IDEA

Innovations Deserving Exploratory Analysis Programs

NCHRP IDEA Program

Automated Data Feature Extraction from Bridge Plans

Final Report for
NCHRP IDEA Project 230

Prepared by:
Behrouz Shafei
Iowa State University

March 2024

NATIONAL ACADEMIES Sciences Engineering Medicine

TRB TRANSPORTATION RESEARCH BOARD

Innovations Deserving Exploratory Analysis (IDEA) Programs Managed by the Transportation Research Board

This IDEA project was funded by the NCHRP IDEA Program.

The TRB currently manages the following three IDEA programs:

- The NCHRP IDEA Program, which focuses on advances in the design, construction, and maintenance of highway systems, is funded by American Association of State Highway and Transportation Officials (AASHTO) as part of the National Cooperative Highway Research Program (NCHRP).
- The Safety IDEA Program currently focuses on innovative approaches for improving railroad safety or performance. The program is currently funded by the Federal Railroad Administration (FRA). The program was previously jointly funded by the Federal Motor Carrier Safety Administration (FMCSA) and the FRA.
- The Transit IDEA Program, which supports development and testing of innovative concepts and methods for advancing transit practice, is funded by the Federal Transit Administration (FTA) as part of the Transit Cooperative Research Program (TCRP).

Management of the three IDEA programs is coordinated to promote the development and testing of innovative concepts, methods, and technologies.

For information on the IDEA programs, check the IDEA website (www.trb.org/idea). For questions, contact the IDEA programs office by telephone at (202) 334-3310.

IDEA Programs
Transportation Research Board
500 Fifth Street, NW
Washington, DC 20001

The project that is the subject of this contractor-authored report was a part of the Innovations Deserving Exploratory Analysis (IDEA) Programs, which are managed by the Transportation Research Board (TRB) with the approval of the National Academies of Sciences, Engineering, and Medicine. The members of the oversight committee that monitored the project and reviewed the report were chosen for their special competencies and with regard for appropriate balance. The views expressed in this report are those of the contractor who conducted the investigation documented in this report and do not necessarily reflect those of the Transportation Research Board; the National Academies of Sciences, Engineering, and Medicine; or the sponsors of the IDEA Programs.

The Transportation Research Board; the National Academies of Sciences, Engineering, and Medicine; and the organizations that sponsor the IDEA Programs do not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the object of the investigation.

AUTOMATED DATA AND FEATURE EXTRACTION FROM BRIDGE PLANS

IDEA Program DRAFT Final Report

NCHRP IDEA 20-30/IDEA 230

Prepared for the IDEA Program
Transportation Research Board
The National Academies of Sciences, Engineering, and Medicine

Principal Investigator
Behrouz Shafei, Ph.D., P.E.
Iowa State University, Ames, Iowa

March 2024

ACKNOWLEDGMENTS

The research team would like to acknowledge the National Cooperative Highway Research Program (NCHRP) Innovations Deserving Exploratory Analysis (IDEA) Program for sponsoring this project. The support and assistance of the NCHRP IDEA Program Manager, Dr. Inam Jawed, are gratefully acknowledged.

Special thanks are due to James Hauber, David Hedeen, Sharon Yen, and Cliff Roblee for their participation in the project’s advisory committee and their invaluable feedback throughout the course of this project. Additionally, the research team would like to recognize the project’s IDEA advisors, Tawney Brennfleck, Nick Burmas, and Ahmad Abu-Hawash, for their time and coordination.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

Machine learning (ML) and artificial intelligence (AI) have significantly impacted numerous fields through their ability to tackle challenges with remarkable computational efficiency. In bridge engineering, ML/AI techniques have been employed to enhance the efficiency of the structural design phase, aid in the selection of optimal bridge types, produce cost estimates, conduct real-time structural health monitoring, predict structural response and deterioration, reconstruct data for comprehensive health assessment, and prioritize maintenance efforts.

This research project applied ML/AI techniques to automate the process of extracting data and features from drawings, tables, and text blocks contained in bridge plan sets using state-of-the-art deep learning algorithms. The research was motivated by the critical need to report bridge inventory information to the Federal Highway Administration (FHWA) in compliance with National Bridge Inspection Standards (NBIS) reporting requirements.

The research ultimately produced a novel computational platform that automates the process of reviewing bridge plans to identify, extract, and report select engineering details. A holistic review of a variety of bridge plans was first performed to identify and categorize engineering details of interest. Convolutional neural network (CNN) algorithms were then used to automate the process of detecting physical objects of interest in bridge plans, identifying the types of the objects, and extracting the objects' main dimensions and details. Because bridge plans often contain various tables with important information about a variety of engineering details, tables were also located, the boundaries of each table's cells were identified, and necessary data points were extracted and reported in an editable format. The process was also extended to enable the extraction of textual information from text blocks in bridge plans. The automated identification and transfer of data and features from bridge plans to spreadsheets greatly facilitated post-processing activities related to querying information or identifying quantities of interest.

The project also involved testing, assessment, and quality control of the developed computational platform using bridge plans provided by the Iowa and California Departments of Transportation. The accuracy and speed of the algorithms in extracting information were systematically evaluated through a verification stage, which ensured that the correct outputs were returned from a set of bridge plans that had been used to train the algorithms, and a validation

stage, which tested the platform’s performance using a different set of bridge plans. The generated outputs were compared to those obtained from manual extraction to identify and address possible errors. The platform was able to extract information successfully for the majority of the queries it was given, demonstrating the feasibility of using a computational platform to automate the extraction of data and features from bridge plan sets.

Although the automated extraction of details from engineering documents can be a complicated task for machines due to the complex nature of plan sets, a combination of several deep learning models and various image processing techniques provided a platform to successfully extract details of interest. Moreover, using the general models and functions developed in this research, the platform can be customized for different transportation agencies, following their formats and practices to provide bridge details in their plan sets.

1. INTRODUCTION

1.1. OVERVIEW

The emergence of machine learning (ML) and artificial intelligence (AI) has significantly impacted various fields, prompting widespread interest in the use of ML/AI techniques to improve processes. Meanwhile, there is growing interest within the bridge engineering community to tap into extensive databases for information, prompting exploration into the application of ML/AI for this purpose across various domains. ML/AI especially stands out for its ability to tackle challenges with remarkable computational efficiency. Recent endeavors in bridge engineering have centered on using ML/AI for preliminary design, construction, structural health monitoring, and maintenance. ML/AI techniques have been employed to enhance the efficiency of the structural design phase, aid in the selection of optimal bridge types, estimate costs, conduct real-time structural health monitoring, predict structural response and deterioration, reconstruct data for comprehensive health assessment, and aid maintenance planning efforts by evaluating structural health conditions and prioritizing maintenance efforts.

1.2. RESEARCH NEED AND MOTIVATION

The motivation for this research project lies in the critical need to report bridge inventory information to the Federal Highway Administration (FHWA) in compliance with National Bridge Inspection Standards (NBIS) reporting requirements. The data collected from each State department of transportation (DOT), Federal agency, and Tribal government contribute to the National Bridge Inventory (NBI) database. This comprehensive database not only enables State- and National-level analyses but also supports Federal funding programs.

The diverse information items that should be documented during bridge inspections are outlined in the American Association of State Highway and Transportation Officials (AASHTO) *Manual for Bridge Evaluation* (MBE). Both the MBE and the FHWA *Bridge Inspector's Reference Manual* (BIRM) provide guidance on inspection procedures and the creation of comprehensive reports pertaining to bridge components. These reports serve as the foundation for reporting values associated with the numerous data items specified in the MBE and BIRM. State DOTs, Federal agencies, and Tribal governments use their own names and codes for many of these data items but follow the standard reporting specifications when reporting NBI data to FHWA. The objective of adhering to these standards is to establish a robust system that not only complies with regulatory requirements but also contributes to the overarching goal of ensuring a safe and efficient highway transportation system (FHWA 2022).

1.3. RESEARCH OBJECTIVES

This project developed a novel computational platform that automates the process of reviewing bridge plans to identify, extract, and report select engineering details. The project focused on automating the data and feature extraction process from drawings and tables contained in bridge plans using state-of-the-art deep learning algorithms. A holistic review of a variety of bridge plans was first performed to identify and categorize engineering details of interest. Convolutional neural network (CNN) algorithms were then used to automate the process of detecting physical objects of interest in bridge plans, identifying the types of the objects, and extracting the objects' main dimensions and details. The details of interest ranged from geometric dimensions (e.g., height and width) to reinforcement properties (e.g., size and length of rebars). Because bridge plans often contain various tables with important information about a variety of engineering details, tables were also located, the boundaries of each table's cells were identified, and necessary data points were extracted and reported in a desired editable format (in this case, an Excel spreadsheet). The process was also extended to enable the extraction of textual information from text blocks in bridge plans. The automated identification and transfer of data and features from bridge plan sets to spreadsheets greatly facilitated post-processing activities related to making queries or finding quantities of interest.

The project also involved testing, assessment, and quality control of the developed computational platform using a variety of bridge plan sets provided by the Iowa DOT and California DOT (Caltrans). The accuracy and speed of the algorithms developed for data and feature extraction from drawings, tables, and text blocks were systematically assessed. This assessment started from the verification stage to ensure that the automated platform was returning correct outputs for the bridge plans used in the training of the algorithms and involved a variety of desired outputs with both single- and multi-source characteristics. The quality control effort was then extended to the validation stage, in which the developed platform was tested on several bridge plans not used for training purposes. The generated outputs were compared to those obtained from manual extraction to identify and properly address possible errors and bugs.

The final report for this project presents relevant data, methods, models, and conclusions and provides guidance on how to use the developed computational platform to automate the process required to extract the data and features of interest from bridge plan sets.

1.4. REPORT OUTLINE

This report is organized into chapters that correspond to the different tasks that were completed in order to accomplish the research objectives. The introduction (Chapter 1) explains the significance of automating bridge engineering tasks, highlights the specific research needs, and outlines the research objectives. Chapter 2 provides a comprehensive literature review covering image processing methods for computer vision tasks, deep learning algorithms relevant to computer vision, and information extraction from engineering drawings. Chapter 3 describes the methodology used for this project, beginning with the data collection process and outlining the development of the deep learning models. Specific attention is given to general functions, including table recognition, dimension line recognition, and text detection. Chapter 4 details the data extraction pipelines, explaining the features and assumptions of bridge plan sets, including general introductory information, bridge properties, main dimensions, and additional details, while describing quality assurance/quality control checks. Chapter 5 provides the conclusions of the research efforts and offers recommendations for next steps.

2. LITERATURE REVIEW

2.1. IMAGE PROCESSING STEPS FOR COMPUTER VISION TASKS

Computer vision is a specialized field in the ML/AI domain dedicated to extracting meaningful information from images. Image processing is a set of different computer vision techniques that manipulates and analyzes the images to improve their quality, extract valuable features, and facilitate automated interpretation. The significance of image processing for computer vision was explored through a literature review, focusing on various common techniques. Image processing employs several fundamental techniques that are useful while dealing with bridge plan sheets. Key image processing techniques include the following:

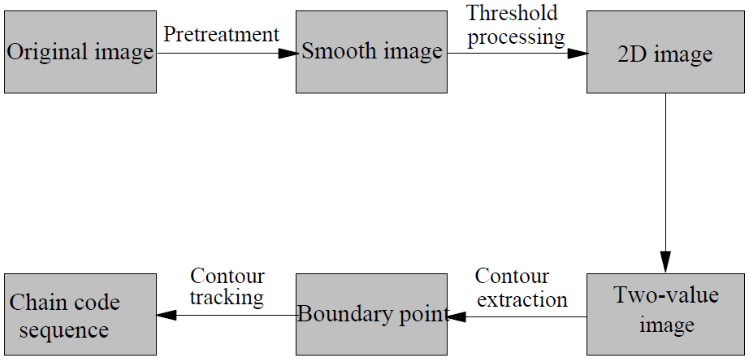
1. **Filtering and Convolution.** Filtering operations, including blurring, sharpening, and reducing noise, are fundamental techniques applied to images through convolution. Convolution involves the systematic application of a filter or kernel over an image that performs mathematical operations on each pixel. This process facilitates improvements such as edge detection and texture removal.
2. **Contour Detection.** Contour detection is a fundamental image processing technique aimed at identifying and highlighting the boundaries of objects within an image. This method is particularly useful for extracting the underlying structure and shape information. Contour detection algorithms focus on locating areas where pixel intensity changes abruptly, revealing the outlines of objects, curves, and shapes.
3. **Edge Detection.** Edge detection algorithms play a crucial role in identifying image boundaries and significant transitions. These algorithms emphasize areas in which intensity changes rapidly, such as edges, curves, or contours.
4. **Image Transformation.** Image transformation techniques involve manipulating the geometric properties of images to achieve specific objectives. Operations such as rotation, scaling, translation, and displacement enable tasks like image alignment, registration, and perspective correction. These transformations facilitate adjustments in image orientation, size, and spatial relationships.
5. **Morphological Transformation.** Morphological transformations are essential image processing operations designed to manipulate the shape and structure of objects within an

image. These operations involve the use of structuring elements like kernels to perform dilation, erosion, opening, and closing operations. Dilation expands the boundaries of objects, while erosion shrinks them. Opening is a combination of erosion followed by dilation, often used to remove noise and fine details, while closing involves dilation followed by erosion, useful for closing small gaps and filling holes in objects. Morphological transformations are crucial for tasks such as noise reduction, feature extraction, and shape analysis.

6. **Image Segmentation.** Image segmentation involves dividing an image into distinct regions based on visual characteristics, allowing separation of objects from the background or division into significant regions. Common segmentation methods include thresholding, region growing, clustering, and graph-based algorithms. This step is essential in various computer vision tasks, such as object detection, image annotation, and semantic understanding.
7. **Template Matching.** Template matching is a powerful image processing technique employed for recognizing specific patterns or objects within an image. This method involves comparing a predefined template, which represents the target pattern, with different regions of the input image. The goal is to identify areas where the template closely matches the local image content. Common similarity measures, such as cross-correlation and normalized cross-correlation, are often used to quantify the resemblance between the template and image regions. While straightforward in concept, template matching requires careful consideration of scale, rotation, and lighting variations to achieve robust and accurate results.

These techniques were employed in various sections of the pipeline developed in this project to enable the platform to address different aspects of feature extraction from complicated images of bridge plan sets.

Contour detection is a core technique used while handling tables to extract crucial information such as reinforcement properties. Contour extraction, essential for precise measurements, involves obtaining the object outline from an image. In the context of computer vision measurement, where images typically feature two targets (work pieces) and a background, the gray threshold method for image segmentation was shown to be a useful method by Cui and Zhang (2013). Mathematical morphology’s nonlinear filtering properties were utilized to eliminate defects and noise in binary images. The contour was extracted from binary images using the hollowed interior points method, and the contour information was saved using chain code tracking. The working process involved initial noise filtering through image preprocessing, followed by threshold-based image segmentation to obtain a binary image. The outline points were derived through flaw repairing treatment, resulting in image contour extraction (target outline), and the final step involved storing the outline in chain code format using the contour tracking algorithm (Figure 2.1). This approach enhanced the reliability and convenience of outline processing and measurement.



Cui and Zhang 2013

Figure 2.1. Contour extraction based on gray threshold

Hashemi et al. (2016) provided an overview of template matching techniques, emphasizing their role in various applications such as image processing, computer vision, and medical image analysis. The techniques were used to match templates (patterns) within larger images, allowing for tasks like object recognition and feature extraction. The key template matching techniques were as follows:

1. **Naive Template Matching.** Naive template matching involves scanning a target image with a template, calculating similarity measures, and identifying potential pattern positions. This straightforward algorithm is efficient when dealing with sub-images from the target image without scaling or manipulation. Error metrics like sum of squared differences (SSD) are commonly used to calculate differences between target and template images.
2. **Image Correlation Matching.** This classic template matching method involves measuring the similarity metric between the target and template images. Cross-correlation and normalized cross-correlation are commonly used metrics. Normalized cross-correlation is invariant to global brightness changes and provides normalized correlation values in the [-1, +1] interval, making it widely used for template matching.

3. **Sequential Similarity Detection Algorithms (SSDAs).** SSDAs offer an efficient alternative to correlation-based methods for translational registration. They calculate the match indirectly based on pixel-wise errors in images under comparison during the registration process.

The referenced study also addressed the challenges in template matching, such as image intensity and scale invariance. Methods like mean intensity level invariance and scale invariance were discussed to address these challenges, providing solutions for certain types of template matching problems.

Edge detection plays a crucial role in image analysis, serving as a fundamental operation for identifying object boundaries. Shah et al. (2020) explored various algorithms aimed at enhancing and detecting edges in images, considering the importance of accurate edge detection that lies in its ability to pinpoint sharp discontinuities in pixel intensity that signifies object boundaries within an image. The study evaluated different edge detection techniques, considering challenges such as false edge detection, missing true edges, and the production of thick or thin lines due to noise. The study compared the following edge detection techniques:

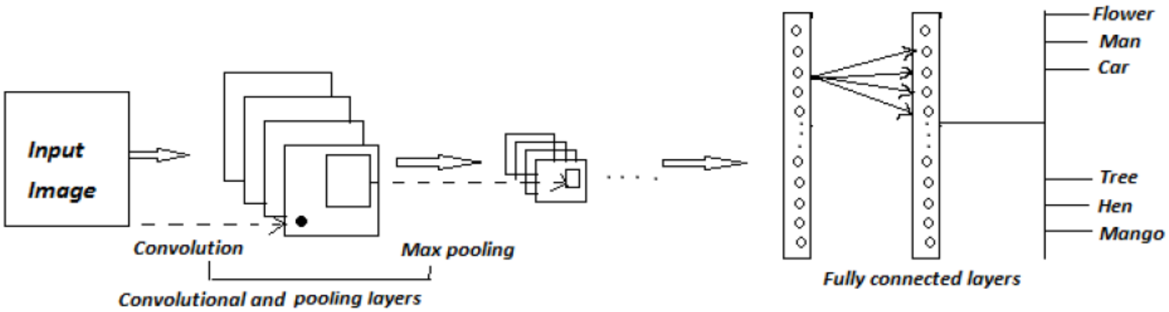
1. **Sobel Operator.** A traditional method in image processing, Sobel edge detection employs separable, small, and integer-valued filters for both *x* and *y* directions. The operator calculates the gradient magnitude and direction for each pixel using 3×3 kernels.
2. **Prewitt Edge Detection.** Similar to Sobel, Prewitt is effective in detecting horizontal and vertical edges, providing the strength of edges based on maximum gradient points.
3. **Canny Edge Detector.** A multi-step algorithm that effectively detects a wide range of edges while suppressing noise. It utilizes Gaussian filtering to smooth images and addresses the challenge of setting appropriate thresholds.
4. **Robert Edge Detection.** A method that executes gradient detection using the difference between adjacent pixels in the diagonal direction. It computes horizontal and vertical edges individually and combines them for the final edge detection.
5. **Laplacian of Gaussian (LOG).** A method that combines the Laplacian and Gaussian filtering techniques to detect edges effectively. It smoothens images and employs zero-crossing to locate edges, with a drawback of not determining edge orientation.

The techniques were categorized into three modules: gradient (first derivative estimate), Laplacian (zero-crossing detectors), and image approximation algorithms. A comparative study was conducted, including a quantitative evaluation using mean squared error (MSE) and peak signal-to-noise ratio (PSNR). The experimental results revealed that the Canny operator exhibited efficiency in detecting edges, outperforming other methods under various conditions. However, the choice of technique depends on the specific characteristics of the image and the desired outcomes.

