AI-driven extraction of electrical circuits from floorplans for BIM

Urbieta, Martin[a,b], Urbieta, Matias[a,b,*], Burriel Guillermo[a]

[a] *Centro de investigación LIFIA, Facultad de Informática, Universidad Nacional de La Plata, Calle 50 y 120, La Plata, 1900, Buenos Aires, Argentina*
[b] *Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Godoy Cruz 2290, Ciudad Autonoma de Buenos Aires, C1425FQB, Buenos Aires, Argentina*

Abstract

BIM solutions require a digital model as a foundation to optimize processes such as maintenance, infrastructure renovation, or demolition. However, a vast number of analog building plans are archived by public entities managing urban development, and manually converting these plans into digital models, which is prohibitively expensive. To address this gap, the paper introduces an approach for organizations who need to convert large datasets of legacy electrical floorplans into a BIM. The approach leverages a Machine Learning model for instance segmentation to detect electrical features, and the line-segment detection model DeepLSD for extracting cable traces. To support model training, a new dataset, referred as IPVBA-ELEC, is provided. The approach assembles circuits by establishing semantic relationships between circuit components and wires, and store them in an IFC file. Case studies were evaluated using quantitative and qualitative techniques yielding promising results and encouraging further research of additional MEP domains.

Keywords:
BIM, machine-learning, automated detection, floor plans, model generation, electrical installations, IFC, raster drawings, and electrical circuits.

⋆Acknowledgments - The research work was supported by the Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)

*Corresponding author
Email address: matias.urbieta@lifia.info.unlp.edu.ar (Urbieta, Matias)

1. Introduction

Building Information Modeling (BIM) is an emerging methodology in the Architecture, Engineering, and Construction (AEC) industry for the creation and management of digital information (i.e. digital twins) for the built asset throughout its life cycle, from planning and design to construction and operations. It represents an opportunity to improve well-being and energy efficiency, flexibility, and resilience of existing buildings. The adoption of BIM is presented as a paradigm shift for the industry, profoundly transforming traditional processes and roles in the AEC industry [1]. The Industry Foundation Classes (IFC) are an open and international standard [2] for describing and exchanging building and infrastructure data. It is widely adopted in BIM processes, ensuring seamless data exchange. IFC can also be used as a means of archiving project information, as an "as-built" collection of information for long-term preservation[1]. Implementing BIM in AEC offers significant advantages by addressing persistent inefficiencies related to the loss of information, coordination, and decision-making throughout the different stages of the project. Eastman et al. [3] highlight the widespread problem of information loss during the transition between stakeholders and stages of traditional design and construction workflows. BIM mitigates this problem by enabling a shared digital representation of a facility, thereby ensuring the maintenance, reuse, and enrichment of critical data throughout the project lifecycle. This efficiency gain is reinforced by the MacLeamy Curve [4], which illustrates that owners who promote full collaboration and early information sharing, concentrating design effort in the early stages when changes are most cost-effective and impactful, make decisions as early as possible, improving the ability to implement changes efficiently and minimizing the cost of rework. BIM enables this collaborative environment and facilitates early-stage decision-making by allowing for higher levels of design detail, simulation, and stakeholder collaboration during the conceptual phase. While BIM offers significant advantages for digital design, construction, and operation, many buildings, particularly those constructed before the adoption of this methodology, still lack BIM models. There is a technical need for BIM in energy retrofit projects during the pre-energy modeling stage [5]. If as-built BIM models of existing buildings are available, they can be effectively integrated into a workflow for building modernization [6]. Approaches that

[1] `https://technical.buildingsmart.org/standards/ifc/`

reduce implementation costs through automation will further enhance the benefits of the BIM methodology.

The ability to obtain an automatic BIM model for floor plans (Fast BIM), including not only architecture and structure but also Mechanical, Electrical, and Plumbing (MEP) systems, enables a comprehensive and multidisciplinary assessment of the building. This broader scope supports collaborative early-stage decisions, especially in retrofit planning or post-disaster recovery, where a richer model leads to more accurate diagnostics. Rather than reducing downstream changes, the value lies in better-informed interventions from the outset.

Machine Learning (ML) techniques have shown considerable potential for extracting information from existing building plans to generate fully or semi-automated BIM models [7, 8, 9, 10, 11, 12, 13], or to enrich existing models with missing information [14, 15]. Although building geometry data is often distributed across multiple drawings or plans, most research focuses on using floor plans for geometric modeling, addressing only common components and overlooking more complex ones (e.g., stairs), without paying sufficient attention to the modeling of semantic information [16]. Although the mentioned contributions help in obtaining BIM versions from existing digitized plans, none of them have addressed the generation of electrical, mechanical, and plumbing elements among others. Therefore, there is a need to address those components and systems that are also part of the construction and are available in the plans.

Unlike architectural and structural floor plans, the electrical floor plans combine dense thin-line topology with heterogeneous symbols that together define electrical circuits.

To make matters worse, circuit lines are drawn together with architectural lines (e.g., walls), so the circuit extraction must disentangle overlapping and interleaved strokes. To address this problem, there are many challenges to be considered. First, drawing styles can vary significantly across countries, regions, firms, and even individual-projects (often including specific abbreviations or ad-hoc sections). This document is targeted to the construction staff and there is no automation involved that could require strict standards. Electric circuits involve wires and devices represented on the floor plan leading to high visual density and reduced readability. Furthermore, the objects on the floor plan appear in different sizes, rotation, configuration, or overlapped each other. Therefore, the complexity is higher than architectural where walls, doors, and windows are more stable and larger foot-

prints. Electrical plans carry functional connectivity using diverse line styles (e.g. solid/dashed, color/thickness). These lines frequently cross, branch, and overlap other circuits. Since circuits are often drawn on top of architectural layers, producing heavy clutter and frequent occlusions; junctions and cross-overs are visually similar. Then, the circuit identification requires disentangling multiple overlapping visual elements. Existing approaches for architectural/structural plans generally do not require this level of graph disentanglement.

Extending existing works on BIM generation using ML necessitates appropriate datasets for model training. The available floor plan datasets [17, 18, 11, 19, 20, 21, 22] primarily focus on architectural features, which do not provide a complete representation of the building. Therefore, structural, mechanical, electrical, and plumbing plans are required to generate a holistic building model.

The representation of electrical, plumbing, and gas components in floor plans shares a common structure: all installations consist of terminal components —such as outlets, faucets, or gas valves— that receive electrical current or fluids. These terminals are connected through lines in the drawings, representing conduits that house electrical cables, cold or hot water pipes, or gas pipes.

This graphical convention reflects a deeper modeling similarity in the IFC schema, where these systems are represented using analogous structures. Terminal elements are typically modeled as instances of IfcFlowTerminal [2], connected through distribution elements like IfcFlowSegment, and grouped into systems via IfcSystem. Considering these parallels, and for the sake of clarity, the discussion here focuses specifically on the electrical installation.

Several contributions are presented. An end-to-end approach is described for converting raster-based electrical floor plans into BIM models using the IFC standard. The approach relies on Machine Learning (ML) for artifacts detection, DeepLSD for line detection, and a circuit assembly framework. The latter connects detected objects via circuit wires based on detected lines, while honoring electrical design requirements.

To support the model training, the first openly available, annotated cor-

[2] `https://standards.buildingsmart.org/IFC/DEV/IFC4_2/FINAL/HTML/schema/ifcsharedbldgserviceelements/lexical/ifcflowterminal.htm`

pus of electrical floor plans, called IPVBA-ELEC [3], is provided.

This contribution enhances the extraction of architectural and structural details introduced in [12] to obtain a comprehensive BIM model. Finally, a performance evaluation of the approach is conducted by comparing the results of plan processing and electrical circuit generation against the ground truth.

The proposed approach is currently limited to detecting circuits and cables, power outlets, lighting fixtures, switches, and main distribution boards. However, it can be readily extended to other technical drawings, such as mechanical and plumbing plans. Given the nature of the task, separate ML models were employed: one for cable detection and another for electrical element detection. In addition, a script was developed to link circuits to power or lighting outlets, enabling the representation of consumption points for each circuit. The paper does not address the processing of non-graphical documentation, such as power panel configurations and single-line diagrams. Although some datasets include labeled electrical devices (e.g., washing machines, water heaters, TVs) in architectural plans, they lack sufficient detail to fully map circuits or accurately define all consumption points. This paper is organized as follows: Section 2 reviews related work in the field, while Section 3 introduces the necessary background. Section 4 details the proposed methodology. Section 5 describes the process of cable detection and circuit assembly. Next, Section 6 explains the generation of an IFC model from the extracted data. Section 7 presents the evaluation of the results, and finally, Section 8 provides concluding remarks.

2. Related Work

This section reviews prior work on detecting elements in AEC plans, the availability of annotated MEP datasets, and machine learning approaches for line-segment detection methods required for reconstructing wiring.

2.1. Element Detection in AEC Plans and Datasets

Most ML implementations have focused on architectural datasets, emphasizing **architectural components and room topology** modeling without addressing electrical plans. CubiCasa5K and MLStructFP enabled strong

[3]Available to the public at `https://ri.conicet.gov.ar/handle/11336/260283`

baselines for walls, slabs, openings, and spaces, yet MEP semantics—especially electrical circuits—remain largely unaddressed.

In the effort to create as-built models for buildings and infrastructures to generate digital twins, the German government supports the BIMKIT project[4], which focuses, among other aspects, on: (1) detecting technical symbols through deep learning (DL) models [23]; (2) detecting text in architectural plans using DL [24]; (3) the extraction of fire safety equipment symbols from escape plans using keypoint-based detection; and (4) developing drawing analysis ontology for standardizing extracted data representation [25]. This approach allows for the integration of different ML models with specific training to provide document analysis services. However, electrical installations have not yet been addressed.

In prior research by Urbieta et al.[12] presented an approach for generating IFC BIM models from architectural and structural plans using Mask R-CNN [26]. However, to achieve a holistic building model, it is essential to incorporate mechanical, electrical, and plumbing (MEP) systems, which include non-geometric components that have traditionally received less attention from researchers [16]. Figure 1 provides a summary of the previous work [12], where an architectural element detector was applied to architectural floor plans, and a structural element detector was also used for structural floor plans. These outputs facilitated the IFC modeling stage, integrating the identified elements into a unified model. In the current work, electrical floor plans are addressed as part of the MEP systems group, with a focus on extracting electrical elements for subsequent modeling to enable their integration into the overall model. The application of this methodology to mechanical and plumbing plans remains a task for future research.

Various experiences have implemented different detection models to identify elements in architectural floor plans, such as YOLO [7], FCN [18], Faster R-CNN [8, 18], DeepLabV3+ [10], U-Net [27], and the combined use of MDA-Unet and MACU-Net [28].

