Automatic floor plan analysis and recognition

Pablo N. Pizarro* (1) Nancy Hitschfeld (2) Ivan Sipiran (2) Jose M. Saavedra (3)

(1) Department of Computer Science, Department of Civil Engineering, Universidad de Chile, pablo.pizarro@ing.uchile.cl, ORCID 0000-0002-8745-6638
(2) Department of Computer Science, Universidad de Chile, nancy@dcc.uchile.cl, ORCID 0000-0003-4923-4679
(2) Department of Computer Science, Universidad de Chile, isipiran@uchile.cl, ORCID 0000-0002-8766-3581
(3) Faculty of Engineering and Applied Science, Universidad de los Andes, jmsaavedrar@miuandes.cl, ORCID 0000-0002-9644-5164

Abstract

Over the last few decades, floor plan analysis and recognition has been an open research topic in computer science, aiming to generate the building's model by automatically extracting meaningful information from diverse sources. Among these, the architectural drawings are one of the most common, typically composed of non-uniform notations, together with their relationship and constraints, defining the structure's layout and usage. Usually, floor plans encompass a high variability in style and semantics, as there is no standard notation to describe each element. Thus, numerous methodologies have been proposed to recognize, vectorize, and model different objects such as walls, doors, and rooms. In this work, we review different procedures from rule-based and learning-based approaches between the years 1995 to 2021, restricting only those considering the plan data as a rasterized image format. Datasets, scopes, and performed tasks were summarized to guide future development within the construction and design industries.

Keywords: *floor plan analysis; image processing; deep machine learning; rule-based methods; object detection; vectorization; segmentation.*

1 Introduction

Architectural floor plans are documents that result from an iterative design process to define the layout, distribution, and usage of a structure, playing a crucial role while designing, understanding, or remodeling indoor spaces [1]. Plans are created from the knowledge and experience of designers and engineers, who use different annotations to integrate the layout, style, use, scale, and external properties of each site, like the environment and regulation. Usually, these documents convey three components to be a valid and complete 2D drawing description of a 3D scene: (1) *geometry*, which defines the shape and dimension of its elements, (2) *topology*, which accounts for the connectivity between building components, and (3) *semantics*, which describes additional characteristics, such as the room function [2, 3]. Moreover, floor plans might also include outer and inner walls, windows, furniture, dimension lines, grids, text, or icons, alongside the constraints and relationships between them, making automatic analysis and information recovery a challenging and open task [1].

Plans have been actively studied in the last 40 years as they are involved in large industries, such as construction, design, property rentals, interior remodeling, or indoor positioning and navigation. Among those, the construction industry, unlike others, has experienced a low growth rate since the late 1960s in major OECD economies, such as the US and UK, or even yielded a negative one (Japan, Germany). Therefore, the declining output per hour worked, and per person employed, became the focus of extensive research [4, 5]. A productivity decrease, in particular for construction, has negative repercussions on the economy, being even one of the key barometers for the 2009 global financial crisis [6]. For these reasons, the computer science community has studied several applications to enhance the design and construction pipelines, simplify the processes, and mitigate losses, eventually reducing the costs and improving productivity.

Although plans are designed and built using advanced vector software, these are frequently stored as raster images in the application process [3]. Similarly, for projects designed before the introduction of computer-aided design (CAD) tools, the architectural documents exist in a paper format that has been manually drawn and scanned to achieve their digital version [7]. Rasterized plans allow non-experts and clients (e.g., home buyers and renters) to understand and acquire information handily. However, these discard semantic and topological metadata like layer or object information, as it is generally considered that only humans will review them [8].

Analyzing these rasterized floor plan images and recognizing their components through an automatic procedure is a long-standing open problem within computer vision, which currently poses four fundamental challenges. First, there is no standard notation among architectural and engineering firms, where colors, line thickness, and symbols usually differ [9]. Second, plans stored as raster images are commonly characterized by complex, fuzzy architectural drawings [10]. Third, the plan structure must satisfy high-level geometric, topologic, and semantic constraints; for example, doors are embedded within walls, generally composed of parallel lines, and walls define the perimeter of rooms, in which their label, furniture, and layout can define its usage. Finally, the floor layout might vary across examples (e.g., houses or apartments can have a different room arrangements) [1].

From a technical standpoint, floor plan analysis research aims to generate the structure model by automatically extracting meaningful information from diverse sources, such as architectural plans or in-scene photographs [10]. This process regularly involves different tasks like the recognition of walls and non-structural elements (e.g., windows, furniture), the detection and classification of rooms, and the building 2D/3D reconstruction. Typically, these procedures cover different disciplines within computer science, like image processing, pattern and symbol recognition, object vectorization, and graph modeling.

