Joint CSCE Construction Specialty & CRC Conference 2025
Conférence conjointe spécialisée en construction de la SCGC et CRC-2025

Montreal, Quebec
July 28-31, 2025 / *28-31 juillet 2025*

Deep Learning-based Detection of Structural Components and Data from Legacy Engineering Drawings

F.T. Nabil[1]* and G. Guven[1]*

[1] Department of Civil Engineering, University of Manitoba, Manitoba, Canada

ABSTRACT: The lack of digitization and efficient utilization of legacy data stored in paper-based engineering drawings pose a significant challenge in the maintenance and management of aging infrastructure, as these documents are critical for assessing structural conditions and planning repairs. Extracting and organizing information from a large volume of drawings is traditionally a manual, time-consuming, and error-prone process. This paper presents an ongoing research on developing an automated asset inventory framework designed to extract, structure, and utilize data from legacy engineering drawings using a state-of-the-art Deep Learning (DL)-based object detection model (YOLOv11). The proposed methodology integrates YOLOv11 to identify and extract geometric and non-geometric data from scanned engineering drawings. The dataset, collected from real-world electrical substations, includes key structural assets information such as asset names, ages, cross-sectional views, plan views, dimensions, and general notes. Preliminary findings demonstrate that the model accurately detects and localizes critical structural components and their attributes, achieving a mean average precision (mAP) of 0.953 and an F1 score of 0.92 at a 0.344 confidence level, validating its effectiveness in automatically extracting information with high accuracy. By automating an image-based asset inventory creation from legacy engineering drawings, this framework facilitates structural condition assessment at electrical substations. Additionally, it assists informed decision-making for repairs, material reuse, and lifecycle optimization. By streamlining data extraction from legacy engineering drawings, this research contributes to the development of more efficient, data-driven infrastructure asset management practices.

1. INTRODUCTION AND BACKGROUND

The structural condition assessment of civil infrastructure assets is critical to ensuring proper maintenance, public safety, efficient operations and the longevity of built environments. An accurate and comprehensive asset inventory management (AIM) system is essential for effective operation and maintenance (O&M) of a large-scale public infrastructure network as 80% of an asset's lifecycle cost are spend in the O&M phase (Rampini et al. 2022; Yuan et al. 2016). The operation, refurbishment and reconstruction of existing structures heavily rely on the availability of up-to-date information on the conditions of structures. Therefore, the importance of AIM systems is widely recognized as one of the most vital aspects of the success of any civil construction project. Legacy engineering drawings, which serve as the primary source of information for many existing structures, particularly aging infrastructure, play a pivotal role in structural condition, a key process in evaluating the performance and safety of structures. This assessment is especially critical for aging infrastructure, which poses a significant challenge in North America. According to the Canadian Infrastructure Report Card (2019) nearly 40% of roads and bridges in Canada are in fair, poor, or very poor condition, with 80% being more than 20 years old. Additionally, in the U.S., approximately 42% of all bridges are at least 50 years old, and more than 45,000 bridges are classified as structurally deficient (ASCE, 2021). Despite their importance in evaluating aging infrastructure, these engineering drawings are largely paper-

based and often outdated, deteriorated and dispersed across various formats, making access and utilization a challenge. Manually extracting information from these drawings is not only time-consuming and error-prone but also a resource-intensive task that requires significant effort, funding, and workforce. Consequently, there is a growing need for automated, robust, and scalable solutions to bridge the gap between legacy documentation and modern digital asset management practices.

The automation of information extraction from engineering drawings has evolved significantly over the past decade, with a growing emphasis on leveraging deep learning (DL) techniques to overcome the limitations of manual methods. A semi-automatic approach was proposed by (Lu and Lee 2017) to extract location information of columns and beams from floor plan drawings through the grid symbol detection and incorporating Optical Character Recognition (OCR) for constructing as-is BIM objects. In another study, to extract information from 2D engineering drawings for reconstructing a 3D model, researchers investigated You Only Look Once (YOLO) object detection model to detect and extract geometry, location and attribute data of structural components (e.g., beam and column) from the preprocessed, scanned CAD floor plan drawings of buildings (Zhao et al. 2020). Although a later investigation suggested that the faster region-based Convolutional Neural Network (R-CNN) provided slightly better results in terms of precision and recall of the model (Zhao et al. 2021). In another study, an approach was proposed to intensively train the CNN on an artificially generated dataset to detect railway infrastructure elements and symbols from the technical drawings (Vilgertshofer et al. 2020). They automated the detection of symbols at railway infrastructures, reducing high manual effort to determine the accuracy of technical drawings (Vilgertshofer et al. 2020).