BIM modeling has also been explored before the rise of ML techniques, as in the prototype presented by Gimenez et al. [29], developed in C++ for generating 3D IFC models from scanned 2D plans, including walls, openings, and spaces. The main drawback of this approach is the need to adjust the algorithm for different architectural styles or adapt it for other objectives,

[4] `https://bimkit.eu/en/welcome-to-bimkit`, accessed on 08/04/2024

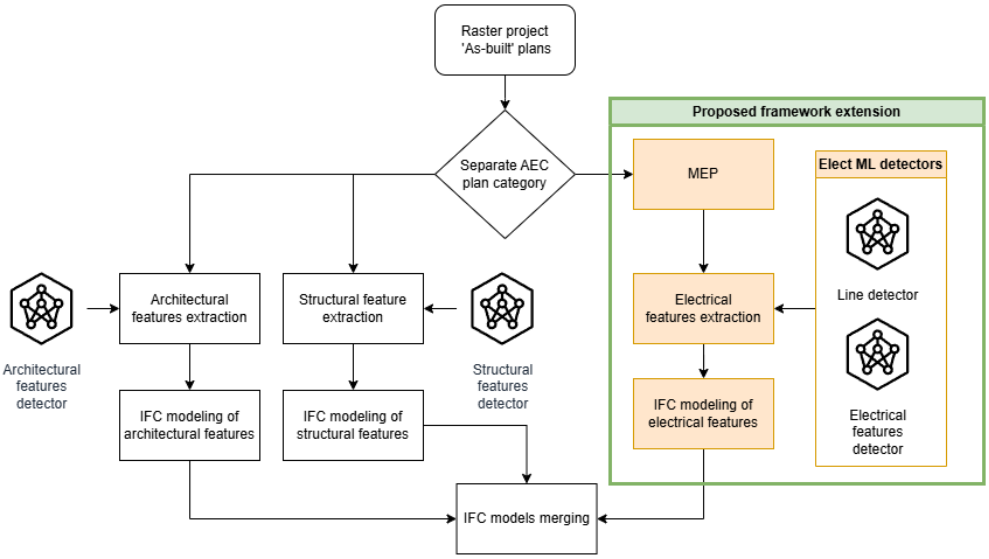


Figure 1: Proposed extension to the original framework by Urbieta et al., incorporating MEP systems.

such as MEP modeling.

There are also precedents for generating BIM models from CAD (Computer-Aided Design) files, processing vector elements from CAD files, and combining them with the Dynamo add-on in Revit to generate architectural [30, 31] and structural [32] BIM models. These techniques require some level of manual intervention, becoming semi-automated when the CAD files need to be preprocessed by a technician.

Liu's method [11] first transforms a raster image into a set of junctions using Convolutional Neural Networks (CNN) and implements Integer Programming (IP) to produce a vectorized floor plans, enabling a 3D model representation. The LIFULL HOME's dataset, which contains 5 million floor plans images, was selected for training, although only 1,000 random samples were used. Other public datasets for architectural floor plans include Rent3D [19], CubiCasa5k [20], CVC-FP [21], and Korea Land and Housing [10]. Pizzarro et al. [27] introduce MLStructFP, a large scale dataset of plans with walls and slabs annotations.

In contrast, the present work targets electrical features using an end-to-end approach, reconstructing circuit graphs through the detection of devices

and wiring. To support further research, IPVBA-ELEC is released as the first open dataset of annotated electrical floor plans.

2.2. Line Detection Using Machine Learning

The approach to identifying and extracting elements from blueprints that require line identification has been widely studied through various methodologies. For instance, [33] use the Hough Transform algorithm and image vectorization. A recent survey by Lin et al.[34] provides a comprehensive overview of image line segment detection.

In this review focuses on machine learning methods for line detection, a computer vision (CV) technique used in 3D reconstruction and segmentation.

For example, Meng et al. [35] present a real-time line segment detection framework called Line Graph Neural Network (LGNN), which uses two modules: a Deep Convolutional Neural Network (DCNN) to generate positions and features of line segments, and a Graph Neural Network (GNN) to reason about their connectivity. Instead of the traditional junction-to-junction line description, they propose a quadruple representation: start junction, end junction, central point of the line, and line offset value. The DCNN is designed to predict a central point heatmap for the line along with a displacement vector map. They observed that in cluttered scenes, the predicted line segments were less fragmented, allowing the endpoints to be reliably assigned to junctions. The GNN module then takes these line segment candidates as vertices and builds a sparse graph to impose structural constraints.

Similarly, Huang et al.[36] present a line segment detector based on three points (TP-LSD), which extends a line in opposite directions from a central point to determine the endpoints, obtaining three keypoints.

Zhao [37] extends the model proposed by Hough [38], which detects shapes in an image, such as ellipses and circles, based on their angles and radii. Once elements are detected, lines are generated to indicate boundaries. While this approach could be useful to suggest the location of a cable after detecting an element, it cannot be applied in this paper, as it does not detect line endpoints. This limitation makes it unsuitable for the present task, where cables have clearly defined start and end points that may not follow a single direction.

In this paper, DeepLSD[39] was selected, which implements LSD [40], an attraction field, and an optimization tool.

3. Background

The proposed approach operates on scanned electrical floorplans and relies on conventional drafting practices, symbol standards, and documentation styles. This section provides the technical background necessary to contextualize the method, including: (i) the representation of electrical plans and their standardized symbols, (ii) machine-learning line-segment detectors relevant to cable-trace extraction, (iii) instance-segmentation models used for electrical component detection, and (iv) the IPVBA-ELEC dataset introduced for training and evaluating the detectors.

3.1. Electrical Plans

Construction records, such as 'as-built documents, typically contain a variety of essential technical documents for project execution. Among these documents are the plans, which can vary in type, such as floor plans, sections, elevations, and details. These plans represent different systems involved in construction, including architectural, structural, electrical, mechanical, and plumbing systems.

In Figure 2, a simple electrical installation floor plan for residential housing is illustrated, showing the location of power outlets, lighting outlets, switches, junction boxes, main distribution board locations, conduit diameters, as well as the number and cross-sectional area of cable conductors.

Electrical plans are accompanied by corresponding legends, as shown in the third column of Table 1, which explain the various symbols, abbreviations, and types of line styles used in the dataset.

Electrical plans also include single-line diagrams that schematically represent elements related to control, operation, measurement, and protection, covering both functional units and general building service systems. In addition, power summary tables quantify all outlets and, through the application of simultaneity coefficients, determine the building's required power. The extraction of elements or semantic information from these tables and diagrams is outside the scope of this paper.

The elements depicted in the floor plans generally do not indicate their height, meaning this data cannot be extracted by the proposed method. Although such data may appear in elevations or detailed drawings, it is often unavailable at the plan level. When height information is required for the model, industry standards can be applied: switches are typically placed between 0.90 – 1.20 m above floor level, while power sockets are positioned

between 0.30 – 0.40 m from the floor, or combined with switches at the specified height.

The electro-technical graphic symbols relevant to detection follow commonly used national conventions in Argentina, historically based on IRAM standards, and international standards such as BS EN 60617 in the UK [41]. Although plan representations vary by region, Table 1 presents a comparison of a selected subset of labeled symbols from the generated dataset, aligned with the Argentine IRAM standard and the British standard, to identify which symbols may perform poorly when switching regions.

Electrical installations represented in as-built plans reflect how they were executed, but they do not always detail the exact position of the cables forming the circuits, particularly when they are embedded within walls or concealed in ceilings. These installations are often modified later due to repairs or increased electrification demands from the addition of appliances. Consequently, this approach cannot assume that representations generated from as-built plans accurately reflect the system's dynamic reality.

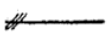
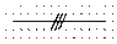
In the CubiCasa5K dataset [20], electrical appliances (e.g., washing machines, refrigerators, dishwashers) are labeled in architectural floor plans, but they do not provide sufficient information to establish circuits or quantify consumption points. In this paper, specific-use circuits will be omitted, although the methodology is extensible [42].

Considering the various efforts to model the architecture and structure of buildings, integrating electrical and plumbing installations will enable a more comprehensive model.

3.2. DeepLSD: An ML Line Detector

DeepLSD, introduced by Pautrat et al.[39], is a learning-based line segment detector designed to achieve robust and accurate performance under noisy and cluttered conditions. The method predicts a dense line attraction field using a deep neural network, which is subsequently transformed into gradient magnitude and orientation maps. These representations enable the extraction of line segments through a refinement stage that incorporates geometric constraints and vanishing point estimation. The approach builds upon the principles of the classical Line Segment Detector (LSD)[40], which is widely used due to its accuracy and efficiency, while overcoming its sensitivity to noise and low-contrast structures. Compared to earlier learning-based detectors such as TP-LSD[36], DeepLSD favors completeness and structural

Table 1: Comparison of Standards and Labeled Classes

Object	Symbol per IRAM 2010	Symbol per BS-EN-60617	Symbol from Dataset	Class
Circuit			[line]	circuit
Meter				meter
Wall-mounted light fixture				wallLightOutlet
Ceiling-mounted light fixture				ceilingLightOutlet
Distribution board		N/A		mainBoardPanel
Switch				singleSwitch
Socket outlet				generalOutlet specialOutlet
Grounding				grounding

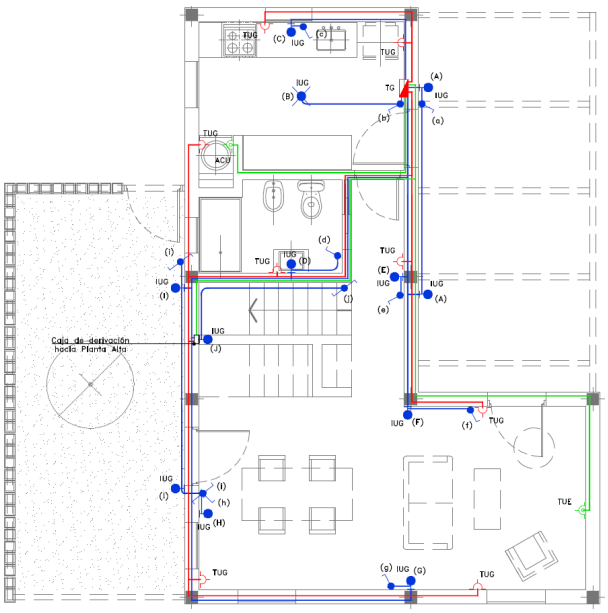


Figure 2: Sample of electrical floor plan

consistency, which is particularly relevant for architectural drawings where line continuity is critical.

Owing to its robustness and ability to recover long and coherent line segments, DeepLSD is well suited for extracting cable traces from scanned electrical floor plans. In this paper, the detected line segments are further filtered and integrated with detected electrical devices to support circuit reconstruction.

3.3. Instance Segmentation Detectors

To accomplish the task of instance segmentation, the Cascade Mask R-CNN model [43], an extension of Cascade R-CNN [44] ,YOLOv11m-seg [45], and YOLOv8m-seg, were selected. In addition to delivering state-of-the-art performance, these models offer broad compatibility with commonly used Python libraries.

Cascade Mask R-CNN integrates object detection and instance segmentation into a unified, multi-stage architecture. Building on Cascade R-CNN, this method incorporates multiple detection stages, each with progressively increasing Intersection over Union (IoU) thresholds, refining the bounding box predictions. In addition to object detection, Cascade Mask R-CNN

introduces a segmentation branch that provides accurate masks for each detected object. At each stage of the cascade, both bounding boxes and masks are progressively refined, enhancing the precision and quality of the segmentation.

The YOLO (You Only Look Once) series has significantly impacted object detection with its real-time capabilities. YOLOv8 balances accuracy and speed through an advanced backbone and neck architecture and an anchor-free split head. Building on this, YOLOv11 is the latest iteration, offering enhanced performance. It refines feature extraction and achieves a higher mean average precision (mAP) with 22% fewer parameters than YOLOv8m, making it highly efficient. Both versions are versatile for various computer vision tasks, including instance segmentation, which is crucial for the work.

3.4. IPVBA-ELEC Dataset

The Housing Institute of the Province of Buenos Aires (IPVBA) is a governmental agency responsible for implementing public policies that publishes architectural and technical documentation for social housing projects. The dataset used in this study, IPVBA-ELEC, consists of 54 electrical floor plans of social housing units, made publicly available through the official website[5]. The floor plans, originally in PDF format, were preprocessed and annotated using the LabelMe[46] JSON format, enabling the structured identification of electrical components. The resolution of the floor plans ranges from 6000 to 7500 pixels in width and from 3500 to 4700 pixels in height, ensuring sufficient detail for component detection and annotation. The table 2 presents the annotated features and their instance counts. Element annotations were performed by researchers and subsequently reviewed by an engineer with expertise in electrical floor plans.