Among plan analysis tasks, wall identification is one of the most common because these objects define the main layout of the building and convey essential information to detect other elements, such as doors or beams [11]. Recognizing walls is also helpful across the spectrum of architecture, engineering, and construction as it provides data for design, analysis, and cost estimation, among others [12]. On the other hand, recovery of the room shape and classification has also played an essential role since it allows for understanding the scene and its layout. Both walls and room information, along with other objects studied in floor plan analysis, has led to many applications within industry, for example, in Building Information Modeling (BIM) reconstruction [13–15], 3D modeling from 2D plans [2, 16, 17], architectural optimization [18, 19], structural design [20–23], plan synthetic description [24], Virtual Reality (VR) exploration [25], indoor navigation and modeling [26–28], 3D reconstruction from in-scene photographs [29–31] and volumetric points [32, 33], floor plan generation [34–36], building search and retrieval [12, 37, 38], architectural symbol spoofing [39, 40], plan sketch interpretation [41–43], apartment price estimation [44], the generation of accessible plans for visually impaired people [45] or the automatic analysis of ancient and historical buildings [46–48].

Within related work, wall and room recovery has been traditionally solved using rule-based image processing methods, which exploit heuristics to locate the object notations in floor plans using shape recognition, text filtering, line scanning, and pixel classification [49]. Nevertheless, relying on hand-crafted features is insufficient, as it lacks generality to handle diverse conditions [50]. Extensive effort is required to choose proper low-level processing operations, tune parameters, and craft rules and grammar based on drawing styles or architectural regularity [16], rules that are still highly dependent on the plan format [10]. In other words, it is not easy to generalize the conventional pipelines to deal with complex annotations and high diversity [27]. For such reasons, several learning-based methodologies have been recently proposed to retrieve and model the building objects, mainly by the application of convolutional neural networks (CNNs), graph neural networks (GNNs), and generative adversarial networks (GANs), improving accuracy while keeping a general approach for handling different input styles [27].

Although some reviewed articles introduced a brief literature revision [10, 50–52], to our knowledge, a comprehensive one is lacking in this area. A review of the methodologies that solve different problems within floor plan analysis can guide future development in the construction, design, and engineering industries, for instance, in BIM and 3D reconstruction [15, 17], or the retrieval of similar plans from large databases [37], because it provides a quick guide into which dataset and algorithm must be used to solve a specific task. In particular, this review conceptualizes the research problem, describes the used and available datasets, details the methodologies and their evolution through decades, and presents the challenges & opportunities for new work to come, providing insights into which areas must be covered by future developers. Upcoming floor plan analysis research, combined with its applications, will reduce costs and increase productivity, especially in these industrial sectors that need to automate and enhance their processes by creating new and better software.

The reasons above motivate this review, in which we present related work to recover, classify, and model the building elements and layout, only considering the raster images of architectural floor plans as input. We organize the datasets, methodologies, and results according to the approach type (rule-based, learning-based) and the commonly performed task. In total, we select 61 peer-reviewed articles from 1995 to 2021 over 118 candidates.

The contributions of this review paper can be classified into four major areas:

1. We explore and describe the available datasets used throughout floor plan analysis.
2. We describe the rule and learning-based methodologies that recognize and model the building objects from rasterized floor plan images, such as walls, non-structural elements (door, windows), and rooms (shape and usage classification).
3. We identify the common tasks performed in the reviewed works, organizing them in tables to help future developers select the most appropriate work according to their needs.
4. We present a selection of floor plan analysis applications within the industry.

Concerning the paper structure, section 2 presents the approach for selecting the reviewed literature and conducting the content analysis. Section 3 describes the article's detail and summary, considering rule-based and learning-based methods. Section 4 discusses the remaining challenges, future directions, and applications. Finally, section 5 provides the conclusions and summary of the work.

2 Research method

The present study used content analysis [53] to select the reviewed literature. Content analysis is commonly used to objectively make valid inferences according to collected data for disclosing central aspects of previous studies, further allowing for qualitative and quantitative operations [54]. In order to direct the review, the following research questions were proposed, which motivated the selection of the related work:

Q1. What datasets exist within the area of plan analysis; what are their properties?

Q2. What methodologies exist within the rule-based and learning approaches?

Q3. What are the common tasks among these methods?

Q4. How has the area of study evolved over the years, considering the rapid development of artificial intelligence?

Q5. What are the challenges and opportunities within research?

Q6. What are the main applications of these algorithms?

