

Detección de circuitos eléctricos en planos de planta mediante Aprendizaje Automático

Guillermo Burriel¹, Martin Urbietta^{1,2}[0000-0001-5824-089X], and Matias Urbietta^{1,2}[0000-0002-4508-1209]

¹ Centro de investigación LIFIA, Facultad de Informática, Universidad Nacional de La Plata

{gburriel,martin.urbieta, matias.urbieta}@lifia.info.unlp.edu.ar

² Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)

Resumen Building Information Modeling (BIM) es una metodología para la gestión del ciclo de vida de edificios. La mayoría de las soluciones BIM requieren un modelo digital como base, para por ejemplo, optimizar el mantenimiento, renovación de infraestructura o su demolición. Sin embargo, estos modelos no están siempre disponibles para edificios existentes. Existe una vasta cantidad de planos analógicos de edificios existentes registrados por los diferentes organismos públicos que gestionan los desarrollos urbanos, y que su modelado manual resultaría muy costoso. Dada esta falta de modelos digitales, se presenta una nueva técnica que permite procesar información en planos de instalación eléctrica y obtener un archivo IFC, basada en Aprendizaje Automático utilizando el modelo de segmentación de instancias Cascade Mask R-CNN para extraer tomas y un modelo detección de segmentos de líneas (LSD) DeepLSD para extraer la traza de los cables que captura la información digitalmente, estableciendo la relación semántica de los componentes del circuito. Los planos eléctricos procesados conforman un nuevo dataset IPVBA-ELEC.

Keywords: Building Information Modeling · BIM · construction · IFC · electrical floorplan · Machine Learning

1. Introducción

El Modelado de la Información en Construcción, también conocido como BIM (por su acrónimo en inglés de Building Information Modeling) es una metodología en auge en la industria de la construcción y el manejo digital de la información. Es utilizada a lo largo de todo el ciclo de vida de un proyecto, desde la planificación y el diseño hasta la construcción, operaciones y mantenimiento. Representa una oportunidad para mejorar la eficiencia energética, la flexibilidad y la resiliencia de los edificios existentes. Industry Foundation Classes (IFC) es un esquema de intercambio de datos para el Diseño Asistido por Computadora (CAD), basado en BIM, utilizado para describir información en la arquitectura, la construcción y la industria de la edificación (Architecture, Engineering and Construction, conocida por su acrónimo en inglés AEC). Es un estándar

internacional abierto, pensado para ser neutral o independiente del proveedor, y utilizable en una amplia gama de dispositivos de hardware, plataformas de software e interfaces para muchos casos de uso diferentes".

A pesar de las numerosas ventajas de realizar el diseño, la construcción y la operación de manera digital y basada en BIM, los modelos aún no están disponibles para muchos edificios, especialmente aquellos construidos antes del surgimiento de esta metodología.

Existe una necesidad técnica de BIM en proyectos de modernización energética para la etapa previa al modelado energético [1]. Si se encuentran disponibles modelos BIM conforme a obra (as-built en inglés) de construcciones existentes, se pueden incorporar efectivamente en un flujo de trabajo para la modernización de edificios [2]. Un enfoque que reduzca el costo de implementación mediante el uso de la automatización aumentará los beneficios del enfoque BIM. Las técnicas de Aprendizaje Automático (ML por su acrónimo en inglés de Machine Learning) han demostrado ser de gran utilidad para la extracción de información de planos de edificios existentes para la generación automática y semi-automática de modelos BIM [3–9] o el enriquecimiento de modelos existentes con información faltante [10, 11]. Para poder entrenar el modelo se requiere contar con un conjunto de datos (datasets en inglés). A pesar del extenso uso de ML se identifican limitaciones y lagunas, como la falta de datasets etiquetados para entrenar los modelos o la falta de transferibilidad del modelo. Siguiendo con las limitaciones, los beneficios del aprendizaje automático pueden no justificar los costos de su implementación y el rendimiento podría no ser confiable y sólido para los objetivos establecidos, ya que el método podría funcionar para algunos edificios pero no podría generalizarse a otros [12]. El objetivo de este trabajo es generar el modelado BIM de componentes eléctricos y sus relaciones semánticas extraídas de los planos de instalaciones eléctricas en planos de planta implementando ML. Para ello, se creó un nuevo dataset de plantas eléctricas que hemos denominado IPVBA-ELEC. La contribución es compatible con la extracción de detalles de arquitectura y estructura introducido en [8] para poder obtener un modelo BIM completo.

Este enfoque, se limita a detectar circuitos/cables, tomacorrientes, bocas de iluminación, interruptores y tableros generales de dichos circuitos. Por su naturaleza, hemos optado por un modelo para detectar los cables, y otro modelo ML para la detección de elementos eléctricos. Adicionalmente, se elaboró un script a efecto de vincular los circuitos a las tomas de energía o iluminación de tal forma de poder representar los puntos de consumo en cada uno. Se excluye de este trabajo el procesamiento de documentación no gráfica tal como los cuadros de potencia, y diagrama unifilar. Si bien existen en algunos datasets etiquetados de artefactos eléctricos (lavarropas, termotanques, TVs, etc) en plantas de arquitectura, no brindan suficiente información para poder establecer los circuitos ni sus puntos de consumo dimensionados.