2.2. DEEP LEARNING ALGORITHMS FOR COMPUTER VISION TASKS

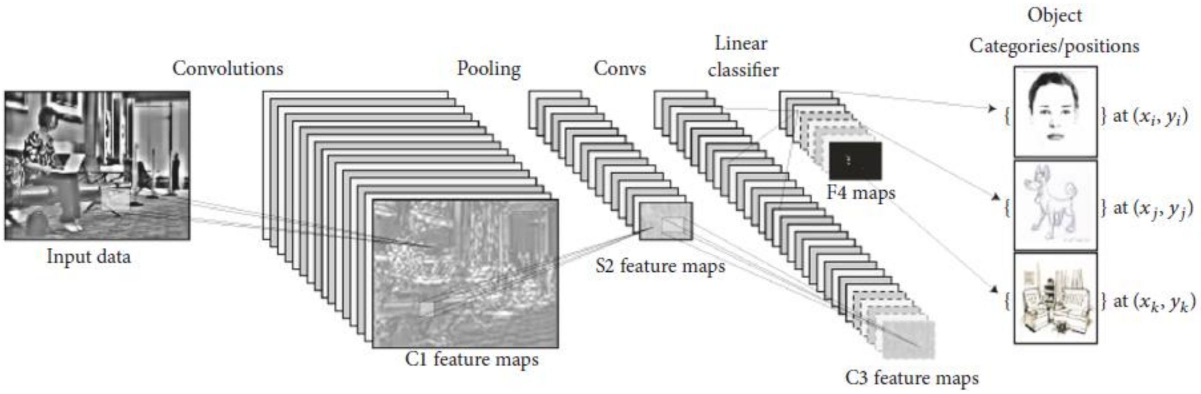
Advances in deep learning and their applications in computer vision were reviewed in Sinha et al. (2017). The significance of deep learning in handling high-level hierarchical data, particularly in improving chip programming on low-cost computing hardware was highlighted. The referenced study categorized deep learning techniques into CNNs, recurrent neural networks (RNN), restricted Boltzmann machines (RBM), autoencoders, and extreme learning. The key points of the mentioned techniques are as follows:

1. **Convolutional Neural Networks.** CNNs are considered significant for training multiple layers in various applications. The architecture includes convolutional layers, pooling layers, and fully connected layers. Convolutional layers reduce parameters through weight sharing, enabling local connectivity. Pooling layers minimize feature map measurements using approaches like stochastic pooling, spatial pyramid pooling, and def pooling. Fully connected layers, forming the final layer, involve a high computational burden during training (Figures 2.2 and 2.3).



Sinha et al. 2017

Figure 2.2. Architecture of CNN

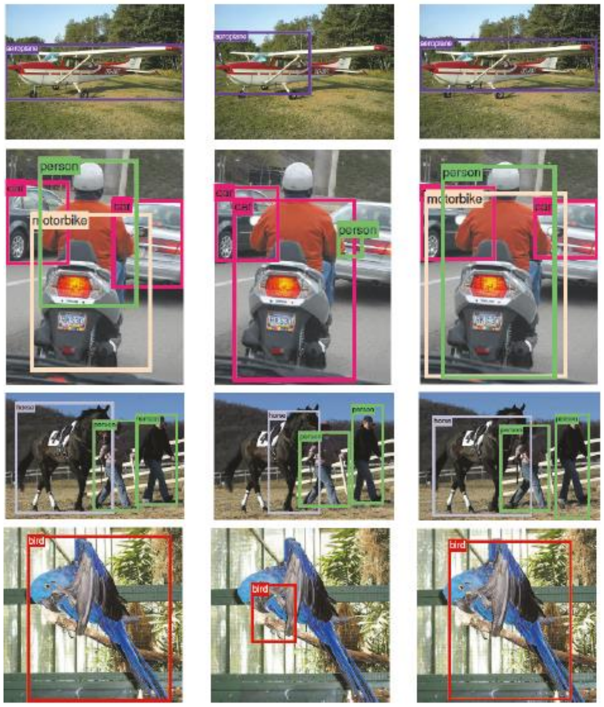


Voulodimos et al. 2018

Figure 2.3. Architecture of CNN for a computer vision problem

2. **Recurrent Neural Networks.** RNNs are employed for handling sequential data and time dependencies. They are suitable for applications such as translating natural languages, music, time-series data, and video processing. RNNs address long-time dependencies and offer good results in image captioning and analysis.
3. **Restricted Boltzmann Machines.** RBMs, a generative random neural network technique, involves hidden and visible layers forming a bipartite graph. RBM architecture is discussed in the context of deep Boltzmann machines (DBMs), deep belief networks (DBNs), and deep energy models (DEMs).
4. **Autoencoders.** Autoencoders are artificial neural networks used for training logical encodings. Various variants of autoencoders, including sparse autoencoder, denoising autoencoder (DAE), and contractive autoencoder (CAE), were discussed in Sinha et al. (2017). Autoencoders learn to reconstruct their own inputs, emphasizing features like sparsity, denoising, and contractive penalty.
5. **Extreme Learning.** Extreme learning is a recent topic in machine learning, focusing on feed-forward neural networks for regression and classification tasks. It involves a solitary layer of masked nodes with random weights for efficient learning.

Object detection, a crucial facet of computer vision, involves the identification of semantic objects within digital images and videos (Figure 2.4).



Voulodimos et al. 2018

Figure 2.4. Example of an object detector

A prevalent methodology in object detection frameworks entails the generation of a comprehensive set of candidate windows, subsequently classified utilizing CNN features. The features are then input into a classifier to ascertain whether the windows detect the target object or not. Notably, methods following the regions with CNN often achieve good detection accuracies. However, ongoing efforts aim to refine the performance of these approaches, with some attempting to determine approximate object positions without precise localization.

A noteworthy subset of methods combines object detection with semantic segmentation, a strategy demonstrated to yield favorable outcomes. The prevalent use of CNNs in object detection is evident in numerous works, where innovative layers and learning strategies are proposed, employing weakly supervised cascaded CNNs, and introducing subcategory-aware CNNs. Despite the prevalence of CNN-based approaches, a limited number of studies explore object detection using alternative deep models. Diao et al. (2016) introduces a coarse object locating method based on a saliency mechanism combined with a DBN for object detection in remote sensing images. A DBN was presented for three-dimensional (3D) object recognition, utilizing a hybrid algorithm that integrates generative and discriminative gradients. Additionally, a fused deep learning approach, deep model representation capabilities in semi-supervised tasks, and autoencoders for multiple organ detection in medical images were explored.

2.3. INFORMATION EXTRACTION FROM ENGINEERING DRAWINGS

Extracting data and features from engineering drawings has gained popularity across engineering fields. This task helps engineers to extract the required information automatically, hence it increases the efficiency and accuracy of dealing with plan sets.

In an early attempt to automate the feature extraction from drawings, Dori and Velkovitch (1998) studied the crucial role of recognizing dimensioning text in engineering drawings, which provided information about the dimensions and tolerances of depicted objects. The focus was on drawings following International Organization for Standardization (ISO) or American National Standards Institute (ANSI) drafting standards. The process involved orthogonal zig-zag vectorization, arc segmentation, arrowhead pair recognition, and initial textbox extraction through a region growing process on text-wire candidates. The drafting standard was determined based on the context of textboxes, and raw textboxes were logically divided and decomposed. A neural network-based optical character recognition (OCR) algorithm was applied to each basic textbox, and results were verified using contextual information and direct drawing measurements. The study presented an algorithm for the recognition of dimensioning text in engineering drawings. It also referenced recent successful work on vector-based segmentation but emphasized the need for combined efforts in segmentation and recognition improvement. Additionally, an automated updating mechanism for neural network weights after each verification process was proposed for self-teaching capabilities.

Xu and Wu (2003) also provided a comprehensive review of the automatic recognition of dimension sets, detailing related computerization processes and outlining the syntax and semantics involved in the elements and classification of dimension sets. Classification of dimension sets are usually based on their semantics including longitude, angle, diameter, chamfer, etc. Arrowheads are an important dimensioning graphic that were classified into four types in the study: regular symmetric (RS), regular asymmetric (RA), irregular symmetric (IS), or irregular asymmetric (IA). The study discussed the application of web representation to illustrate dimension sets. This included dimensioning grammar and text sub-grammar, hierarchically describing dimensioning components and their logical relationships. By recognizing text and arrowheads as inherent features of dimension sets, a specific vocabulary was defined to characterize components and their patterns. This vocabulary, along with dimensioning and text grammar, enabled the establishment of clear parsing rules. The study also provided the background on dimension set recognition techniques starting with text extraction. Pixel-level connected component (CC) on raster images to extract eight-connected components as text in addition to vector-based, morphological, and multi-stage relaxation methods were detailed.

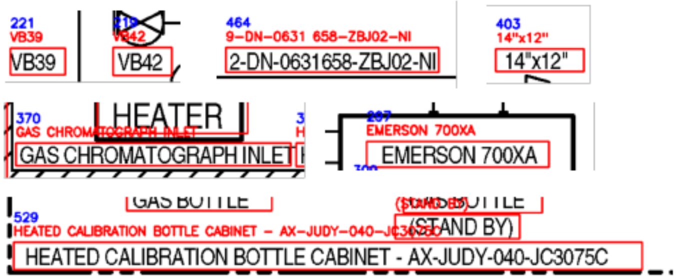
Ondrejcek et al. (2009) conducted a study to identify and maintain connections between engineering drawings and their modern equivalents, such as 3D computer-aided design (CAD) models. The study introduced a comprehensive prototype system named File2Learn, designed to (1) extract file system information, (2) conduct content-based analyses of two-dimensional (2D) engineering drawings and 3D CAD models, and (3) establish relationships among files through a semi-automated framework. The study assessed challenges related to the automation of information extraction, the organization and representation of extracted information, and information quality control. The automation challenge was addressed by scripting OCR operations using AutoHotKey scripts with the ABBYY OCR package. The organization and representation of information involved the adoption and extension of existing ontologies and the use of resource description framework. Information quality control was approached through an exploratory framework enabling editing of image areas and corresponding OCR strings. The prototype system was tested on 784 images of engineering drawings and over 22 CAD models. The results highlighted the manual and automated extraction of block and field coordinates, creation of master text files, and OCR processes. A prototype for information extraction was presented, emphasizing the need for manual preprocessing of real datasets due to the existence of non-standard blocks or block fields. The prototype, tested with engineering drawings and CAD models, helped detecting file relationships in a broader system.

De et al. (2011) introduced an algorithm designed for the recognition of electrical symbols in digitized documents. The algorithm utilized morphological operations and geometric analysis to identify various classes of symbols. A distinctive feature of the algorithm involved the creation and utilization of three spaces: H-space, V-space, and C-space. These spaces contained horizontal line segments, vertical line segments, and circuit symbols, respectively. Morphological operations built these spaces, and during geometric analysis, they were systematically searched and scanned to recognize symbols by validating the structural combination of their basic parts. The algorithm's robustness, efficiency, and versatility were verified as well. Testing was conducted on digitized documents sourced from various books and periodicals. The morphological operations and geometric analyses in H-space, V-space, and C-space were successfully executed on these document sets. The study provided details on the segmentation algorithm's application to a document page, resulting in the extraction of an engineering drawing and the associated H-, V-, and C-spaces. The detection of certain electrical symbols, such as earthing and multiple series-connected batteries, was not covered in their implementation but could be addressed with appropriate geometric analysis.

Datta et al. (2015) continued the previous study, focusing on the applications of decomposing and representing digital logic circuit drawings into a suitable vector format, critical for tasks such as data compression, storage, analysis, and editing. The study introduced an efficient method centered on segmenting and recognizing logic gate symbols from circuit drawing images. This segmentation process relied on morphological operations, followed by symbol identification using a decision tree classifier. The proposed method aimed to facilitate vectorization of circuit drawings by detecting information from segmented symbols and their connectivity matrices. The method's effectiveness was evaluated on a dataset comprising 53 scanned images of various digital logic circuit drawings, demonstrating successful symbol detection and identification. Notably, the algorithm offered advantages such as independence from symbol orientations and a straightforward detection process based on three distinct features (Euler number, spike, and circularity). However, a limitation of the method was its inability to detect broken symbols.

Ravagli et al. (2019) introduced a methodology for recognizing text in floor plan images, emphasizing the localization, reading, and categorization of text within these images to extract relevant building information. The primary focus was on comparing conventional text detection techniques, based on image processing, with contemporary methods such as CNNs. The paper aimed to enhance results by combining multiple methods to surpass the performance of individual ones. Text detection and text classification were two main modules of the system. Three different approaches were used to extract a list of rectangles around detected text regions, and then, a combined method was used to discard false positives. Additionally, text regions were classified into four semantic classes based on their intended purpose. The experiments involved two datasets with distinct characteristics, including variations in quality and size. The study comprehensively addressed text extraction, classification, and recognition in floor plan images, showcasing the improvement of original methods through tailored pre and post-processing steps designed for this specific task.

Jamieson et al. (2020) discussed the digitization of engineering drawings, particularly focusing on piping and instrumentation diagrams (P&IDs). These diagrams, crucial for identifying shapes, pipeline activities, and tags, often existed in an undigitized format. The paper reviewed recent advancements in deep learning, employing models for text detection and recognition to digitize text from complex engineering diagrams. A total of 172 complex P&IDs containing symbols, text, and connector lines were used as the dataset. For text detection in P&IDs, the Efficient and Accurate Scene Text (EAST) detector, which is a deep learning method, was employed. The EAST detector, chosen for its reported superior performance on various text detection tasks, outperforms other state-of-the-art methods in terms of F-score. In contrast to existing methods with multiple stages, EAST follows the process by directly producing text predictions from a single neural network, eliminating the need for intermediate steps like candidate proposals. The detector utilizes a fully convolutional network (FCN) based on the design of DenseBox, comprising a feature extractor stem, feature merging branch, and an output layer. The loss function used in training incorporates a balanced cross-entropy loss for the score map and a scale-invariant loss for geometry predictions. The EAST network is trained end-to-end using the Adam optimizer. For text recognition, long short-term memory (LSTM) networks were employed for recognizing text strings. The smallest available LSTM network in Tesseract was chosen for its processing speed. The combination of the EAST detector for text detection and LSTM-based text recognition was applied to digitize text from complex engineering diagrams, particularly P&IDs. The results indicated a 90% success rate in detecting text strings, including vertical instances, but challenges persisted, especially in scenarios with complex representations of text in close proximity to other drawing elements. The EAST detection model and LSTM-based text recognition showed promising results, as demonstrated by the experiments on a real-world P&ID dataset (Figure 2.5).

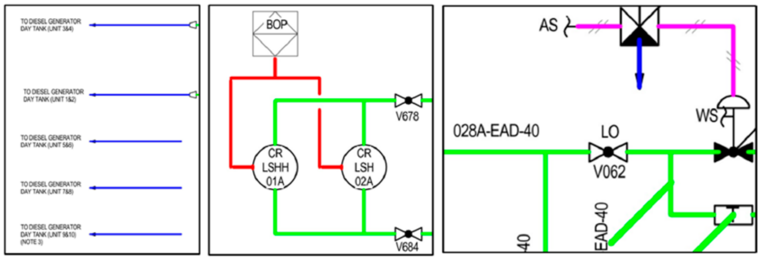


Jamieson et al. 2020

Figure 2.5. Example of correct text detection and recognition

The analysis of selected P&IDs revealed an average of 415 text instances per diagram, with 90% successfully detected by the EAST model. However, challenges included false positive detections, where non-text elements were misidentified as text, occurring in approximately 4% of output detections. Additionally, 11% of text instances were not detected on average. The study highlighted challenges related to text detection, such as difficulties in distinguishing text from non-text elements and issues with bounding box accuracy. Technical annotations, including line numbers, pose challenges when located close to other components. The study also addressed issues with text recognition, including instances of incorrect recognition, especially in vertical text instances.

As part of the research on technology for the automatic conversion of image-format P&ID into digital P&ID, the study by Moon et al. (2021) proposed a method to recognize various types of lines and flow arrows within images. The proposed approach involved three main steps: preprocessing, detection, and post-processing. In the preprocessing step, the outer border and title box were removed. The detection step involved identifying continuous lines, line signs, and flow arrows indicating flow direction. An object-detection based technique was exploited for the second step that outperformed pixel processing-based methods. The RetinaNet algorithm was used to detect line signs and flow arrows. Post-processing adjusted line types based on detected line signs and merged recognized lines with flow arrows. The verification of the proposed method through a prototype system yielded high recognition performance, with an average precision of 96.14% and an average recall of 89.59% for nine test P&IDs. The uniqueness of this research was its capability to recognize various types of lines and flow arrows, extending beyond continuous lines. The method incorporated techniques such as line thinning, pixel processing, and Hough transform for detection. Figure 2.6 shows the results of line recognition techniques for a test image.


TO DIESEL GENERATOR DAY TANK (UNIT 3&4)
TO DIESEL GENERATOR DAY TANK (UNIT 1&2)
TO DIESEL GENERATOR DAY TANK (UNIT 5&6)
TO DIESEL GENERATOR DAY TANK (UNIT 7&8)
TO DIESEL GENERATOR DAY TANK (UNIT 9&10) (NOTE 3)
BOP
CR LSHH 01A
CR LSH 02A
V678
V684
AS
WS
028A-EAD-40
LO
V062
40
EAD-40

Moon et al. 2021

Figure 2.6. Test example of the line recognition model

Automatic floor plan analysis has gained attention, with many studies focusing on experiments utilizing simplified floor plan datasets characterized by low resolution and a small housing scale. However, for practical applications, especially in the context of large-scale complex buildings, there is a need to shift the emphasis toward utilizing indoor structures, such as reconstructing multi-use buildings for indoor navigation. A study by Kim et al. (2021) addressed this gap by introducing a framework employing CNN models to analyze floor plans across various scales of complex buildings. The framework divided floor plans into normalized patches, allowing the CNN model to process inputs of varied scale or high resolution, a challenge faced by existing methods. The model detected building objects per patch, assembled them into a unified result using corresponding translation matrices, and vectorized the detected building objects, considering compatibility in 3D modeling. The model was trained on 200 Seoul National University building floor plan images and tested on 30 extra images. The framework included normalized patch extraction, floor recognition based on patches, and indoor model generators. Despite the complexity of the data used, the framework exhibited similar performance to existing studies with a detection rate of above 87% and recognition accuracy of above 85%. The practical implications of the study included (1) enabling the automatic extraction of indoor elements from complex and varied floor plan images, (2) extracting elements for both interior space geometry and connectivity with other floors, and (3) facilitating the reconstruction of indoor space in a standardized format for various purposes.

Scheibel et al. (2021) introduced DigiEDraw, presenting both a conceptual approach and a prototype aimed at extracting dimensioning information from drawings and integrating this data into the production process for enhanced and optimized quality control. The extraction process was founded on 2D clustering, utilizing density-based spatial clustering of applications with noise (DBSCAN) as a method to distinguish clusters representing different dimensioning information and achieving a recall value exceeding 88%. The goal of the proposed algorithm was to detect and extract text elements from drawings into dimension sets. The model consisted of preprocessing, clustering, and postprocessing steps. For the preprocessing step, the portable document format (PDF) version of the drawing was fed into the model, and then it was converted to HyperText Markup Language (HTML) to extract the text elements and bounding boxes. Several methods were evaluated for the clustering task, resulting in choosing DBSCAN as the desired method to combine extracted elements from a previous step into connected groups. The clusters were then post-processed in terms of data cleaning and element sorting.

Van Daele et al. (2021) introduced a tool designed to automate the interpretation of various components within technical drawings, enabling automatic reasoning and machine learning on a large technical drawing database. The tool aimed to facilitate tasks such as identifying designs and recognizing patterns in successful designs. The proposed method incorporated both neural and symbolic approaches. Neural methods were employed for visual image interpretation and recognition of parts within 2D drawings, while symbolic methods handled the relational structure and comprehension of data within complex tables present in technical drawings. The output from this method could be used to build a similarity-based search algorithm. A CNN model was trained on 318 images and tested on 53 examples. The study introduced five key contributions: (1) the use of inductive logic programming (ILP) to learn parsers for data extraction from tables, (2) a bootstrapping learning strategy for ILP, (3) a deep learning architecture for meaningful summarization of CAD drawings, (4) a similarity measure for identifying related technical drawings in a database, and (5) the demonstration of the method’s efficacy through experiments on a real-world dataset.