Sample collection was performed in this study by searching and selecting peer-reviewed articles related to the research questions. Articles were collected from academic databases and cited works within them, considering their impact, contributions, and relationship with the review guidelines. The procedure of literature search and selection for this study can be summarized as follows:

- The academic databases Web of Science, Scopus, IEEE/IET Xplore, Science Direct, ACM Digital Library, ASCE Library, ProQuest, and Springer were used for article search and selection. Also, Semantic Scholar and Connected Papers were employed to retrieve similar articles powered by AI and interactive graphs.

- Keywords such as "floor plan analysis", "floor plan recognition and interpretation", "floor plan segmentation", "floor plan image", "apartment structure", "architectural plan vectorization", "room and wall retrieval", "apartment graph", "object detection in floor plans", and "parsing floor plan images" were used to search the databases. The search date period ranged from 1995 to December 1st, 2021. For each article, its cross-references and similar works were also considered for revision.

- The inclusion criteria correspond to English-only and peer-reviewed articles that used rasterized architectural floor plans of houses or apartments to perform the analysis. The recognized objects were walls or other non-structural elements (e.g., window, door) and rooms alongside their shape and classification, accounting for rule-based and learning-based techniques. Articles that vectorized or modeled a graph of the structure were also included.

- Works within floor plan analysis that recognized objects from sketches, volumetric points, CAD/XML-vector files, in-scene photographs, or examined other structures such as archaeologic or industrial complexes were excluded. Articles that did not consider the building semantics in the recognition, spotting, or vectorization of objects were also discarded; however, those that applied their algorithms to architectural plans were mentioned without further detail. Finally, articles that were only abstracts, minor revisions of previous authors' work, or did not contemplate the evaluation or validation of their methodologies were also discarded. In total, 118 candidates were selected for further revision.

- Following the inclusion/exclusion criteria, a two-round selection technique was employed. In the first round, the titles, abstracts, and keywords of the noted articles were checked to ascertain if they met the criteria. The second round consisted of reading and analyzing the entire document, thus ensuring that all papers were closely related to the aforementioned objectives. Finally, 61 articles were selected and analyzed for the present review.

The analysis of each selected article considers the classification of its tasks, recognized objects, implemented models, used datasets, and a summary of the overall procedure. These features allow reviewed work to be represented in aggregated form within tables and figures, detailed in the following section, to quickly examine the methodologies, leading future developers to choose the appropriate one for their purposes.

3 Architectural floor plan analysis and recognition

Architectural floor plan analysis combines sequential processes that generate building models by automatically extracting meaningful information from rasterized floor plans [10, 13]. As these documents contain a large quantity of heterogeneous information, along with its constraints and interactions, most processes involve different tasks to clean the images and extract valuable data [85]. For example, the pipelines usually pre-process the image to remove distortions, grids, decorations, or titles through binarization. Text extraction and classification [86, 87], or line detection [9], are also common. Typically, pattern recognition, line scanning, or segmentation approaches are used to retrieve objects such as walls and doors, some of which are also vectorized to convert the recognized objects into a vector representation to be editable, scale-independent, and compact [88]. Room space is detected through geometry and semantic information, including textual data [89]. Symbol recognition is also an important part of building plan processing, which extracts labels to identify dimensions, room usages, and objects such as doors or windows [13, 90–92].

Although floor plan analysis considers many tasks and processes, they can be classified into four broad categories: (1) *Graphics separation*, a pre-processing for object recognition, which removes graphic elements from floor plans such as furniture or grids that do not bring new semantic information to the analysis, (2) *Object recognition*, a process which recognizes building elements like walls, openings, and rooms, being the core of the floor plan research, (3) *Vectorization*, stage in which the structural elements are transformed into a vector form for their 2D/3D representation and analysis, and (4) *Structural modeling*, a process that aims to create a mathematical model of the floor topology, generally as a connected graph by constructing an adjacency matrix based on the relationship among plan objects.

Rule-based methods, such as text filtering and line scanning, were initially proposed to recognize and vectorize elements like walls and rooms [49]. Traditionally, a pre-processing pipeline was carried out as the first step to separate graphic elements, for example, by distinguishing between lines of different thicknesses [10]. Nevertheless, relying on hand-crafted features is insufficient, as it lacks generality to handle diverse conditions [50]. Moreover, rule-based algorithms depend heavily on notation and empirical parameters, performing well in specific formats but having limitations in copying others. Extensive effort is required to choose proper low-level processing operations, tune parameters, and craft rules and grammar based on drawing styles or architectural regularity [16].