Este trabajo se encuentra estructurado de la siguiente forma: En la sección 2 presentamos un resumen del estado del arte en el tema. En la Sección 3 se presenta la contribución de este trabajo: el enfoque. En la Sección 4 se describe

cómo se entrenó una red neuronal para la detección de elementos. En la Sección 5 se introduce la solución a la detección de cables. Posteriormente, la Sección 6 menciona como generar un modelo IFC a partir de los datos capturados. En la Sección 7 se presenta una evaluación de resultados. Finalmente la Sección 8 presentan las conclusiones.

2. Estado del arte

2.1. Detección de elementos en planos AEC y Datasets

La mayor parte de las implementaciones de ML han sido sobre datasets arquitectónicos, con foco en el modelado geométrico sin abordar los planos eléctricos. En el esfuerzo de modelado conforme a obra de edificios e infraestructuras utilizando IA para generar gemelos digitales, el gobierno Alemán apoya el proyecto BIMKIT³, que abordan la detección de símbolos técnicos con modelos de aprendizaje profundo (Deep Learning, DL) [13], detección de texto en plantas arquitectónicas con DL [14], extracción de equipamiento contra incendio usando detección de símbolos en planos de escape basado en puntos-clave (keypoint-based). Este enfoque permite integrar diferentes modelos ML con entrenamiento específico y prestar servicios de análisis de documentación. Sin embargo las instalaciones eléctricas aún no han sido abordadas. En la investigación previa de Urbietta et al. [8] se presenta un enfoque novedoso para generar un modelo BIM IFC a partir de planos de arquitectura y estructura, implementando Mask R-CNN [15]. Avanzando en la misma línea de investigación resulta necesario poder incorporar los planos de instalaciones eléctricas, mecánicas y sanitarias (Mechanical, Electrical and Plumbing, sus siglas en inglés MEP) que permitan incorporar el modelado de los componentes no geométricos que han recibido menor atención por parte de los investigadores [16]. Existen variadas experiencias implementando diferentes modelos de detección, como YOLO [3], FCN [17], Faster R-CNN [4, 17] y DeeplabV3+ [6]. Existen experiencias de modelado BIM previo al auge de las técnicas de ML, como el prototipo presentado por Gimenez et al. [18] desarrollado en C++ para la generación de un modelo IFC 3D a partir de planos escaneados en 2D, los cuales incluyen muro, aberturas y espacios. El principal inconveniente de este enfoque es que requiere realizar ajustes del algoritmo para adaptarlo a diferentes estilos arquitectónicos o ajustarlo para otros objetivos como el modelado MEP. Hay antecedentes de generación de modelos BIM a partir de archivos creados mediante Diseño Asistido por Ordenador (CAD acrónimo en inglés de Computer Aided Design), procesando los elementos vectoriales de archivos CAD y combinando con el complemento Dynamo en Revit generando modelo BIM arquitectónico [19, 20] y estructural [21]. Estas técnicas adquieren un carácter semi-automático cuando requieren el preprocesado por parte de un técnico del archivo CAD.

³ <https://bimkit.eu/en/welcome-to-bimkit> Accedido el 08/04/2024

2.2. Detección de líneas utilizando aprendizaje automático

El abordaje de la identificación y extracción de elementos de planos que requieren la identificación de líneas ha sido ampliamente estudiado con diferentes metodologías. Por ejemplo, Macé et al. [22] utiliza un algoritmo de la Transformada de Hough y vectorización de imagen. En esta revisión nos enfocaremos en métodos de Aprendizaje Automático para la detección de líneas, técnica de Visión por Computador (CV) utilizada en el campo de la reconstrucción 3D y segmentación.

En su investigación Meng et al. [23] presenta un novedoso esquema de detección de segmentos de línea en tiempo real llamado Line Graph Neural Network (LGNN) que utiliza dos módulos, una Red neuronal convolucional profunda (DCNN por sus siglas en inglés) para generar posiciones y características de segmentos de línea y una red neuronal gráfica para razonar sus conectividades. En lugar de la tradicional descripción de la línea como Cruce - Cruce (Junction - Junction), proponen una representación cuádruple: cruce inicio, cruce fin, punto central de la línea y valor de desplazamiento de la línea. La DCNN se propone predecir un mapa de calor de punto central de línea junto con un mapa vectorial de desplazamiento de línea. Y observaron que, para escenas desordenadas, los segmentos de línea predichos están menos fragmentados, donde pueden asignar de manera confiable sus puntos finales a los cruces o uniones. Luego, el módulo GNN toma estos candidatos a segmentos de línea como vértices y construye un gráfico disperso para imponer restricciones estructurales.

Para la detección de líneas o cables también se emplean la detección de puntos claves (key-points en inglés). Por ejemplo Zhiyong et al. [24] propone utilizar puntos claves en lugar de cajas (bounding boxes en inglés) de una sola vez, aplicando una política de detección de cinco puntos, que son aproximados por una polinómica. Esta polinómica permite posteriormente aumentar las imágenes etiquetadas, seleccionando otros nuevos cinco puntos en la polinómica.