Despite these advancements, the application of DL for extracting information from civil engineering drawings remains underexplored as most of the existing studies focus on extracting data from floor plan drawings, which differ significantly in terms of complexity and annotation styles compared to cross-sectional and detailed drawings of structures. Additionally, the integration of extracted information into AIM systems has been largely overlooked in the literature. There is no study that have addressed the challenges to develop an automated AIM system specifically designed for extracting and managing data from civil engineering drawings. While there have been a few attempts to develop an automated AIM system for roadway assets and electricity transportation infrastructure components of the power grid (Kala et al. 2022; Kargah-Ostadi et al. 2020), none of these utilized engineering drawings, and rather relied on object images and drone-captured photos.

This research addresses these gaps by proposing a methodology of extracting geometric and non-geometric data from cross-sectional detailed drawings of structural elements using YOLOv11 (Jocher and Qiu, 2024). While DL-based YOLO models have shown promise in object detection tasks, to the best of the authors' knowledge, the latest version YOLOv11 has not yet been applied to data extraction from civil engineering drawings, making this study a novel contribution. The model is trained on a real-world dataset that consists of detailed foundation drawings from two electrical substations. The model identifies the critical information required for conducting structural condition assessment for aging infrastructures. In the future phases of the study, an additional dataset will be aggregated with the original dataset to enhance the model's robustness and improve its training and generalization capabilities. While the integration of extracted information into an AIM system is beyond the scope of this study, it is identified as a critical direction for future work. The rest of this paper is organized as follows: Section 2 discusses the data preprocessing and labeling, while Section 3 describes the methodology. Section 4 presents the results to evaluate the performance of the proposed methodology and Section 5 wraps up with a conclusion.

2. DATASET

The dataset consists of detailed structural drawings of civil assets including foundations and site plans for steel structures and cable trenches from two real-world electrical substations, stored in various formats (e.g., PDF, TIFF). This study aims to automatically extract geometric and non-geometric data from foundation drawings. The dataset is divided into training, testing, and validation datasets. The images from the datasets of Substations A and B were combined to generalize the trained YOLO model for varying drawing formats.

2.1 Dataset Exploration

Each detailed structural drawing of the foundation contains both geometric and non-geometric data. The non-geometric data includes a title “FOUNDATION”, date of construction which indicates the age of the foundations, individual foundation markings, electrical apparatus attached to the foundations, unique drawing reference number, date of revisions made on drawings and some notes on standards. Geometric data is comprised of longitudinal section and cross section of foundations, plan view of foundation, pile, pile cap, different types of anchor bolts, anchor bolt layout, dimensions of foundations, and reinforcement details.

The anchor bolts in foundations are a critical information for structural conditional assessment. In the dataset images, anchor bolts appear tiny and are often difficult to identify due to the low quality of the scanned images. To address this, an additional dataset was created by cropping anchor bolt samples from detailed foundation drawings to improve the generalization capability of the object detection model. The original dataset was then augmented with these newly created images to help the model learn the patterns of small anchor bolt objects. This step was designed to maximize insights from the dataset, ensuring that both geometric and non-geometric data receive equal importance. This approach not only improves the model’s representation of the dataset but also enhances its performance in detecting complex and tiny components within the drawing images.

2.2 Data Preprocessing

To prepare the dataset for analysis, a comprehensive data preprocessing pipeline is implemented to standardize the image data. The dataset, originally stored in various formats (e.g., PDF, TIFF) were converted to a single image format (i.e., JPG). Upon converting, some images reached sizes as large as 10224x7137 pixels which are way higher than the compatible image input sizes (e.g., 640x640, 960x960) for object detection model training. Directly annotating these high-resolution images can lead to significant feature distortion when the images are later downscaled during model training. This is especially problematic with small objects, which may become blurry or disappear entirely. It is also not ideal to perform annotations on very low-resolution images that have been resized to match the input size required for model training. This is because important fine details can be lost, especially when the original resolution is significantly higher than the resized version, making accurate annotation difficult. Therefore, to preserve as much detail as possible during labeling, the images were resized to the dataset's average dimensions of 7000x5700 pixels which ensured consistent scaling, preserved object visibility, prevented feature distortion and optimized labeling precision.