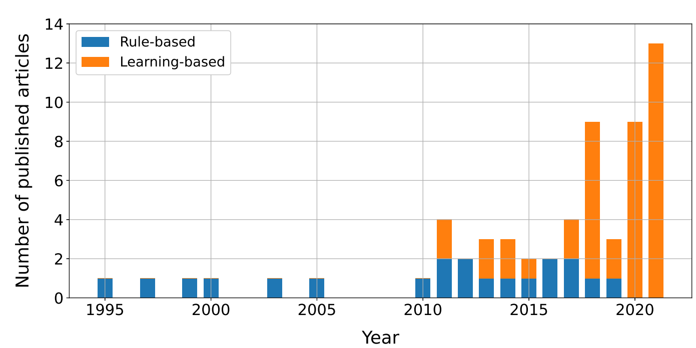


Figure 1: Articles published per year within reviewed works regarding rule-based (20) and learning-based (41) approaches.

By contrast, learning-based methods have gained much attention in recent years as they are trained to achieve the same goals

Table 1: Datasets used by floor plan analysis research.

Dataset Name, reference (year)	Public access	Annotation	Number of plans
FPLAN-POLY, [55] (2010)	✓ [56]	Walls, doors, windows, and furniture in vectorized format	42
SESYD, [57] (2010)	✓ [58]	Walls, doors, windows, and 6 different furniture types; 10 different synthetic apartment configurations, designed to study symbol recognition. Res 1,837–6,775	1,000
CVC-FP, [9, 59] (2010–2015)	✓ [60]	Walls, doors, windows, and rooms without type; 4 different subsets. Res 905–7,383	122
R3D – Rent3D, [61] (2015)	✓ [62]	Walls, doors, windows, and room types	215
SydneyHouse, [63] (2016)	✓ [64]	Walls, doors, and windows of multi-unit house floor plans; several styles. Res 404–4,678	174
R-FP – Rakuten, [65] (2017)	✓ [66]	Walls. Res 156–1,427	500
ROBIN, [12] (2017)	✓ [67]	Synthetic 3–5 room apartments; designed to study plan retrieval. Res 1,837–6,775	510
R2V, [1] (2017)	✓ [68]	Walls, openings, and room types. Res 96–1,920	815
CubiCasa5K, [69] (2019)	✓ [70]	80 object categories such as doors, windows, and walls. Res 50–8,000	5,000
RPLAN, [34] (2019)	✓ [71]	Wall, room, boundary, and inside masks; designed to study plan generation	80,788
Korea LH, [72] (2019)	✓ [73]	None. Res 230–5,092	343
BRIDGE, [74] (2019)	✗	Windows, doors, along with other 14 object types. Include region and paragraph descriptions	13,000
HouseExpo, [75] (2020)	✓ [76]	Binary house wall masks; designed to study indoor-layout learning. Res 110–10,086	35,126
BTI, [77] (2020)	✗	None	2,000
EAIS, [28, 78] (2020)	✗	Walls, doors	450
ZSCVFP, [79] (2021)	✗	Walls, rooms, entry, door, window, and balcony objects	10,800
RFP, [80] (2021)	✗	Walls, doors, windows, doorways, and 7 rooms types. Res 180–3,615	7,000
RuralHomeData, [81] (2021)	✗	Walls, doors, windows, stairs, slopes, text, and 21 room types. Res 1,600–2,560	800
RUB, [82] (2021)	✓ [83]	Segment nodes classified as door or non-door, both in image and CAD format. Res 500–18,000	74
LIFULL, [84] (–)	✗	None	5,300,000+

Note: Res – Resolution in pixels (px).

but with better accuracy, handling different input styles [27]. In the early learning approach, graphic separation and specific segmentation rules were needed. However, as deep learning was introduced, the applications have undergone rapid development or were simplified to a few steps. For example, many used the floor plan images directly to train the models without the need for complex image pre-processing pipelines, increasing the analysis versatility [10]. Compared to the rule-based works, the research community has extensively focused on learning-based methods in the last five years, mainly due to the advances in machine learning models and the accessibility to richer and more extensive databases. This trend is illustrated in Figure 1, which compares the number of published articles per year from 1995 to December 1st, 2021.

Although there has been a significant improvement in processing algorithms over the last years, floor plan analysis and recognition is still considered an open and challenging task [1, 9]. Rule-based algorithms rely on particular plan styles that are hard to generalize or require expert knowledge to readjust for other formats. Learning-based models trained on various input floor plan datasets may have great adaptability. Still, their outputs may be blurry as they perform pixel-level segmentation, creating problems as some entities might have unconnected lines [93]. General-purpose object detection algorithms, such as Faster R-CNN [94] and YOLO [95], as well as other anchor-based frameworks, cannot retrieve curved or sloped walls or have problems recognizing objects in different conditions, as there is no suitable annotation to describe the complex geometrical characteristic of these architectural primitives [79]. Moreover, room detection and recognition depend heavily on structural elements in the floor plan, such as walls, doors, or windows. Thus, if a particular plan misses an element or some object polygons are not closed, it will considerably affect the room formation process [93]. Despite these difficulties and challenges, current works within the area have tackled many problems, from recognition to vectorization, with several applications for the construction and design industries while improving accuracy and generalization to process diverse and complex floor plans.