Xie at al. (2022) proposed a computational framework aimed at automatically identifying appropriate manufacturing methods for queried engineering drawings, such as lathing, sheet metal bending, and milling. The framework involved a series of preprocessing steps and a graph neural network to accurately determine manufacturing methods. These steps included a line tracing algorithm for transforming complex geometries into vectorized line segments with minimal information loss. An efficient image segmentation network isolated shape contours, followed by a graph neural network to detect and remove dimension lines. CascadeTabNet, which is a neural network, was used to segment drawing shapes from boxed text. Line thinning and vectorization processes were employed to convert shapes into vectorized line segments, enhancing storage efficiency and eliminating blank spaces while retaining geometric details. Dimension lines were then removed using graph-based methods to avoid interference in subsequent manufacturing process identification. A graph neural network classified the part's manufacturing process based on identified centerlines, employing differentiable pooling to extract global features. This hierarchical graph neural network provided accurate classification results. The proposed solution aimed to automate the quoting process for rapid manufacturing platforms, addressing the challenge of efficiently sorting engineering drawings by their required manufacturing methods. The framework was validated using a dataset compiled to demonstrate its accuracy in sorting engineering drawings by manufacturing methods.

3. METHODOLOGY

3.1. DATA COLLECTION

Datasets play a pivotal role in the success and efficacy of deep learning projects, serving as the foundation upon which neural networks learn and generalize patterns. The quality, size, and diversity of a dataset directly impact the model’s ability to understand complex relationships within the input data. A well-curated dataset ensures that the model encounters a representative sample of the real-world scenarios it is designed to handle, allowing it to generalize effectively to new, unseen data. Additionally, a diverse dataset helps the model adapt to various conditions, enhancing its robustness and applicability. The process of training a deep learning model involves adjusting its parameters based on the information present in the dataset, making it imperative to have a comprehensive and accurately labeled dataset.

The data collection process for this project involved sourcing information from the bridge plan sets from Iowa DOT and Caltrans. The initial phase of data acquisition involved extracting information from PDF files. To facilitate the deep learning model training, the PDFs were converted into image formats, providing a standardized and computationally efficient input for the neural network models. This conversion allowed for the preservation of the visual information embedded in the plans, ensuring that the models could learn and recognize structural features effectively. Converting all plan sets to individual images provided the data points required to perform various deep learning trainings.

To create a well-organized and informative dataset, different classes were defined based on the distinctive features and attributes present in the bridge plan sets. These classes could include various structural elements or design specifications relevant to the analysis goals of the project. The careful definition of classes ensures that the models can categorize the diverse components within the dataset accurately. Subsequently, multiple deep learning models were developed, each tailored to address specific classification tasks within the defined classes. The diversity of the dataset enables the models to learn and generalize effectively, contributing to their overall robustness in handling various scenarios and structural configurations. The iterative development process involved continuous refinement of the models. Figure 3.1 illustrates examples of images, i.e., bridge plan sheets, from two different States: Iowa and California.

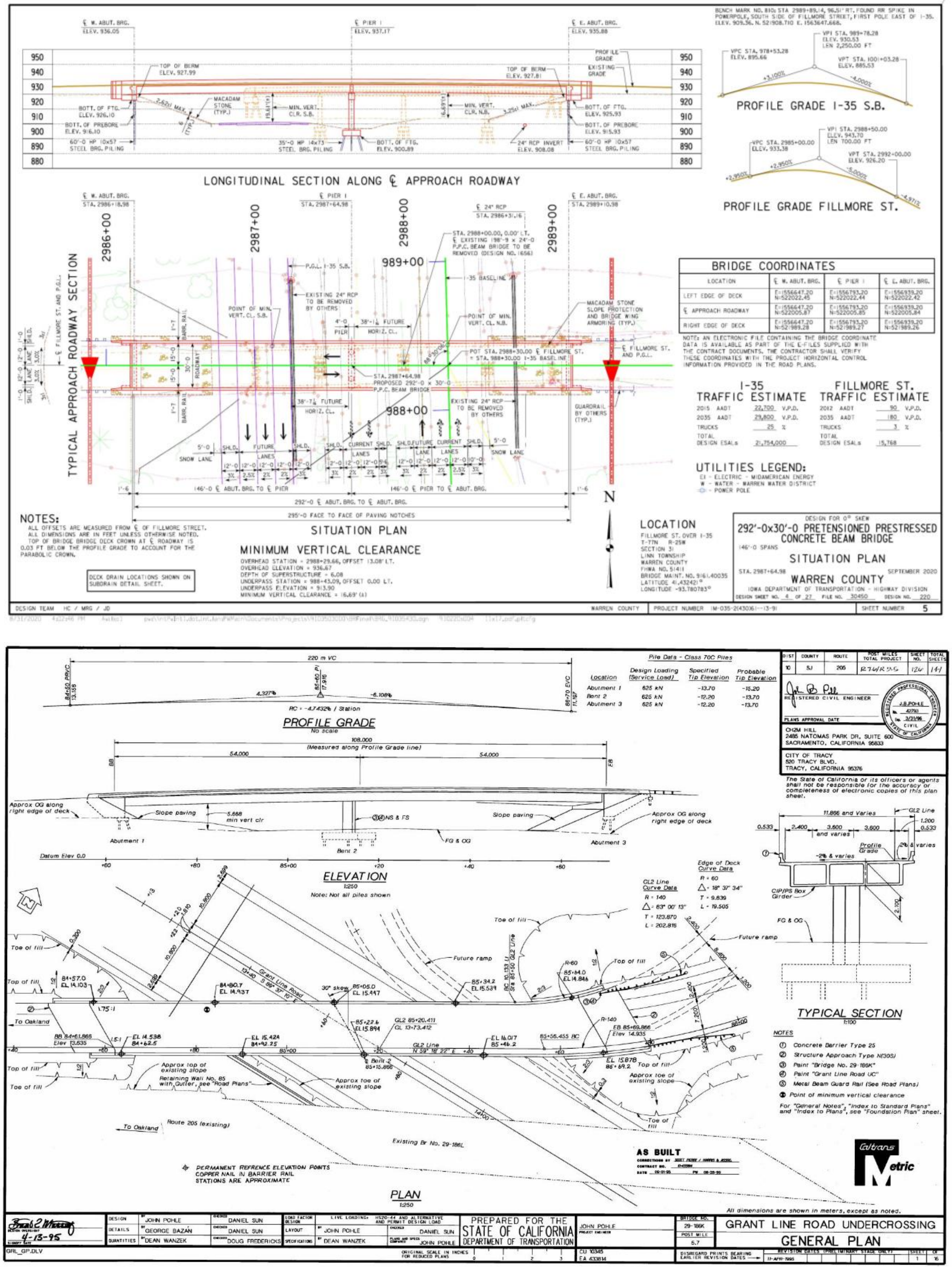

LONGITUDINAL SECTION ALONG ℄ APPROACH ROADWAY
PROFILE GRADE I-35 S.B.
PROFILE GRADE FILLMORE ST.
BRIDGE COORDINATES
TYPICAL APPROACH ROADWAY SECTION
SITUATION PLAN
NOTES:
MINIMUM VERTICAL CLEARANCE
I-35 TRAFFIC ESTIMATE
FILLMORE ST. TRAFFIC ESTIMATE
UTILITIES LEGEND:
LOCATION
292'-0x30'-0 PRETENSIONED PRESTRESSED CONCRETE BEAM BRIDGE
SITUATION PLAN
WARREN COUNTY
IOWA DEPARTMENT OF TRANSPORTATION - HIGHWAY DIVISION
PROFILE GRADE
ELEVATION
PLAN
TYPICAL SECTION
AS BUILT
GRANT LINE ROAD UNDERCROSSING
GENERAL PLAN
STATE OF CALIFORNIA
DEPARTMENT OF TRANSPORTATION

Figure 3.1. Data point examples from Iowa (top) and California (bottom)

3.2. DEEP LEARNING MODELS

Data labeling is a crucial step in the training process of computer vision models. It involves annotating raw data to provide supervised learning algorithms with labeled examples that facilitate pattern recognition. Data labeling typically involves outlining and identifying objects of interest within images, defining their boundaries and categories. Accurate and comprehensive data labeling is essential for the model to learn and generalize effectively. The quality of labeled data directly influences the model's ability to detect and classify objects accurately during inference. Manual labeling by human annotators or automated labeling tools is common, and the choice depends on factors such as dataset size, complexity, and the required precision.

The methodology employed in this study utilizes a comprehensive platform designed for image dataset management, annotation, and preprocessing. The integration of this platform into the image labeling process is fundamental to enhancing the efficiency of dataset preparation for subsequent deep learning model training. Image labeling functionality relies on a user-friendly annotation interface. Annotators utilize this interface to mark objects of interest within images, defining bounding boxes or segmentation masks as necessary. The platform supports multiple annotation formats, ensuring flexibility and compatibility with various deep learning frameworks. The dataset versioning and management capabilities are also critical for maintaining version control and ensuring the reproducibility of experiments. The platform enables users to track changes, revert to previous dataset versions, and maintain a comprehensive history of modifications, contributing to overall project organization and reproducibility.

Dataset augmentation is a critical component of the data preprocessing pipeline, enhancing the robustness and diversity of training datasets for computer vision models. The augmentation capabilities involve the application of various transformations to input images, such as rotation, scaling, flipping, and changes in brightness and contrast. These augmentations create additional variations in the dataset, effectively expanding its size and ensuring that the trained model becomes more resilient to different environmental conditions and scenarios. The exported dataset is organized and formatted based on different computer vision models. Typically, the dataset includes the original images along with corresponding annotations, where objects of interest are precisely labeled with bounding boxes and associated class labels.

The first step in extracting structural features is to detect and identify the objects. For this purpose, a plan sheet should be divided into individual parts, each indicating one structural component and its details. The first deep learning model aims to detect objects of interest within a plan sheet to enhance the pipeline’s capability for more refined feature extraction. For this purpose, the first model was created to detect and localize different objects such as a plan view of the bridge to measure the general dimensions or a tabular structure to extract reinforcement properties. The annotated dataset included distinct classes, such as “elevation,” “layout,” “nameplate,” “plan,” “section,” “symbol,” “table,” and “textblock.”

The images were augmented to increase the size of dataset with different filters including crop, cutout, and mosaic. Cutout is a technique where random sections of the input images are “cut out” or masked during training. Also, mosaic is a technique where multiple images are combined into a single training example. This can help the model learn to recognize objects in different contexts and configurations. The dataset was then classified into training, validation, and testing sets. This process helped automatically detect various bridge details and components in a plan sheet. The model was trained on the training set, hyperparameters were fine-tuned on the validation set, and the final model was tested on the test set. Figure 3.2 shows the metrics and losses during the training of the model. The figure shows three different bounding box regressions (MSE), object presence confidence, and classification (cross entropy) losses.

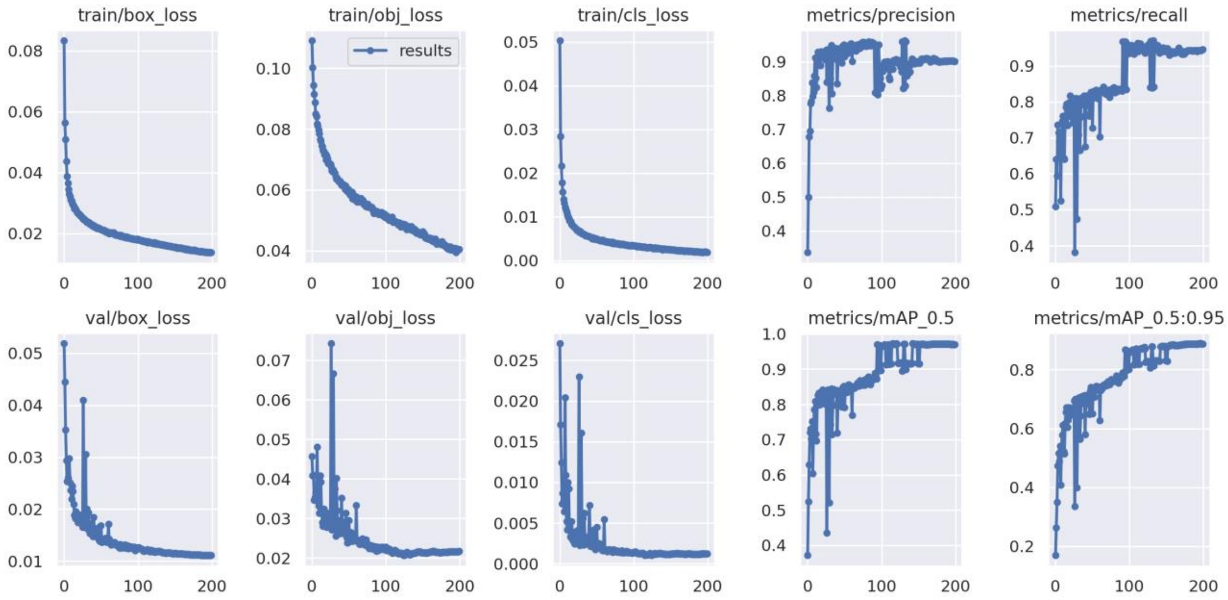


Figure 3.2. Metrics and losses of the first model

Bounding box regression measures how well the predicted bounding box coordinates match the ground truth bounding box coordinates. The bounding box regression loss penalizes the model for the discrepancy between predicted and actual bounding box positions. A decrease in the bounding box regression loss indicates that the model is getting better at accurately predicting the location and size of objects. As training progresses, the model learns to refine its bounding box predictions, leading to a reduction in the MSE. The object presence confidence loss penalizes the model for incorrectly predicting the presence or absence of an object in a cell. A decrease in the object presence confidence loss indicates that the model is becoming more accurate in determining whether an object is present in a specific grid cell. As training advances, the model learns to assign higher confidence scores to correct object predictions and lower scores to incorrect predictions. The classification loss measures the difference between the predicted class probabilities and the ground truth class labels using cross-entropy. A decrease in the classification loss suggests that the model is improving its ability to correctly classify objects. As training continues, the model refines its understanding of object categories, leading to a reduction in the cross-entropy loss.

Also, three metrics of precision, recall, and mean average precision (mAP) results are shown in Figure 3.2. Precision shows the number of true positives to the number of positives, indicating the amount of bounding box correct predictions. Recall also measures the number of correct predictions of the true bounding boxes by calculating the number of true positives over the sum of true positives and false negatives. The extent of overlap of two boxes is measured by the intersection over union (IoU) metric. The mAP at the IoU of 0.5 is indicated by mAP_0.5, and the average of mAP over an IoU range of 0.5 to 0.95 is shown by mAP_0.5:0.95 in the figure. All three metrics show increasing trends over the epochs, representing a good convergence of the base algorithm on the custom data.

A confusion matrix is often employed to evaluate the performance of the object detection model. The confusion matrix is constructed based on the comparison between predicted bounding boxes and the ground truth bounding boxes. Each entry in the matrix represents a count of how many instances fall into specific categories. Given the nature of object detection tasks, the diagonal of the confusion matrix typically captures the instances where predictions align with the ground truth. The elements in a confusion matrix are as follows:

1. **True Positive (TP):**
   Definition: The number of correctly predicted bounding boxes.
   Interpretation: The model correctly identified and localized objects, and these instances are counted along the diagonal.
2. **True Negative (TN):**
   Definition: Not applicable for object detection tasks, as the model's objective is to identify and localize objects, not to predict the absence of objects in specific regions.
3. **False Positive (FP) - Type I Error:**
   Definition: The number of predicted bounding boxes where there is no corresponding ground truth bounding box.
   Interpretation: The model predicted the presence of an object where none exists in reality.
4. **False Negative (FN) - Type II Error:**
   Definition: The number of ground truth bounding boxes with no corresponding predicted bounding box.
   Interpretation: The model failed to detect and predict the presence of an object that actually exists in reality.

The confusion matrix for the first model is shown in Figure 3.3.

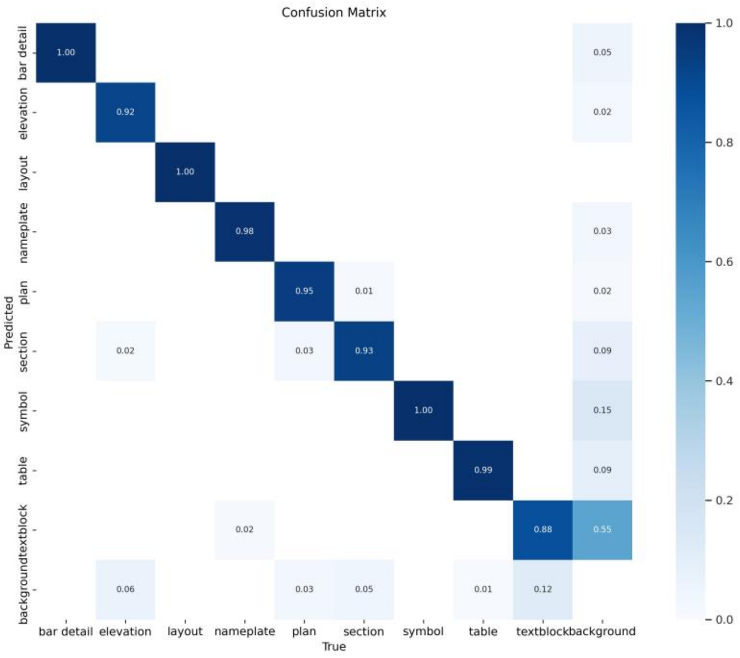


Figure 3.3. Confusion matrix of the first model

As seen, high values on the diagonal indicates that the base model is performing well in terms of correct detections. The diagonal represents instances where the predicted bounding boxes match the ground truth, signifying successful localization and classification of objects. The rightmost column could be used for detecting instances of false positives, where there are predicted bounding boxes with no matching ground truth. As seen, there is considerable chance for false positives in the textblock class and symbols might be prone to false positives in some cases. All in all, the model shows very high chances for correct predictions for the majority of the classes.

The performance diagrams of the base model are illustrated in Figure 3.4.