En forma similar, Huang et. [25] al presenta un detector de segmento de líneas basados en tres puntos (TP-LSD por su acrónimo en inglés), que esencialmente a partir de un punto de la línea lo extiende en direcciones opuestas hasta determinar los extremos, obteniendo tres *keypoints*.

Zhao [26] extiende el modelo planteado por Hough [27]. Detecta formas en una imagen, como pueden ser elipses y círculos a partir de los ángulos y radios de la misma. Una vez detectados los elementos, genera líneas para indicar delimitaciones. Esto podría servir para indicar, una vez detectado un elemento, donde puede haber un cable. Pero no se puede utilizar ya que no detecta los puntos finales de las líneas, sino que detecta elementos y a partir de ellos las genera. En la tarea a desarrollar en este trabajo, los cables tienen un comienzo y un final marcado, además de que no necesariamente van a ser siempre en el mismo sentido. En este trabajo, hemos optado por DeepLSD [28] que implementa LSD [29], un campo de atracción y una herramienta de optimización.

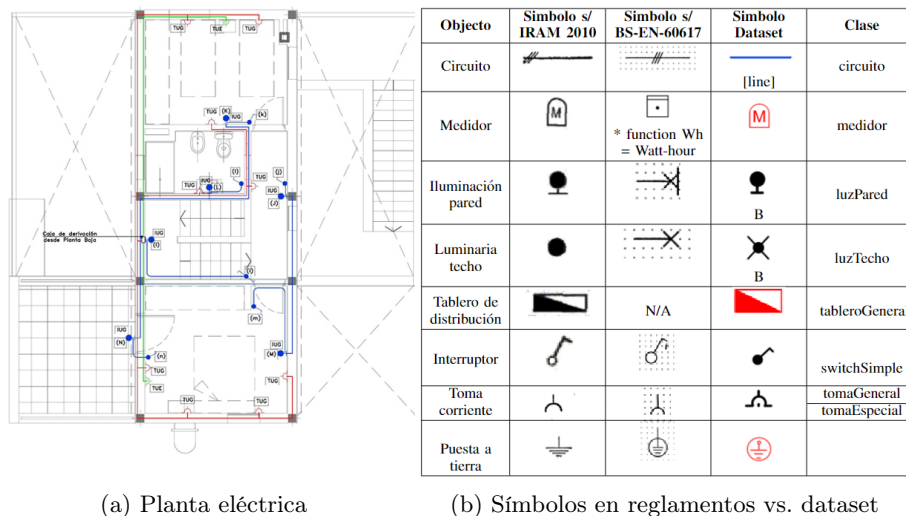


Figura 1: Planos eléctrico y referencias

3. Enfoque

El enfoque consiste en detectar elementos correspondientes a instalaciones eléctricas tal como toma corriente, de iluminación, y cables de circuito eléctrico en planos para luego generar un archivo IFC. El modelo obtenido podría ser utilizado tanto para generar un nuevo documento IFC o enriquecer uno ya existente. Para ello, en primer lugar se detectan los elementos del sistema de una imagen de plano con un detector de segmentación de instancias, y luego sobre el mismo plano se detectan las distintas trazas líneas. Solo algunas de estas líneas se corresponden a cables de circuitos eléctricos coherentes con los elementos que van vinculándose para dar energía. En este último caso, la dificultad radica en inferir correctamente cuales trazos de líneas en un plano conectan diferentes elementos eléctricos. Es decir, encontrar la traza de circuitos de cables y sus componentes. El proceso completo se presenta en la Figura 2.

En una primera etapa se procedió a entrenar un detector de segmentación de instancias con el objetivo de detectar las diferentes clases de elementos del circuito. Para ello se creó un dataset específico de planos eléctricos y se procedió a su etiquetado supervisado por un ingeniero. Con los planos etiquetados se generaron subimágenes de 512x512, separando un 80 %-20 % Se utilizó el modelo Cascade Mask R-CNN pre entrenado con el dataset COCO, y aplicando transfer learning, se realizó un entrenamiento adicional específico con nuestro dataset.

Para la segunda etapa, se integra el modelo anterior con un modelo detector de líneas, con el objetivo de crear un modelo IFC. Para la detección de cables, a diferencia de la etapa anterior, no fue necesario aplicar *transfer learning*, implementándose directamente el modelo pre-entrenado Deep LSD para detectar