4. Approach

The proposed approach focuses on detecting electrical installation components—such as power outlets, lighting fixtures, and electrical circuit cables - in floor plans, with the objective of generating an IFC model. This constitutes an extension of the framework proposed by Urbieta et al., incorporating

[5]IPVBA - Tender Plans `https://www.gba.gob.ar/vivienda/licitaciones` Accessed September 20, 2022

Table 2: IPVBA-ELEC dataset features summary

Nr.	Features	Instances
1	Dual General-Purpose Electrical Outlets	620
2	Wall-Mounted Lighting Outlet	449
3	Ceiling-Mounted Lighting Outlet	282
4	Single Switch	261
5	Dual Special-Purpose Electrical Outlets	169
6	Double Three-Way Switch Assembly	83
7	Main Electrical Distribution Panel	44
8	Momentary Pushbutton Switch	36
9	Triple single-switch assembly	27
10	Two-way (or three-way) switch	13
11	Electrical Junction Box	9
12	Electricity Meter	6
13	Four single-switch assembly	6
	Total	2005

MEP systems, as illustrated in Figure 1. The resulting model can serve either as the basis for a new IFC model or as an enhancement to an existing one.

To this end, the method simultaneously applies instance segmentation to identify electrical components and a line detector to extract all line traces present in the floor plan image. Among these lines, only a subset corresponds to electrical circuit cables that interconnect with identified components to deliver power. The key challenge is to accurately infer which of these line traces represent actual circuit connections between electrical components; that is, to reconstruct the wiring layout and its associated components.

The overall workflow is illustrated in Figure 3, and can be summarized as follows:

- **Feature Detector Training:** A feature detector for electrical components was initially trained (Step ①).
- **Data Acquisition and Curation:** Data was acquired from the IPVBA source (Step ②), as detailed in Section 3.4 . Electrical floor plans were manually selected based on content suitability. Under the supervision of a qualified engineer, manual polygon annotations (Step ③) of circuit elements were performed using a labeling tool [46].
- **Dataset:** This curated dataset served as the ground truth for model training and has been made publicly available to support reproducibility (Step ④).

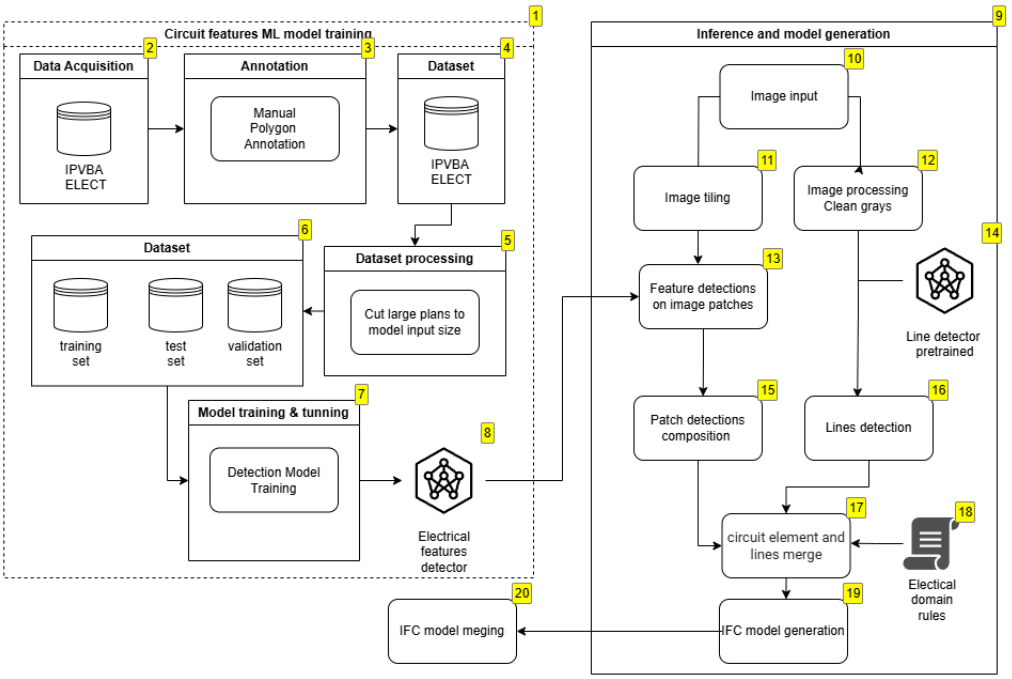


Figure 3: Approach for BIM model generation from electrical floor plans

- **Dataset Pre-processing and Splitting:** To meet the input requirements of the detector model and avoid image downsizing, the dataset images were cropped (Step ⑤). These were then split into training (70%), test (15%), and validation (15%) sets (Step ⑥).

- **Model Fine-Tuning:** Transfer learning techniques were applied to fine-tune a pretrained model (Step ⑦), optimizing its performance as the primary detector of electrical features. The best-performing model was selected for further use (Step ⑧).

During the inference and model generation pipeline (Step ⑨), the following steps are executed:

- **Input Image & Pre-processing:** An electrical floor plan image (Step ⑩) serves as the primary input. The image is prepared and minor changes are applied such as grayscale content removal to suppress architecture details (Step ⑫).

- **Image Division for Feature Detection:** The high-resolution electrical floor plan is divided into sub-images (Step (11)) to facilitate feature detection. The detector processes each sub-image independently (Step (13)), producing a JSON object that records the classes and locations of the detected elements. These outputs are then reassembled into the original image coordinate system (Step (15)).

- **Line Detection and Preprocessing:** Simultaneously, the complete floor plan image is preprocessed — such as removing grayscale elements — before being passed to a line detector (Step (14)). In this stage, no fine-tuning was necessary; instead, the pretrained DeepLSD model was directly employed to detect line segments within the image.

- **Feature and Line Integration:** After running the line detector, a post-processing stage is used to determine which of the detected lines represent electrical cables, as many may correspond to architectural elements or other non-electrical elements within the image. The detected features and lines are then processed by a specialized script that applies electrical design rules (Step (18)) to group elements logically and infer the underlying circuits (e.g., outlets, switches, cables). This step is discussed in detail in a subsequent section.

- **IFC Model Generation:** An Industry Foundation Classes (IFC) model is generated (Step (19)), representing the inferred electrical system layout.

- **Model Merging:** Finally, the generated IFC model can be integrated with architectural and structural BIM models (Step (20)) to produce a comprehensive, multidisciplinary BIM representation.

In this paper, Cascade Mask R-CNN and YOLOv11 were used for the object detection phase, and DeepLSD model for the line segment detection task. In Figure 4, it is depicted the approach steps (10) – (19) and their interaction with ML models. In the image, you can see the usage of two different models that run independently each other, and then the outcomes of both tasks are combined to produce an IFC.

When an approach instantiation in a different context (i.e country,or region) that requires using a new training dataset, the accuracy could be

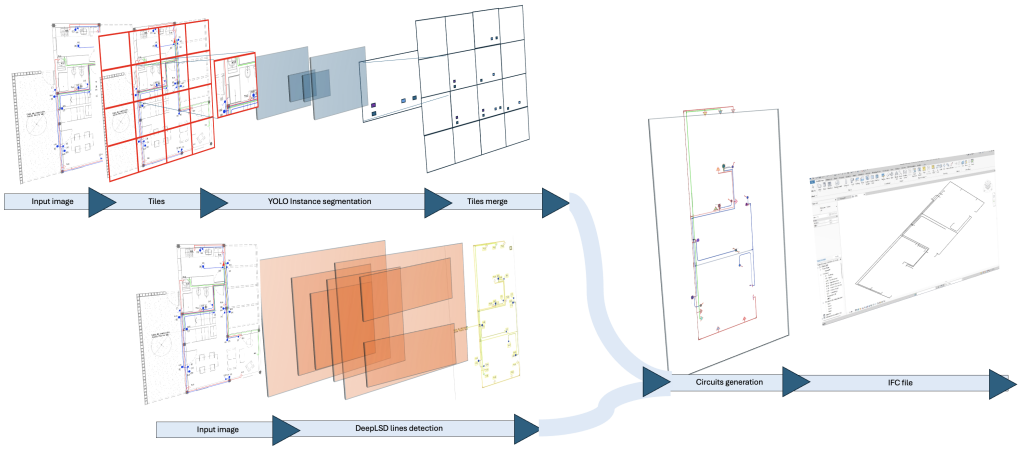


Figure 4: Models and data flow

negatively affected. A guidance for improving the approach's accuracy in Appendix A.

4.1. Training

The methodology is model-agnostic and modular, enabling the replacement of instance segmentation or line detection components as improved models become available. In line with the evaluation strategy proposed by König et al. [23], , the performance of multiple deep learning models for symbol segmentation in technical floor plans is explored. Specifically, three architectures were tested: Cascade Mask R-CNN (ResNet-50 FPN backbone), YOLOv8m-seg, and YOLOv11m-seg. All models were implemented using the MMDetection[6] and Ultralytics[7] libraries, and trained with a custom-labeled dataset of electrical floor plan features.

Experiments were carried out at three different image resolutions: 256×256, 512×512, and 1024×1024. Additionally, both a full version of the dataset (100%) and a reduced version (10%) were used to analyze model robustness under limited data conditions. Each configuration followed a 70% training, 15% validation, and 15% test split. The Cascade Mask R-CNN model was

[6] `https://github.com/open-mmlab/mmdetection`
[7] `https://github.com/ultralytics/ultralytics`

fine-tuned from a pretrained checkpoint on the COCO dataset[8] using transfer learning.

Training was performed on an NVIDIA GeForce RTX 4070 Ti GPU. Model performance was evaluated using the standard metrics mAP@0.5 and mAP@0.5:0.95, providing a consistent basis for comparison across architectures and dataset configurations.

Although the dataset enabled effective experimentation, it was derived entirely from a single institutional source (IPVBA), which introduces a bias in symbol styles and design patterns. Future work could mitigate this limitation through synthetic dataset generation techniques, such as those proposed by Vilgertshofer et al. [47], which would allow the injection of underrepresented features and promote class balance.

Model	**Input Size**	$\text{mAP}^{\text{IoU}}_{0.50}$	$\text{mAP}^{\text{IoU}}_{0.50:0.95}$	**Epochs**
YOLOv11m-seg	256x256	0.8934	0.6004	96
YOLOv8m-seg	256x256	0.8930	0.6078	124
Cascade R-CNN	256x256	0.7660	0.5410	250
YOLOv11m-seg	512x512	0.9874	0.7186	150
YOLOv8m-seg	512x512	0.9896	0.7235	150
Cascade R-CNN	512x512	0.9440	0.6970	249
YOLOv11m-seg	**1024x1024**	**0.9910**	**0.7368**	**84**
YOLOv8m-seg	1024x1024	0.9870	0.7310	72
Cascade R-CNN	1024x1024	0.9150	0.6600	237

Table 3: Model comparison across input sizes using 100% of the dataset.

The results in Table 3 show a consistent performance improvement with increasing image resolution for all models. YOLOv11m-seg achieved the best overall performance, reaching a peak $\text{mAP}^{\text{IoU}}_{0.50:0.95}$ of 0.7368 at 1024×1024, outperforming YOLOv8m-seg and Cascade R-CNN in both accuracy and training efficiency.