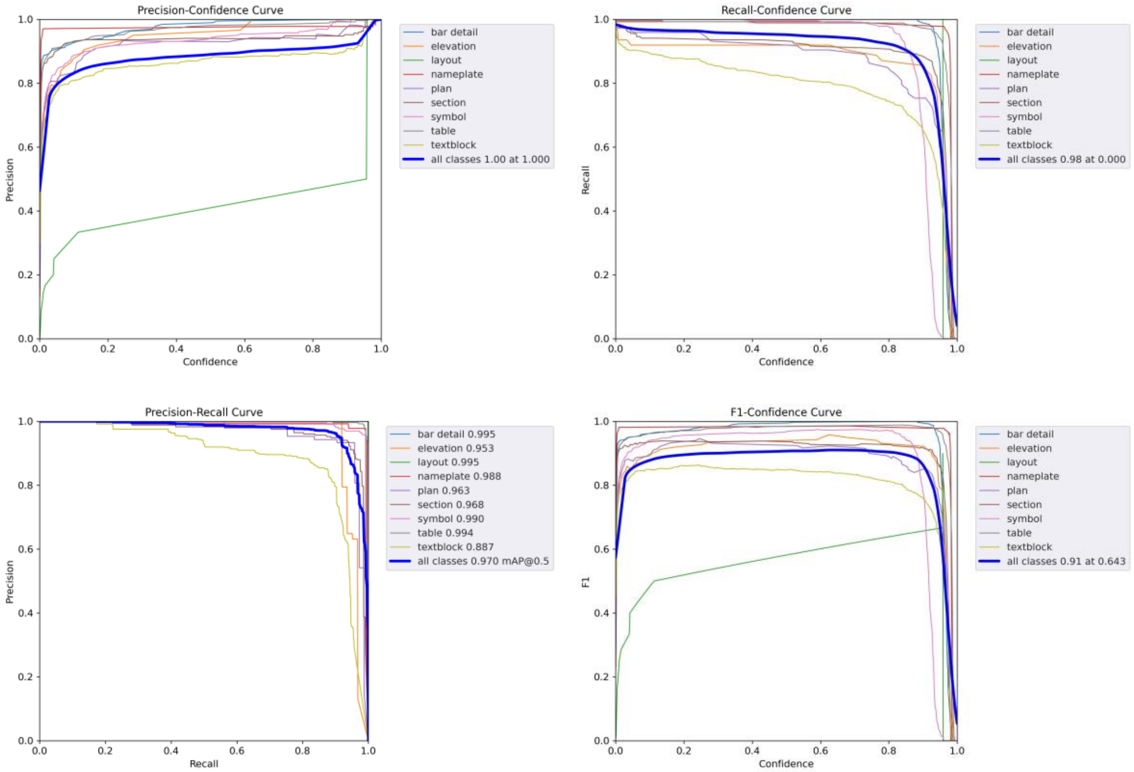


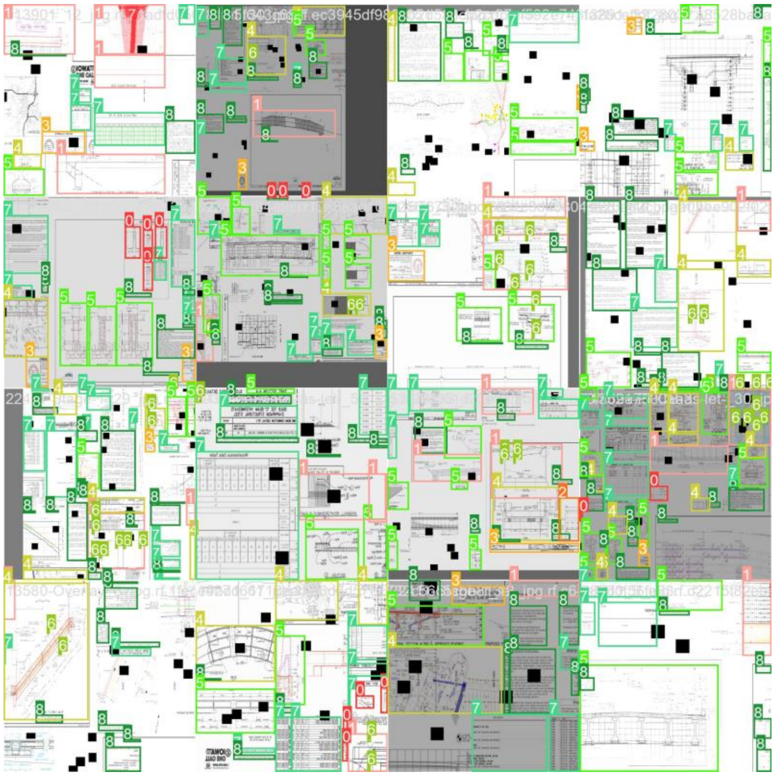
Figure 3.4. Performance diagram of the first model

The graphs include precision, recall, and F1 score confidence trends as well as precision-recall curves. The confidence curves show the precision, recall, and F1 score for different confidence values.

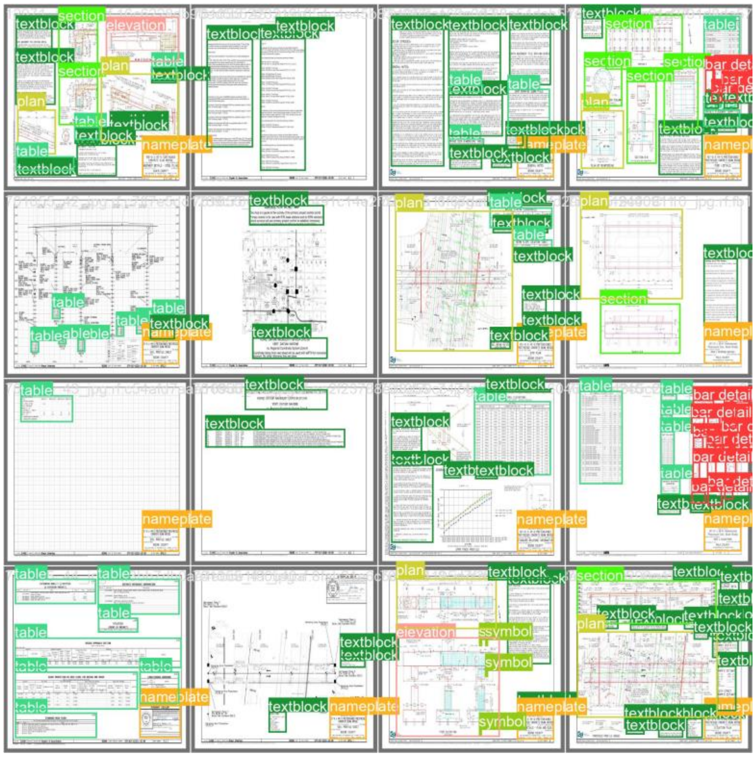
1. **Precision-Confidence Curve.** The precision-confidence curve shows how precision varies with changing confidence thresholds. Precision measures the accuracy of positive predictions made by the model. An increasing curve indicates that, as the confidence threshold rises, the model becomes more conservative in making positive predictions, resulting in higher precision. This can be beneficial when aiming to reduce false positives at the cost of potentially missing some detections.
2. **Recall-Confidence Curve.** The recall-confidence curve depicts how recall changes as the confidence threshold for positive predictions varies. As the model increases the confidence threshold for positive predictions, the model is becoming more conservative in its predictions. It is still able to maintain a relatively high recall, indicating that it is capturing most of the relevant instances with reasonable confidence.

3. **Precision-Recall Curve.** The precision-recall curve illustrates the trade-off between precision and recall across different confidence thresholds. As such, the precision should remain high as the recall increases, maximizing the area under the curve. The curve shows a high area under the curve for most of the classes, indicating the good performance of the first mode.
4. **F1 Score-Confidence Curve.** The F1 score-confidence curve illustrates the relationship between the F1 score and the confidence threshold for object detection across different classes. An F1 score balances precision and recall, providing a single metric that considers both false positives and false negatives. The figure shows high F1 scores at the specified confidence thresholds, indicating a good trade-off between precision and recall for the given classes.

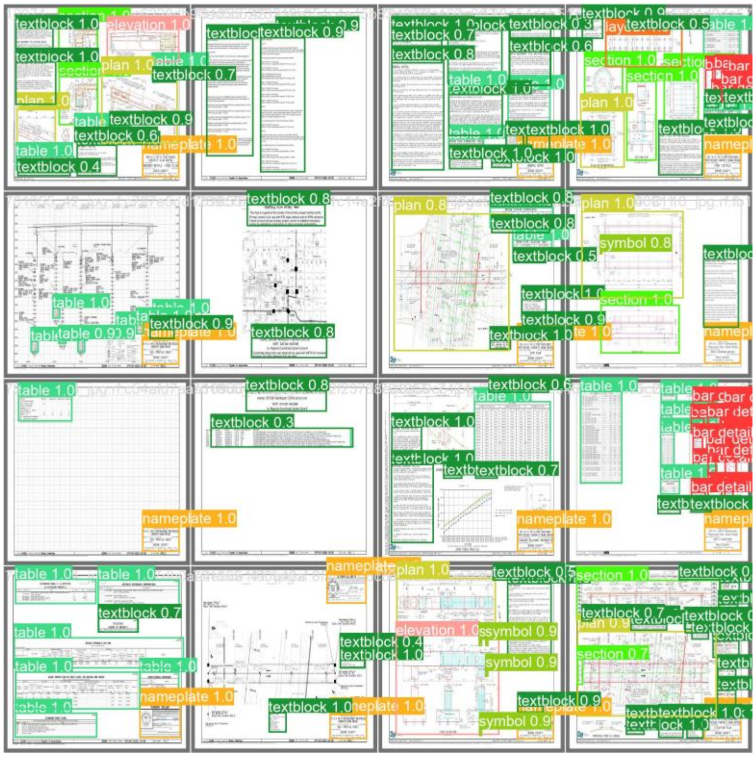
Figure 3.5 shows some photos from the training batches containing different classes and some examples of the validation set. As seen, the model can detect most of the classes of the validation set with high accuracy.



a. Training set



b. Validation set – Labels



c. Validation set – Predictions

Figure 3.5. Examples of the developed base model's performance on training and validation sets

Based on the desired properties requested to be extracted from the bridge plan sets, the base model was expanded with more refined classes. For this purpose, different objects such as substructure units, columns in a bent, and girders were identified and located. The images were augmented to increase the size of the dataset with different filters, including horizontal and vertical flip, rotation, and noise. The dataset was then classified into training, validation, and testing sets. The expanded model was trained on the training set, hyperparameters were fine-tuned on the validation set, and the final model was tested on the test set. Figure 3.6 shows the metrics and losses during the training of the expanded model. The figure shows three different bounding box regressions (MSE), object presence confidence, and classification (cross entropy) losses.

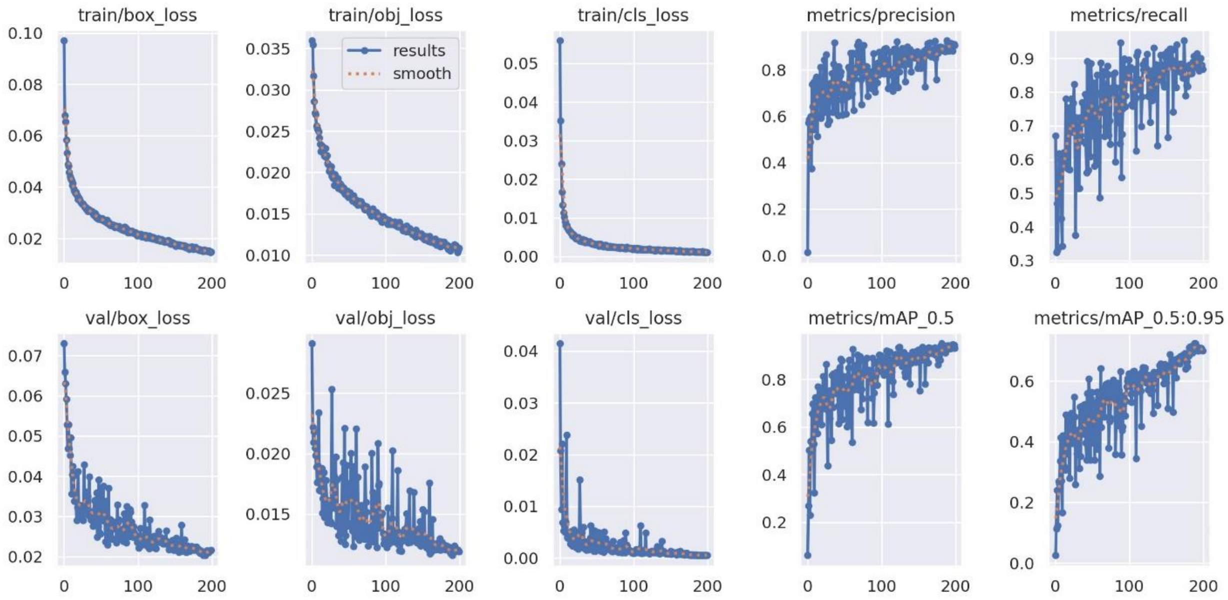


Figure 3.6. Metrics and losses of the expanded model

The decrease in the bounding box regression loss indicates that the model is getting better at accurately predicting the location and size of objects. As training progresses, the model learns to refine its bounding box predictions, leading to a reduction in the MSE. The decrease in the object presence confidence loss also indicates that the model is becoming more accurate in determining whether an object is present in a specific grid cell. As training advances, the model learns to assign higher confidence scores to correct object predictions and lower scores to incorrect predictions. In addition, the decrease in the classification loss suggests that the model is improving its ability to correctly classify objects. As training continues, the model refines its understanding of object categories, leading to a reduction in the cross-entropy loss.

Also, three metrics of precision, recall, and mAP results are evaluated. All the three metrics show increasing trends over the epochs, representing a good convergence of the developed algorithm on custom data.

The confusion matrix for the expanded model is shown in Figure 3.7.

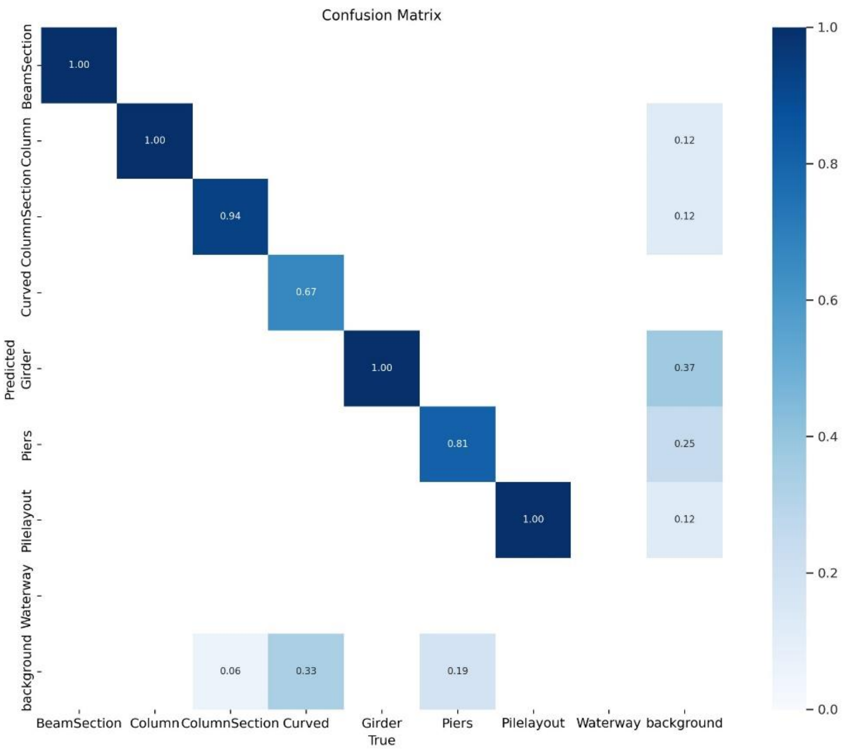


Figure 3.7. Confusion matrix of the expanded model

As seen, high values on the diagonal indicates that the expanded model is performing well in terms of correct detections. The diagonal represents instances where the predicted bounding boxes match the ground truth, signifying successful localization and classification of objects. The rightmost column could be used for detecting instances of false positives, where there are predicted bounding boxes with no matching ground truth. As seen, there are some instances of false positives in the Girder class. Also, the model might miss some cases of Curved class, as there are few instances of this class in the dataset. All in all, the model shows very high chances for correct predictions for the majority of the classes.

The performance diagrams of the expanded model are illustrated in Figure 3.8.

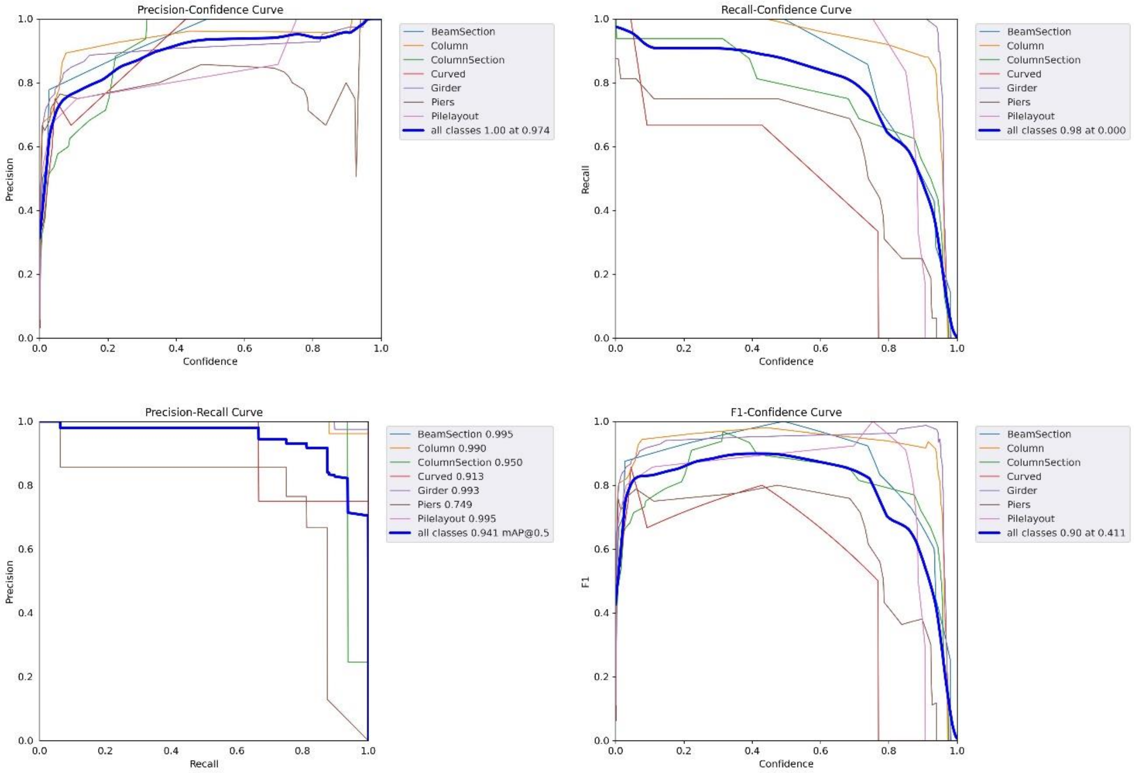
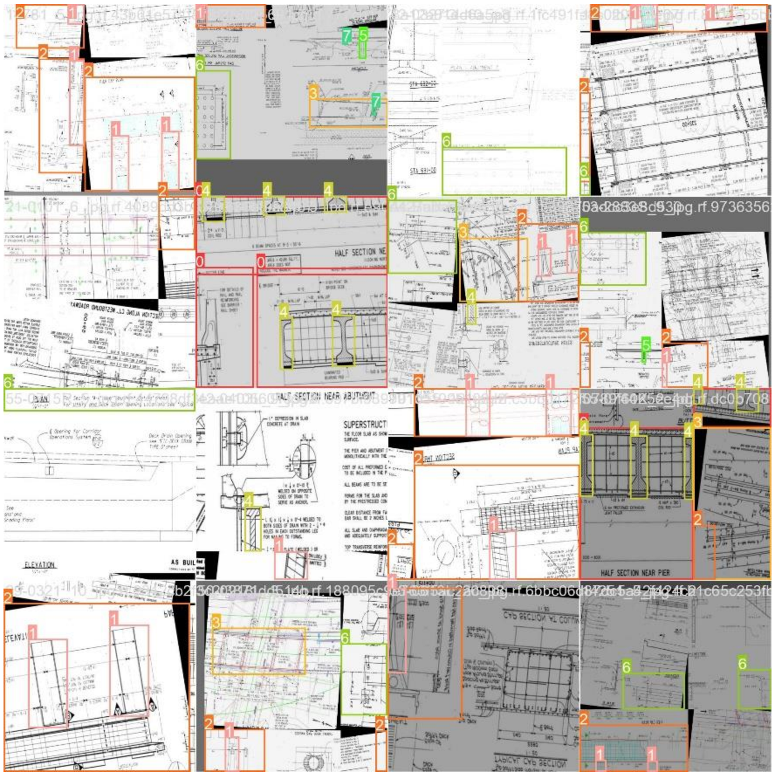
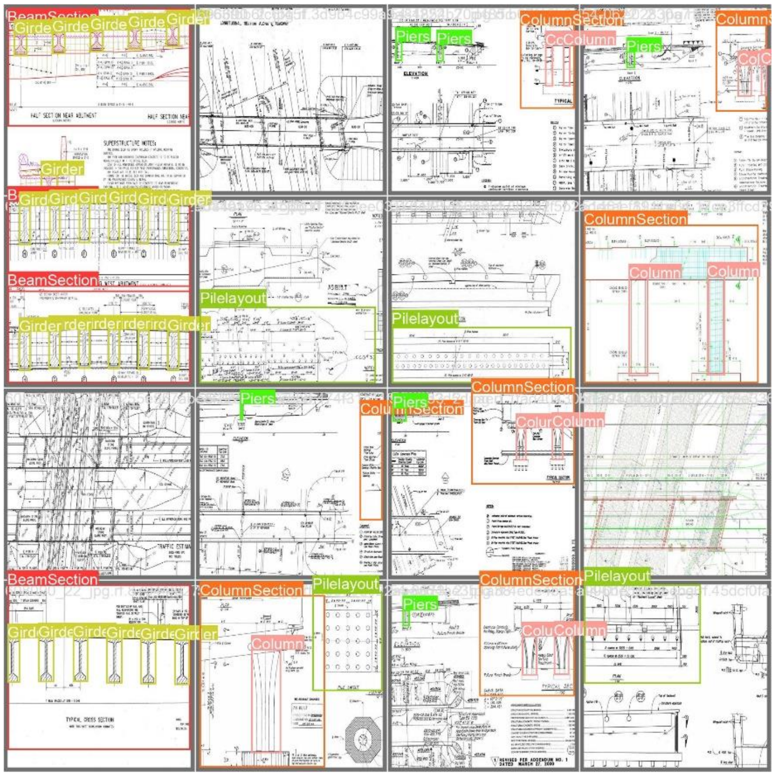


