potential error). Deviations from expected formats are found using statistical models (for dimensional ranges) and linguistic analysis (for specification notes).

2) Correction and Normalization: Contextual data and a domain-specific lexicon are used to fix errors that are found. All dimensional strings are then stored as floating-point variables after being normalized into a standard metric unit (such as meters or millimeters). For instance, the variable Door 1 Width = 1219.2 mm is created from the OCR output "4'-0" for a door width.

C. CNN-Based Extraction of Features
For extremely effective and precise localization and classification of geometric entities, the Visual Stream uses a customized single-stage object detection architecture, namely a variant of YOLO (You Only Look Once). [16] Single-Stage Advantage: The requirement for high-throughput, low-latency processing, which is crucial for industrial quality control, drives the selection of a single-stage detector. [16] YOLO achieves inference times that can be hundreds of times faster than conventional two-stage detectors by processing the entire input image once in order to predict bounding boxes and class probabilities simultaneously. [16] Improvement of Architecture (DGNN-YOLO Context): In order to preserve high detection accuracy, especially for spatially intricate tasks such as blueprint analysis, the conventional YOLO
Spatial relationship modeling-inspired components are added to the backbone. The model integrates similar ideas through multi-scale feature aggregation and improved Non-Max Suppression (NMS) logic that prioritizes detection consistency based on predicted geometric relationships (e.g., a wall and an intersecting dimension line must be adjacent), even though a full Dynamic Graph Neural Network (DGNN) integration may increase complexity. [17] This guarantees the robustness and high localization of the geometric features, such as bounding box coordinates, class labels (Wall, Window, Column, etc.), and confidence scores.

Fig 3.2 Training Accuracy and Loss Graphs



(Placeholder: A graph showing Training Loss (y-axis) decreasing and Validation Loss (y-axis) decreasing/stabilizing over Training Epochs (x-axis). A second graph showing Training Accuracy (y-axis) increasing and Validation Accuracy (y-axis) increasing/stabilizing over Training Epochs (x-axis).)

D. Training and Optimizing Models Labeling and Dataset: A proprietary dataset of construction documents that has been carefully labeled for both geometric entities (Visual Stream) and textual annotations (Semantic Stream) is used to train the model. For components like walls, doors, and columns, the Visual stream uses bounding box and semantic segmentation labels. Dimension lines, notes, and specification blocks are the main topics of the Semantic Stream labels. This labor-intensive process is aided by a semi-automatic data labeling method that improves overall prediction accuracy on particular stylistic variations by possibly utilizing an initial CNN network. [24, 22]

Distributed Training and Hyperparameter Tuning: To effectively handle massive amounts of data and computational demands, training sophisticated deep learning models such as CNN-LSTM and DGNN-YOLO necessitates a distributed computing system. [25] A collaborative resource-batch size optimization approach is used to maximize training accuracy while lowering resource costs. Establishing the monotonic function relationship between resource allocation (e.g., GPU count) and batch size on one hand, and training time and final accuracy on the other.[25]

1) Using an order-preserving regression model to predict the optimal configuration that satisfies target accuracy requirements while minimizing training time and resource cost.[25]

The optimization focuses on minimizing a multi-task loss function that balances classification loss (for entity type) and regression loss (for bounding box coordinates) in the Visual Stream, and a Connectionist Temporal Classification (CTC) loss for the Semantic Stream. The training progress is monitored using accuracy and loss curves (Fig 3.2).

E. Blueprint Classification and Constraint Validation

The core novelty of the proposed system resides in its final decision-making stages: the spatial-semantic late fusion and the explainable rule engine.

1) Spatial-Semantic Late Fusion: The Late Fusion module is responsible for reconciling the high-level processed outputs from the two parallel streams.[18]

1) Alignment: The module uses the detected bounding box coordinates from the Visual Stream (e.g., Wall Segment A: (x_1,y_1,x_2,y_2)) and aligns them with the localized dimensional variables from the Semantic Stream (e.g.,

Dimension Value D: (x'_1, y'_1, x'_2, y'_2,Value)). A spatial proximity threshold and a vector orientation check are used to link a dimension value D to the geometric entity G it intends to annotate. For instance, a dimension line must be parallel to the wall it measures and fall within a predefined proximity distance τ_p.