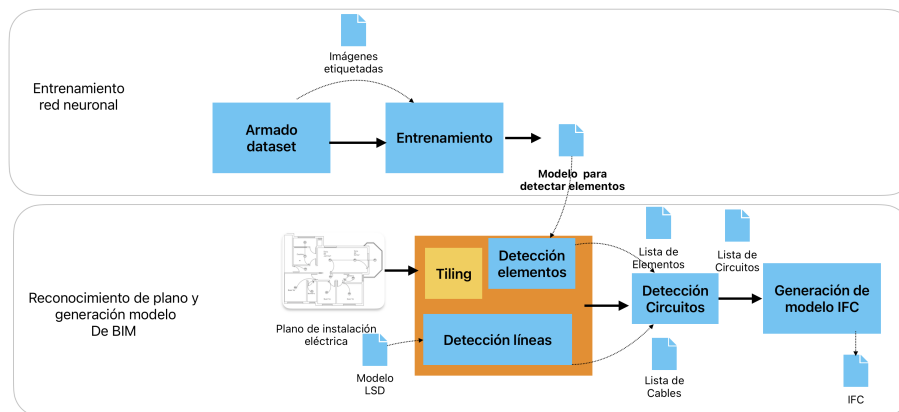


Figura 2: Esquema de la metodología desarrollada

líneas en imágenes de planos. Dado un plano de planta eléctrica de gran resolución, se corre el detector de segmentación de instancias, en conjunto con una librería que permite la subdivisión de grandes imágenes en parches, e inferir en cada subimagen de instancia. Se obtiene como resultado un objeto JSON de intercambio que contiene cada clase de elemento y su ubicación. Posteriormente, se ejecuta el modelo LSD y obtener identificadas las líneas en el plano, el siguiente paso es ver cuáles de esas líneas son un cable ya que algunas líneas corresponden a detalles de arquitectura u otros elementos de la imagen. Esto se logra a través de un post procesamiento que se verá en detalle en un capítulo posterior. Por último, una rutina toma la lista de componentes ya procesados, y genera un modelo IFC.

4. Entrenando un modelo detector de elementos eléctricos

En esta sección presentaremos brevemente las consideraciones efectuadas para entrenar el modelo de red neuronal que sea capaz de identificar elementos específicos de una red de instalación eléctrica. Los pasos son bien conocidos y consisten en: confección de un dataset, etiquetado de elementos en las imágenes, entrenamiento de la red y verificación.

4.1. Dataset

Origen de datos El Instituto de la Vivienda de la Provincia de Buenos Aires (IPVBA) implementa políticas públicas para garantizar el acceso a la vivienda y a un hábitat justo a las familias bonaerenses⁴.

⁴ IPVBA - Planos licitatorios <https://www.gba.gov.ar/vivienda/licitaciones> Accedido 20 septiembre 2022

Preprocesado de planos para generar el dataset Lo primero es obtener imágenes de planos de planta de instalaciones eléctricas. Del conjunto de planos, se procesaron 24 planos, que consisten en PDFs donde se puede ver la imagen del plano junto a sus referencias, como se observa en Figura 1a. Se etiquetaron las siguientes clases con sus respectivas cantidad de ocurrencias en los planos en paréntesis: bocaDosTomasGeneral(620), bocaIluminacionPared(449), bocaIluminacionTecho(282), switchSimple(261), bocaDosTomasEspecial(169), switchDoble(83), tableroGeneral(44), pulsador(36), switchTriple(27), switchCombinada(13), cajaDerivacion(9), medidor(6), switchTetra(6).

Para maximizar la eficiencia, las redes neuronales se entrenan para procesar imágenes de tamaño reducido, típicamente de unos cientos de píxeles en alto y ancho, mientras que las imágenes de planos suelen tener una resolución significativamente mayor, alcanzando miles de píxeles en ambas dimensiones. Para evitar la pérdida de precisión al reducir la resolución de las imágenes de entrada al tamaño de la red, implementamos una técnica que preserva la calidad de los planos tanto como sea posible. La solución consiste en dividir el plano en varios parches de imágenes del tamaño de la red, un proceso conocido como "patching", con zonas de superposición. Cada parche es procesado por la red, lo que permite detectar elementos con una precisión óptima en cada uno. Posteriormente, se componen las detecciones de los parches en la imagen original de mayor resolución, asegurando así que la precisión de la detección se mantenga en todo el plano original de mayor resolución.

Una vez que la imagen se ha dividido en parches, el modelo, puede procesar cada parche de manera individual. Esto permite que el modelo maneje imágenes grandes o de alta resolución de manera más eficiente, ya que solo necesita procesar una fracción de la imagen a la vez, reduciendo así los requisitos de memoria y cómputo.

Después de que el modelo ha procesado cada parche, se pueden fusionar las predicciones de cada uno para obtener la salida final del modelo para la imagen completa. Este proceso de fusión puede implicar la combinación de las predicciones de los parches mediante técnicas como la interpolación o el solapamiento de máscaras para evitar discontinuidades en las regiones de superposición, entre otros métodos.

La resolución de los planos de instalaciones varía entre 6000 y 7500 píxeles de ancho con 3500 a 4700 píxeles de alto. Se generó un dataset para entrenamiento con un subconjunto de imágenes de 512x512 teniendo en consideración experiencias de patching anteriores [13].

Para la inferencia de parches, adoptamos el enfoque presentado por Akyon [30], y se utiliza la librería de SAHI (acrónimo en inglés de Slicing Aided Hyper Inference) que ofrece soporte para diversos modelos, que permiten inferir las detecciones de imágenes de gran resolución.

4.2. Entrenamiento

Se utilizó Cascade Mask R-CNN como detector de las clases etiquetadas en el dataset. Para el desarrollo de este trabajo se utilizó la implementación de

la librería MMDetection ⁵ basada en PyTorch y tensorpack basada en Tensorflow. Específicamente, se seleccionó el modelo combinado con una arquitectura en cascada con una red backbone ResNet-50 y una Feature Pyramid Network, entrenado en el conjunto de datos COCO.⁶ Aplicando la técnica de transfer learning, el modelo se volvió a entrenar con el dataset preparado.