The following subsections describe the public and private datasets, as well as the rule-based and learning-based methodologies. In both cases, the reviewed works were cataloged according to the categories they satisfy (graphics separation, object recognition, vectorization, structural modeling), the objects they recognize (wall, door/window, rooms, OCR/dimension), and the model or algorithm implemented.

3.1 Datasets

Datasets have played an essential role within floor plan analysis because there is no standard notation for their composition; therefore, designed models must incorporate specific rules for each particular style. Typically, the implementations face a high variability in their design due to three main reasons:

1. The plan representation, where, in best cases, only 70 % of the graphical information is compliant with a standard rule [96].
2. The nature of these documents, where the total possible configurations and relationships between plan elements are virtually infinite.

(a) **FPLAN-POLY** [55] (b) **SESYD** [57] (c) **CVC-FP** [59] (d) **R3D – Rent3D** [61] (e) **SydneyHouse** [63]

(f) **R-FP – Rakuten** [65] (g) **ROBIN** [12] (h) **R2V** [1] / **LIFULL** [84] (i) **CubiCasa5K** [69] (j) **RPLAN** [34]

(k) **BTI** [77] (l) **EAIS** [28, 78] (m) **ZSCVFP** [79] (n) **RFP** [80] (o) **RuralHomeData** [81]

Figure 2: Floor plan image examples from datasets.

3. The way the information is visually represented, for example, in different styles, formats, or symbols [52].

Moreover, each floor plan dataset has limitations regarding quantity or complexity. Thus, researchers opt to utilize the datasets suitable for their purposes, including specific processing steps that could not be generalized to others [10].

For such datasets to be helpful in floor plan analysis, there must be annotations for objects such as walls and rooms. Annotating floor plans, despite other document types, is a complex and expensive task, as it requires high-level expertise to recognize the different elements due to ambiguity in notation [9, 52]. For example, in some plans, windows can be overlapped with beams, or the slab can contain paths, shafts, or custom symbols defined by the architectural and structural firms. Even though several practical tools have been developed to annotate them conveniently [59, 97, 98], it is difficult to do so because there is no way to guarantee the same annotations from different experts, especially for complicated plans [10].

The reviewed datasets are summarized in Table 1, considering their source article, public availability, annotation, and the number of plans. Figure 2 illustrates a selection of images from the datasets considered within the review. It can be noted that there are distinct drawing styles among the apartment and house plans; some have color and textures (Cases *f, h, m, n*), room type labels (Cases *c-f, h, i*), icons (Cases *d, f*), dimension lines (Cases *c, l-n*), furniture (Cases *a-i, l, o*), and walls with several styles, angles, and complex arrangements. These diverse settings were exploited by rule-based methods, described in section 3.2, which recognize walls, doors, windows, furniture, and rooms by defining algorithms that considered different approaches specific to each style; or by learning-based ones (section 3.3), that trained models to automatically recognize the objects.

3.2 Rule-based methods

Early research within floor plan analysis studied the object recognition and modeling from CAD files, as these vector documents already contain the exact and accurate geometry of their elements in separate layers. However, the topologic and semantic properties are usually not present or exist as icons or text. Examples of early studies are Cherneff *et al.* [99], which proposed an interpretation method to extract the plan structure, i.e., walls, doors, windows, rooms, and its associated spatial relations considering a limited drawing grammar. Shape grammar (SGs) was a popular rule-based approach within automatic floor plan analysis, comprising a set of rules that can be applied consecutively to generate a geometrical shape, reproducing the particular architectural styles [100]. Other early works are the vector segment conversion from line drawings [101], the hand-sketched plan interpretation [102], and the recognition of symbols and structural textures from printed or hand-drawn plan sketches [103]. Despite these examples, this preliminary research did not analyze the plans concerning the semantics and functional interaction of the elements, for example, the relationship between walls and rooms or that openings (doors and windows) are usually embedded between two wall segments. Furthermore, these examples did not consider raster floor plans, which are common when storing and distributing to customers [3], or processed simplified sketches. Therefore, the scope was restricted to analyzing vector-based CAD files or retrieving individual elements from simple format plans.