2.3 Dataset labeling

The dataset was meticulously labeled with categories representing both geometric and non-geometric data to train the object detection model. These labels capture key features and critical information of foundation drawings to develop an efficient asset inventory system required for structural condition assessment. The labeling process systematically identifies, localizes and categorizes all relevant elements for accurate detection and classification during model training and evaluation. The labeling schema is designed to comprehensively annotate various elements in the dataset. Key labels include $title\_box$, which identifies the title box containing document metadata, such as age, dr_title (drawing title), $dr\_reference$ (unique drawing reference number), f_mark (foundation name), $elec\_apparatus$ (electric apparatus attached to the foundation). The $foundation$ label identifies the area encapsulating the longitudinal section view of foundations, depicting the geometric and structural details. These details include anchor bolts attached to the foundation, labeled as $anchor$, and f_depth (foundation depth), which aligns with the $pile$ and $pile\_cap$ labels. The f_plan label identifies the plan view of foundation, while f_len and f_width mark its length and width in the plan view. The $pile\_cross$ label represents cross-sectional view of the pile, and $pile\_dia$ captures the dimension of the pile cross-section. Additionally, the $anchor\_des$ label marks table containing anchor bolt information, while the $notes$ label captures some general notes on standard work procedures at the site. This comprehensive labeling schema ensures that all the key features within the image are covered, supporting a comprehensive model training for precise object detection.

The labeling process was conducted manually by drawing bounding boxes around the corresponding elements in the images. After annotation was completed, a text file was generated for each labeled image, containing the class number and four coordinates of the bounding box. These coordinates represent the horizontal position of the bounding box center, the vertical position of the bounding box center, its width and its height. The DL model was trained on the bounding box coordinates over the images to learn patterns and accurately localize objects. The consistency and accuracy of the bounding boxes was verified through quality checks. This structured approach enables the model to distinguish between various elements more precisely during training, leading to more accurate predictions.

3. METHODOLOGY

This study employs the latest release of YOLOv11-m object detection model developed by Ultralytics to detect geometric and non-geometric data from the structural drawings (Jocher and Qiu, 2024). This version offers a balanced trade-off between detection accuracy and computational efficiency, achieving strong performance with a moderate model size. As shown in Figure 1 below, the methodology follows a pipeline that takes preprocessed and labeled images, along with the bounding box coordinates for different categories in each image, and a configuration file containing dataset paths, class number and names. The model utilizes the output of this pipeline to train and validate itself and to make predictions on unseen datasets. YOLO divides input images into a grid of cells, each responsible for predicting a fixed number of bounding boxes for objects whose centers fall within that cell, including their bounding box coordinates. The exported custom-trained detection model enables efficient data retrieval from the drawings, supporting the development of an automated asset inventory system. The detailed methodology, including the model architecture and workflow, is explained in the following subsections.

3.1 YOLOv11 Model Architecture

YOLOv11 is a state-of-art iteration of YOLO series with enhanced capabilities of feature extraction with significant improvements in accuracy and speed (Zou and Hu 2024). The architecture of the model consists of mainly three components: the backbone, the neck, and the head.

3.1.1 Backbone

The backbone (Figure 1.a) of the YOLOv11 primarily extracts features from input images of different scales by leveraging CNNs through multi-scale training. It consists of multiple CNN blocks, each containing a 2D convolutional (Conv) layer, a 2D batch normalization layer (BatchNorm2D), and a Sigmoid Linear Unit (SiLU) activation function. The C3K2 blocks in the backbone handle feature extraction and optimize information flow through the network by applying a series of smaller (3x3) kernel convolution layers (C3K), along with two Conv blocks at the beginning and end. Finally, the Conv block and C3K block outputs are concatenated. This structure is designed to balance speed and accuracy while maintaining low computation costs.

3.1.2 Neck

The neck (Figure 1.b) of YOLOv11 architecture retains a Spatial Pyramid Pooling Fast (SPPF) module that improves the model's ability to efficiently extract the multi-scale features by its max pooling layers, enabling the detection of objects with varying sizes (e.g., anchor bolts, foundation depth annotations) across different resolutions. YOLOv11 introduces a Cross Stage Partial with Special Attention (C2PSA) mechanism that improves the model's ability to capture and emphasizes critical region within an image by prioritizing spatial attention in the feature maps. The C2PSA block uses to two Partial Spatial Attention (PSA) modules which enhance the feature extraction and processing capabilities. The neck aggregates multiscale features using a series of convolution layers with C3K2 blocks, which process features after upsampling and concatenation before passing them to the head for final prediction.