Para esta etapa de entrenamiento, utilizó una placa de video NVIDIA GeForce RTX 4070 Ti. La duración del entrenamiento en modo GPU fue de 12h, sobre un dataset aumentado, en un conjunto de sub-imagenes de 512x512, 80%-20% entrenamiento y validación.

5. Detección de cables y circuitos

Para la detección de cables, como ya se mencionó en la Sección 3, se utilizó el modelo Deep LSD, un modelo pre-entrenado para detectar líneas. En unas primeras pruebas se corroboró que este modelo detectaba líneas adecuadamente en los planos, y la tarea en esta parte, a diferencia de la detección de elementos, consistió en realizar un post procesamiento a las líneas detectadas por el modelo para ver cuales de ellas corresponden a cables. Luego, a partir de estas líneas detectadas hay que lograr identificar circuitos eléctricos. A continuación se describen en detalle los pasos necesarios para realizar estas tareas.

- El primer paso consiste en, previo a que el modelo detecte las líneas en el plano, realizar un pre procesamiento de la imagen para obtener una versión más limpia del mismo donde se encuentren solo los detalles que interesan para la detección de circuitos. Por ejemplo en la Figura 3a se presenta un plano tipo y en la Figura 3b el mismo plano donde se han eliminado los grises y/o los negros del plano. Gracias a obtener una versión mucho más limpia del plano, como se puede observar en la Figura 3c, al utilizar el modelo DeepLSD, se obtendrán líneas de interés y se logrará una reducción del tiempo de ejecución de todo el algoritmo de detección de circuitos ya que se trabajará con menos líneas.
- Se post procesaron las clases de elementos detectadas por el modelo de segmentación de instancias, donde se combinaron clases muy próximas, porque en algunos casos, un mismo elemento estaba subdividido en varias instancias adyacentes e incluso de tamaño reducido. También se validaron solo aquellas clases inferidas con un confianza de 0.9 o superior.
- El siguiente paso es utilizar el modelo DeepLSD para detectar líneas en el plano. A pesar de haber limpiado el plano en el paso anterior, el modelo detectará algunas líneas que no son de interés, por ejemplo las propias líneas que componen un elemento las cuales habrá que eliminarlas, así como también detectará líneas superpuestas que habrá que unir las para que se consideren una sola y posteriormente no tener problemas a la hora de armar los circuitos. Esto implica que, una vez detectadas todas las líneas del

⁵ <https://github.com/open-mmlab/mmdetection>

⁶ `cascade_mask_rcnn_r50_fpn_1x_coco_20200203-9d4dcb24.pth`

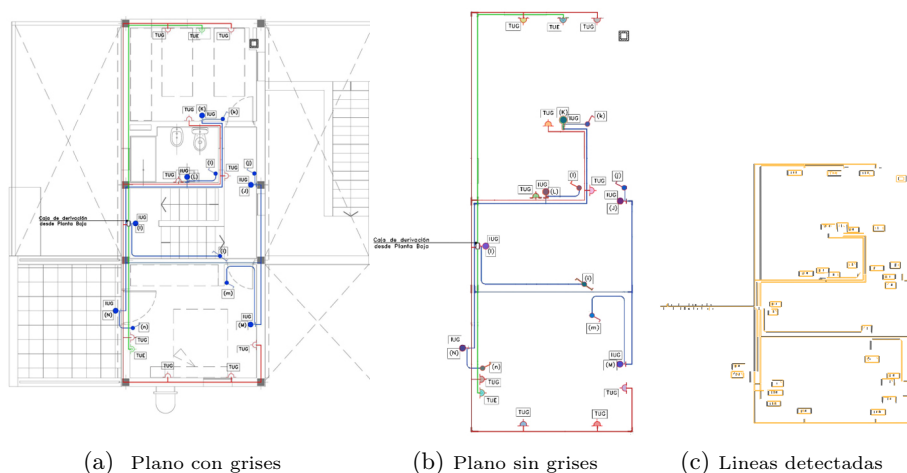


Figura 3: Mismo plano con y sin filtrado de grises | Lineas detectadas

plano, habrá que realizar un post procesamiento, como se puede observar en la Figura 3c para mejorar aún más los resultados que serán de crucial importancia para luego elaborar los circuitos.

- Una vez que se tienen listas las líneas, lo que se hace es, a partir de cada elemento detectado en la primera parte de este trabajo, crear un grafo donde se irán agregando las líneas que representan a los cables siguiendo diversas funciones que provee la librería de Python `shapely`⁷ para verificar las intersecciones y superposiciones, entre otras cosas. De esta forma, se logrará conectar los distintos elementos de un mismo circuito, a partir de agregar todas las líneas que los unen. Luego de una serie de iteraciones, que se harán hasta que todos los elementos detectados se hayan asignado a su circuito correspondiente, se tendrá un grafo por cada circuito presente en el plano. La reglamentación vigente en Argentina, establece el diferentes circuitos según su uso. En consecuencia se implementó esta regla, de forma tal que, por ejemplo, un circuito de toma corrientes no puede contener iluminación y viceversa. Crear estos circuitos lleva mucho procesamiento, ya que en cada iteración se agregan nuevas líneas que pueden llevar a elementos que, bajo criterios establecidos, se llega a la conclusión de que no son parte del circuito. Para poder solucionar esto es que se trabajó con la librería `networkx`⁸ de Python que permite utilizar grafos. Estos son de gran utilidad ya que permiten, a partir de la evaluación de sus nodos y aristas, hacer un camino inverso, para así eliminar las líneas en caso de haber llegado a un elemento que pertenece al circuito que se está detectando.