Among the first works that considered the analysis directly on raster plans is Ryall *et al.* [104]. They proposed an early semi-automatic room segmentation method, which finds regions using a proximity metric. Despite its significant drawbacks, such as retrieving false positives from slab shafts, doors, or staircases, it serves as a first approach to extracting objects directly from images, proving that algorithms can recognize the building semantics and constraints even if they are not apparent from a low-level standpoint.

A major improvement to Ryall's work happened along with the contributions of Tombre's group that studied the automatic reconstruction of 3D structures from scanned plans [96, 105, 106], whose main idea is illustrated in Figure 3. Their approach estimated tiling the high-resolution images, dividing them into independent and overlapped patches to overcome memory issues, and segmenting the pixels of thin and thick lines by morphological filtering [107] after separating graphics and text. The overall process considers two kinds of walls represented by thick parallel or single lines. Doors are sought by detecting arcs, windows by finding small loops, and rooms by even bigger loops. Finally, the segmented pixels are skeletonized to assemble a vectorized format, which leads to the 3D reconstruction of a single level [96]. Moreover, a multi-level building reconstruction is possible if floors are matched by finding special symbols like corners, staircases, pipes, and bearing walls [105].

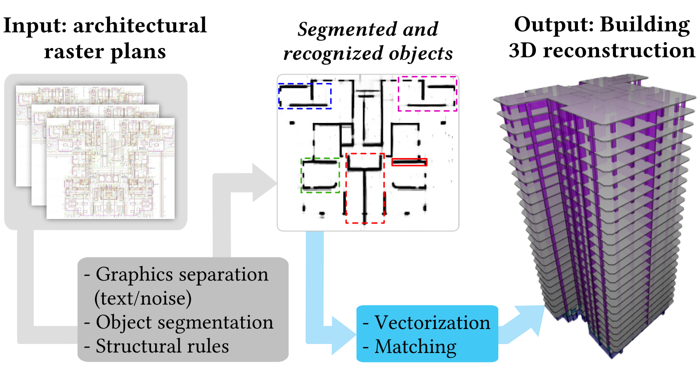


Figure 3: Reconstruction methodology of a 3D building structure from rasterized input plans.

Tombre's group has also intensively studied several rule-based methods for symbol detection, text separation, and graphics vectorization [107–110]. These proposed pipelines completely revolutionized floor plan analysis, contributing methods to assemble and reconstruct the overlying topologic and semantic constraints embedded in floor plans. However, the implemented symbol detection strategies are oriented to one specific and limited notation, same as their 3D building reconstruction method. Thus, a hypothetical change of the floor plan style might imply reconsidering part of the algorithms, requiring a new set of threshold values for each case.

Or *et al.* in [111] also focused on 3D model generation from rasterized plans for one-story buildings. After separating text and vectorizing the graphical layer with QGAR tools [97], they manually detected object symbols unrelated to the plan structure, such as cupboards, sinks, among others. Once the remaining lines only belong to walls, doors, and windows, a set of polygons is generated using each vectorized image's polyline. Walls are represented by thick lines, windows by rectangles inside them, and doors by arcs. Similarly, Gimenez *et al.* [2] proposed another method to reconstruct the 3D models from image plans. After separating graphics using a couple of Tombre's work [112] and QGAR tools, various building elements were detected based on structural rules, like assigning the wall to two parallel lines within a certain distance. Finally, 3D building models were generated by properly assembling the vectorized elements. Even though the models achieved a good performance concerning their plan style, these methods have many predefined hyper-parameters, manual pre-processing heuristics, or assumed a specific notation for wall segments; thus, these methods lack generalizability.

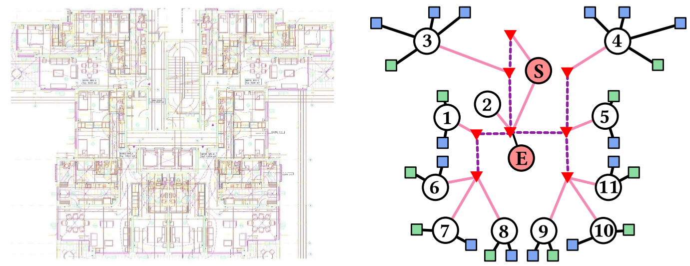
Macé *et al.* [9] also focused on extracting the structure from scanned plans and proposed an algorithm to detect rooms. Like previous examples, text/graphic pre-processing is performed with QGAR, followed by a thin/thick separation from graphic components based on coupling the Hough transform [113] and image vectorization. The thick lines extracted from this algorithm are regarded as wall contours, which authors expected to be parallel, and are used as the candidates for the wall detection. Finally, doors and windows are identified to detect rooms through recursive decomposition until convex-shaped regions are found from the wall borders. Similar to previous works, this approach also considers manual thresholds and is limited to a specific notation; thus, the wall detector must be re-designed to deal with other plan styles.