A comparison between Table 3 (100% of the dataset) and Table 4 (10%) reveals a consistent degradation in performance across all models when trained

[8] `https://github.com/open-mmlab/mmdetection/blob/main/configs/cascade_rcnn/cascade_mask_rcnn_r50_fpn_1x_coco.py`

Model	Input Size	$\text{mAP}^{\text{IoU}}_{0.50}$	$\text{mAP}^{\text{IoU}}_{0.50:0.95}$	Epochs	Performance drop
YOLO11m-seg	256x256	0.7244	0.4873	37	18.8%
YOLO8m-seg	256x256	0.7379	0.4675	35	23.1%
Cascade R-CNN	256x256	0.5430	0.3080	207	43.1%
YOLOv11m-seg	**512x512**	**0.9194**	**0.6380**	**138**	**11.2%**
YOLO8m-seg	512x512	0.8971	0.5688	46	21.4%
Cascade R-CNN	512x512	0.5400	0.3270	246	53.1%
YOLO11m-seg	1024x1024	0.8239	0.5515	25	25.1%
YOLOv8m-seg	1024x1024	0.8784	0.5778	52	21.0%
Cascade R-CNN	1024x1024	0.7620	0.5060	243	23.3%

Table 4: Model comparison across input sizes using 10% of the dataset.

on the reduced dataset. The most notable drop is observed in the $\text{mAP}^{\text{IoU}}_{0.50:0.95}$ metric, highlighting the sensitivity of detection accuracy to data availability.

At 512×512, YOLOv11m-seg demonstrated the strongest resilience, retaining 88.8% of its original performance when trained on only 10% of the dataset (compared to 100), while Cascade R-CNN fell to just 46.9%.

Because of the good performance reported by assessed models, they were used directly without additional fine-tuning or architectural improvement. In the event that local drawing styles yield unsatisfactory results due to local codes of practice, the object detection model could profit from existing efforts on technical drawing [48, 49, 50, 51].

5. Circuit Assembly and Interpretation

For cable detection, as described in Section 4, the pretrained DeepLSD model was employed to detect lines. Initial tests confirmed that this model reliably identified lines in the floor plans. Unlike element detection, the task here involved a post-processing stage to determine which of the detected lines corresponded to electrical cables. Once the relevant lines were identified, the next step was to assemble and interpret the electrical circuits. The detailed methodology is outlined below:

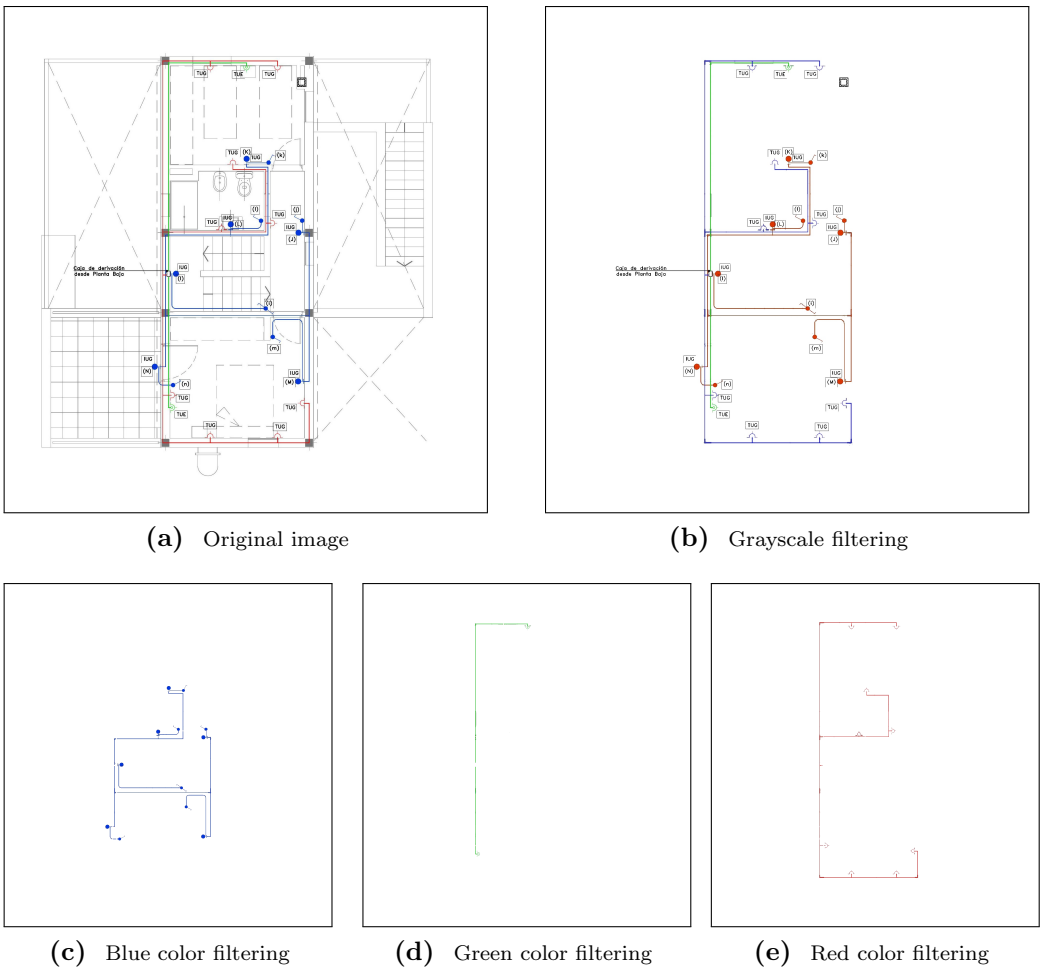
- *Image preprocessing.* Prior to line detection, image preprocessing is a crucial step. This process refines the floor plan by removing irrelevant

architectural elements, thus simplifying the image for subsequent analysis. In the dataset used, background elements such as walls and other structural features are typically drawn in light grayscale, while electrical circuits are represented using thicker or colored lines, typically blue or red in the dataset. This distinction proved advantageous, as it allowed for the effective removal of grayscale elements during preprocessing without affecting the visibility of circuit-related lines. For instance, Figure 5a displays a typical raw drawing of a floor plan, while Figure 5b shows the same plan with grayscale content removed. In some locations the circuits could be documented using specific colors accordingly to their goal. In figures 5c,5e, and 5d you can see three different circuits in blue, red and green accordingly. This filtering helps removing unnecessary lines form technical drawings and reducing false positives during the detection process.

As a result, the DeepLSD model processes a reduced and more relevant set of lines (Figure 6a), which improves both detection accuracy and computational efficiency. However, in cases where electrical circuits are not distinctly represented using color or thicker strokes, alternative preprocessing strategies may need to be explored.

- *Post-processing of segmentation results.* In some cases, the instance segmentation process identifies different element instances but they actually refer to the object (it is unexpected split). This typically occurs when adjacent pixels of the same object are assigned to separate regions. As illustrated in Figure 7a, the model initially identifies one object as three separate instances. To resolve this, a merging strategy was applied using an offset-based proximity criterion. Segments that are close or overlapping were consolidated into a single detection, as shown in Figure 7b. Only merged segments with a confidence score greater than or equal to 0.9 were retained for subsequent processing.

- *Feature-Aware Line Filtering.* Once the initial set of lines is detected using DeepLSD, an additional filtering stage is performed. Despite pre-processing, some lines unrelated to electrical wiring—such as those forming part of annotated components—are still detected. Moreover, overdetections of the same line frequently occur, often resulting in overlapping or closely adjacent segments. These redundancies must be resolved to ensure a clean and coherent set of lines for accurate circuit



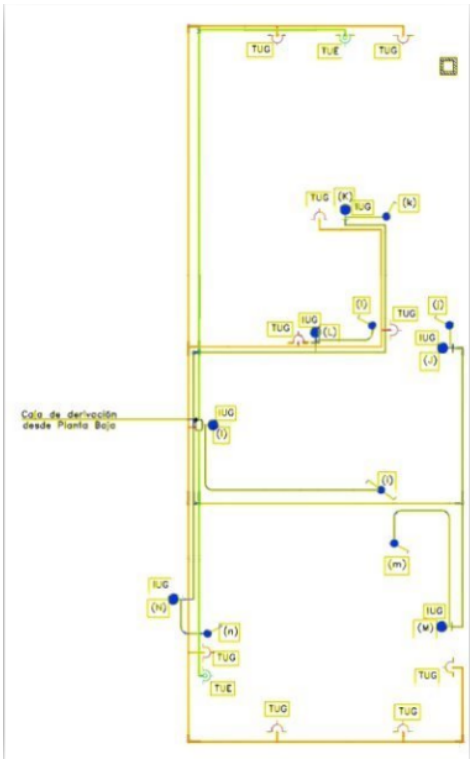
(a) Original image **(b)** Grayscale filtering

(c) Blue color filtering **(d)** Green color filtering **(e)** Red color filtering

Figure 5: Example of color filtering strategies on a floor plan.

assembly. Figure 8a shows an example where lines corresponding to internal features (highlighted in yellow within the polygon) must be excluded. After this cleanup step (Figure 8b), the remaining lines more accurately represent potential cables.

Moreover, overlapping or redundant lines were merged to preserve consistency during graph construction. Importantly, polygonal masks were preferred over bounding boxes for this filtering step, as bounding boxes may exclude valid but adjacent wiring elements. Therefore, a final refinement step ensures that only the necessary lines remain, as shown



(a) Detected lines by DeepLSD

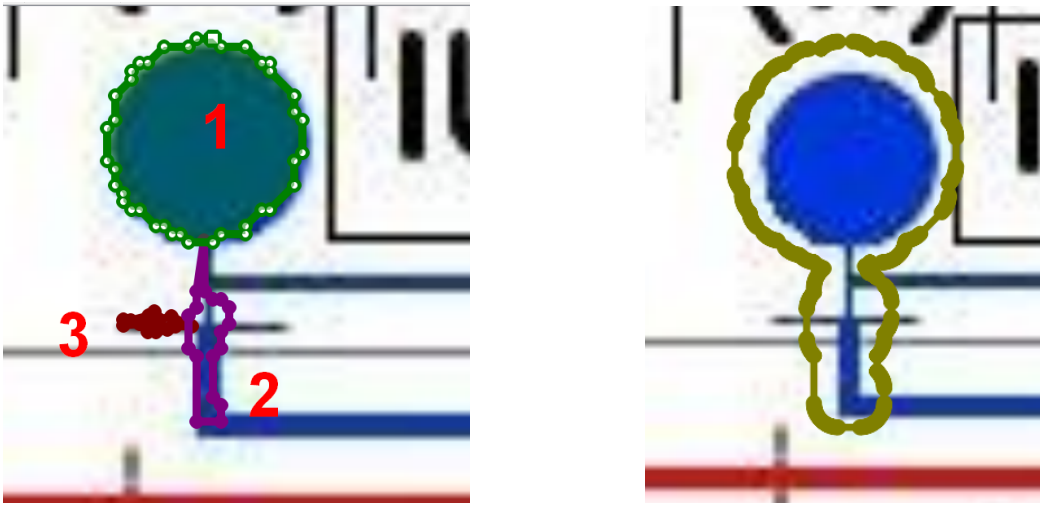
Figure 6: Example of grayscale filtering on a floor plan and subsequent line detection.

in Figure 6a, enabling accurate circuit reconstruction.

- *Circuit Graph Construction.* Once the filtered lines were obtained, a connectivity graph was constructed, linking all detected elements using the Python library Shapely[9]. This graph represents potential circuits by connecting elements via the lines representing electrical cables.