Figure 3.8. Performance diagram of the expanded model

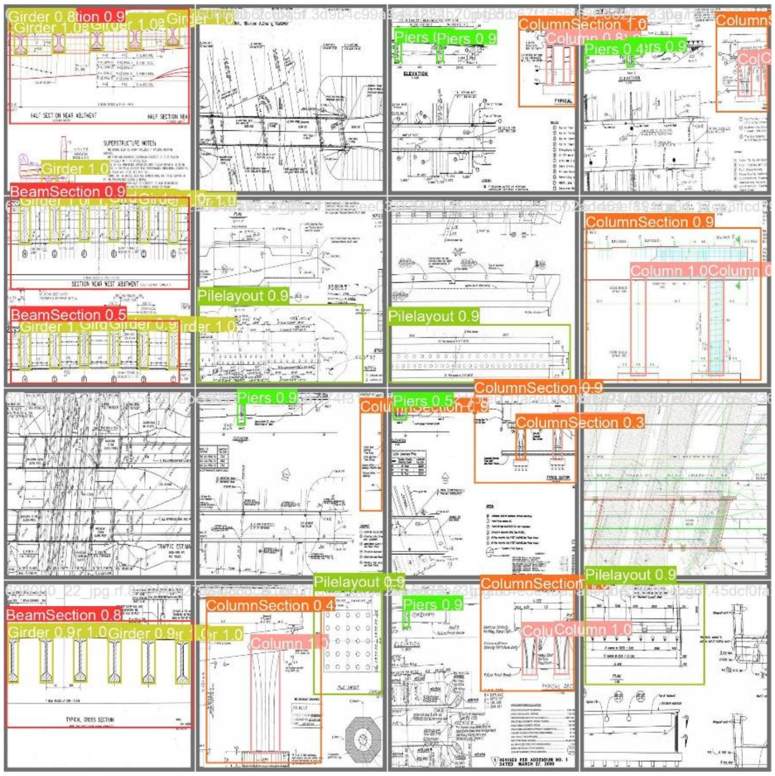
The graphs include precision, recall, and F1 score confidence trends as well as precision-recall curves. The confidence curves show the precision, recall, and F1 score for different confidence values. Figure 3.9 shows some photos from the training batches containing different classes and some examples of the validation set. As seen, the model can detect most of the classes of the validation set with high accuracy.



a. Training set



b. Validation set – Labels



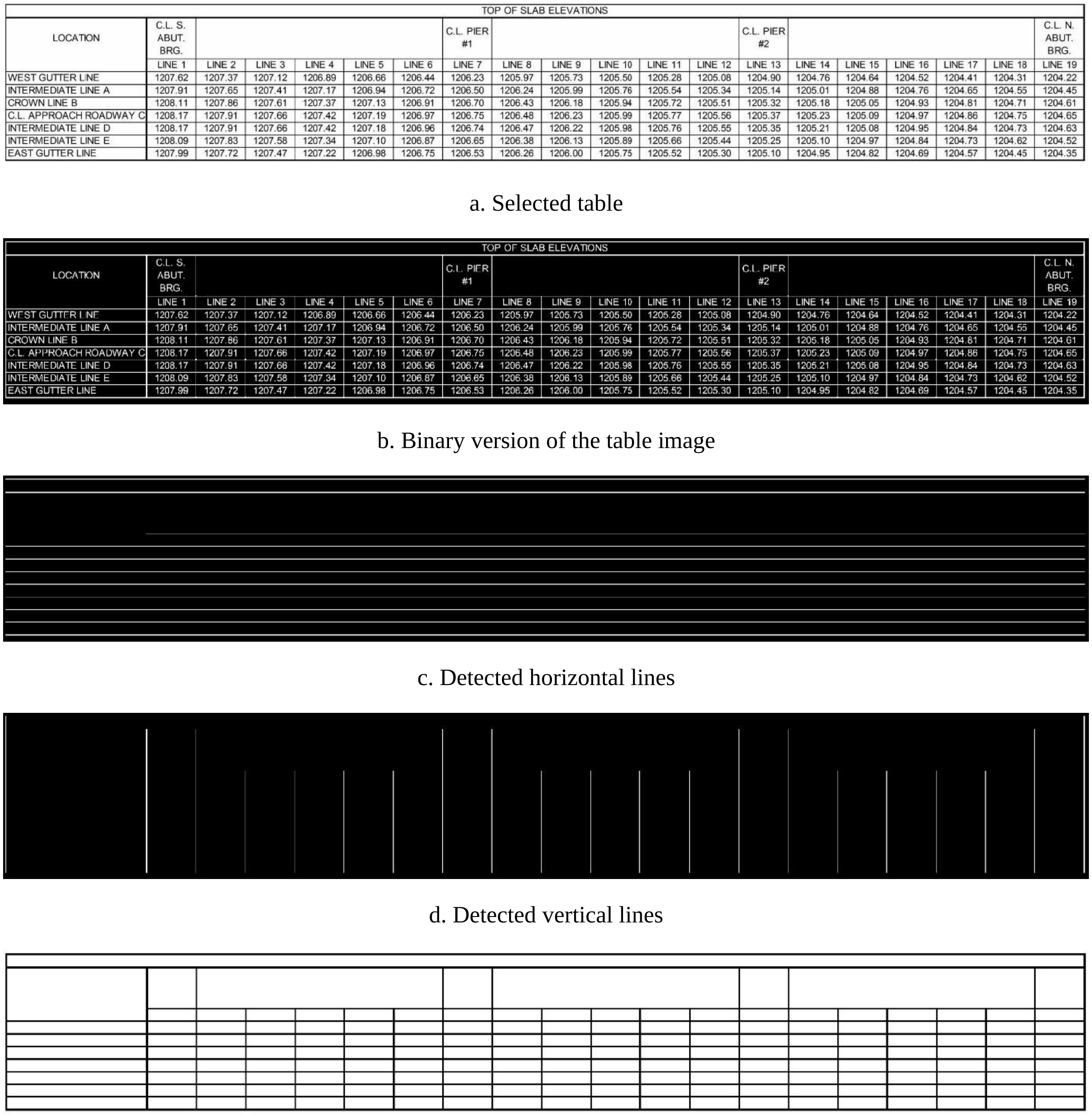
c. Validation set – Predictions

Figure 3.9. Examples of the expanded model’s performance on training and validation sets

3.3. ADDITIONAL FUNCTIONS

As the expanded model can iterate through all the images to detect and localize tabular structures, a subsequent step is required to extract the information from the detected tables and convert the table’s image to an editable version such as an Excel spreadsheet.

Recognition of the tabular structure of a table begins by reading the input image to create a binary representation. Additional operations are subsequently applied to detect and enhance vertical and horizontal lines in the image. Figure 3.10 shows a table reporting top of slab elevations, along with its binary version and horizontal and vertical lines.



TOP OF SLAB ELEVATIONS																			
LOCATION	C.L. S. ABUT. BRG.						C.L. PIER #1						C.L. PIER #2						C.L. N. ABUT. BRG.
	LINE 1	LINE 2	LINE 3	LINE 4	LINE 5	LINE 6	LINE 7	LINE 8	LINE 9	LINE 10	LINE 11	LINE 12	LINE 13	LINE 14	LINE 15	LINE 16	LINE 17	LINE 18	LINE 19
WEST GUTTER LINE	1207.62	1207.37	1207.12	1206.89	1206.66	1206.44	1206.23	1205.97	1205.73	1205.50	1205.28	1205.08	1204.90	1204.76	1204.64	1204.52	1204.41	1204.31	1204.22
INTERMEDIATE LINE A	1207.91	1207.65	1207.41	1207.17	1206.94	1206.72	1206.50	1206.24	1205.99	1205.76	1205.54	1205.34	1205.14	1205.01	1204.88	1204.76	1204.65	1204.55	1204.45
CROWN LINE B	1208.11	1207.86	1207.61	1207.37	1207.13	1206.91	1206.70	1206.43	1206.18	1205.94	1205.72	1205.51	1205.32	1205.18	1205.05	1204.93	1204.81	1204.71	1204.61
C.L. APPROACH ROADWAY C	1208.17	1207.91	1207.66	1207.42	1207.19	1206.97	1206.75	1206.48	1206.23	1205.99	1205.77	1205.56	1205.37	1205.23	1205.09	1204.97	1204.86	1204.75	1204.65
INTERMEDIATE LINE D	1208.17	1207.91	1207.66	1207.42	1207.18	1206.96	1206.74	1206.47	1206.22	1205.98	1205.76	1205.55	1205.35	1205.21	1205.08	1204.95	1204.84	1204.73	1204.63
INTERMEDIATE LINE E	1208.09	1207.83	1207.58	1207.34	1207.10	1206.87	1206.65	1206.38	1206.13	1205.89	1205.66	1205.44	1205.25	1205.10	1204.97	1204.84	1204.73	1204.62	1204.52
EAST GUTTER LINE	1207.99	1207.72	1207.47	1207.22	1206.98	1206.75	1206.53	1206.26	1206.00	1205.75	1205.52	1205.30	1205.10	1204.95	1204.82	1204.69	1204.57	1204.45	1204.35

a. Selected table

TOP OF SLAB ELEVATIONS																			
LOCATION	C.L. S. ABUT. BRG.						C.L. PIER #1						C.L. PIER #2						C.L. N. ABUT. BRG.
	LINE 1	LINE 2	LINE 3	LINE 4	LINE 5	LINE 6	LINE 7	LINE 8	LINE 9	LINE 10	LINE 11	LINE 12	LINE 13	LINE 14	LINE 15	LINE 16	LINE 17	LINE 18	LINE 19
WEST GUTTER LINE	1207.62	1207.37	1207.12	1206.89	1206.66	1206.44	1206.23	1205.97	1205.73	1205.50	1205.28	1205.08	1204.90	1204.76	1204.64	1204.52	1204.41	1204.31	1204.22
INTERMEDIATE LINE A	1207.91	1207.65	1207.41	1207.17	1206.94	1206.72	1206.50	1206.24	1205.99	1205.76	1205.54	1205.34	1205.14	1205.01	1204.88	1204.76	1204.65	1204.55	1204.45
CROWN LINE B	1208.11	1207.86	1207.61	1207.37	1207.13	1206.91	1206.70	1206.43	1206.18	1205.94	1205.72	1205.51	1205.32	1205.18	1205.05	1204.93	1204.81	1204.71	1204.61
C.L. APPROACH ROADWAY C	1208.17	1207.91	1207.66	1207.42	1207.19	1206.97	1206.75	1206.48	1206.23	1205.99	1205.77	1205.56	1205.37	1205.23	1205.09	1204.97	1204.86	1204.75	1204.65
INTERMEDIATE LINE D	1208.17	1207.91	1207.66	1207.42	1207.18	1206.96	1206.74	1206.47	1206.22	1205.98	1205.76	1205.55	1205.35	1205.21	1205.08	1204.95	1204.84	1204.73	1204.63
INTERMEDIATE LINE E	1208.09	1207.83	1207.58	1207.34	1207.10	1206.87	1206.65	1206.38	1206.13	1205.89	1205.66	1205.44	1205.25	1205.10	1204.97	1204.84	1204.73	1204.62	1204.52
EAST GUTTER LINE	1207.99	1207.72	1207.47	1207.22	1206.98	1206.75	1206.53	1206.26	1206.00	1205.75	1205.52	1205.30	1205.10	1204.95	1204.82	1204.69	1204.57	1204.45	1204.35

b. Binary version of the table image

c. Detected horizontal lines

d. Detected vertical lines

e. Combined vertical and horizontal lines

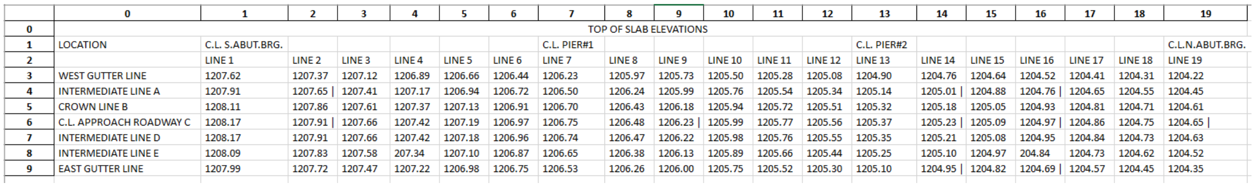
Figure 3.10. Example of a table transferred to a readable format

The resulting lines are combined, and additional operations refine the image further for contour detection. The contours are extracted and sorted top to bottom. The code then iterates through the contours, creating bounding boxes around potential cells within the table. These boxes are organized into rows and columns based on their spatial relationships, using mean height analysis.

With a final sorting of values, the resulting text is structured and organized, forming the content of the table.

The final step involves converting the extracted data into an Excel file using appropriate libraries. The resulting Excel file is saved to a specified output path, providing a tangible and accessible representation of the tabular information present in the original plan sheet. Table 3.1 shows the result of performing the outlined steps on the table image.

Table 3.1. Regenerated table in an Excel file.

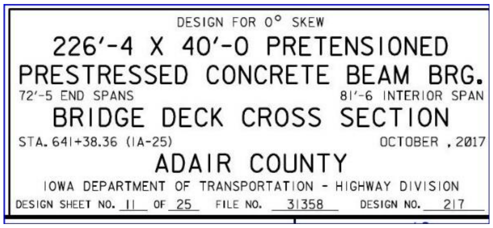


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	TOP OF SLAB ELEVATIONS																			
1	LOCATION	C.L. S.ABUT.BRG.						C.L. PIER#1						C.L. PIER#2						C.L.N.ABUT.BRG.
2		LINE 1	LINE 2	LINE 3	LINE 4	LINE 5	LINE 6	LINE 7	LINE 8	LINE 9	LINE 10	LINE 11	LINE 12	LINE 13	LINE 14	LINE 15	LINE 16	LINE 17	LINE 18	LINE 19
3	WEST GUTTER LINE	1207.62	1207.37	1207.12	1206.89	1206.66	1206.44	1206.23	1205.97	1205.73	1205.50	1205.28	1205.08	1204.90	1204.76	1204.64	1204.52	1204.41	1204.31	1204.22
4	INTERMEDIATE LINE A	1207.91	1207.65 |	1207.41	1207.17	1206.94	1206.72	1206.50	1206.24	1205.99	1205.76	1205.54	1205.34	1205.14	1205.01 |	1204.88	1204.76 |	1204.65	1204.55	1204.45
5	CROWN LINE B	1208.11	1207.86	1207.61	1207.37	1207.13	1206.91	1206.70	1206.43	1206.18	1205.94	1205.72	1205.51	1205.32	1205.18	1205.05	1204.93	1204.81	1204.71	1204.61
6	C.L. APPROACH ROADWAY C	1208.17	1207.91 |	1207.66	1207.42	1207.19	1206.97	1206.75	1206.48	1206.23 |	1205.99	1205.77	1205.56	1205.37	1205.23 |	1205.09	1204.97 |	1204.86	1204.75	1204.65 |
7	INTERMEDIATE LINE D	1208.17	1207.91	1207.66	1207.42	1207.18	1206.96	1206.74	1206.47	1206.22	1205.98	1205.76	1205.55	1205.35	1205.21	1205.08	1204.95	1204.84	1204.73	1204.63
8	INTERMEDIATE LINE E	1208.09	1207.83	1207.58	207.34	1207.10	1206.87	1206.65	1206.38	1206.13	1205.89	1205.66	1205.44	1205.25	1205.10	1204.97	204.84	1204.73	1204.62	1204.52
9	EAST GUTTER LINE	1207.99	1207.72	1207.47	1207.22	1206.98	1206.75	1206.53	1206.26	1206.00	1205.75	1205.52	1205.30	1205.10	1204.95 |	1204.82	1204.69 |	1204.57	1204.45	1204.35

As reflected in the reported values, the model was able to detect, localize, and report the cell values with maximum accuracy. This methodology offers a systematic and automated approach to transform image-based tables into an editable format.

A crucial aspect of feature extraction from bridge plan sets is thorough and accurate text detection as almost all of the desired features and information lies within the textual patterns of the plan sheets. Therefore, a text detector with high performance is required to manipulate data within a bridge plan sheet.

As the plan sets are available in different formats including scanned images and PDF versions, various state-of-the-art algorithms were employed to extract textual information. Figure 3.11 shows an example of the reported text of a nameplate object.



DESIGN FOR O° SKEW

226′-4 X 40′-O PRETENSIONED

PRESTRESSED CONCRETE BEAM BRG.

72′-5 END SPANS 81′-6 INTERIOR SPAN

BRIDGE DECK CROSS SECTION

STA. 641+38.36 (IA-25) OCTOBER , 2017

ADAIR COUNTY

IOWA DEPARTMENT OF TRANSPORTATION - HIGHWAY DIVISION

DESIGN SHEET NO. 11 OF 25 FILE NO. 31358 DESIGN NO. 217

a. Original image

```
DESIGN FOR O° SKEW

226′-4 X 40′-O PRETENSIONED

PRESTRESSED CONCRETE BEAM BRG.


72′-5 END SPANS 81′-6 INTERIOR SPAN

BRIDGE DECK CROSS SECTION

STA. 641+38.36 (1A-25) OCTOBER , 2017

ADAIR COUNTY

IOWA DEPARTMENT OF TRANSPORTATION - HIGHWAY DIVISION

DESIGN SHEET NO. || OF _25 FILE NO. __ 31358 DESIGN NO. 217
```

b. Converted text using Tesseract-OCR

Figure 3.11. Example of text detection

4. DATA EXTRACTION PIPELINES

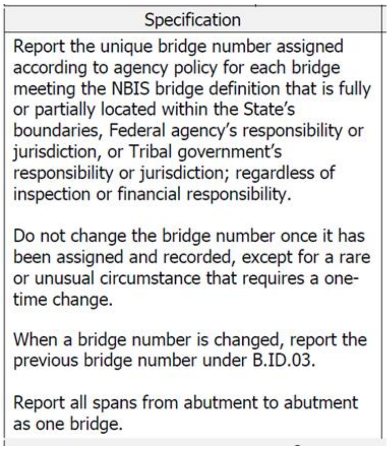
The *Engineering Details Extraction Pipeline* is a comprehensive solution designed in this project to extract crucial engineering information from bridge plan sets. Using deep learning models and advanced image processing techniques, along with additional functions introduced in the previous chapter, this pipeline automates the extraction process, enhancing efficiency and accuracy in handling engineering documents. The pipeline combines information extracted from different components, creating a holistic understanding of the engineering details present in the plan sets. The pipeline pulls data from diverse sources, including bridge plan sets, NBIS guidelines, and relevant websites. By integrating information from multiple channels, the system enriches the extracted data, providing a comprehensive pipeline for various bridge-related details.