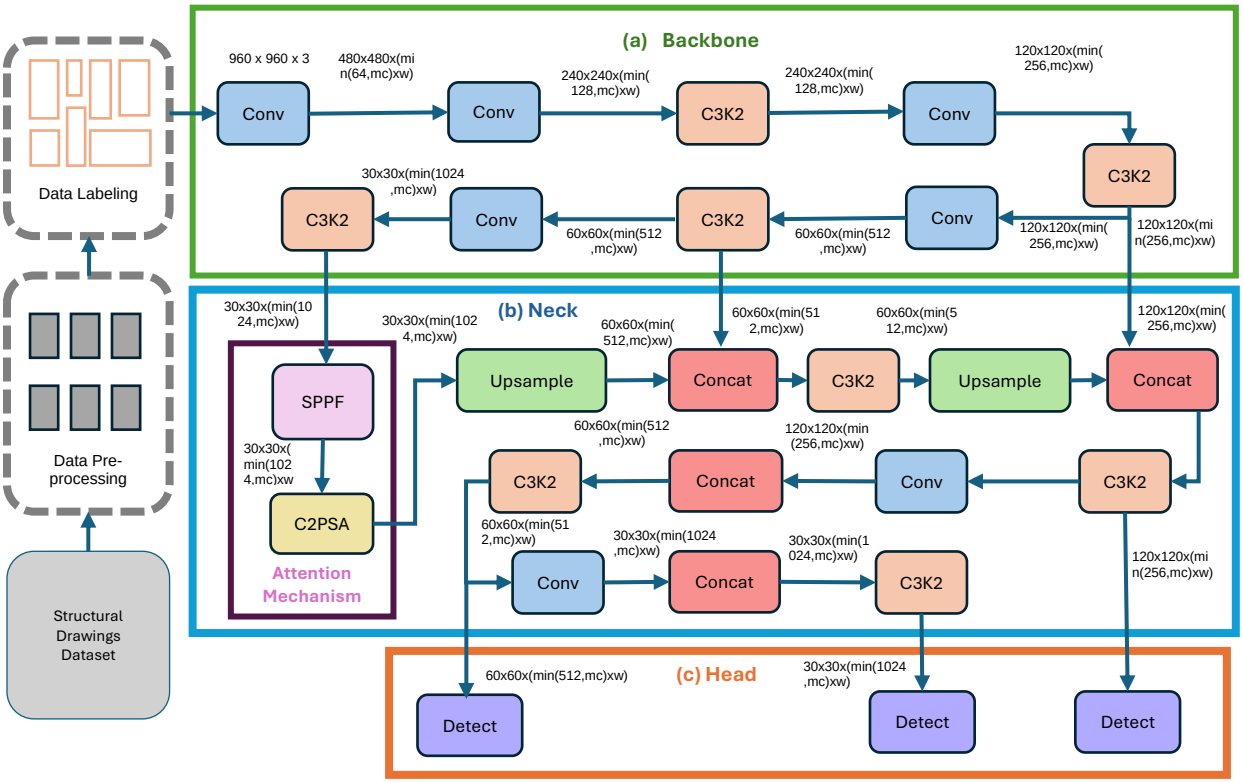


Figure 1: Schematic View of the Methodology Incorporating YOLOv11 Model

3.1.3 Head

The YOLOv11 uses a multi-scale prediction head (Figure 1.c) which performs the final classification and bounding box regression by leveraging the three different scales (low, medium, high) of feature maps produced by the backbone and neck of the model. The prediction output from the three feature maps ensures the detection of small objects in finer details.

3.1.4 Optimizer and Loss Function

The model is trained with Adaptive Moment Estimation (Adam) optimization algorithm, an adaptive gradient-based optimization method that utilizes both momentum and scaling and combines the benefits of Root Means Square Propagation (RMSprop) and Stochastic Gradient descent (SGD) with momentum (Kingma and Ba 2017). Adam is highly effective for training DL models because its learning rate adapts for each parameter based on the estimates of the first and second moments of the gradients. Adam optimizer is particularly efficient in scenarios involving non-stationary objectives and noisy gradients. Adam ensures that parameters with large gradients are scaled down, while those with small gradients are scaled up, leading to efficient training across diverse features. Additionally, the use of momentum helps accelerate convergence, while the adaptive learning rate mechanism reduces oscillations during optimization. This makes Adam particularly effective for training the model to detect structural elements from engineering drawings, where the gradients can vary significantly across different features and training iterations (Kingma and Ba 2017).
The loss function of YOLOv11 is comprised of three components (i.e., distributed focal loss, bounding box regression loss, class probability loss) (Redmon and Farhadi 2018). The loss function formula is expressed by the following equation (Eq. 1):

$$[1]\ L_{v11}=L_{cls}+L_{box}+L_{dfl}$$

Where L_{v11}represents the combined loss function of YOLO v11, L_{cls} represents class probability loss, that measures the difference between the predicted class probabilities and ground truth using cross-entropy principles. It is important to improve the model's accuracy across various categories. L_{box} minimized the loss in the predicted bounding box and actual bounding box using metrics such as intersection over union (IoU) to improve precision. L_{dfl} is the distribution focal loss that enhances the object localization capability of the model by improving the alignment between predicted bounding boxes and actual bounding boxes by leveraging the distribution-based predictions (Li et al. 2020; Lin et al. 2017). The loss function integrates all the individual loss components to optimize the overall accuracy and efficiency of the model's object detection performance.