  To ensure semantic correctness, circuit classification rules based on Asociación Electrotécnica Argentina (AEA) guidelines were applied. For example, circuits for lighting must not include outlets, and vice versa. These rules are enforced through an iterative graph-building algorithm using NetworkX[10]. During each iteration, new lines are added, and

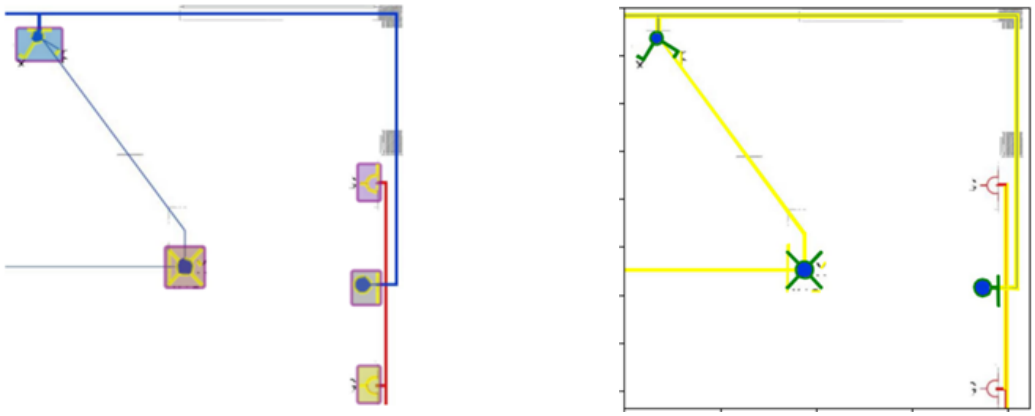
[9] `http://shapely.readthedocs.io/`
[10] `http://networkx.org/`



(a) Three detected instances

(b) Merged detected instances

Figure 7: Example of Segmentation Merging Strategy.



(a) Lines in the object bounding box

(b) Lines in the bounding box removed

Figure 8: Removal of lines overlapping with detected features.

invalid connections—such as a lighting component incorrectly linked to a power outlet—are removed by tracing the paths and eliminating conflicting branches. When present, the main distribution board is considered as the circuit's starting point.

- *Circuit representation and data structuring.* Once the circuits have been correctly assembled and structured according to applicable design rules and standards, the resulting information is stored in a structured JSON file. This file acts as a comprehensive data representation of the electrical layout, capturing each element's class, geometry, and connections to other components. It serves as the input for the next phase: generating a BIM model in the IFC format.

  Establishing the logical relationships between lighting switches and their corresponding fixtures typically requires the use of OCR recognition, as these connections are often denoted by matching letters (e.g., switch (a) – lighting (A)) to indicate functionality for the IFC model. The implementation of such OCR-based associations is beyond the scope of the present study and is left for future work.

As noted above, the line detection step identifies all lines in the document, including both circuit and non-circuit elements (e.g., text, frames, dimensions). While the approach can exclude non-circuit lines, poor drawing quality may still hinder accurate circuit line detection. Faint, broken, or overlapping lines are common problems, although the DeepLSD method has shown strong performance in many cases. Line continuity is another challenge. Broken or intersecting strokes often fragment circuit lines. This is mitigated by using an algorithm that reconnects segments and resolves ambiguities caused by crossings and strokes. Overlapping circuits, however, remain a difficult case.

6. Generation of the IFC Model

Following the construction of electrical circuits and the organization of the detected elements into a structured JSON format, the next step involves generating a Building Information Modeling (BIM) representation. This JSON file serves as the input for producing an IFC (Industry Foundation Classes) file, enabling integration with BIM platforms.

To achieve this, Python was used alongside the open-source IfcOpenShell library[11], which allows the creation and export of IFC-compliant models. IfcOpenShell is also the core engine behind the BlenderBIM plugin, a free

[11] `http://ifcopenshell.org/`

and open-source BIM authoring tool. In this context, both the BIM model and the generated IFC file are interchangeably referred to as the output of the modeling process.

The resulting IFC model contains the detected electrical components, their spatial relationships, and the circuits to which they belong. Although these elements are currently limited to their 2D footprint (with no height data), the structure is compatible with further enrichment using cross-sectional drawings or parametric definitions in future developments. The present study complements earlier research [12], which focused on modeling other architectural components such as walls, doors, windows, beams, and columns.

The BIM model generation code then creates instances of each lighting outlet or power outlet, and then creates the circuits that connect them. These circuits converge at the location corresponding to the electrical distribution board.

The elements are modeled according to the standards established by buildingSMART[12] for the electrical domain. Outlets are represented using *IfcOutlet*, lighting fixtures with *IfcLightFixture*, and cables with *IfcCableSegment*. Additional elements such as electrical distribution boards and lighting switches are instantiated using *IfcElectricDistributionBoard* and *IfcSwitchingDevice*, respectively. Each instantiated element is assigned one or more *IfcDistributionPort* instances, which are connected using *IfcRelConnectPorts*, ensuring logical consistency beyond geometric appearance.

It is important to note that the location of electrical elements in architectural plans inherently includes a positional tolerance. For instance, outlets are often depicted outside the walls to improve visual clarity, even though their correct placement during modeling must be on the wall surface itself.

The evaluation was performed on a basic subset of BIM elements, and the approach can be extended with finer electrical details to increase the level of detail (LOD) [52]. The level of detail is LOD 200 (cable, main distribution board) and LOD 290 [53] (outlets, lighting), where the model element is graphically represented as a generic system or object, with approximate quantities, size, shape, and their exact positions subject to change (i.e. outlet height). Non-graphical information may also be included in the modeled elements. From the generated model, its element properties can be inspected,

[12] `https://standards.buildingsmart.org/IFC/RELEASE/IFC4/FINAL/HTML/annex/annex-d/ifcelectricaldomain/index.htm`

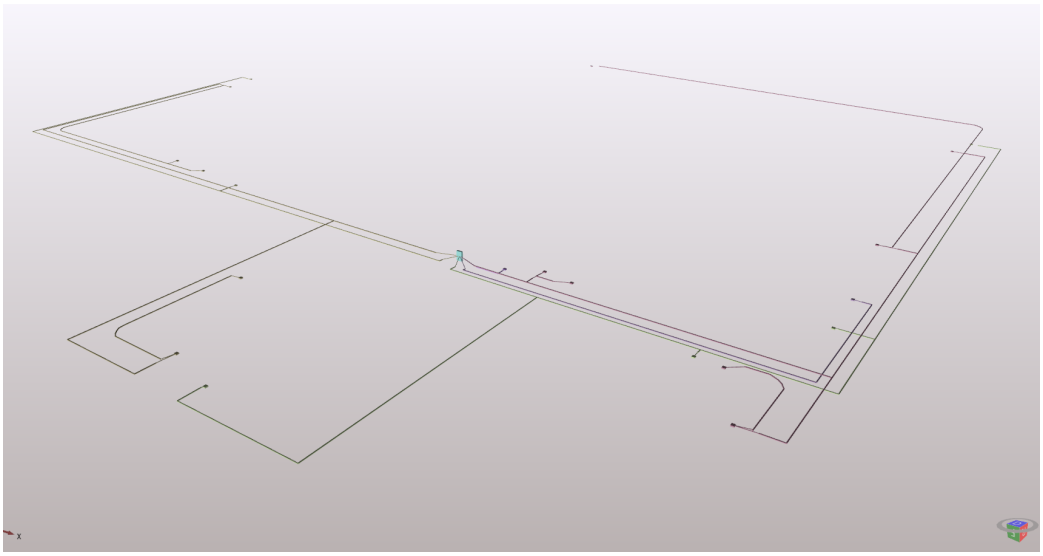


Figure 9: Generated IFC model

such as geometric dimensions, area, and volume audited. Figure 9 illustrates a representative view of the resulting model.

In future developments, the model could be further enriched by incorporating additional electrical properties such as cable cross-sectional area, operating voltage, and estimated power consumption. These attributes could be extracted using advanced text recognition techniques or inferred from additional detected elements.

In this step, any errors from the object detection or line detection stages propagate into the assembly process. Skipped components or missing circuit lines affect the accuracy of the final circuit definition. The merging algorithm can handle many scenarios, such as line disambiguation near elements, multi-line connections, and circuit–component composition. However, inaccuracies in the earlier stages result in incomplete or incorrect input data. Misclassified objects or missing lines lead to orphaned elements that are not connected to any circuit. And then they are discarded. Fortunately, false positives are rare. Each circuit must satisfy several constraints allowing the system to filter invalid circuit combinations such as including both a start and end electrical component, avoiding mixing components from different component families, etc.

7. Evaluation

No established methodology was found in the literature for evaluating the performance of automatically generated electrical circuits. Consequently, the assessment in this paper focuses on three complementary aspects: (i) the quality of line detection representing circuit traces, (ii) feature detections performance (iii) the topological correctness of the generated BIM model. A key contribution of the proposed approach is the implementation of a pipeline that reconstructs circuit connectivity from line segments extracted by a line detection model such as DeepLSD.

7.1. Circuit Line Detection Evaluation

In conventional line detection benchmarks [34, 54], the line detection evaluation relies on one-to-one matching between predicted and ground-truth segments, typically using the Hungarian algorithm, and a binary indicator based on overlap thresholds. However, in this case, this approach is not directly applicable because the prediction process often yields on multiple fragmented segments, while the ground truth is represented by a single continuous polyline. Under such circumstances, applying segment-level matching would artificially penalize correct detections that are only split due to rasterization effects, or the extractor's handling of line thickness. To address this, the well-known Precision, Recall, IoU, and F1-Score evaluation metrics are adapted to operate on *sets of lines* rather than on individual segments. Specifically, both the predicted lines $\mathcal{L}_P$ and the ground-truth lines $\mathcal{L}_{GT}$ are expanded into *buffer regions* of thickness τ, which accounts for line width and positional tolerance. The comparison then reduces to measuring the overlap between these buffered regions, expressed in terms of area. Importantly, τ is set to match the line thickness of the original raster floor plan and is the same parameter employed by the circuit extraction script when unifying duplicated or fragmented line segments. The adapted metrics are defined as follows:

$$\text{Precision} = \frac{\left|\text{Buffer}(\mathcal{L}_P, \tau) \cap \text{Buffer}(\mathcal{L}_{GT}, \tau)\right|}{\left|\text{Buffer}(\mathcal{L}_P, \tau)\right|} \tag{1}$$

$$\text{Recall} = \frac{\left|\text{Buffer}(\mathcal{L}_P, \tau) \cap \text{Buffer}(\mathcal{L}_{GT}, \tau)\right|}{\left|\text{Buffer}(\mathcal{L}_{GT}, \tau)\right|} \tag{2}$$

$$\mathrm{IoU}(\mathcal{L}_P, \mathcal{L}_{GT}) = \frac{\left|\mathrm{Buffer}(\mathcal{L}_P, \tau) \cap \mathrm{Buffer}(\mathcal{L}_{GT}, \tau)\right|}{\left|\mathrm{Buffer}(\mathcal{L}_P, \tau) \cup \mathrm{Buffer}(\mathcal{L}_{GT}, \tau)\right|} \tag{3}$$

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

where $\mathrm{Buffer}(\mathcal{L}, \tau)$ denotes the polygonal region obtained by applying a morphological buffer of thickness τ around the set of lines $\mathcal{L}$. These metrics represents:

- **Precision** quantifies the proportion of the predicted coverage that corresponds to ground truth, thus penalizing false positives.
- **Recall** quantifies the proportion of the ground-truth coverage captured by the prediction, thus penalizing false negatives.
- **IoU** provides a balanced measure of overall overlap between predicted and ground-truth line regions.

This formulation provides a stricter yet more realistic evaluation criterion for this scenario, as it naturally accommodates cases where a single ground-truth line corresponds to multiple predicted segments, avoiding artificial penalties due to fragmentation. In this evaluation, the ground-truth annotations are further enriched by incorporating the corresponding traces as *LineStrings*, where the `group-id` field explicitly specifies the circuit to which each trace belongs. This enhancement provides a reliable GT reference, enabling consistent operations for computing the evaluation metrics. In Figure 10, it is shown the prediction of a circuit (in blue at left-hand side) and the expected ground-truth (in red at right-hand side). For the sake of space, 4 plans were selected and 2 circuits per plan were reported in the visual report. The reader can verify that the layout of the predicted circuit is similar to the ground-truth.