Mace's work was later expanded by Ahmed *et al.* [43, 114], where they introduced new processing steps like wall edges extraction and boundary detection, designed for plan retrieval tasks. Their process starts with wall detection and text/graphics segmentation [115] to separate graphic components into thin, thick, and, as a novelty, medium lines. Walls are assembled from thick and medium ones, while thin lines are considered to form symbols; components outside the convex hull of the outer walls were also removed. Then, doors, windows, and rooms were spotted using SURF [116], which is a method that provides a good discriminative translation, rotation, and scale-invariant representation of symbols. Finally, the text inside the rooms was used for their labeling. According to its distribution, the authors further enhance this method by splitting rooms into many parts as labels are inside them, vertically or horizontally [86]. It is important to note that these works [43, 86, 114, 115] consider some structural and semantic information as they assembled the wall contours of each room, labeled them with their name, and verified their composition using the door and window positions. However, as before, these methods might have to be revisited when dealing with floor plans of different graphical conventions.

Several other studies have also considered a line representation to recognize structural elements from floor plans. Park and Kwon [7] recognized the main walls of apartments using the auxiliary dimension line, where windows can be retrieved as a subproduct. Feltes *et al.*'s work [117] is capable of finding the object's corners in wall-line drawing images by filtering out unnecessary points without changing the overall structure, especially those that appeared through over-segmentation of diagonal lines; also, a wall-gap filling is possible while performing a heuristic criterion. Tang *et al.* [118] automatically generated vector drawings by applying various filters, such as gradient, length, gap-filling, line-merging, and connectivity under several millimeter sizes, assuming walls are represented by parallel lines in both vertical and horizontal axis. Pan *et al.* [119] detected walls and windows considering empirical rules regarding their pixel layouts, where the user must adjust the methodology's thresholds; bearing walls corresponded to black areas, non-bearing walls to unfilled parallel rectangles, and windows are composed of three to four closer parallel lines. De [120] also assumed that only walls are illustrated as thick black lines in a floor plan layout. Thus, thick and thin lines can be distinguished using a morphological transformation; thick lines can be considered walls, whereas arc lines represent doors. On the other hand, in an effort to overcome the lack of a standard notation, de las Heras *et al.* [11] presented an unsupervised wall segmentation method that assumes walls as repetitive rectangular elements, placed in orthogonal directions, filled with the same pattern and naturally distributed across the plan. Although assumptions might work over a specific notation, they do not consider semantical relationships or require new rules to adopt for other plan styles.

Graph-based solutions also have been presented to describe the underlying structure of floor plans. Sharma *et al.* [121] proposed a room layout segmentation and adjacent room detection algorithm to represent the layout as an undirected graph. The model was developed to retrieve similar plans from a large database by calculating a matching score that considered fine-grained features computed from an assembled Room Adjacency Graph (RAG), where the room area and furniture types were identified [38]. Similarly, Barducci *et al.* [122] described floor plan images by building a RAG, identifying room purpose from the furniture recognized by graph matching, without considering textual labels. Their work was further expanded by Goncu *et al.* [45], extending the wall, door, and room identification. Walls were binarized, whose straight-line segments were identified by the Hough transform and polygonized with the Ramer-Douglas-Peucker algorithm [123]. Hough transform was also used to detect arcs, which were later assigned to doors, as previous examples did.

Figure 4 illustrates an example of a graph model from a complex rasterized floor plan. The circular numbered nodes represent apartments, red nodes represent the stairs (S) and elevators (E), and the red inverted triangles stand for hall joints. The squared nodes belong to bedrooms (blue) and dinner rooms (green). Finally, edges represent the connectivity between elements.



(a) Rasterized floor plan. (b) Graph model.

Figure 4: Example of a graph model from a rasterized input plan.

Different low-level geometrical vectorization methods have also been developed to obtain the objects from generic line drawings, for instance, by separating and skeletonizing layers of homogeneous thickness [124] or by an energy-based approach [125]. Nevertheless, recognizing floor plan elements without considering their semantic relationship is error-prone, as each element can be a constraint for other objects. For example, walls delineate the perimeter of rooms, defining the layout and conveying essential information to detect other structural elements, like doors, windows, or openings [11]. For such reason, and as a means to avoid the elaboration of complex recognition rules, learning-based models have been widely studied, especially in the last five years. Learning models, detailed in the following section, allow extracting the complex and hidden relationships embedded in plan documents directly from the training data, synthesizing the experience of several architects and structural engineers, the construction regulation, and human creativity.