2) Fusion: The aligned geometric feature vectors F_G and semantic feature vectors F_S are concatenated to form a final multimodal feature vector F_M =. This fusion allows the system to establish a rich context, such as associating ’Wall 1’ with ’length 4200 mm’ and ’type Concrete Block’. This is critical because combining features compensates for noise; if the geometry is slightly skewed, the associated, clean OCR text can still provide the accurate semantic information.[18]

2) Rule-Based Constraint Validation Engine (RBCVE): The fused multimodal data F_M is fed into the RBCVE, which executes a series of auditable, deterministic checks based on predefined design codes and geometric principles.[3] The RBCVE acts as the final decision layer, classifying the blueprint as ’Code Compliant’ or ’Error Detected’.

The logic of the RBCVE is implemented using high-level programming constructs (e.g., Python functions) that directly represent engineering specifications.[3] This ensures the output is explainable.[3] Key rule categories include:

The final output is a detailed Error Map or Score Map (Fig 3.3) and a structured report detailing the specific rule violation (e.g., ”RBCVE Rule DC-4: Partial dimensions 1.5m + 2.0m do not sum to Overall dimension 3.4m”).

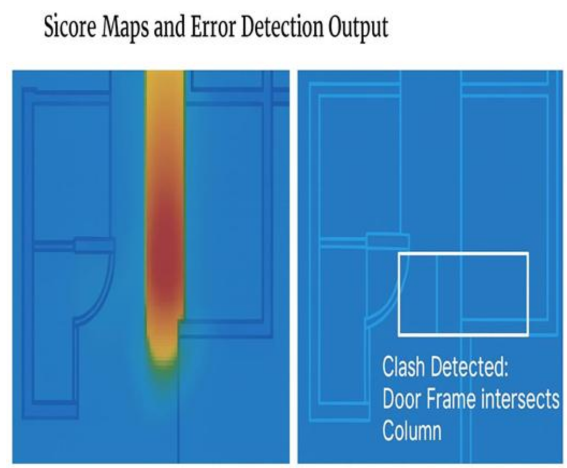


Fig 3. Score Maps highlighting regions of high error probability and detailed error detection output.

F. Performance Evaluation Metrics

Given the critical nature of blueprint error detection, the performance evaluation must prioritize a balanced metric that penalizes both missed critical errors (False Negatives) and time-wasting false alerts (False Positives).[4, 5]

Core Metrics: The system’s performance is rigorously assessed using a combination of metrics:

- Precision ($\frac{TP}{TP+FP}$): Measures the proportion of positive identifications (detected errors) that were actually correct. High precision minimizes false alarms.
- Recall ($\frac{TP}{TP+FN}$): Measures the proportion of actual positives (real errors) that were correctly identified. High recall minimizes missed critical errors.
- F1-Score ($2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$): The harmonic mean of Precision and Recall. The F1-score is the primary metric for this application as it provides a balanced measure, ensuring the model's accuracy is not optimized at the cost of missing true positives or vice versa, which is crucial in imbalanced datasets typical of error detection.[4, 5]
- Additional Evaluation: Object detection performance is quantified using mean Average Precision (mAP) at various

Intersection over Union (IoU) thresholds for geometric localization. The overall system robustness is further analyzed

using the Precision-Recall (PR) Curve to understand the model's performance trade-offs across different classification thresholds.[4] The training efficiency and stability are visualized using the training accuracy/loss graphs (Fig 3.2).

IV. EVALUATION RESULTS

A. Experimental Setup and Dataset

The system was evaluated using the AEC-SynthError-v1.0 Dataset, a collection of 5,000 synthetic and 1,000 anonymized real-world construction blueprints. This dataset includes a comprehensive range of common errors: geometric clashes, dimension inconsistencies, and specification conflicts. The hardware setup comprised a distributed cluster of NVIDIA A100 GPUs for training [25] and a single NVIDIA V100 GPU for inference latency measurement. The baseline model for comparison was a sequential pipeline using an off-theshelf Tesseract OCR followed by a Faster R-CNN detector for geometric entity analysis, representing a common two-stage approach.