4.1. BRIDGE DETAILS OF INTEREST

In this project, a "bridge plan set" is considered to be a set of bridge plan sheets in either an image or PDF format that contains extensive information about the sizing and materials used in a bridge in the form of text, tables, or dimension lines. The plan set offers a full view of the "plan" and "elevation" of the bridge. Each plan sheet has a bounded area that contains at least the names of the included bridge components for the purpose of navigating through the plan sheets to find the pages that have information about the objects of interest. After setting up all the necessary libraries, models, and functions, the pipeline starts using these components to extract the desired data from the bridge plan sets.

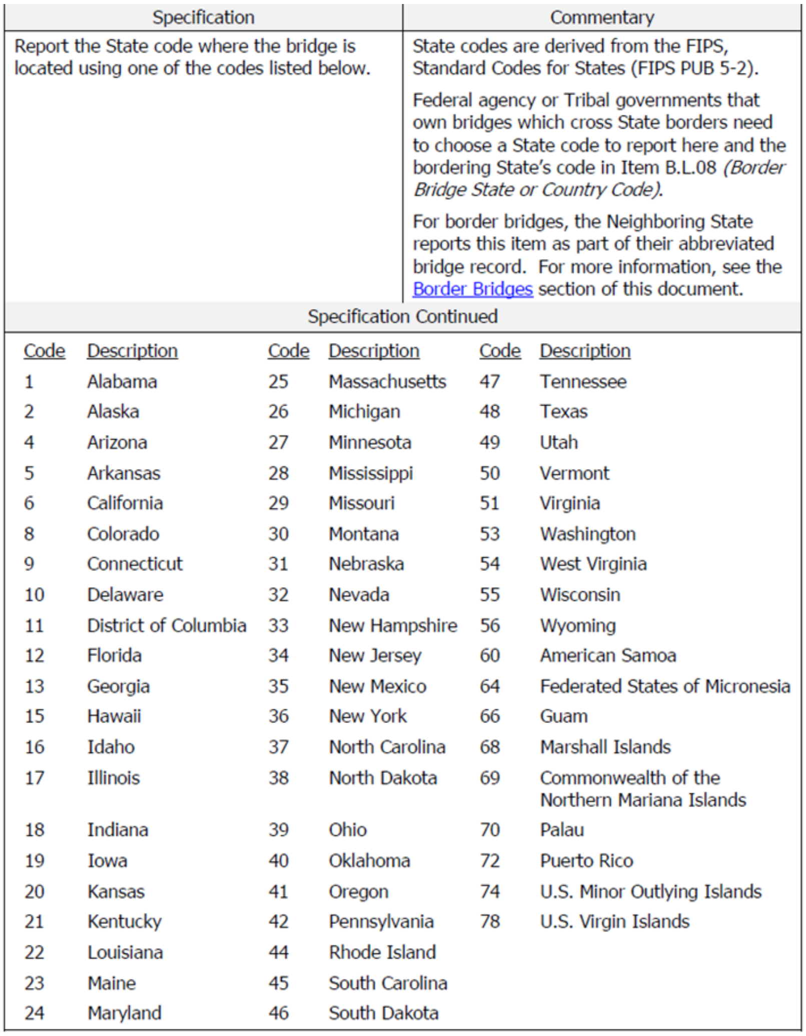
The pipeline was developed to extract primarily a set of selected bridge details (with the possibility of expansion). Extra details of interest can be easily added to the pipeline based on the specific features of the detail and the characteristics of the plan sets. The current pipeline can successfully capture a wide range of details using multiple sources of information. Following the NBIS, some of the main details are as follows:

1. B.ID.01: Bridge Number



Specification
Report the unique bridge number assigned according to agency policy for each bridge meeting the NBIS bridge definition that is fully or partially located within the State's boundaries, Federal agency's responsibility or jurisdiction, or Tribal government's responsibility or jurisdiction; regardless of inspection or financial responsibility. Do not change the bridge number once it has been assigned and recorded, except for a rare or unusual circumstance that requires a one-time change. When a bridge number is changed, report the previous bridge number under B.ID.03. Report all spans from abutment to abutment as one bridge.

2. B.L.01: State Code

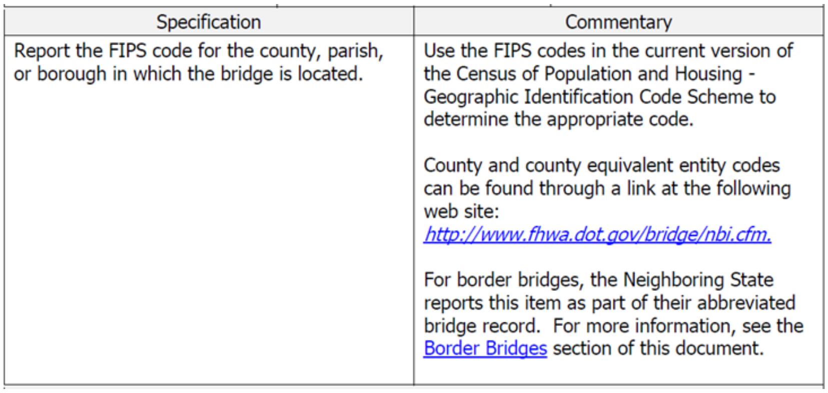


Specification	Commentary
Report the State code where the bridge is located using one of the codes listed below.	State codes are derived from the FIPS, Standard Codes for States (FIPS PUB 5-2). Federal agency or Tribal governments that own bridges which cross State borders need to choose a State code to report here and the bordering State's code in Item B.L.08 *(Border Bridge State or Country Code).* For border bridges, the Neighboring State reports this item as part of their abbreviated bridge record. For more information, see the Border Bridges section of this document.

Specification Continued

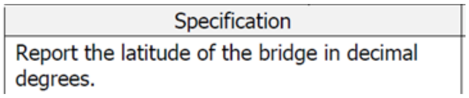
Code	Description	Code	Description	Code	Description
1	Alabama	25	Massachusetts	47	Tennessee
2	Alaska	26	Michigan	48	Texas
4	Arizona	27	Minnesota	49	Utah
5	Arkansas	28	Mississippi	50	Vermont
6	California	29	Missouri	51	Virginia
8	Colorado	30	Montana	53	Washington
9	Connecticut	31	Nebraska	54	West Virginia
10	Delaware	32	Nevada	55	Wisconsin
11	District of Columbia	33	New Hampshire	56	Wyoming
12	Florida	34	New Jersey	60	American Samoa
13	Georgia	35	New Mexico	64	Federated States of Micronesia
15	Hawaii	36	New York	66	Guam
16	Idaho	37	North Carolina	68	Marshall Islands
17	Illinois	38	North Dakota	69	Commonwealth of the Northern Mariana Islands
18	Indiana	39	Ohio	70	Palau
19	Iowa	40	Oklahoma	72	Puerto Rico
20	Kansas	41	Oregon	74	U.S. Minor Outlying Islands
21	Kentucky	42	Pennsylvania	78	U.S. Virgin Islands
22	Louisiana	44	Rhode Island		
23	Maine	45	South Carolina		
24	Maryland	46	South Dakota		

3. B.L.02: County Code



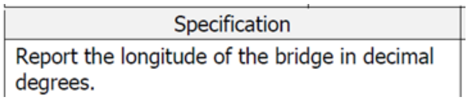
Specification	Commentary
Report the FIPS code for the county, parish, or borough in which the bridge is located.	Use the FIPS codes in the current version of the Census of Population and Housing - Geographic Identification Code Scheme to determine the appropriate code. County and county equivalent entity codes can be found through a link at the following web site: *http://www.fhwa.dot.gov/bridge/nbi.cfm.* For border bridges, the Neighboring State reports this item as part of their abbreviated bridge record. For more information, see the Border Bridges section of this document.

4. B.L.05: Latitude



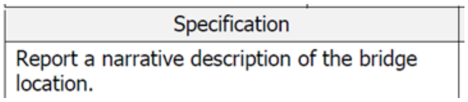
Specification
Report the latitude of the bridge in decimal degrees.

5. B.L.06: Longitude



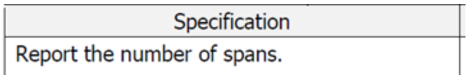
Specification
Report the longitude of the bridge in decimal degrees.

6. B.L.11: Bridge Location



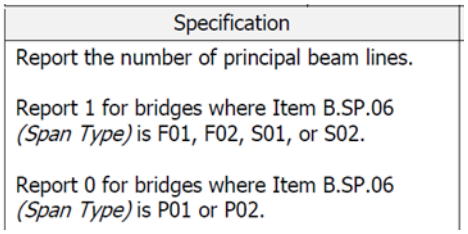
Specification
Report a narrative description of the bridge location.

7. B.SP.02: Number of Spans



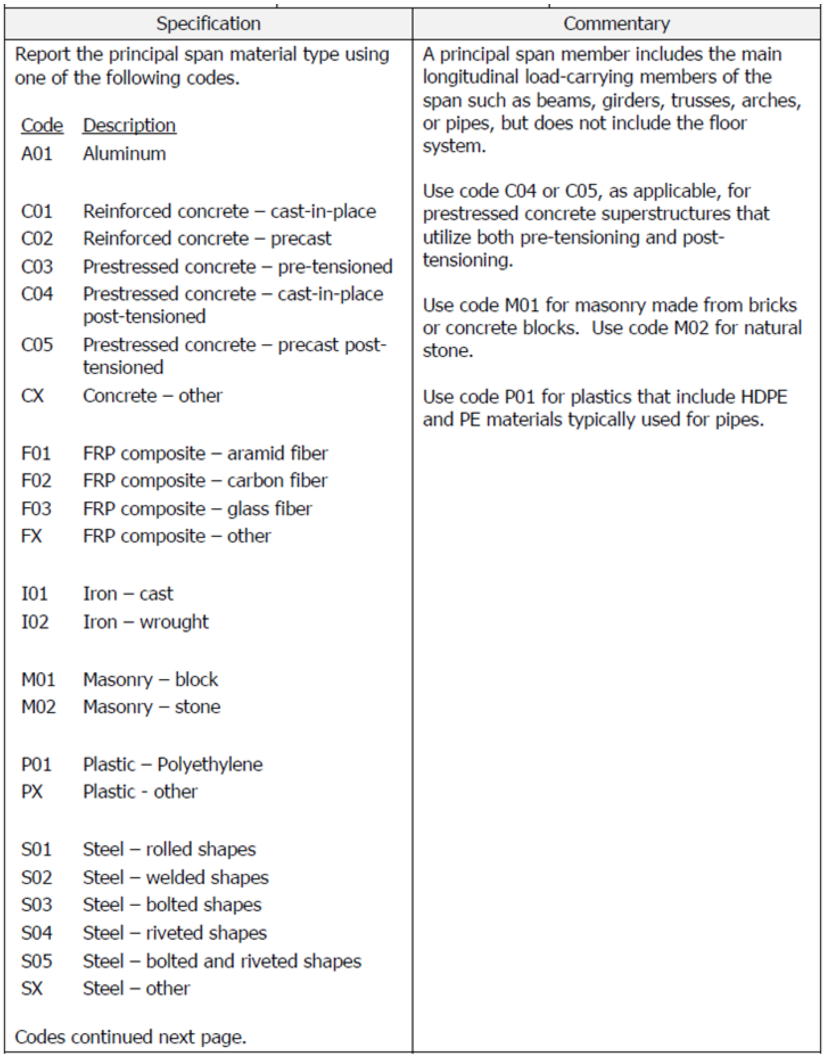
Specification
Report the number of spans.

8. B.SP.03: Number of Beamlines



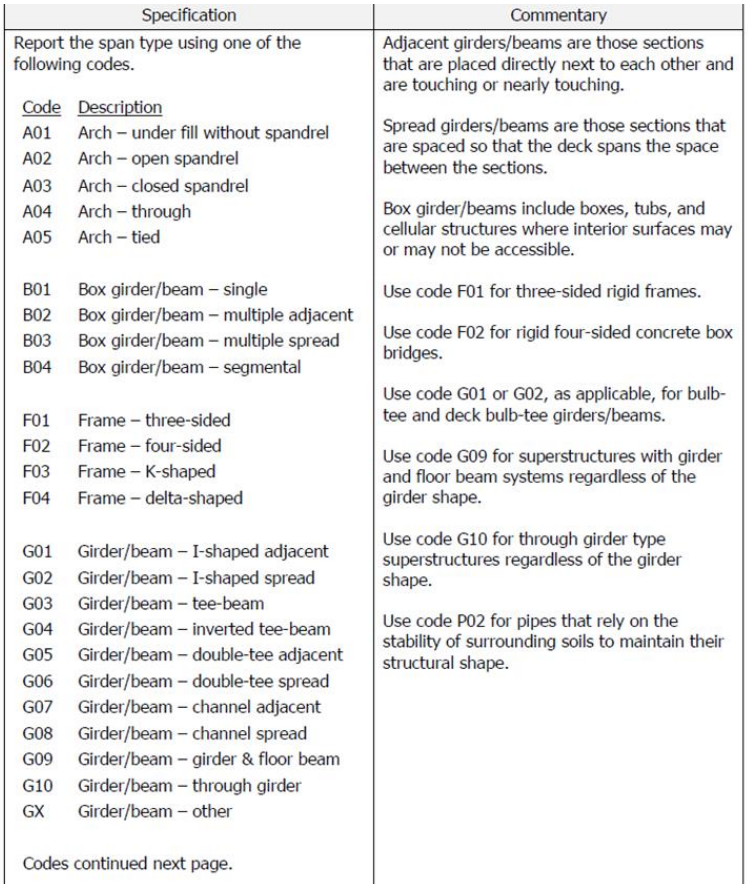
Specification
Report the number of principal beam lines. Report 1 for bridges where Item B.SP.06 *(Span Type)* is F01, F02, S01, or S02. Report 0 for bridges where Item B.SP.06 *(Span Type)* is P01 or P02.

9. B.SP.04: Span Material (Steel Girder, PPCB, and CCS only)

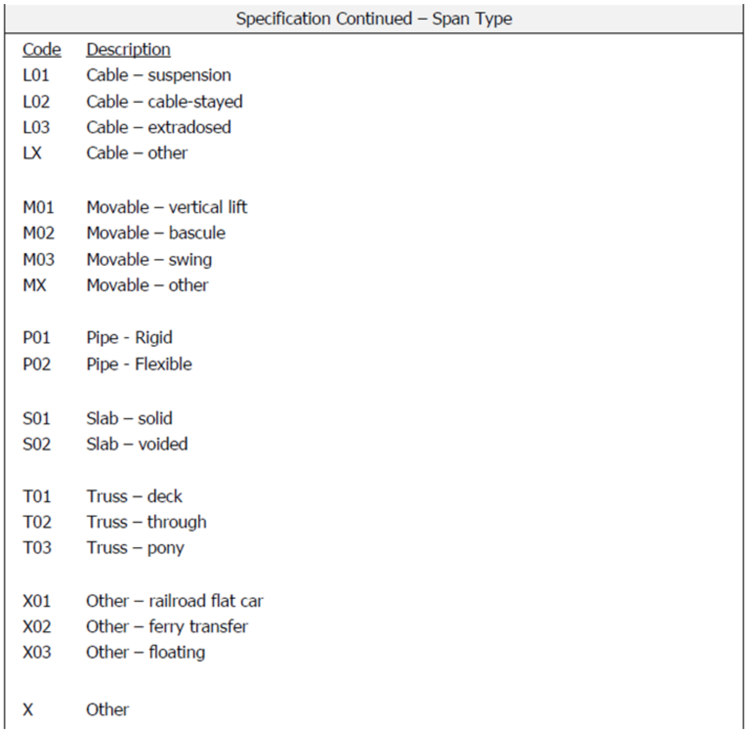


Specification	Commentary
Report the principal span material type using one of the following codes. Code Description A01 Aluminum C01 Reinforced concrete – cast-in-place C02 Reinforced concrete – precast C03 Prestressed concrete – pre-tensioned C04 Prestressed concrete – cast-in-place post-tensioned C05 Prestressed concrete – precast post-tensioned CX Concrete – other F01 FRP composite – aramid fiber F02 FRP composite – carbon fiber F03 FRP composite – glass fiber FX FRP composite – other I01 Iron – cast I02 Iron – wrought M01 Masonry – block M02 Masonry – stone P01 Plastic – Polyethylene PX Plastic - other S01 Steel – rolled shapes S02 Steel – welded shapes S03 Steel – bolted shapes S04 Steel – riveted shapes S05 Steel – bolted and riveted shapes SX Steel – other Codes continued next page.	A principal span member includes the main longitudinal load-carrying members of the span such as beams, girders, trusses, arches, or pipes, but does not include the floor system. Use code C04 or C05, as applicable, for prestressed concrete superstructures that utilize both pre-tensioning and post-tensioning. Use code M01 for masonry made from bricks or concrete blocks. Use code M02 for natural stone. Use code P01 for plastics that include HDPE and PE materials typically used for pipes.

10. B.SP.06: Span Type (Box Girder/ Beam, Girder/Beam, and Slab only)

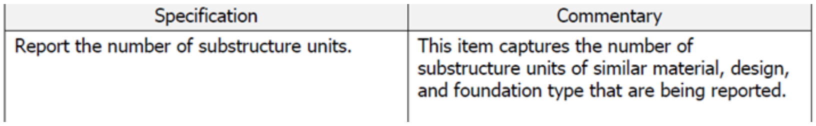


Specification	Commentary
Report the span type using one of the following codes. Code Description A01 Arch – under fill without spandrel A02 Arch – open spandrel A03 Arch – closed spandrel A04 Arch – through A05 Arch – tied B01 Box girder/beam – single B02 Box girder/beam – multiple adjacent B03 Box girder/beam – multiple spread B04 Box girder/beam – segmental F01 Frame – three-sided F02 Frame – four-sided F03 Frame – K-shaped F04 Frame – delta-shaped G01 Girder/beam – I-shaped adjacent G02 Girder/beam – I-shaped spread G03 Girder/beam – tee-beam G04 Girder/beam – inverted tee-beam G05 Girder/beam – double-tee adjacent G06 Girder/beam – double-tee spread G07 Girder/beam – channel adjacent G08 Girder/beam – channel spread G09 Girder/beam – girder & floor beam G10 Girder/beam – through girder GX Girder/beam – other Codes continued next page.	Adjacent girders/beams are those sections that are placed directly next to each other and are touching or nearly touching. Spread girders/beams are those sections that are spaced so that the deck spans the space between the sections. Box girder/beams include boxes, tubs, and cellular structures where interior surfaces may or may not be accessible. Use code F01 for three-sided rigid frames. Use code F02 for rigid four-sided concrete box bridges. Use code G01 or G02, as applicable, for bulb-tee and deck bulb-tee girders/beams. Use code G09 for superstructures with girder and floor beam systems regardless of the girder shape. Use code G10 for through girder type superstructures regardless of the girder shape. Use code P02 for pipes that rely on the stability of surrounding soils to maintain their structural shape.