Moreover, Table 5 summarizes the overall processing results of 7 different plans. The metrics reported a good performance showing IoU of 0.89 on average with a minimum of 0.88. Unlike object detection, here IoU reflects the spatial agreement of reconstructed traces with the ground truth. Regarding line tracing (Length ratio), the total detected line length may differ from the original ground truth. An overestimation may occur when the reconstructed endpoints slightly deviate from their true positions, generating small diagonal connections or virtual joints to reach incomplete elements. Conversely,

underestimation can arise when unconnected or isolated segments are discarded during the circuit assembly process. A dedicated control parameter prevents the inclusion of duplicated or overlapping segments. While these variations remain acceptable for wire length estimation, they require caution when modeling conduits, since a single conduit typically carries multiple wires that may be represented as separate traces. The IoU analysis of buffered line geometries (4th column) confirms that the reconstructed traces generally preserve the original topology and spatial distribution. When analyzed on a circuit-by-circuit basis, this metric also reveals occasional annexed or merged circuits caused by tolerance settings in the geometry processing pipeline. In practice, these minor deviations are considered satisfactory, as they maintain the overall structural and topological coherence of the electrical layout.

7.2. Features detections and circuit assembly qualitative evaluation

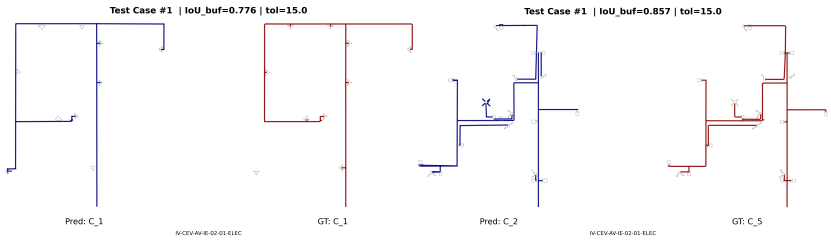
This section focuses on the evaluation of the generated IFC circuits, taking into account local regulations and codes of practice. For brevity, the evaluation is conducted on three electrical installation floor plans, covering a total of 10 circuits, assessing both the component detection performance (Table 6) and the automatically generated circuits (Table 7), listing the floor plan ID, filename, element counts from manual annotation (*Ground Truth*), and inferred counts (*Result*). The column "Result - Circuit IDs" reports the number of circuits generated from detected components and line associations.

The floor plans were evaluated using the detectors described in previous sections, which produced various predictions, each with a confidence score above 0.9 for the respective element classes. The sensitivity of these predictions was assessed using three well-established metrics: precision, recall, and F1-Score, all of which have a range of values from 0 to 1. The results presented in Table 6.

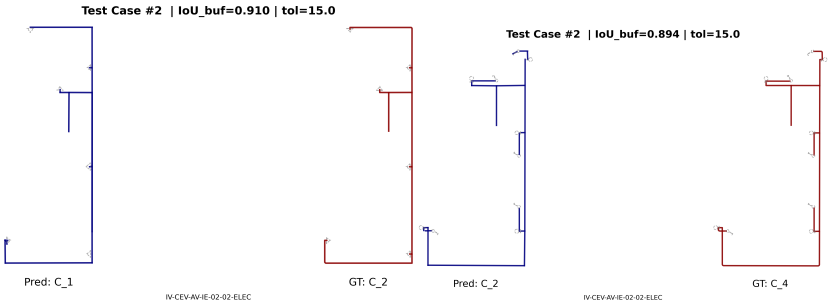
To complete the evaluation of the proposed method, the annotated ground truth of three control floor plans was compared against the outputs predicted

Table 5: Circuit lines detection evaluation on floor plans

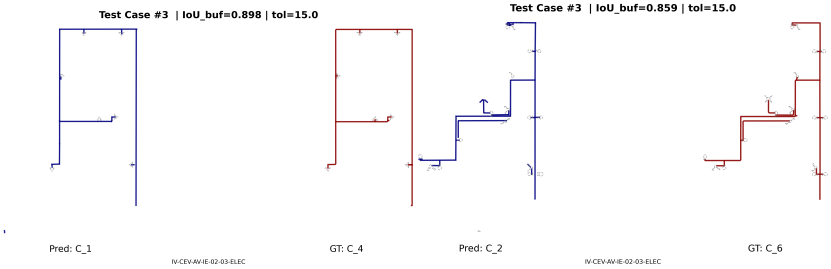
Nr.	Case	Length ratio	IoU(buf)	Precision IoU	Recall IoU	F1-Score
1	IV-CEV-AV-IE-02-01	1.049	0.88	0.91	0.94	0.92
2	IV-CEV-AV-IE-02-02	1.094	0.90	0.94	0.91	0.94
3	IV-CEV-AV-IE-02-03	0.992	0.89	0.94	0.94	0.94
4	IV-CEV-AV-IE-02-04	1.074	0.91	0.96	0.96	0.96
5	IV-FRA-IE-PL-11c	0.975	0.89	0.91	0.91	0.96
6	IV-ROM-IE-PL-BT-29b-01	1.043	0.90	0.95	0.95	0.95
7	IV-ROM-IE-PL-BT-29b-02	1.109	0.89	0.94	0.95	0.94
8	Global	1.084	0.89	0.94	0.95	0.94



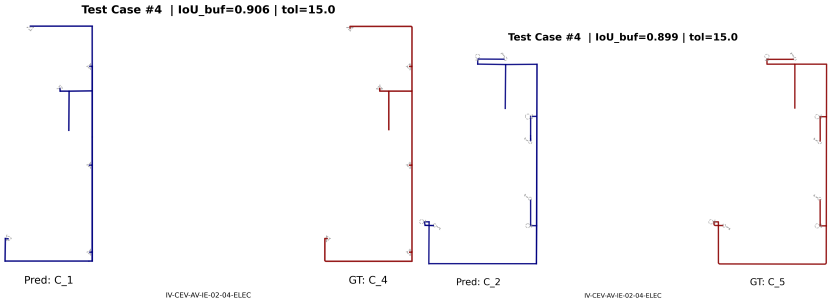
(a) Plan 1 - Circuit 1 - IoU=0.776 **(b)** Plan 1 - Circuit 2 - IoU=0.857



(c) Plan 2 - Circuit 1 - IoU=0.910 **(d)** Plan 2 - Circuit 2 - IoU=0.894



(e) Plan 3 - Circuit 1 - IoU=0.898 **(f)** Plan 3 - Circuit 2 - IoU=0.859



(g) Plan 4 - Circuit 1 - IoU=0.906 **(h)** Plan 4 - Circuit 2 - IoU=0.899

Figure 10: Circuit line detection analysis

by the detection and circuit-generation models, as summarized in Table 7. This evaluation includes two complementary aspects: (1) A quantitative assessment, based on the number of elements correctly detected across 13 predefined categories; and (2) A topological validation, based on whether the detected elements are correctly associated with their corresponding circuits. The table lists, for each case, the floor plan ID, filename, element counts from manual annotation (*Ground Truth*), and the counts inferred by the model (*Result*).The column "Result - Circuit IDs" indicates how many distinct circuits were generated based on the detected elements and inferred line connections.

In the first case, which consists of a single circuit identified as 1A, the approach initially generates an additional circuit (+1) to the one present in the floor plan. This outcome reveals a limitation in the circuit generation logic: the strict separation of circuits based on usage, in accordance with AEA regulations, which state that lighting and power outlets must be placed on separate circuits. This rule was embedded into the script to ensure compliance. However, in real-world scenarios, it is common to find circuits that combine both types of loads, even if such configurations do not align with regulatory standards. This inconsistency is evident in Circuit '1A', where the methodology splits what was originally a single circuit into two: one for outlets and another for lighting. Based on this finding, the script was improved to support both strict and flexible circuits to accommodate such real-world deviations from the current regulatory norms. After tweaking this parameter, the circuit was detected correctly. In the second case, there are three circuits named 2A, 2B, and 2C. In all cases, the number of circuits was correctly generated, and the electrical components were associated, except in one case in 2C, where a switch was not included (-1). In the last case, six circuits are identified: 3A to 3F. In cases 3A, 3B, 3E, and 3F, the elements were correctly generated and assigned to the circuits. However, the circuit 3D was not generated (✗), and its only component (-1) was mistakenly associated with the circuit 3C (+1). This issue stemmed from post-processing logic (see Section 5), in which the cable leading to the main panel was removed. In this particular case, the main panel was included within the element's bounding box, leading to the error. To mitigate this, the methodology was improved by using the segmentation polygon for element detection instead of relying solely on the bounding box.

Description	Precision								Recall								F1-Score							
	specialDoubleOutlet	generalDoubleOutlet	wallLightOutlet	junctionBox	twoWaysSwitch	dobleSwitch	singleSwitch	mainBoardPanel	specialDoubleOutlet	generalDoubleOutlet	wallLightOutlet	junctionBox	twoWaysSwitch	dobleSwitch	singleSwitch	mainBoardPanel	specialDoubleOutlet	generalDoubleOutlet	wallLightOutlet	junctionBox	twoWaysSwitch	dobleSwitch	singleSwitch	mainBoardPanel
Case 1	1	1	1	-	1	1	1	1	1.0	0.9	0.9	-	1.0	0.8	1.0	1	1.0	1.0	0.9	-	1.0	0.9	1.0	1.0
Case 2	0.5	1	1	1	1	1	1	-	1.0	1.0	0.9	1.0	1.0	1.0	0.7	0	0.7	1.0	0.9	1.0	1.0	1.0	0.8	-
Case 3	1	1	1	-	1	1	1	1	1.0	1.0	0.8	-	1.0	0.8	1.0	1	1.0	1.0	0.9	-	1.0	0.9	1.0	1.0
Case 4	1	1	1	-	1	1	1	-	1.0	1.0	0.9	-	1.0	1.0	1.0	0	1.0	1.0	0.9	-	1.0	1.0	1.0	-
Case 5	-	1	1	-	0	-	1	1	-	1.0	1.0	-	-	-	1.0	1	-	1.0	1.0	-	-	-	1.0	1.0
Case 6	1	1	1	-	1	-	1	-	1.0	1.0	1.0	-	1.0	-	1.0	0	1.0	1.0	1.0	-	1.0	-	1.0	-
Case 7	-	1	1	-	-	-	1	-	-	1.0	1.0	-	-	-	1.0	0	-	1.0	1.0	-	-	-	1.0	-

Table 6: Precision, Recall and F1-Score for lines extraction in evaluation plans.