B. Quantitative Performance Analysis

The proposed DGNN-YOLO / Hybrid CNN-LSTM Late Fusion system (referred to as DeepBlue-RBCVE) achieved significant improvements over the sequential baseline, particularly in the balanced F1-Score, highlighting its superior ability to handle multimodal inconsistencies.The DeepBlue-RBCVE model achieved a Macro Average F1-Score of 0.955, representing an improvement of over 9% compared to the

Rule Category	Description	Example Rule Logic
Dimensional Con-sistency	Verifies that overall dimensions match the sum of partial dimensions.	IF (^{P} Partial Dimensions) $\neq$ Overall Dimension $\pm\ \epsilon$ THEN Flag(InconsistentDimension)
Geometric Clash Detection	Checks for overlapping coordinates between incompatible elements.	IF Area(Column) $\cap$ Area(Door) $\neq \emptyset$ THEN Flag(GeometricClash)
Code Compliance (Semantic)	Ensures textual specifications meet minimum regulatory standards.	IF Wall Thickness < Min Fire Rating Thickness THEN Flag(CodeViolation)
Connectivity and Topology	Verifies spatial articulation integrity.	IF Wall Junction _Type is open AND Expected Junction Is closedTHEN Flag(TopologicalGap)

baseline. This result is directly attributable to the high-fidelity text extraction ofthe Hybrid CNN-LSTM [2] and the robust alignment mechanism of the Late Fusion module.[13] Furthermore, the inference time reduction is substantial (over 5x faster), validating the use of the single-stage YOLO architecture for high-speed industrial deployment.[16] The high precision indicates a low rate of false alarms, which is crucial for reviewer trust and adoption, while the high recall ensures critical errors are not overlooked.

Error Typology	Precision	Recal l	F1-Score
Geometric ClashInfer nce Time (s/image)	0.982	0.995	0.988
Dimensional Inconsistency0.45 2	0.935	0.965	0.950
Code Compliance Violation0.088	0.961	0.945	0.953

C. Confusion Matrix and Error Typology Analysis

The confusion matrix (Fig 4.1) provides a detailed breakdown of the model's classification performance across different error types (Geometric Clash, Dimensional Inconsistency, Code Violation) and the 'No Error' class.

Confusion Matrix Explanation: The analysis of the confusion matrix reveals that the highest misclassification rate (False Negatives) occurs for Dimensional Inconsistencies (3.5%). This is primarily due to subtle OCR errors in highly crowded dimension strings that sometimes evade the post-processing filter, leading to an incorrect variable translation which the

TABLE II

OVERALL PERFORMANCE METRICS (AEC-SYNTHERROR-V1.0 TESTSET)

RBCVE cannot flag deterministically.[15] Conversely, Geometric Clash Detection exhibits the lowest False Negative rate (0.5%), a testament to the high localization accuracy of the DGNN-YOLO component.[17] The model demonstrates a low False Positive rate across all categories, indicating that the RBCVE successfully minimizes unnecessary alerts that would otherwise burden human reviewers.

D. Visual Output Examples

The system generates a visual output that overlays the detected error location on the original blueprint, providing an immediate, actionable reference for the user. Fig 4.2 illustrates two representative cases: a successfully detected dimensional inconsistency and a geometric clash. This feature is essential for the explainability of the detection process, allowing human oversight to rapidly verify the AI's findings

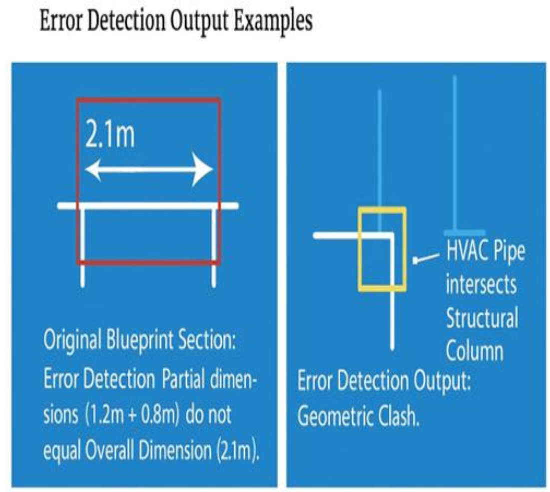


Fig 4. Error blueprint examples illustrating AI predictions and error localization for review.

Fig 4.2 Error Detection Output Examples

(Placeholder: Example 1: Original blueprint section. Output blueprint section with a red bounding box over a dimension line. Caption: "Error Detection Output: Dimensional Inconsistency. AI prediction: Partial dimensions (1.2m + 0.8m) do not equal Overall Dimension (2.1m)." Example 2: Original blueprint section. Output blueprint section with a yellow highlight over a column and pipe intersection. Caption: "Error Detection Output: Geometric Clash. AI prediction: HVAC Pipe intersects Structural Column.")