4. EXPERIMENT AND RESULTS

This section discusses the hyperparameter settings and evaluation metrics used for YOLO model in this study. The results include an analysis of the graphs generated from the training and validation process, demonstrating how these evaluation parameters contributed to achieving optimal performance with YOLO v11. For this experiment, a laptop equipped with an Intel Core i7-1185G7 @ 3.00GHz was used. Google Colab was selected as a development environment, and the NVIDIA Tesla T4 Graphical Processing Unit (GPU) was utilized for training the YOLO model. A configuration file (data.yaml) was created for the model, specifying the file paths for the training, validation, and testing dataset directories, along with the total number of classes and their corresponding labels.

4.1 Hyperparameter Settings

The proposed model is trained and evaluated with a carefully chosen set of hyperparameters to ensure optimal learning capability of the model and precise detection. YOLOv11 resizes the input images at a fixed image size which is by default 640x640 pixels. The training cannot be performed on the previously resized dimensions because it would become computationally heavy, inefficient and disrupt the training process. To make it computationally efficient and stable, the input image size was set to 960 pixels so that it still retained better visual details for accurate object detection. The training epochs was set to 400 and the batch size was set to 10 to optimize the training process while maintaining an effective computational load. The $multi\_scale$ parameter was enabled to handle multiple image sizes and resolutions during training, improving its generalization and detection accuracy for objects of various scales. Additionally, the plot function was enabled to generate and save graphs of evaluation metrics throughout the training and validation process.

The cosine learning rate scheduler (cos_lr) parameter was set true to dynamically modify the learning rate according to a cosine decay curve over the training iterations, ensuring a smooth convergence and achieving better generalization capability. For the Adam optimizer, an initial learning rate ($lr0$) of 0.001 was specified which is the standard setting for optimizers such as Adam (Ruder 2016). A final learning rate fraction (lrf) of 0.001 was used to ensure slow adjustment of model parameters that lessen the risk of overshooting the optimal solution. This combination of hyperparameters helps the model adjusting its training weights gradually, leading to a better feature extraction and learning efficiency while balancing the training speed and stability.

4.2 Model Evaluation Metrics

To assess the performance of the model on the structural drawings, five widely accepted metrics are evaluated: mean average precision (mAP), precision, recall, F1 score and confusion matrix. These metrics provide a broad view of the model's ability to precisely detect the geometric and non-geometric data from the structural drawings. Some of the basic terms utilized to compute the evaluation metrics are True Positive (TP) which means positively identifying the correct object, True Negative (TN) means positively identify the incorrect object, False Positive (FP), means failing to detect an object that is not present in the image, and finally, False Negative (FN), means failing to detect an object that is present in the image.

Precision is the proportion of the TP predictions out of the total number of instances predicted as positive (TP & FP). Precision provides an idea of how many of the predicted positive labels are actually correct. It is particularly important to reduce the FP. Recall, also referred to as the true positive rate, measures the proportion of actual positive instances correctly identified by the model. Recall indicates how many of the actual instances of a particular class in the dataset were correctly predicted by the model. Recall is calculated as the ratio of true positive detections to the total actual positives (TP + FN). The F1-score is the harmonic mean of the precision and recall, providing a good balance between precision and recall. Hence, it is very useful where the class distributions are imbalanced. The mathematical representation of F1 score is stated by the formula (Eq.2) below:

$$[2]\ \text{F1 Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

The mean average precision (mAP) measures the overall performance of the model by computing the mean of the average precision (AP) scores across all object categories. It provides a comprehensive assessment of how well the model predicts both the classification and location. The mathematical expression of mAP (Eq. 3) is below:

$$[3]\ \text{mAP} = \frac{1}{N}\sum_{i=1}^{N} AP_i$$

Two types of mAP are evaluated based on intersection of union (IoU) thresholds. IoU measures the overlap between the predicted bounding box and the actual bounding box created during data labeling. The mAP@50 evaluates AP at 0.5 IoU threshold while mAP@50:95 computes the average AP over multiple IoU thresholds ranging from 0.5 to 0.95. A higher mAP indicates better performance while a lower value suggests lower accuracy in detection and localization. These evaluation metrics help assessing the performance of the YOLO model in detecting the structural component as well as the associated geometric and non-geometric data from the drawings.