⁷ <http://shapely.readthedocs.io/>

⁸ <http://networkx.org/>

6. Generación del modelo IFC

Para generar un modelo BIM se utiliza el lenguaje de programación Python junto con la librería IfcOpenShell ⁹ para producir documentos intercambiables. La librería código abierto (LGPL) IfcOpenShell es utilizada por el complemento BlenderBIM que está diseñado para ser una plataforma OpenSource para la creación de BIM completa. De ahora en adelante, nos referiremos al modelo BIM y el archivo IFC para la representación de la salida de nuestro proceso. Como fue mencionado, los resultados obtenidos luego de identificar en el plano eléctrico los objetos, se genera un archivo JSON con los elementos detectados como una colección del tipo diccionario del tipo clave-valor para definir la clase del elemento, los pares de puntos definidos como píxeles X,Y del polígono de forma, y la relación con otros elementos. Un circuito, puede relacionarse con varias tomas, pero una toma no puede relacionarse con varios circuitos. El modelado de los elementos, se limita a los elementos detectados, en continuidad con la investigación previa [8] que complementa el modelado de otros elementos como muros, paredes, puertas, ventanas, vigas y columnas. Los elementos detectados carecen de altura, que puede ser extraída los planos de corte respectivos.

7. Evaluación

Con el propósito de evaluar la calidad de las detección de elementos de los planos de instalaciones eléctrica y los circuitos eléctricos, procedimos a realizar una comparación sobre los elementos reales presentes en los tres planos de control contra la predicción obtenida por las redes neuronales descriptas anteriormente reportados en el Cuadro 1 . En primer lugar, identificamos los elementos reales en de 13 categorías en planos. El cuadro cuenta con las siguientes columnas: un numero de plano (Columna *ID*), nombre del archivo (Columna *Nombre de Plano*), la cantidad de elementos reales por clase identificados por un profesional agrupados en la columna *REAL*, y la cantidad de elementos inferidos por la solución agrupados en la columna *RESULTADO* . Dado que cada plano puede contar con circuitos eléctricos independientes, que vinculan las clases de los elementos eléctricos, en la tabla se puede apreciar la cantidad de elementos por clase que existen para cada par *ARCHIVO-CIRCUITO*. Por ejemplo, el circuito B del plano número 2 cuenta con solo 2 elementos de tipo *bocaDosTomasGenerales*.

En segundo lugar, se procedió a evaluar los planos utilizando las redes neuronales descriptas en secciones anteriores obteniendo diferentes predicciones para un *score* mayor a 0.9 para cada inferencia correspondiente a las clases de elementos. La evaluación de la sensibilidad de la predicción se llevo a cabo utilizando tres métricas bien conocidas: Precision, Recall y F1-Score, donde sus rangos de valores es de 0 a 1. Los resultados se pueden apreciar en el Cuadro 2 donde podemos notar que la predicción fue satisfactoria. Sin embargo, en dos casos se detectaron *pulsador* en condición de Falso Positivo (FP), invadiendo la región de otros dos elementos detectados, situación que impacta en la siguiente fase,

⁹ <http://ifcopenshell.org/>

Cuadro 1: Comparación de resultados de elementos reales en tres plantas de evaluación

ID	Nombre de plano	REAL					RESULTADO									
		Circuitos ID	bocaDosTomasGeneral	bocaDosTomasEspecial	bocaIluminacionPared	tableroGeneral	switchSimple	switchCombinada	Circuitos ID	bocaDosTomasGeneral	bocaDosTomasEspecial	bocaIluminacionPared	bocaIluminacionTecho	tableroGeneral	switchSimple	switchCombinada
1	IV-ROM-IE-PL-BT-29b-02-GT.jpg	1A	2		3	2	+1	✓		✓				✓		
		2A	9					✓	✓							
2	IV-ROM-IE-PL-BT-29b-01-GT.jpg	2B	2					✓		✓						
		2C		6	5	1	✓		✓					-1	✓	
3	IV-FRA-IE-PL-11c-GT.jpg	3A		3	1	2	✓		✓		✓	✓	✓			
		3B	2				✓	✓								
		3C	4					✓	+1							
		3D	1					✗	-1							
		3E		4		3		✓		✓					✓	
		3F		1				✓		✓						

debiendo ser filtrados previa determinación de circuitos. En vista de los resultados obtenidos, el número de planos del dataset resultó apropiado, sin embargo, del análisis del conjunto de planos sin incluir, se considera necesario ampliar el etiquetado, para incluir nuevas clases presentes en otros documentos.