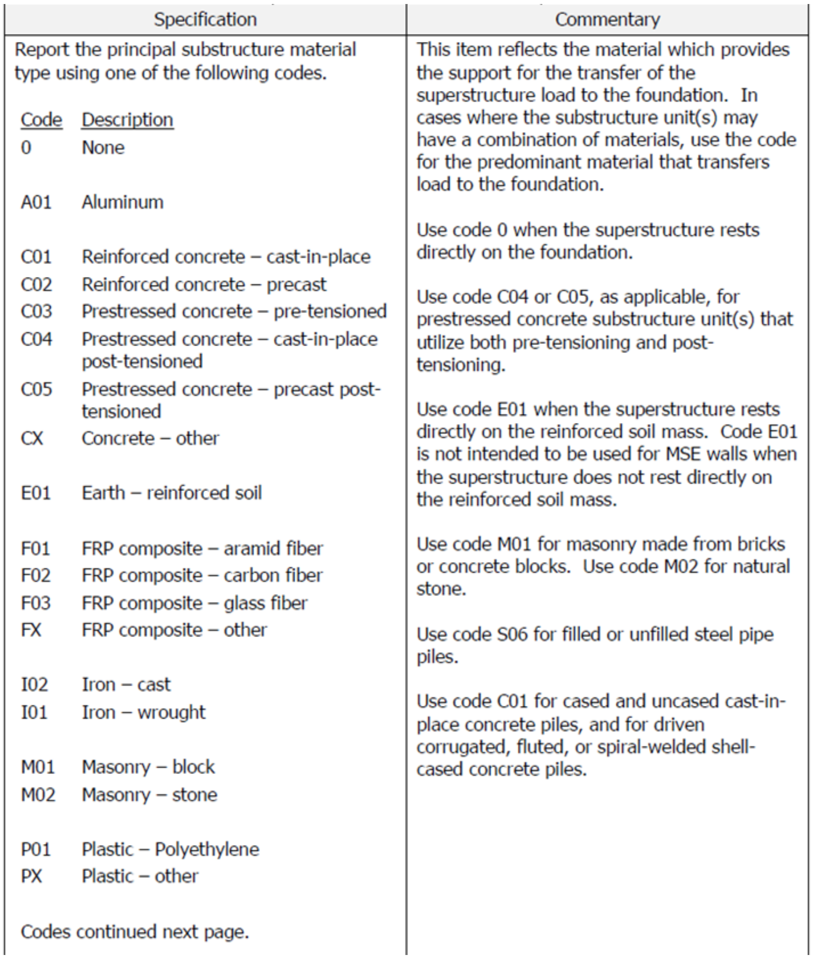
Specification Continued – Span Type	
Code	Description
L01	Cable – suspension
L02	Cable – cable-stayed
L03	Cable – extradosed
LX	Cable – other
M01	Movable – vertical lift
M02	Movable – bascule
M03	Movable – swing
MX	Movable – other
P01	Pipe - Rigid
P02	Pipe - Flexible
S01	Slab – solid
S02	Slab – voided
T01	Truss – deck
T02	Truss – through
T03	Truss – pony
X01	Other – railroad flat car
X02	Other – ferry transfer
X03	Other – floating
X	Other

11. B.SB.02: Number of Substructure Units

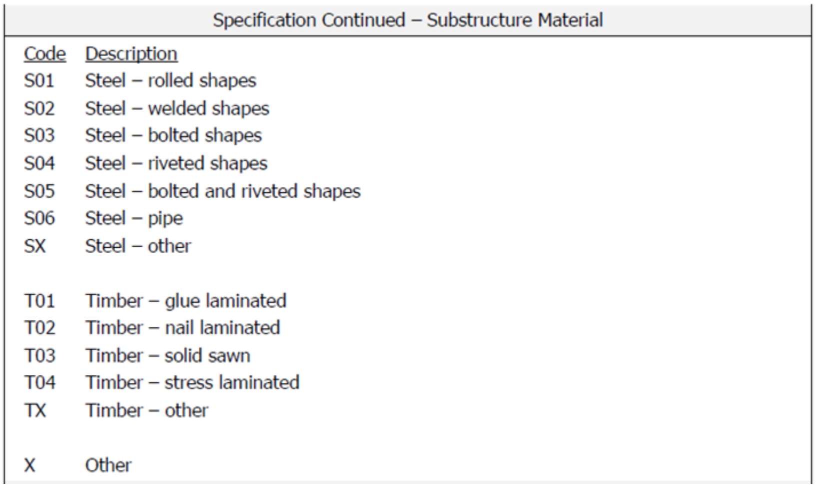


Specification	Commentary
Report the number of substructure units.	This item captures the number of substructure units of similar material, design, and foundation type that are being reported.

12. B.SB.03: Substructure Material (Steel and Concrete only)



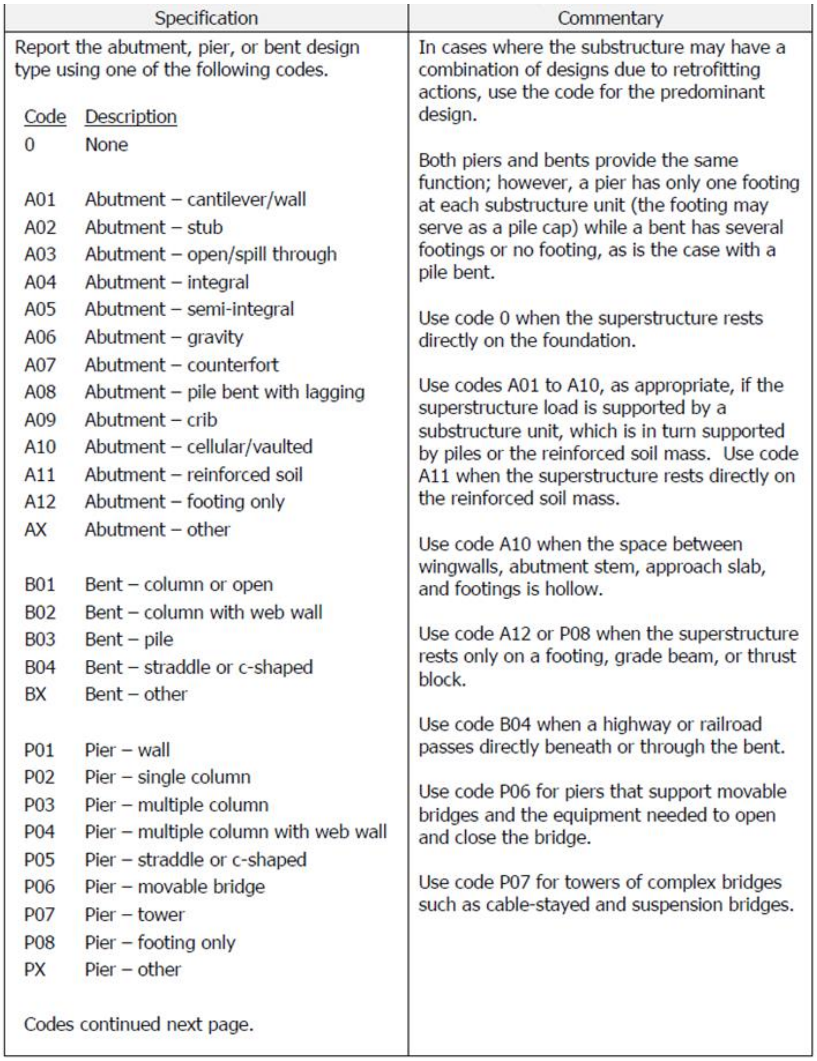
Specification	Commentary
Report the principal substructure material type using one of the following codes. Code Description 0 None A01 Aluminum C01 Reinforced concrete – cast-in-place C02 Reinforced concrete – precast C03 Prestressed concrete – pre-tensioned C04 Prestressed concrete – cast-in-place post-tensioned C05 Prestressed concrete – precast post-tensioned CX Concrete – other E01 Earth – reinforced soil F01 FRP composite – aramid fiber F02 FRP composite – carbon fiber F03 FRP composite – glass fiber FX FRP composite – other I02 Iron – cast I01 Iron – wrought M01 Masonry – block M02 Masonry – stone P01 Plastic – Polyethylene PX Plastic – other Codes continued next page.	This item reflects the material which provides the support for the transfer of the superstructure load to the foundation. In cases where the substructure unit(s) may have a combination of materials, use the code for the predominant material that transfers load to the foundation. Use code 0 when the superstructure rests directly on the foundation. Use code C04 or C05, as applicable, for prestressed concrete substructure unit(s) that utilize both pre-tensioning and post-tensioning. Use code E01 when the superstructure rests directly on the reinforced soil mass. Code E01 is not intended to be used for MSE walls when the superstructure does not rest directly on the reinforced soil mass. Use code M01 for masonry made from bricks or concrete blocks. Use code M02 for natural stone. Use code S06 for filled or unfilled steel pipe piles. Use code C01 for cased and uncased cast-in-place concrete piles, and for driven corrugated, fluted, or spiral-welded shell-cased concrete piles.



Specification Continued – Substructure Material

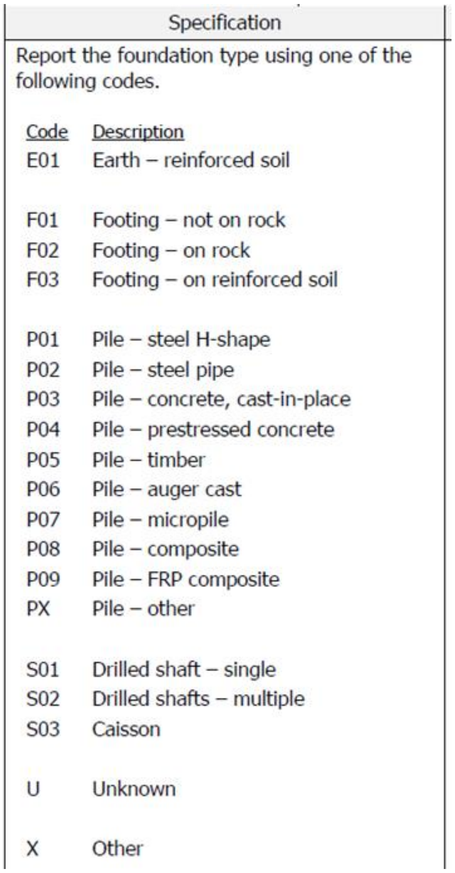
Code	Description
S01	Steel – rolled shapes
S02	Steel – welded shapes
S03	Steel – bolted shapes
S04	Steel – riveted shapes
S05	Steel – bolted and riveted shapes
S06	Steel – pipe
SX	Steel – other
T01	Timber – glue laminated
T02	Timber – nail laminated
T03	Timber – solid sawn
T04	Timber – stress laminated
TX	Timber – other
X	Other

13. B.SB.04: Substructure Type



Specification	Commentary
Report the abutment, pier, or bent design type using one of the following codes. Code Description 0 None A01 Abutment – cantilever/wall A02 Abutment – stub A03 Abutment – open/spill through A04 Abutment – integral A05 Abutment – semi-integral A06 Abutment – gravity A07 Abutment – counterfort A08 Abutment – pile bent with lagging A09 Abutment – crib A10 Abutment – cellular/vaulted A11 Abutment – reinforced soil A12 Abutment – footing only AX Abutment – other B01 Bent – column or open B02 Bent – column with web wall B03 Bent – pile B04 Bent – straddle or c-shaped BX Bent – other P01 Pier – wall P02 Pier – single column P03 Pier – multiple column P04 Pier – multiple column with web wall P05 Pier – straddle or c-shaped P06 Pier – movable bridge P07 Pier – tower P08 Pier – footing only PX Pier – other Codes continued next page.	In cases where the substructure may have a combination of designs due to retrofitting actions, use the code for the predominant design. Both piers and bents provide the same function; however, a pier has only one footing at each substructure unit (the footing may serve as a pile cap) while a bent has several footings or no footing, as is the case with a pile bent. Use code 0 when the superstructure rests directly on the foundation. Use codes A01 to A10, as appropriate, if the superstructure load is supported by a substructure unit, which is in turn supported by piles or the reinforced soil mass. Use code A11 when the superstructure rests directly on the reinforced soil mass. Use code A10 when the space between wingwalls, abutment stem, approach slab, and footings is hollow. Use code A12 or P08 when the superstructure rests only on a footing, grade beam, or thrust block. Use code B04 when a highway or railroad passes directly beneath or through the bent. Use code P06 for piers that support movable bridges and the equipment needed to open and close the bridge. Use code P07 for towers of complex bridges such as cable-stayed and suspension bridges.

14. B.SB.06: Foundation Type

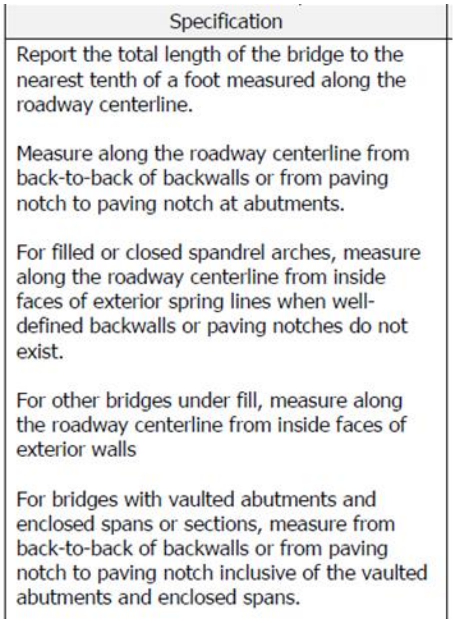


Specification

Report the foundation type using one of the following codes.

Code	Description
E01	Earth – reinforced soil
F01	Footing – not on rock
F02	Footing – on rock
F03	Footing – on reinforced soil
P01	Pile – steel H-shape
P02	Pile – steel pipe
P03	Pile – concrete, cast-in-place
P04	Pile – prestressed concrete
P05	Pile – timber
P06	Pile – auger cast
P07	Pile – micropile
P08	Pile – composite
P09	Pile – FRP composite
PX	Pile – other
S01	Drilled shaft – single
S02	Drilled shafts – multiple
S03	Caisson
U	Unknown
X	Other

15. B.G.02: Total Bridge Length



Specification

Report the total length of the bridge to the nearest tenth of a foot measured along the roadway centerline.

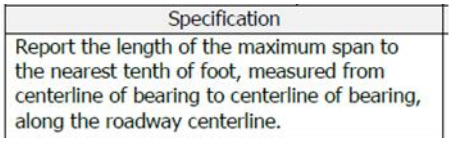
Measure along the roadway centerline from back-to-back of backwalls or from paving notch to paving notch at abutments.

For filled or closed spandrel arches, measure along the roadway centerline from inside faces of exterior spring lines when well-defined backwalls or paving notches do not exist.

For other bridges under fill, measure along the roadway centerline from inside faces of exterior walls

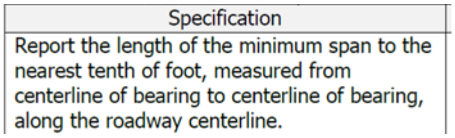
For bridges with vaulted abutments and enclosed spans or sections, measure from back-to-back of backwalls or from paving notch to paving notch inclusive of the vaulted abutments and enclosed spans.

16. B.G.03: Maximum Span Length



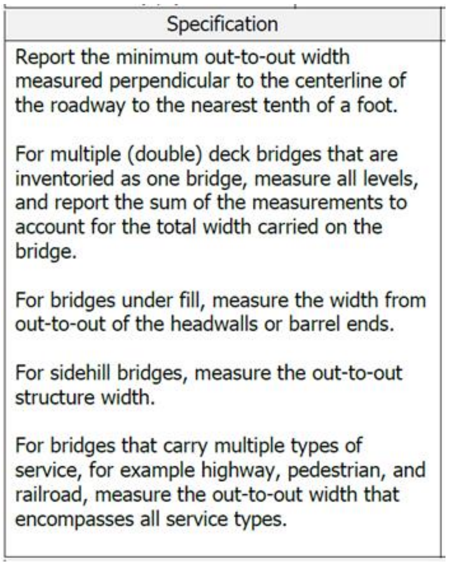
Specification
Report the length of the maximum span to the nearest tenth of foot, measured from centerline of bearing to centerline of bearing, along the roadway centerline.

17. B.G.04: Minimum Span Length



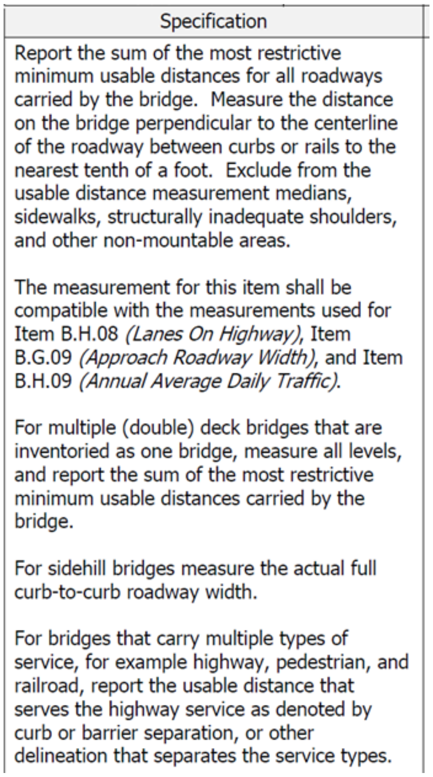
Specification
Report the length of the minimum span to the nearest tenth of foot, measured from centerline of bearing to centerline of bearing, along the roadway centerline.

18. B.G.05: Bridge Width Out-to-Out



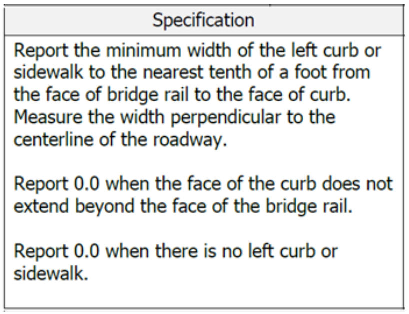
Specification
Report the minimum out-to-out width measured perpendicular to the centerline of the roadway to the nearest tenth of a foot. For multiple (double) deck bridges that are inventoried as one bridge, measure all levels, and report the sum of the measurements to account for the total width carried on the bridge. For bridges under fill, measure the width from out-to-out of the headwalls or barrel ends. For sidehill bridges, measure the out-to-out structure width. For bridges that carry multiple types of service, for example highway, pedestrian, and railroad, measure the out-to-out width that encompasses all service types.

19. B.G.06: Bridge Width Curb-to-Curb



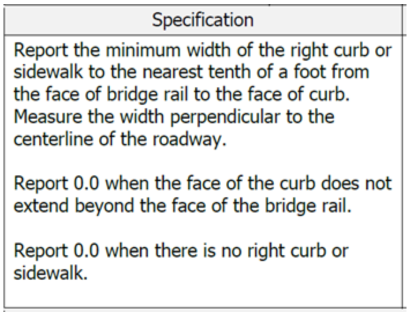
Specification
Report the sum of the most restrictive minimum usable distances for all roadways carried by the bridge. Measure the distance on the bridge perpendicular to the centerline of the roadway between curbs or rails to the nearest tenth of a foot. Exclude from the usable distance measurement medians, sidewalks, structurally inadequate shoulders, and other non-mountable areas. The measurement for this item shall be compatible with the measurements used for Item B.H.08 *(Lanes On Highway)*, Item B.G.09 *(Approach Roadway Width)*, and Item B.H.09 *(Annual Average Daily Traffic)*. For multiple (double) deck bridges that are inventoried as one bridge, measure all levels, and report the sum of the most restrictive minimum usable distances carried by the bridge. For sidehill bridges measure the actual full curb-to-curb roadway width. For bridges that carry multiple types of service, for example highway, pedestrian, and railroad, report the usable distance that serves the highway service as denoted by curb or barrier separation, or other delineation that separates the service types.