Table 7: Circuit assembly results generated for three floor plans

ID	floor plan name	GROUND TRUTH							RESULT							
		Circuit IDs	generalDoubleOutlet	specialDoubleOutlet	wallLightOutlet	mainBoardPanel	singleSwitch	combinedSwitch	Circuit IDs	generalDoubleOutlet	specialDoubleOutlet	wallLightOutlet	ceilingLightOutlet	mainBoardPanel	singleSwitch	combinedSwitch
1	IV-ROM-IE-PL-BT-29b-02-GT.jpg	1A	2		3		2		+1	✓		✓			✓	
2	IV-ROM-IE-PL-BT-29b-01-GT.jpg	2A	9						✓	✓						
		2B		2					✓		✓					
		2C			6		5	1	✓			✓			-1	✓
3	IV-FRA-IE-PL-11c-GT.jpg	3A			3	1	2		✓			✓		✓	✓	
		3B	2						✓	✓						
		3C	3						✓	+1						
		3D	1						✗	-1						
		3E			4		3		✓			✓			✓	
		3F		1					✓		✓					

7.3. BIM Test Cases Studies

To evaluate the performance of the proposed methodology under realistic BIM modeling conditions, a dedicated BIM test case was developed for each evaluation plan using a widely adopted architectural modeling platform[13]. For each case, the electrical circuits were manually modeled within the authoring environment, and the resulting BIM models were exported to IFC. This enabled a direct comparison between (i) the IFC generated by the method and (ii) the IFC exported from the authoring tool. Across all test cases, a high degree of consistency was observed in the elements most relevant to practical BIM workflows. Device-level entities—such as *IfcLightFixture*, *IfcSwitchingDevice*, and *IfcOutlet*—as well as their circuit assignments represented through *IfcSystem* and *IfcRelAssignsToGroup*, were reliably comparable between the IFC generated by the approach and the IFC obtained from the reference models. This confirms both the correctness of the component detection stage and the robustness of the circuit-grouping process implemented in the methodology. These findings are fully consistent with the results summarized in Table 7. Differences arose primarily in the handling of wiring information.

Two common modeling practices were observed in the authoring tool. In the first, circuits are defined only at a logical level (device–board assignments) without explicit modeling of wire paths; in such cases, both IFC outputs contain equivalent element types and relationships, leading to consistent comparability. In the second practice, wire paths are explicitly modeled; however, the native IFC exporter does not translate these wire elements into *IfcCableSegment* instances. As a result, wire routing cannot be compared, even though the corresponding device-to-board relations remain available. Although the exporter includes *IfcDistributionPort* objects, the relational links (*IfcRelConnectsPorts* or *IfcRelConnectsPortToElement*) are not generated. The absence of these relationships leaves ports unconnected and prevents an IFC-level verification of logical connectivity. Importantly, this limitation reflects the current behavior of the specific exporter rather than a restriction of the IFC schema itself, which fully supports this type of connectivity modeling. Overall, the BIM case studies show strong agreement between the elements and relationships exported to IFC by the authoring environment and those generated by the proposed methodology. The remaining

[13]Available at `https://bitbucket.org/plans2bim/mep2bim`

gaps—primarily involving wire segments and port-to-port connectivity—are attributable to current exporter implementations and are expected to diminish as IFC support in authoring tools continues to evolve. These limitations do not affect the validity of the proposed method and highlight, instead, the importance of ongoing efforts to strengthen interoperability between IFC and commercial BIM platforms.

8. Conclusions and Future Work

This paper addressed the challenge of processing electrical installation floorplans using machine learning to obtain a fast BIM model, in IFC format, facilitating informed decision-making. Instance segmentation models (such as Cascade Mask R-CNN, YOLOv8m-seg, and YOLOv11m-seg) were combined with the DeepLSD line segment detection model to infer electrical circuits and their components.

A key finding was the superior performance of YOLOv11m-seg at 1024x1024 resolution, demonstrating robust detection capabilities even with reduced datasets. A critical aspect highlighted in this research is that accurately detecting and assembling electrical circuits requires a careful adaptation of the logical relationships between components, alongside a thorough consideration of regional regulatory standards. As evidenced by the evaluation, these regulations, which typically vary by region, introduce complexities that necessitate flexible validation and adaptation within the methodology. Currently, these local rules and standards are incorporated directly into the processing script, requiring manual updates for different regions or evolving regulations. A significant future challenge, however, is to develop a machine learning model capable of automatically inferring and extracting these complex regulatory rules from diverse documentation, thus enabling more adaptive and scalable circuit generation across various geographical contexts.

Furthermore, the effectiveness of the approach was significantly bolstered by specific image preprocessing techniques (e.g., grayscale removal) and post-processing strategies (e.g., merging fragmented detections and filtering lines on features). These steps proved indispensable for generating a clean and coherent set of elements and lines for circuit assembly. In addition, the paper introduces IPVBA-Elec, a dataset specifically curated for electrical floorplans in Argentina.

This paper introduces the following contributions on the electrical floorplan processing:

- An approach to process electric floorplan and extract BIM models. The reproducible workflow allows to process technical hand-drawn or digital raster images into BIM models using IFC standard. It combines instance segmentation for electrical symbols detection, and DeepLSD for line detection.

- A new annotated electrical floorplan dataset so-called IPVBA-ELEC. This dataset will support research on electrical installation understanding. The dataset includes the curation, annotation protocol, class schema, splits, and license.

- Approach evaluation. Several evaluations were conducted to demonstrate the applicability of the approach using YOLOv8, YOLOv11, and Cascade Mask R-CNN models across different resolutions and dataset sizes. The report presents promising preliminary results.

These contributions can be considered as a baseline for any other pipe or wire-based technical floorplan like gas or plumbing installations.

Further work will include comparing different networks and hyperparameters to improve detection precision and by adding new post-processing techniques for circuit assembly.

The heterogeneity of graphical representation in electrical installations, resulting from the practice codes of each country, invites an analysis of the various variants and an extension of the approach to accommodate them. Even within a country, codes of practice vary over time as standards evolve. The use of synthetic datasets will be evaluated to enhance the training process according to different countries' codes of practice. In addition, the IPVBA-ELEC dataset is planned to be extended to include legacy Argentinean codes of practice.

References

[1] A. Borrmann, M. König, C. Koch, J. Beetz, Building information modeling: Why? what? how?, in: A. Borrmann, M. König, C. Koch, J. Beetz (Eds.), Building Information Modeling: Technology Foundations and Industry Practice, Springer International Publishing, Cham, 2018, pp. 1–24. doi:10.1007/978-3-319-92862-3_1.

[2] International Organization for Standardization (ISO), ISO 16739-1: 2024: Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries— Part 1: Data schema (2024).
URL https://www.iso.org/standard/84123.html

[3] C. Eastman, P. Teicholz, R. Sacks, K. Liston, BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors, John Wiley & Sons, 2008. doi:10.1002/9780470261309.

[4] C. U. R. (Firm), Collaboration, Integrated Information and the Project Lifecycle in Building Design, Construction and Operation, Construction Users Roundtable, 2004.
URL https://books.google.com.ar/books?id=xUbjtgAACAAJ

[5] M. Khaddaj, I. Srour, Using bim to retrofit existing buildings, Procedia Engineering 145 (2016) 1526–1533, iCSDEC 2016 – Integrating Data Science, Construction and Sustainability. doi:10.1016/j.proeng.2016.04.192.
URL https://www.sciencedirect.com/science/article/pii/S1877705816301990

[6] L. D'Angelo, M. Hajdukiewicz, F. Seri, M. M. Keane, A novel bim-based process workflow for building retrofit, Journal of Building Engineering 50 (2022) 104163. doi:10.1016/j.jobe.2022.104163.
URL https://www.sciencedirect.com/science/article/pii/S2352710222001760

[7] Y. Zhao, X. Deng, H. Lai, A deep learning-based method to detect components from scanned structural drawings for reconstructing 3d models, Applied Sciences (Switzerland) 10 (3 2020). doi:10.3390/app10062066.

[8] Y. Zhao, X. Deng, H. Lai, Reconstructing bim from 2d structural drawings for existing buildings, Automation in Construction 128 (2021) 103750. doi:10.1016/j.autcon.2021.103750.

[9] Z. Zeng, X. Li, Y. K. Yu, C.-W. Fu, Deep floor plan recognition using a multi-task network with room-boundary-guided attention, in: Proceedings of the IEEE/CVF International Conference on Computer Vision,

arXiv, 2019, pp. 9096–9104. doi:10.48550/ARXIV.1908.11025.
URL https://arxiv.org/abs/1908.11025

[10] J. Seo, H. Park, S. Choo, Inference of drawing elements and space usage on architectural drawings using semantic segmentation, Applied Sciences 10 (20) (2020). doi:10.3390/app10207347.
URL https://www.mdpi.com/2076-3417/10/20/7347

[11] C. Liu, J. Wu, P. Kohli, Y. Furukawa, Raster-to-vector: Revisiting floorplan transformation, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2214–2222. doi:10.1109/ICCV.2017.241.

[12] M. Urbieta, M. Urbieta, T. Laborde, G. Villarreal, G. Rossi, Generating bim model from structural and architectural plans using artificial intelligence, Journal of Building Engineering 78 (2023) 107672. doi:10.1016/j.jobe.2023.107672.
URL https://www.sciencedirect.com/science/article/pii/S2352710223018521

[13] Y. Wu, J. Shang, P. Chen, S. Zlatanova, X. Hu, Z. Zhou, Indoor mapping and modeling by parsing floor plan images, International Journal of Geographical Information Science 35 (6) (2021) 1205–1231. doi:10.1080/13658816.2020.1781130.

[14] T. Bloch, R. Sacks, Comparing machine learning and rule-based inferencing for semantic enrichment of bim models, Automation in Construction 91 (2018) 256–272. doi:10.1016/j.autcon.2018.03.018.

[15] P. Schönfelder, A. Aziz, F. Bosché, M. König, Enriching bim models with fire safety equipment using keypoint-based symbol detection in escape plans, Automation in Construction 162 (2024) 105382. doi:10.1016/j.autcon.2024.105382.
URL https://www.sciencedirect.com/science/article/pii/S0926580524001183

[16] C. Zhang, Y. Zou, J. Dimyadi, A systematic review of automated bim modelling for existing buildings from 2d documentation, in: ISARC. Proceedings of the International Symposium on Automation and

Robotics in Construction, Vol. 38, IAARC Publications, 2021, pp. 220–226. doi:10.22260/ISARC2021/0032.

[17] P. N. Pizarro, N. Hitschfeld, I. Sipiran, J. M. Saavedra, Automatic floor plan analysis and recognition, Automation in Construction 140 (2022) 104348. doi:10.1016/j.autcon.2022.104348.
URL https://www.sciencedirect.com/science/article/pii/S0926580522002217

[18] S. Dodge, J. Xu, B. Stenger, Parsing floor plan images, in: 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA), IEEE, 2017, pp. 358–361. doi:10.23919/MVA.2017.7986875.

[19] C. Liu, A. G. Schwing, K. Kundu, R. Urtasun, S. Fidler, Rent3d: Floorplan priors for monocular layout estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3413–3421. doi:10.1109/CVPR.2015.7298963.

[20] A. Kalervo, J. Ylioinas, M. Häikiö, A. Karhu, J. Kannala, Cubicasa5k: A dataset and an improved multi-task model for floorplan image analysis, in: Scandinavian Conference on Image Analysis, Springer, 2019, pp. 28–40. doi:10.1007/978-3-030-20205-7_3.

[21] L.-P. de las Heras, O. R. Terrades, S. Robles, G. Sánchez, Cvc-fp and sgt: a new database for structural floor plan analysis and its groundtruthing tool, International Journal on Document Analysis and Recognition (IJDAR) 18 (1) (2015) 15–30. doi:10.1007/s10032-014-0236-5.

[22] R. Luo, Z. Liu, T. Cheng, J. Wang, T. Wang, X. Wei, H. Wang, Y. Li, F. Chai, F. Cheng, S. Ye, W. Wang, Y. Zhang, Y. Qiao, H. Zhang, X. Zhao, Archcad-400k: An open large-scale architectural cad dataset and new baseline for panoptic symbol spotting (2025). arXiv:2503.22346, doi:10.48550/arXiv.2503.22346.

[23] B. Faltin, D. Gann, M. König, A comparative study of deep learning models for symbol detection in technical drawings, in: CONVR 2023 - Proceedings of the 23rd International Conference on Construction Applications of Virtual Reality: Managing the Digital Transformation of Construction Industry, 2023, pp. 877–886. doi:10.36253/979-12-215-0289-3.87.