4.3 Results

The training process of the model generates fitting curves of various loss functions to effectively evaluate the performance of the custom-trained model. Figure 2 illustrates the three primary components of the loss function, box_loss, class_loss and dfl_loss along with mAP@50, mAP@50-95, precision and recall metrics. The loss functions for both the training and validation sets decreases exponentially as the number of epochs increases and significantly slows down by 350-400 epochs. The validation performance is also monitored alongside training and no significant improvement is observed by the end of 400 epochs. The downward trends of the loss functions indicate that the model is near convergence. The training evaluation results show that the precision stabilizes at around 0.932 and recall at 0.919, which reflect the model's learning capability with high accuracy in effectively detecting the target object while maintaining low misidentification and missed detection rates. The overall detection performance of the model is evaluated with mAP metrics. At the end of 400 epochs, the mAP@50 value stabilizes at 0.944 suggesting strong detection capability of the model at 0.5 IoU threshold. The average value of mAP@50-95 stabilizes at 0.739 providing insight on the detection capability under stricter IoU threshold. All start getting saturated and stabilized by 400 epochs, indicating the model has learned the underlying features effectively. While a marginal decrease in training loss may still be possible with more epochs, further training would likely result in slow performance gains and could lead to overfitting. Therefore, the training is stopped at the end of 400 epochs to optimize training efficiency with improved bounding box localization and classification performance in prediction.

The precision-recall (PR) curve and the F1-confidence curve are generated by performing the validation test of the trained model on a 60-image dataset. A trade off between precision and recall is critical to make a decision on selecting an effective confidence threshold for prediction.

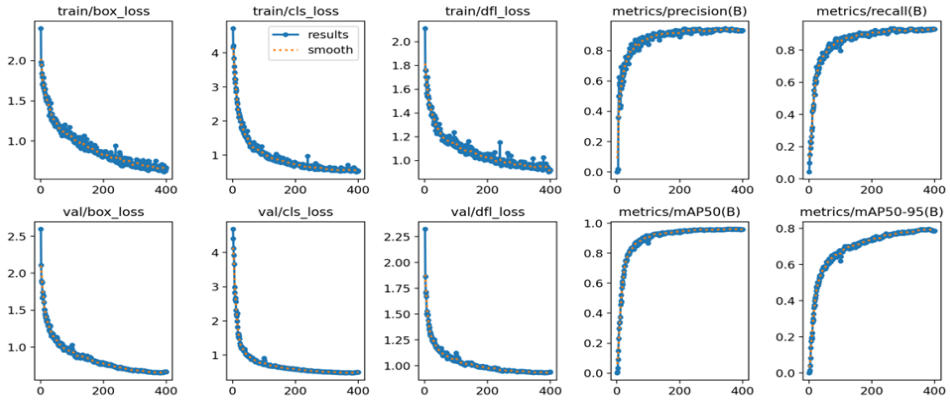
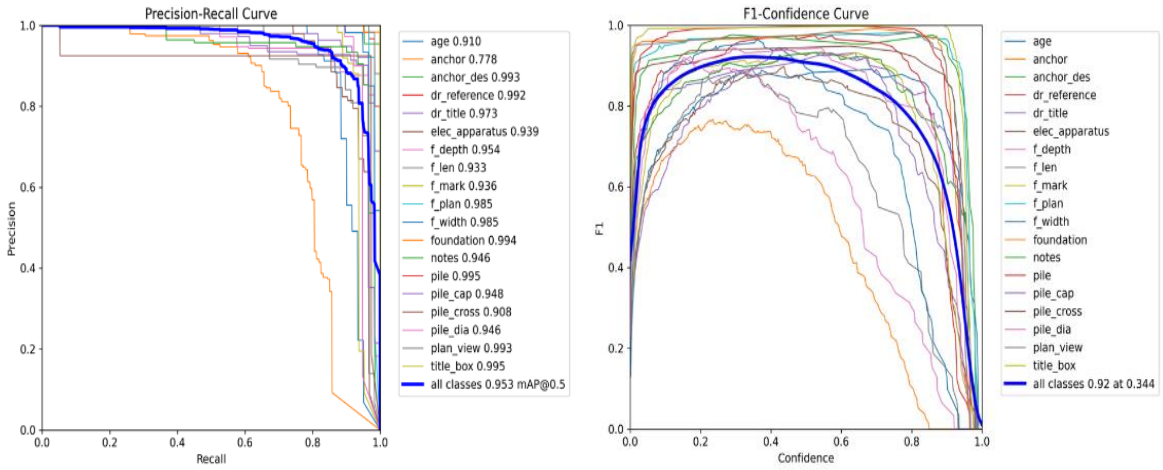


Figure 2: Evaluation metrics results from training

In Figure 3 (a), the thick blue line represents the average PR curve across all classes and average mAP@50 is 0.953, indicating a strong overall performance of the model. In the PR graph, most of the individual categories having a mAP more than the overall average mAP shows a good balance between precision and recall. The categories having lower mAP value than the overall average indicate room for improvement to balance the parameters. This can be improved by increasing the amount of data for those particular instances. The model achieves an optimal F1 score of 0.92 at 0.344 confidence level, indicates a strong trade off between minimizing FPs and FNs. Most of the classes demonstrate a very high F1 score, reflecting a strong detection and localization capabilities of the model.