En el Cuadro 2 se encuentran los resultados de evaluar la detección de líneas en los planos de control, y correr posteriormente el código para relacionar las líneas detectadas con los elementos de los circuitos inferidos en la segunda etapa. Una limitación observada al generar los circuitos, fue considerar la separación de circuitos según su uso. Esto implica que un toma no puede pertenecer a un circuito eléctrico, adoptándose esta restricción en la lógica. Sin embargo, en la práctica, hay casos en que coexisten en un mismo circuito ambos usos. Esta situación se refleja en el caso del circuito "1A" del cuadro sea dividido en dos circuitos: uno de tomas y otro de iluminación. En futuras mejoras, deberán considerarse estos casos reales fuera de reglamento vigente al momento de determinar los circuitos. El circuito "3D" no fue creado y su único componente fue asociado al circuito "3C", situación provocada porque al hacer el post procesado mencionado en la Sección 5, se elimina el cable que conecta al tablero general que, en este caso en particular, se encuentra incluido dentro del bounding box del elemento. Una situación similar ocurre con el circuito "2C", donde el componente *switchSimple* no fue asociado al circuito, sin ser asociado a ningún otro circuito. Esta situación podría ser mitigada utilizando el polígono de segmentación en lugar del bounding box del elemento detectado. Para el caso de interruptores de iluminación y su respectivo artefacto, resulta necesario incorporar reconocimiento de OCR para establecer la relación entre ambos que en estos casos se establece con letras, por ejemplo llave (a) -iluminación (A) para establecer en el

modelo IFC el funcionamiento. Donde el tablero general estuviera representado, fue posible establecer el inicio de los circuitos en dicho punto.

Por ultimo, el modelado en IFC permite realizar un conteo de los elementos eléctricos, y por lo tanto, resulta suficiente para calcular la demanda eléctrica a partir de asociar consumos típicos de la región para cada elemento. Para los elementos modelados, se asignaron valores de altura en forma manual, ya que dicha información no se encuentra disponible en los planos de planta eléctrica, pero es posible establecer un valor según las reglas del buen arte.

Cuadro 2: Resultado de Precision, Recall y F1-Score en plantas de evaluación.

Archivo	Precision						Recall						F1 Score					
	bocalluminacionPared	bocaDosTomasGeneral	bocaDosTomasEspecial	tableroGeneral	switchSimple	switchCombinada	bocalluminacionPared	bocaDosTomasGeneral	bocaDosTomasEspecial	tableroGeneral	switchSimple	switchCombinada	bocalluminacionPared	bocaDosTomasGeneral	bocaDosTomasEspecial	tableroGeneral	switchSimple	switchCombinada
IV-ROM-IE-PL-BT-29b-01-GT.jpg	1.0	1.0	1.0	-	1.0	1.0	1.0	1.0	1.0	-	1.0	1.0	1.0	1.0	1.0	-	1.0	1.0
IV-ROM-IE-PL-BT-29b-02-GT.jpg	1.0	1.0	-	-	0.5	-	1.0	1.0	-	-	1.0	-	1.0	1.0	-	-	0.67	-
IV-FRA-IE-PL-11c-GT.jpg	1.0	1.0	1.0	1.0	1.0	-	1.0	1.0	1.0	1.0	1.0	-	1.0	1.0	1.0	1.0	1.0	-

8. Conclusiones y trabajos futuros

En este trabajo se abordó la problemática de cómo procesar planos de instalaciones eléctricas utilizando machine-learning para obtener una representación digital del mismo en BIM; en particular en formato IFC, combinando un modelo detector de instancias de segmentos como Cascade Mask R-CNN y el modelo detector de segmentos de líneas DeepSLD para inferir las líneas que conforman los circuitos. La detección de los circuitos y sus componentes requiere una apropiada adaptación de la lógica que relaciona los mismos y consideraciones reglamentarias que normalmente varían de región a región, y por lo tanto deben ser validadas. A su vez, se presenta un dataset específico IVPBA-Elec para planos eléctricos de Argentina. Este enfoque puede ser utilizado para procesar planos con mismas características tal como instalación de gas e instalación sanitaria.

Por otro lado, profundizaremos el estudio comparando diferentes redes e hiperparámetros para mejorar la precisión en la detección de elementos, y añadiendo nuevas técnicas de post procesamiento para el armado de los circuitos.

Referencias

1. Maggie Khaddaj and Issam Srour. Using bim to retrofit existing buildings. *Procedia Engineering*, 145:1526–1533, 2016. ICSDEC 2016 – Integrating Data Science, Construction and Sustainability.