20. B.G.07: Left Curb or Sidewalk Width



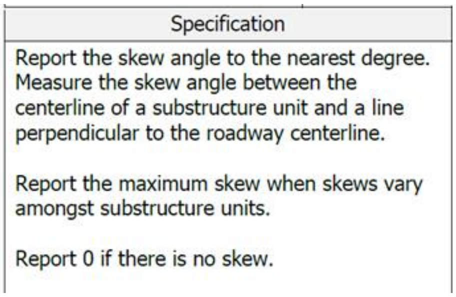
Specification
Report the minimum width of the left curb or sidewalk to the nearest tenth of a foot from the face of bridge rail to the face of curb. Measure the width perpendicular to the centerline of the roadway. Report 0.0 when the face of the curb does not extend beyond the face of the bridge rail. Report 0.0 when there is no left curb or sidewalk.

21. B.G.08: Right Curb or Sidewalk Width



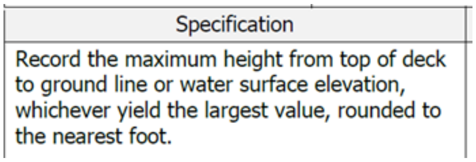
Specification
Report the minimum width of the right curb or sidewalk to the nearest tenth of a foot from the face of bridge rail to the face of curb. Measure the width perpendicular to the centerline of the roadway. Report 0.0 when the face of the curb does not extend beyond the face of the bridge rail. Report 0.0 when there is no right curb or sidewalk.

22. B.G.11: Skew



Specification
Report the skew angle to the nearest degree. Measure the skew angle between the centerline of a substructure unit and a line perpendicular to the roadway centerline. Report the maximum skew when skews vary amongst substructure units. Report 0 if there is no skew.

23. B.G.13: Maximum Bridge Height



Specification
Record the maximum height from top of deck to ground line or water surface elevation, whichever yield the largest value, rounded to the nearest foot.

24. Reinforcement Properties: Number, Length, and Diameter of the Bars

4.2. ADDITIONAL DETAILS OF INTEREST

A set of additional details are also provided through the platform to show the capabilities of the pipeline to handle data and feature extraction from different plan sets. The additional details are as follows:

1. Bridge Name
2. Number of Piles
3. Column Section Sizes
4. Deck Thickness
5. Column Height
6. Number of Columns in the Bent
7. Column Reinforcement
8. Pile Data Table

The previous details are extracted based on the specific details and drawing methods of specific plan sets and could be adjusted to fit the requirements of various transportation agencies. As the main functions and models of the pipelines have been developed, some extra properties could be potentially extracted from the bridge plan sets if adequate data are available. The potential properties are as follows:

1. The main dimensions of individual bridge components
2. Whether the bridge is a curved structure or not
3. Bridge median
4. Approach roadway width
5. Bridge railing properties
6. Bridge material quantities

4.3. QUALITY ASSURANCE/QUALITY CONTROL CHECKS

The pipeline was tested on several bridge plan sets to ensure the accuracy of the feature extraction process, providing a robust platform. As such, the results of extracting some details and features from Iowa DOT bridge plan sets are reported in Table 4.1. The table shows the extracted features for different bridges with various structural properties including different deck slabs, abutments, skew angles, and dimensions. Accordingly, the pipeline has been successful in detecting and extracting most of the desired details accurately.

Bridge number 44721 includes a slab with varying girder distances along the bridge length. Hence, the pipeline was not able to extract the bridge deck properties across the bridge width. As for bridge number 701060, the text recognition process occasionally fails to extract vertical characters such as number “1,” reporting variations of “I” or “!” instead. Some errors could arise due to the few false recognitions of “1,” which also resulted in reporting 46 instead of 146 as the minimum span length of the mentioned bridge. This can be easily detected and fixed by having a second check point to match the summation of the span lengths with the total bridge length. The table also indicates that the maximum bridge height was prone to no output. This is because the bridge height is currently extracted as the difference between the top of the deck and any bed elevations that the pipeline identifies. If the bed elevation indicators, i.e., “streambed elevation” or “top of rail elevation,” are not mentioned in the plan sheet, the pipeline cannot find the bed elevation. This can be resolved by opting for other bed elevation indicators.

Table 4.1. Data extraction examples for 10 bridges in Iowa

Bridge Number	14251	46071	12971	28431	018891	44721	701105	700535	701060	51411
State Code	19	19	19	19	19	19	19	19	19	19
County Code	11	157	1	87	33	155	71	113	113	181
Latitude	41.963669	41.695614	41.429913	40.975053	42,984,666	41,242,322	40.682081	41.9072136	42.058964	41.432421
Longitude	-92.149071	-92.720993	-94.452471	-91.677947	-93.20184	-95.904281	-95.82029	-91.3760785	-91.699906	-93.780783
Number of Substructure Units	2	2	2	3	2	1	5	1	1	1
Number of Spans	3	3	3	4	3	2	6	2	2	2
Number of Beamlines	6	8	1	5	1		1	4	8	4
Span Material	PPCB	PPCB	CCS	PPCB	CCS	Steel Girder	CCS	Steel Girder	PPCB	PPCB
Span Type	Girder/Beam	Girder/Beam	Slab	Girder/Beam	Slab	Girder/Beam	Slab	Girder/Beam	Girder/Beam	Girder/Beam
Pier Material	Concrete	Concrete	Steel	Concrete	Steel	Concrete	Concrete	Concrete	Concrete	Concrete
Abutment Material	Steel	Steel	Steel	Steel	Steel	Steel	Steel	Steel	Steel	Steel
Pier Type	Tee Pier	Frame Pier	Pile Bent Pier	Tee Pier	Pile Bent Pier	Diaphragm Pier	Encased Pile Bent Pier	Frame Pier	Wall Pier	Frame Pier
Abutment Type	Integral Abutment	Integral Abutment	Integral Abutment	Stub Abutment	Integral Abutment	Semi-Integral Abutment	Integral Abutment	Integral Abutment	Integral Abutment	Integral Abutment
Pier Foundation Type	HP10x57	HP10x57	HP14x73	Spread Footing	HP 14x73	HP14x89	HP10x57	HP10x57	HP 12x84	HP 14x73
Skew	30	5	15	20	0	5	0	15	3°43'59.40"	0
Total Bridge Length	246-5	222-0	150-10	561-4	150-10	311-08	327-10	283-6	295-01	295-0
Maximum Span Length	81.5	117	59	142	59	174	59	160	146	146
Minimum Span Length	80.75	41	45.5	131	45.5	134	45.5	120	46	146
Bridge Width Curb-to-Curb	40	66	40	40	44	VARIES	40	30	49-9	30
Curb width	1-7	1-7	1-4 , 1-7	1-7	1-4 , 1-7		1-7	1-7	1-2	1-7
Bridge Width Out-to-Out	43.17	69.17	42.92	43.17	46.92	VARIES	43.17	33.17	51.33	33.17
Maximum Bridge Height	21.93	29.97	24.2	51.22	22.46		16.86	38.43		36.28

Note: The highlighted rows show the few instances that do not match those in the plan sets. The rest of the values are completely consistent with those presented in the source plan sets.

5. CONCLUSIONS

The significant impacts of applying ML/AI algorithms to optimize various aspects of bridge engineering highlights the critical need to automate data and feature extraction from bridge plan sets. This research project addressed the required steps to automate the extraction and reporting of engineering details of interest from bridge plans, particularly to comply with NBIS reporting requirements. While the automated extraction of details from engineering drawings can be a complicated task for machines due to the complex nature of plan sets, a combination of several deep learning models and image processing techniques provided a novel platform to successfully extract target details. Thus, the project's outcome directly contributes to improving both the accuracy and efficiency of data and feature extraction from bridge plan sets commonly used by transportation agencies.

Deep learning models were employed as powerful object detectors to construct a visual understanding of the plan set for the machine, providing an approach that was able to navigate through different plan sheets. The deep learning models covered objects of various scales, ranging from objects as large as the general plan of the bridge to objects as small as stirrups. The models were trained on large datasets and enhanced through rigorous testing and validation. Specifically, the base model was trained using nine distinct classes of bridge parameters with the purpose of main object localization. During the pipeline compilation stage, the need for an expanded model emerged to find additional details efficiently. Using a similar approach to that used to develop the base model, the expanded model was developed on refined classes defined for various queries, such as the number of spans or the number of beamlines. All of the developed models provided satisfactory performance metrics over the training period, demonstrating the promise of the pipeline to be used for new plan sets not used for training purposes.

Various processing techniques were developed and used in several parts of the pipeline, especially for table and text recognition. Using a hybrid of detection processes, text recognition methods, and data frame packages, the plan set details were successfully converted to spreadsheets for further analysis. Combining all of the outlined models and techniques, a pipeline was developed to extract the desired engineering details from bridge plan sets. With general models and functions developed, the pipeline is customizable to match the approaches that different State transportation agencies use to provide details in their bridge plan sets.

The project provided example results of customization of the developed pipeline to deliver the characteristics of bridge plan sets from the Iowa DOT and Caltrans. The desired details ranged from general introductory information to bridge properties and main dimensions. Details were compared side by side for several bridge plan sets with distinct properties such as span type, abutment type, and foundation type. The pipeline was able to successfully extract the majority of the queries, proving the feasibility of preparing a software prototype to automate data and feature extraction from bridge plan sets. This software prototype is expected to offer a powerful tool that can be implemented by various transportation agencies to improve the accuracy and speed of the process to obtain bridge details of interest.

INVESTIGATOR'S PROFILE

Behrouz Shafei, Ph.D., P.E. Associate Professor, Institute for Transportation, Department of Civil, Construction, and Environmental Engineering, Iowa State University, Ames, IA 50011. Phone: (515) 294-4058; E-mail: shafei@iastate.edu.

Dr. Behrouz Shafei contributes to various research and educational activities related to bridge projects. He has extensive experience in the design and assessment of bridge structures subjected to natural and manmade hazards, in addition to the development of condition-based strategies for the inspection and maintenance of aging structural components. His research efforts primarily aim to address the issues associated with the design and construction of bridges. He has completed multiple research projects sponsored by various State and National agencies. His contribution to the field has been recognized through the James D. Cooper Award from the International Bridge Conference (2011), Young Engineer Award from the Orange County Engineering Council (2012), the Wiley Award for Innovation in Computing (2014), and the Charles W. Schafer Award for Excellence in Teaching, Research, and Service (2020). He teaches bridge design and serves as a research advisor for undergraduate and graduate students at Iowa State University. The Iowa State University research team for this project included Shadi Azad, a graduate research assistant, and Dr. Alireza Razavi, a professional and scientific staff member.

REFERENCES

Cui, Z., and H. Zhang. 2013. A study on contour extraction method in computer vision measurement technology. *Computer Modeling and New Technologies*, 17(5B), pp. 88–91.

Datta, R., P. De, S. Mandal, and B. Chanda. 2015. Detection and identification of logic gates from document images using mathematical morphology. *Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*. IEEE, Piscataway, NJ.

De, P., S. Mandal, and P. Bhowmick. 2011. Recognition of electrical symbols in document images using morphology and geometric analysis. *International Conference on Image Information Processing*. IEEE, Piscataway, NJ.

Diao, W., X. Sun, X. Zheng, F. Dou, H. Wang, and K. Fu. 2016. Efficient saliency-based object detection in remote sensing images using deep belief networks. *IEEE Geoscience and Remote Sensing Letters*, Vol. 13, No. 2, pp. 137–141.

Dori, D., and Y. Velkovitch. 1998. Segmentation and recognition of dimensioning text from engineering drawings. *Computer Vision and Image Understanding*, Vol. 69, No. 2, pp. 196–201.

Hashemi, N. S., R. B. Aghdam, A. S. B. Ghiasi, and P. Fatemi. 2016. Template matching advances and applications in image analysis. *arXiv*: 1610.07231.

Jamieson, L., C. F. Moreno-Garcia, and E. Elyan. 2020. Deep learning for text detection and recognition in complex engineering diagrams. *International Joint Conference on Neural Networks (IJCNN)*. IEEE, Piscataway, NJ.

Kim, H., S. Kim, and K. Yu. 2021. Automatic extraction of indoor spatial information from floor plan image: A patch-based deep learning methodology application on large-scale complex buildings. *ISPRS International Journal of Geo-Information*, Vol. 10, No. 12, Article ID 828.

Moon, Y., J. Lee, D. Mun, and S. Lim. 2021. Deep learning-based method to recognize line objects and flow arrows from image-format piping and instrumentation diagrams for digitization. *Applied Sciences*, Vol. 11, No. 21, Article ID 10054.

Ondrejcek, M., J. Kastner, R. Kooper, and P. Bajcsy. 2009. *Information Extraction from Scanned Engineering Drawings*. Report No: NCSA-ISDA09-001. National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Image Spatial Data Analysis Group, Urbana, IL.

Ravagli, J., Z. Ziran, and S. Marinai. 2019. Text recognition and classification in floor plan images. *International Conference on Document Analysis and Recognition Workshops (ICDARW)*, Vol. 1. IEEE, Piscataway, NJ.

Scheibel, B., J. Mangler, and S. Rinderle-Ma. 2021. Extraction of dimension requirements from engineering drawings for supporting quality control in production processes. *Computers in Industry*, Vol. 129, Article ID 103442.

Shah, B. K., V. Kedia, R. Raut, S. Ansari, and A. Shroff. 2020. Evaluation and comparative study of edge detection techniques. *IOSR Journal of Computer Engineering*, Vol. 22, No. 5, pp. 6–15.

Sinha, R. K., R. Pandey, and R. Pattnaik. 2017. Deep learning for computer vision tasks: A review. *International Conference on Intelligent Computing and Control (I2C2)*, Coimbatore, India.

FHWA. 2022. *Specifications for the National Bridge Inventory*. FHWA-HIF-22-017. Federal Highway Administration, Office of Bridges and Structures, Washington, DC.

Van Daele, D., N. Decleyre, H. Dubois, and W. Meert. 2021. An automated engineering assistant: Learning parsers for technical drawings. *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35, No. 17, pp. 15195–15203.

Voulodimos A, N. Doulamis, A. Doulamis, and E. Protopapadakis. 2018. Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, Vol. 2018, Article ID 7068349.

Xie, L., Y. Lu, T. Furuhata, S. Yamakawa, W. Zhang, A. Regmi, L. Kara, and K. Shimada. 2022. Graph neural network-enabled manufacturing method classification from engineering drawings. *Computers in Industry*, Vol. 142, Article ID 103697.

Xu, X. W., and Y. Wu. 2003. Dimension set recognition methodologies. *Conference Proceedings: ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, New York, NY.

APPENDIX

TITLE

Data and Feature Extraction from Bridge Plans

SUBHEAD

Development of a novel computational platform that automates the process of extracting and reporting information from bridge plan sets.

WHAT WAS THE NEED?

Governmental agencies with bridges in their jurisdictions are required to report bridge inventory information to the Federal Highway Administration (FHWA) in compliance with National Bridge Inspection Standards (NBIS) reporting requirements. The reported information contributes to the National Bridge Inventory (NBI) database, which enables State- and National-level analyses and supports Federal funding programs. However, the sheer magnitude of bridge details and the fact that many agencies use their own labels and codes for many data items make inventory analysis and subsequent reporting difficult.

The emergence of machine learning (ML) and artificial intelligence (AI) has led to widespread interest within the bridge engineering community to use automated techniques to tap into extensive databases, such as the NBI, for improved bridge maintenance and management.

WHAT WAS OUR GOAL?

The goal of this project was to develop a computational platform that automates the process of reviewing bridge plans to identify, extract, and report select engineering details while increasing the accuracy and efficiency of feature extraction from bridge plan sets.

WHAT DID WE DO?

Deep learning models were employed as powerful object detectors to construct a visual understanding of the plan set for the machine, providing an approach that was able to navigate through different plan sheets. The deep learning models covered objects of various scales, ranging from objects as large as the general plan of the bridge to objects as small as stirrups. The models were trained on large datasets and enhanced through rigorous testing and validation. Specifically, the base model was trained using nine distinct classes of bridge parameters with the purpose of main object localization. During the pipeline compilation stage, the need for an expanded model emerged to find additional details efficiently. Using a similar approach to that used to develop the base model, the expanded model was developed on refined classes defined for various queries, such as the number of spans or the number of beamlines. All of the developed models provided satisfactory performance metrics over the training period, demonstrating the promise of the pipeline to be used for new plan sets not used for training purposes. The contribution and involvement of Iowa, Minnesota, and California Departments of Transportation are gratefully acknowledged.

WHAT WAS THE OUTCOME?

The significant impacts of applying ML/AI algorithms to optimize various aspects of bridge engineering highlights the critical need to automate data and feature extraction from bridge plan sets. This research project addressed the required steps to automate the extraction and reporting of engineering details of interest from bridge plans, particularly to comply with NBIS reporting requirements. While the automated extraction of details from engineering drawings can be a complicated task for machines due to the complex nature of plan sets, a combination of several deep learning models and image processing techniques provided a novel platform to successfully extract target details. Thus, the project's outcome directly contributes to improving both accuracy and efficiency of data and feature extraction from bridge plan sets commonly used by transportation agencies.

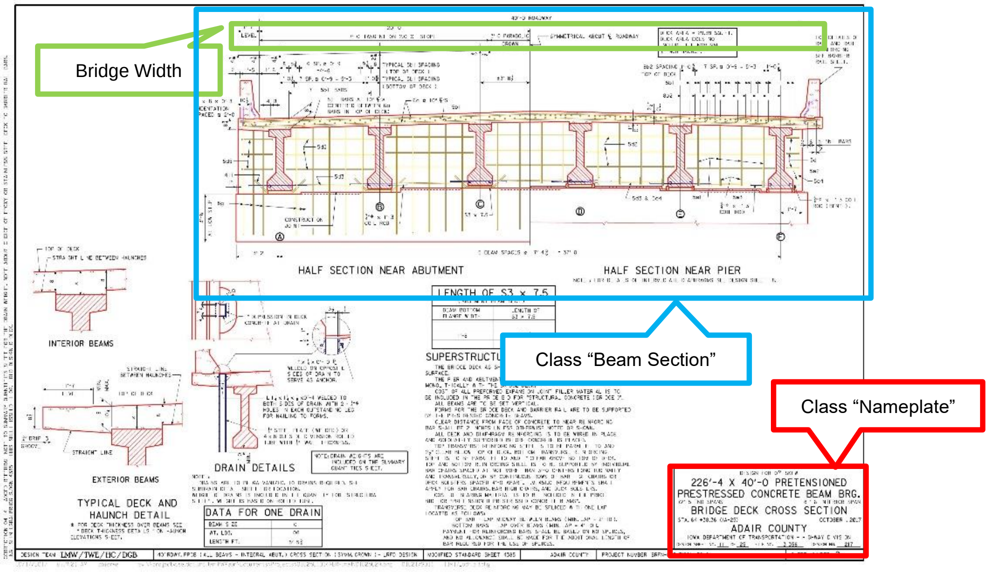
WHAT IS THE BENEFIT?

The benefit will be a drastic reduction in the time and effort that State highway agencies spend on reviewing, finding, extracting, and reporting bridge details. Through the developed platform, the entire process for data and feature extraction will be shortened significantly. Given that knowledge of bridge details is essential for planning purposes, the time gained at the beginning will expedite and improve various bridge management activities. Further to the identified benefits, the developed automated platform will free up the time of engineers and staff members of State highway agencies, giving them the opportunity to further focus on the tasks that require their expertise. In addition, the information generated through this automated process is expected to be of higher quality than information obtained manually, as many common human errors are eliminated.

LEARN MORE

A link to the final report for this project may be found on the TRB's NCHRP IDEA website. The principal investigator, Behrouz Shafei (shafei@iastate.edu) may also be contacted for additional information.

IMAGES



Detection of bridge components and their main details

SIDEBAR INFO

Program Steering Committee: NCHRP IDEA Program Committee
Month and Year: February 2024
Title: Data and Feature Extraction from Bridge Plans
Project Number: NCHRP IDEA 20-30/IDEA 230
Start Date: 07/01/2021
Completion Date: 03/31/2024
Product Category: Bridge Structures
Principal Investigator: Behrouz Shafei, Iowa State University
E-Mail: shafei@iastate.edu
Phone: 515-294-4058