[24] P. Schönfelder, F. Stebel, N. Andreou, M. König, Deep learning-based text detection and recognition on architectural floor plans, Automation in Construction 157 (2024) 105156. doi:10.1016/j.autcon.2023.105156.
URL https://www.sciencedirect.com/science/article/pii/S0926580523004168

[25] P. Schönfelder, M. König, Ontology-based reasoning in automatic floor plan analysis, Advanced Engineering Informatics 68 (2025) 103761. doi:10.1016/j.aei.2025.103761.
URL https://www.sciencedirect.com/science/article/pii/S1474034625006548

[26] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, Vol. 2015 International Conference on Computer Vision, ICCV 2015, 2015, pp. 1440–1448. doi:10.1109/ICCV.2015.169.

[27] P. N. Pizarro, N. Hitschfeld, I. Sipiran, Large-scale multi-unit floor plan dataset for architectural plan analysis and recognition, Automation in Construction 156 (2023) 105132. doi:10.1016/j.autcon.2023.105132.
URL https://www.sciencedirect.com/science/article/pii/S0926580523003928

[28] L. Kratochvila, G. de Jong, M. Arkesteijn, S. Bilik, T. Zemcik, K. Horak, J. S. Rellermeyer, Multi-unit floor plan recognition and reconstruction using improved semantic segmentation of raster-wise floor plans (2024). arXiv:2408.01526, doi:10.48550/arXiv.2408.01526.
URL https://arxiv.org/abs/2408.01526

[29] L. Gimenez, S. Robert, F. Suard, K. Zreik, Automatic reconstruction of 3d building models from scanned 2d floor plans, Automation in Construction 63 (2016) 48–56. doi:10.1016/j.autcon.2015.12.008.

[30] B. Bortoluzzi, I. Efremov, C. Medina, D. Sobieraj, J. McArthur, Automating the creation of building information models for existing buildings, Automation in Construction 105 (2019) 102838. doi:10.1016/j.autcon.2019.102838.

URL https://www.sciencedirect.com/science/article/pii/S0926580518308239

[31] M. Yin, L. Tang, T. Zhou, Y. Wen, R. Xu, W. Deng, Automatic layer classification method-based elevation recognition in architectural drawings for reconstruction of 3d bim models, Automation in Construction 113 (2020) 103082. doi:10.1016/j.autcon.2020.103082.
URL https://www.sciencedirect.com/science/article/pii/S0926580519303735

[32] B. Yang, B. Liu, D. Zhu, B. Zhang, Z. Wang, K. Lei, Semiautomatic structural bim-model generation methodology using cad construction drawings, Journal of Computing in Civil Engineering 34 (3) (2020) 04020006. doi:10.1061/(ASCE)CP.1943-5487.0000885.

[33] S. Macé, H. Locteau, E. Valveny, S. Tabbone, A system to detect rooms in architectural floor plan images, in: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, DAS '10, Association for Computing Machinery, New York, NY, USA, 2010, p. 167–174. doi:10.1145/1815330.1815352.

[34] X. Lin, Y. Zhou, Y. Liu, C. Zhu, A comprehensive review of image line segment detection and description: Taxonomies, comparisons, and challenges (2024). arXiv:2305.00264, doi:10.48550/arXiv.2305.00264.

[35] Q. Meng, J. Zhang, Q. Hu, X. He, J. Yu, Lgnn: A context-aware line segment detector, in: Proceedings of the 28th ACM International Conference on Multimedia, MM '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 4364–4372. doi:10.1145/3394171.3413784.

[36] S. Huang, F. Qin, P. Xiong, N. Ding, Y. He, X. Liu, Tp-lsd: Tri-points based line segment detector (2020). arXiv:2009.05505, doi:10.48550/arXiv.2009.05505.

[37] K. Zhao, Q. Han, C.-B. Zhang, J. Xu, M.-M. Cheng, Deep hough transform for semantic line detection, IEEE Transactions on Pattern Analysis and Machine Intelligence (2021) 1–1doi:10.1109/tpami.2021.3077129.

[38] P. V. Hough, Method and means for recognizing complex patterns, uS Patent 3,069,654 (Dec. 18 1962).

[39] R. Pautrat, D. Barath, V. Larsson, M. R. Oswald, M. Pollefeys, Deeplsd: Line segment detection and refinement with deep image gradients (2023). arXiv:2212.07766, doi:10.48550/arXiv.2212.07766.

[40] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, G. Randall, Lsd: a line segment detector, Image Process. Line 2 (2012) 35–55. doi:10.5201/ipol.2012.gjmr-lsd.
URL https://api.semanticscholar.org/CorpusID:5847178

[41] British Standards Institution, Graphical symbols for diagrams part 11: Architectural and topographical installation plans and diagrams, British standard, British Standards Institution, London, UK (1997).
URL https://knowledge.bsigroup.com/products/graphical-symbols-for-diagrams-symbol-elements-qualifying-symbols-and-other-s

[42] A. E. Argentina, Reglamentación para la ejecución de instalaciones eléctricas en inmuebles aea 90364 - parte 7: Reglas particulares para las instalaciones en lugares y locales especiales - sección 770: Viviendas (unifamiliares hasta 63 a; clasificaciones ba2 y bd1), Standard, Asociacion Electrotecnica Argentina (2017).
URL https://aea.org.ar/wp-content/uploads/2017/10/90364-7-770-1.pdf

[43] Z. Cai, N. Vasconcelos, Cascade r-cnn: High quality object detection and instance segmentation (2019). arXiv:1906.09756, doi:10.48550/arXiv.1906.09756.
URL https://arxiv.org/abs/1906.09756

[44] Z. Cai, N. Vasconcelos, Cascade r-cnn: Delving into high quality object detection, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 6154–6162. doi:10.1109/CVPR.2018.00644.

[45] G. Jocher, J. Qiu, A. Chaurasia, Ultralytics YOLO (Jan. 2023).
URL https://github.com/ultralytics/ultralytics

[46] K. Wada, mpitid, M. Buijs, Z. C. N., B. M. Kubovčík, wkentaro/labelme: v4.6.0 (nov 2021). doi:10.5281/zenodo.5711226.

[47] S. Vilgertshofer, D. Stoitchkov, A. Borrmann, A. Menter, C. Genc, Recognising railway infrastructure elements in videos and drawings using neural networks, Proceedings of the Institution of Civil Engineers - Smart Infrastructure and Construction 172 (1) (2019) 19–33. doi:10.1680/jsmic.19.00017.

[48] T. Reddy, N. Kumar, N. Ponnappa, N. Mohana, P. Vinod, M. Herbert, S. Rao, Intelligent gd&t symbol detection in mechanical drawings: a comparative study of yolov11, faster r-cnn, and retinanet for quality assurance, Journal of Intelligent Manufacturing (2025) 1–19doi:10.1007/s10845-025-02669-3.

[49] L. Jamieson, E. Elyan, C. F. Moreno-García, Few-shot symbol detection in engineering drawings, Applied Artificial Intelligence 38 (1) (2024) 2406712. doi:10.1080/08839514.2024.2406712.

[50] T. Schlagenhauf, M. Netzer, J. Hillinger, Text detection on technical drawings for the digitization of brown-field processes, Procedia CIRP 118 (2023) 372–377, 16th CIRP Conference on Intelligent Computation in Manufacturing Engineering. doi:10.1016/j.procir.2023.06.064.
URL https://www.sciencedirect.com/science/article/pii/S2212827123002883

[51] H. Bhanbhro, Y. K. Hooi, Z. Hassan, Modern approaches towards object detection of complex engineering drawings, in: 2022 International Conference on Digital Transformation and Intelligence (ICDI), IEEE, 2022, pp. 01–06. doi:10.1109/ICDI57181.2022.10007400.

[52] BIMForum, Level of development (lod) specification part i & comentary (2020).
URL https://bimforum.org/wp-content/uploads/2022/06/BIMForum_LOD-Spec-2020.zip

[53] P. Uusitalo, O. Seppänen, E. Lappalainen, A. Peltokorpi, H. Olivieri, Applying level of detail in a bim-based project: An overall process for lean design management, Buildings 9 (5) (2019). doi:10.3390/buildings9050109.
URL https://www.mdpi.com/2075-5309/9/5/109

[54] I. Suárez, J. M. Buenaposada, L. Baumela, Elsed: Enhanced line segment drawing, Pattern Recognition 127 (2022) 108619. doi:10.1016/j.patcog.2022.108619.
URL https://www.sciencedirect.com/science/article/pii/S0031320322001005

Appendix A. Troubleshooting guide

In this section presents a troubleshooting guide that summarizes common issues and corresponding recommendations.

Table A.8: Main Challenges and Recommendations Across Stages

Stage	Challenge	Impact	Recommendations
Floorplan raw image	Poor image quality	High	Poor detection performance. Suggest enhancing pre-processing (denoising, contrast).
	Background lines	Medium	Apply filters to remove gray lines/features.
	Circuit color code trace	Medium	Apply color filters to generate per-color images for line detection.
	Elevation extraction from 2D floorplans	Low	Elevation can be set by element or device type.
Dataset / Training	Target electrical symbols missing	High	Enrich dataset to include the expected object clases.
	Color-coded symbol sensitivity	Low	Deactivate color augmentation during training.
	Few class instances to train	Medium	Implement synthetic dataset augmentation to get more samples for the class.
Object detection	Fine-tuning / benchmarking	Medium	Missing components or misclassification. Suggest implement SOTA models & retrain.
	Feature location errors	Low	Incorrect component position. Place as terminal in line endpoint.
	Feature segregation	Low	Merge close fragments into one.
	Feature size mismatch	None	The dimension is not considered in the outcome model .
Line detection & Postprocessing	Low readability because high line density	Medium	Suggest pre-pruning. and filters to remove unnecessary elements
	Line style (continuous, dashed, etc.)	Medium	Circuit assembly requires tolerance setup.
	Excessive line width (pixels vs vectors)	Low	Missed endpoints ¡ wall width; merging parallel lines.
	Many parallel lines	Low–Medium	Apply merging tolerance to get one single line
	Line overlaps	Low	Apply merging tolerance to unify lines.
	Objects occluding lines (symbols, text, hatches)	Low–Medium	Tweak connectivity tolerance parameter or implement new strategies.
	Undetected segments gaps	Low–Medium	"Apply endpoint joining tolerances. Switch to SOTA line detection models."
Circuit graph & IFC generation	Fragmented circuit traces	Critical	Incorrect topology. Suggest stricter continuity enforcement.
	Zigzag trace segments	Very Low	Loss of line straightness. Minor visual impact.
	Gaps between objects and lines	Low	Apply distance-link tolerance.
	Node identification errors	Critical	Incorrect topology (tees, crosses). Suggest rule-based validation.
	Missing features	Critical	Branch discarded in topological graph.
	Orphan elements	Critical	Excluded from topological graph. Suggest error-logging for review.
	Electrical standards validation	Critical	Invalid circuits. Suggest rule-checker integration.

Table A.8: (continued)

Stage	Main challenges	Error propagation impact	Recommendations
	Geometry placement errors	Medium	Wrong length calculations, rendering errors.
	Wires modeled independently (no conduit sharing)	Low	Missing conduit elements. Suggest conduit grouping logic.
	Circuit graph errors	Critical	Incorrect IFC logic model. Suggest consistency check before export.

Highlights

AI-driven extraction of electrical circuits from floorplans for BIM

Urbieta, Martin, Urbieta, Matias, Burriel Guillermo

- Introduction of IPVBA-ELEC, a dataset of electrical floorplans from social housing projects.
- AI-based extraction of electrical components and circuit traces to generate an IFC BIM model.

Graphical Abstract

AI-driven extraction of electrical circuits from floorplans for BIM

Urbieta, Martin, Urbieta, Matias, Burriel Guillermo