2. Letizia D'Angelo, Magdalena Hajdukiewicz, Federico Seri, and Marcus M. Keane. A novel bim-based process workflow for building retrofit. Journal of Building Engineering, 50:104163, 2022.
3. Yunfan Zhao, Xueyuan Deng, and Huahui Lai. A deep learning-based method to detect components from scanned structural drawings for reconstructing 3d models. Applied Sciences (Switzerland), 10, 3 2020.
4. Yunfan Zhao, Xueyuan Deng, and Huahui Lai. Reconstructing bim from 2d structural drawings for existing buildings. Automation in Construction, 128:103750, 2021.
5. Zhiliang Zeng, Xianzhi Li, Ying Kin Yu, and Chi-Wing 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, pages 9096–9104. arXiv, 2019.
6. Jihyo Seo, Hyejin Park, and Seungyeon Choo. Inference of drawing elements and space usage on architectural drawings using semantic segmentation. Applied Sciences, 10(20), 2020.
7. Chen Liu, Jiajun Wu, Pushmeet Kohli, and Yasutaka Furukawa. Raster-to-vector: Revisiting floorplan transformation. In 2017 IEEE International Conference on Computer Vision (ICCV), pages 2214–2222, 2017.
8. Martin Urbietta, Matias Urbietta, Tomas Laborde, Guillermo Villarreal, and Gustavo Rossi. Generating bim model from structural and architectural plans using artificial intelligence. Journal of Building Engineering, 78:107672, 2023.
9. Yijie Wu, Jianga Shang, Pan Chen, Sisi Zlatanova, Xuke Hu, and Zhiyong Zhou. Indoor mapping and modeling by parsing floor plan images. International Journal of Geographical Information Science, 35(6):1205–1231, 2021.
10. Tanya Bloch and Rafael Sacks. Comparing machine learning and rule-based inferring for semantic enrichment of bim models. Automation in Construction, 91:256–272, 7 2018.
11. Phillip Schönfelder, Angelina Aziz, Frédéric Bosché, and Markus König. Enriching bim models with fire safety equipment using keypoint-based symbol detection in escape plans. Automation in Construction, 162:105382, 2024.
12. Tianzhen Hong, Zhe Wang, Xuan Luo, and Wannu Zhang. State-of-the-art on research and applications of machine learning in the building life cycle. Energy and Buildings, 212:109831, 2020.
13. Benedikt Faltin, Damaris Gann, and Markus 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, pages 877–886, 01 2023.
14. Phillip Schönfelder, Fynn Stebel, Nikos Andreou, and Markus König. Deep learning-based text detection and recognition on architectural floor plans. Automation in Construction, 157:105156, 2024.
15. R. Girshick. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, volume 2015 International Conference on Computer Vision, ICCV 2015, pages 1440–1448, 2015.
16. Cheng Zhang, Yang Zou, and Johannes 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, volume 38, pages 220–226. IAARC Publications, 11 2021.

17. Samuel Dodge, Jiu Xu, and Björn Stenger. Parsing floor plan images. In 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA), pages 358–361. IEEE, 2017.
18. Lucile Gimenez, Sylvain Robert, Frédéric Suard, and Khaldoun Zreik. Automatic reconstruction of 3d building models from scanned 2d floor plans. Automation in Construction, 63:48–56, 03 2016.
19. Brandon Bortoluzzi, Ivan Efremov, Clarice Medina, Daniel Sobieraj, and J.J. McArthur. Automating the creation of building information models for existing buildings. Automation in Construction, 105:102838, 2019.
20. Mengtian Yin, Llewellyn Tang, Tongyu Zhou, Ya Wen, Ruohan Xu, and Wu Deng. Automatic layer classification method-based elevation recognition in architectural drawings for reconstruction of 3d bim models. Automation in Construction, 113:103082, 2020.
21. Bin Yang, Boda Liu, Dayu Zhu, Binghan Zhang, Zhichen Wang, and Ke Lei. Semiautomatic structural bim-model generation methodology using cad construction drawings. Journal of Computing in Civil Engineering, 34(3):04020006, 05 2020.
22. Sébastien Macé, Hervé Locteau, Ernest Valveny, and Salvatore Tabbone. A system to detect rooms in architectural floor plan images. pages 167–174, 06 2010.
23. Quan Meng, Jiakai Zhang, Qiang Hu, Xuming He, and Jingyi Yu. Lgnn: A context-aware line segment detector. In Proceedings of the 28th ACM International Conference on Multimedia, MM '20. ACM, October 2020.
24. Zhiyong Dai, Jianjun Yi, Yajun Zhang, Bo Zhou, and Liang He. Fast and accurate cable detection using cnn. Applied Intelligence, 50:4688–4707, 2020.
25. Siyu Huang, Fangbo Qin, Pengfei Xiong, Ning Ding, Yijia He, and Xiao Liu. Tp-lsd: Tri-points based line segment detector, 2020.
26. Kai Zhao, Qi Han, Chang-Bin Zhang, Jun Xu, and Ming-Ming Cheng. Deep hough transform for semantic line detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, page 1–1, 2021.
27. Paul VC Hough. Method and means for recognizing complex patterns, December 18 1962. US Patent 3,069,654.
28. Rémi Pautrat, Daniel Barath, Viktor Larsson, Martin R. Oswald, and Marc Pollefeys. Deeplsd: Line segment detection and refinement with deep image gradients, 2023.
29. Rafael Grompone von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall. LSD: a Line Segment Detector. Image Processing On Line, 2:35–55, 2012. <https://doi.org/10.5201/ipo1.2012.gjmr-lsd>.
30. Fatih Cagatay Akyon, Sinan Onur Altinuc, and Alptekin Temizel. Slicing aided hyper inference and fine-tuning for small object detection. 2022 IEEE International Conference on Image Processing (ICIP), pages 966–970, 2022.