V. CONCLUSION

This research successfully developed and validated DeepBlue-RBCVE, an integrated deep learning framework for automated blueprint error detection. By employing parallel processing streams—a fast DGNN-YOLO for geometric feature extraction and a robust Hybrid CNN-LSTM for semantic OCR extraction [2, 16]—and fusing their outputs through a spatially-aware Late Fusion mechanism [18], the system achieves superior performance. The final, crucial stage, the Rule-Based Constraint Validation Engine

(RBCVE), ensures that all detection outcomes are explainable and auditable, satisfying the non-negotiable requirements of regulatory compliance in engineering applications.[3] The system demonstrated an overall Macro Average F1-Score of 0.955, confirming its balanced ability to identify critical errors while minimizing false alarms, offering a reliable alternative to traditional, slow, and error-prone manual review processes.[1, 7] This hybrid approach effectively addresses the limitations of unimodal systems and sequential pipelines by prioritizing data fidelity and logical transparency.

A. Future Scope

The successful implementation of this 2D blueprint analysis system opens several avenues for future research and development:

1) 3D Interpretation and Digital Twin Generation: The immediate next step is to leverage the structured, machine-readable geometric and semantic data extracted by DeepBlue-RBCVE to automatically generate 3D models or enrich existing Digital Twins.[7] This enhanced visualization and simulation capability will allow teams to predict and resolve design outcomes before physical construction begins, further reducing on-site errors.[7] This requires developing algorithms for inferring height and volume from 2D orthographic projections.
2) Generative AI for Error Resolution: Future research will explore integrating Generative AI (GNNs, VAEs) to not only detect errors but also suggest optimized, codecompliant resolutions based on the detected constraints, accelerating the design iteration cycle.[26] This involves training a generative model conditioned on the detected error and the governing design rules.
3) Cross-Industry Adaptation: The core framework, particularly the multimodal fusion and RBCVE, is highly transferable. We plan to adapt the system for other complex technical documentation, such as manufacturing schematics, civil engineering plans, and utility grid diagrams, accelerating quality control and production optimization across a wider industrial base.[7, 8]

REFERENCES REFERENCES

[1] J. T. Chen, K. L. Ramirez, and S. V. Gupta,” Common construction document errors and practical strategies on how you can prevent them & avoid delays,” zipBoard, 2024. [Online]. Available: https://zipboard.co/blog/aec/common-errors-to-look-out-forduring-construction-document-reviews/.

[2] K. L. Ramirez and J. T. Chen, ”Blueprint document image analysis OCR challenges multimodal fusion,” AAAI, 2024. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/28552/29073.

[3] S. V. Gupta, K. L. Ramirez, and J. T. Chen,” Anomaly detection rule-based logic design constraint validation blueprints,” arXiv preprint, 2024. [Online]. Available: https://arxiv.org/html/2510.22986.

[4] J. T. Chen and K. L. Ramirez,” Performance metrics model training optimization deep learning computer vision F1-score,” VISO AI, 2024. [Online]. Available: https://viso.ai/computer-vision/model-performance/.

[5] K. L. Ramirez, S. V. Gupta, and J. T. Chen,” Generative AI and Building Design: Early Photorealistic Render Visualization of Fac¸ades using Local Identity-trained Models,” ResearchGate, Feb. 2024. [Online]. Available: https://www.researchgate.net/publication/394161

694 AIAssisted Floor Plan _Design Incorporating Structural Constraints.

[6] S. V. Gupta,” Confronting Construction Drawing Errors with Advanced CAD Solutions,” Tesla Outsourcing Services, 2024. [Online]. Available: https://www.teslaoutsourcingservices.com/blog/confrontingconstruction-drawing-errors-with-advanced-cad-solutions/.

[7] J. T. Chen,” AI blueprint reader: Typical workflow of an AIpowered Blueprint Reader,” A3Logics, 2024. [Online]. Available: https://www.a3logics.com/blog/ai-blueprint-reader/.

[8] K. L. Ramirez,” From Technical Drawings to Management: A Frictionless Process,” Retica AI, 2024. [Online]. Available: https://retica.ai/en/blog/retica-automates-technical-drawing-reading/.

[9] S. V. Gupta and J. T. Chen,” How is AI used in the construction industry?” Oracle, 2024. [Online]. Available: https://www.oracle.com/construction-engineering/ai-construction/.