(a) (b)

Figure 3: (a) Precision-recall curve, (b) F1-confidence curve

In Figure 4, the normalized confusion matrix shows the performance across different classes. The distribution highlights the TPs and misclassification across the classes, demonstrating the model's performance of accurately classifying the target objects while revealing patterns in its errors and misclassifications. Notably, classes such as, $anchor\_des$, $dr\_reference$, dr_title, f_plan, $foundation$, $notes$, $pile$, $pile\_cap$, $pile\_cross$, $plan\_view$ and $title\_box$ attain near-perfect classification with a value close to 1, solidifying the model's competence to accurately predict these categories. On the other hand, the $anchor$ category shows 73% detection accuracy, with the remaining predictions misclassified as background. This suggests challenges in distinguishing this class due to limited representation in the training dataset. All other classes achieved a strong classification accuracy with values ranging from 0.85 to 0.95. Overall, the confusion matrix highlights the model's strong generalization across most categories while identifying areas for improvement to further enhance the accuracy of the model.

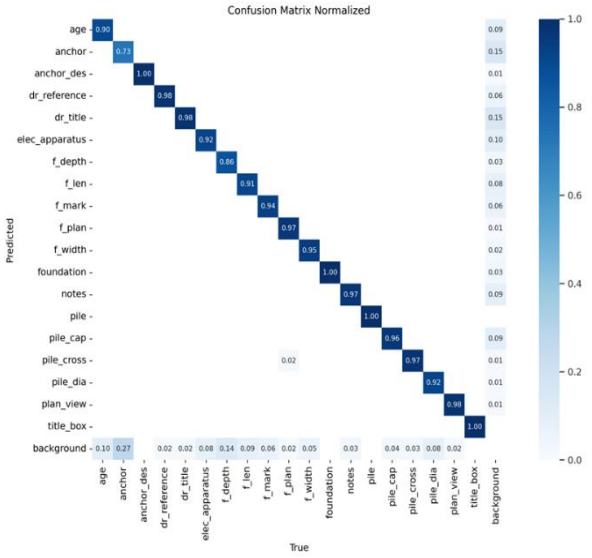


Figure 4: Normalized confusion matrix

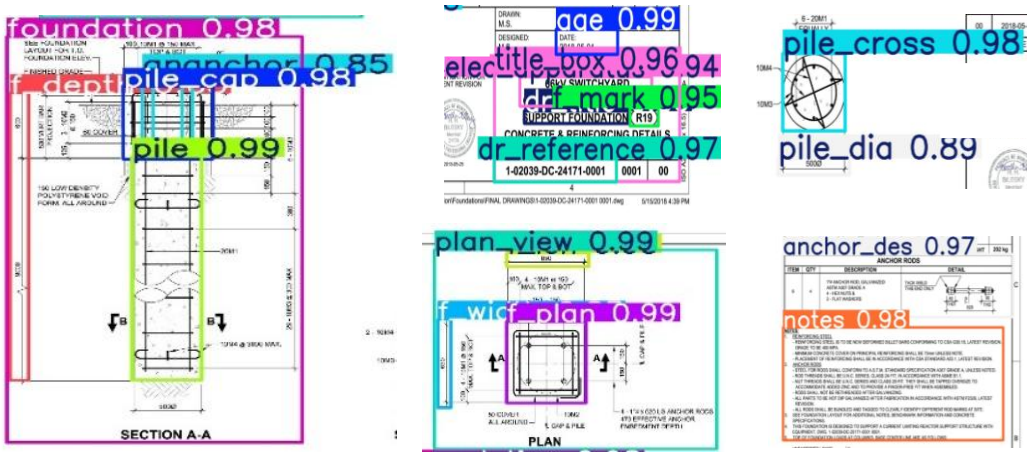


Figure 5: Prediction outcomes

Figure 5 illustrates the prediction results obtained during inference on the test using the custom-trained YOLO model, with a confidence threshold of 0.7 and an IoU threshold of 0.6. Majority of the predictions demonstrate high confidence level in detecting and classifying objects across multiple categories. This shows that the model is robust and reliable in recognizing objects with significant detection accuracy.

5. CONCLUSIONS

This paper presented the preliminary results from an ongoing study aimed at developing an automated asset inventory framework that leverages a DL-based object detection model (YOLOv11) for the automatic identification and extraction of data from legacy engineering drawings. This framework is designed to automate asset inventory creation from engineering drawings, assisting structural condition assessment practices in aging infrastructure, particularly in electrical substations. The object detection model is trained on aged and low-quality scanned images of paper-based structural drawings of foundations that are collected from two real-world substations. The model shows significant accuracy in detecting geometric (e.g., pile, pile cap, foundation dimensions, anchor bolt detail, etc.) and non-geometric data (e.g., age, drawing reference, foundation name, etc.) from the drawings, achieving an overall precision of 93.2% and

recall of 91.9% using the Adam optimizer. The evaluation metrics, including loss functions, precision-recall curves, F1-confidence curves, and the normalized confusion matrix, collectively show the robustness of the model in detecting and classifying targeted object information. Although the model was trained with limited data, the evaluation indexes show that the model performs well in mAP and F1 score at 0.7 confidence level and 0.6 IoU threshold. The consistent low loss values and high mAP scores reflect effective learning and convergence, while F1 scores show a strong balance between minimizing FPs and maximizing TPs across most classes. For the future work, the DL model training will be scaled across other civil infrastructure asset drawings, such as steel structures, trench cable as well as site plan drawings to develop a comprehensive asset inventory. Furthermore, a framework will be developed in which all predicted results within the bounding boxes will undergo a character recognition process using an Optical Character Recognition (OCR) model to extract textual information from the engineering drawings. The OCR model's output will be processed by a Large Language Model (LLM) and the extracted information will be organized into a structured format. This framework will automate the process of developing an asset inventory management system and enhance the structural condition assessment for large-scale and aging infrastructure facilities. This framework can potentially improve efficiency by reducing time spent on locating changes, documenting modifications, minimizing costs, and optimizing manpower.

6. REFERENCES

American Society of Civil Engineers. 2021. *2021 report card for America's infrastructure: Bridges.*

Canadian Infrastructure Report Card. 2019. *Canadian infrastructure report card 2019.*

Jocher, G., & Qiu, J. 2024. Ultralytics YOLO11. Version 11.0.0. [Computer software]. https://github.com/ultralytics/ultralytics

Kala, J. R., Kre, D. M., Gnassou, A. N., Kala, J. R. K., Akpablin, Y. M. A., and Coulibaly, T. 2022. "Assets management on electrical grid using Faster-RCNN." *Ann Oper Res*, 308 (1–2): 307–320.

Kargah-Ostadi, N., Waqar, A., and Hanif, A. 2020. Automated Real-Time Roadway Asset Inventory using Artificial Intelligence. *Transportation Research Record,* 2674 (11): 220–234.

Kingma, D. P., and Ba, J. 2015. Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations,* ICLR 2015, San Diego, CA, USA, May 7-9

Li, X., Wang, W., Wu, L., Chen, S., Hu, X., Li, J., Tang, J., and Yang, J. 2020. Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. *In Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS '20).* Curran Associates Inc., Red Hook, NY, USA, Article 1763, 21002–21012.

Lin, T. -Y., Goyal, P., Girshick, R., He, K. and Dollár, P. 2017. Focal Loss for Dense Object Detection. *IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017, pp. 2999-3007

Lu, Q., and Lee, S. 2017. A Semi-Automatic Approach to Detect Structural Components from CAD Drawings for Constructing As-Is BIM Objects*. Computing in Civil Engineering*, ASCE, 84-91.

Rampini, L., and Cecconi, F. R. 2022. Artificial Intelligence in Construction Asset Management: A Review of Present Status, Challenges, and Future Opportunities. *Journal of Information Technology in Construction*, 27, 884–913.

Redmon, J., and A. Farhadi. 2018. "YOLOv3: An Incremental Improvement." *Computer vision and pattern recognition*, Springer, Berlin/Heidelberg, Germany, 1804: 1–6.

Ruder, S. 2016. An overview of gradient descent optimization algorithms. *arXiv*, arXiv:1609.04747

Vilgertshofer, S., Stoitchkov, D., Borrmann, A., Menter, A., and Genc, C. 2020. Recognising railway infrastructure elements in videos and drawings using neural networks. *Proceedings of the Institution of Civil Engineers: Smart Infrastructure and Construction*, ICE Publishing, 172 (1): 19–33.

Yuan, C., McClure, T., Dunston, P., and Cai, H. 2016. Leveraging Construction Inspection and Documentation for Asset Inventory and Life Cycle Asset Management. *Journal of Information Technology in Construction (ITcon)*, 21, no. 5: 72-85.

Zhao, Y., Deng, X., and Lai, H. 2020. A deep learning-based method to detect components from scanned structural drawings for reconstructing 3D models. *Applied Sciences*, MDPI, 10 (6): 2066.

Zhao, Y., Deng, X., and Lai, H. 2021. Reconstructing BIM from 2D structural drawings for existing buildings. *Automation in Construction*, Elsevier,128: 103750.

Zou, X., and Hu, Y. 2024. Hidden Danger Detection and Identification System of Power Transmission Tower Based on YOLOV11. *Academic Journal of Science and Technology*, 13 (1): 224–231.