Table 2 details the reviewed rule-based methods within floor plan recognition, considering the datasets used (Table 1) and the four categories of tasks, such as (1) *Graphics separation*, (2) *Object recognition*, (3) *Vectorization*, and (4) *Structural modeling*.

3.3 Learning-based methods

3.3.1 Machine learning in floor plan analysis

Since rule-based methods were introduced in floor plan analysis, many tasks have been solved with reasonable accuracies, such as recognizing the structural objects and symbols. Nevertheless, methods have still been confined to a few concise, simplified, and abridged versions of architectural plans [10]. The lack of a standard notation and the limited number of public datasets forced the pipelines to deal with different styles, for example, by applying new rules that were hard to implement or needed in some cases to fine-tune certain variables requiring expert design knowledge. These drawbacks, especially those related to the lack of public data, maintained a limited development within the area.

However, since 2017, an explosive research of learning-based methodologies (Figure 1) happened alongside the increase of public datasets (Table 1) and general-purpose models within the computer vision field. In contrast to the rule-based methods previously detailed, learning-based pipelines automatically learn the relationship between floor plan elements by exploiting new low-level and high-level features directly from hundreds of validated floor plans, improving results while simplifying the analysis. However, learning methods require a larger volume of data for training and parameter tuning, which can be challenging to access, extremely expensive, or unnecessary if only a few concise plans are required to be processed.

Table 2: Rule-based research, sorted by year, considering its tasks and datasets used.

Reference (year)	Dataset (number of plans used)	Strategy	G. Sep.[a]	Object recognition				Vect.[d]	Mod.[e]
				Wall	Door/W.[b]	Room	OCR/Dim.[c]		
[104] (1995)	Defined in paper (1)	Proximity field	–	–	–	✓	–	–	–
[96, 105, 106] (1997–2000)	Defined in paper (2)	Tiling, Morphological operations, Skeletonization, Feature matching	✓	✓	✓	✓	–	✓	–
[7] (2003)	Defined in paper (1)	Auxilary dimension line, Binarization	✓	✓	✓	–	✓	✓	–
[111] (2005)	Defined in paper (-)	QGAR, Segment matching, Predefined rules	✓	✓	✓	–	–	✓	–
[9] (2010)	CVC-FP (80)	QGAR, Hough transform, image vectorization, recursive decomposition	✓	✓	✓	✓	–	✓	–
[115] (2011)	CVC-FP (90)	Morphological operations, connected component analysis	✓	✓	–	–	✓	–	–
[114] (2011)	CVC-FP (80)	Text/graphics segmentation, line separation, SURF	✓	✓	✓	✓	✓	✓	–
[86] (2012)	CVC-FP (80)	SURF, post-processing room split, predefined rules	✓	–	–	✓	✓	✓	–
[122] (2012)	SESYD (1000), FPLAN-POLY (42)	Graph matching, adaptive thresholding, morphological operations, Hough transform	✓	✓	✓	✓	–	–	✓
[11] (2013)	CVC-FP (122)	Predefined rules	✓	✓	–	–	–	–	–
[117] (2014)	CVC-FP (90)	Corner detection and filtering, wall gap closing	–	✓	–	✓	–	✓	–
[45] (2015)	CVC-FP (90)	Adaptive thresholding, Hough transform, Ramer-Douglas-Peucker, Voronoi partition, RAG	✓	✓	✓	✓	✓	✓	✓
[121] (2016)	SESYD (1000)	Boundary extraction, morphological operations, graph spectral embedding	✓	✓	✓	✓	–	–	✓
[2] (2016)	CVC-FP (90)	Text and Geometry separation, Hough transform, QGAR, predefined rules	✓	✓	✓	–	✓	✓	✓
[118] (2017)	Defined in paper (-)	Rule-based filters	✓	✓	✓	–	–	✓	–
[119] (2017)	Defined in paper (100)	OTSU binarization, predefined rules	✓	✓	✓	–	–	–	–
[38] (2018)	ROBIN (510)	Topological adjacency graph, furniture categorization	–	–	✓	✓	–	–	✓
[120] (2019)	Defined in paper (80)	OTSU binarization, thin/thick morphological separation, skeletonization	✓	✓	✓	–	–	✓	–

[a] Graphics separation [b] Door/Window/Furniture/Others [c] OCR or object dimensions were recognized [d] Vectorization [e] Modeling (Graph, other)

Among the first approaches, de las Heras and Sánchez [126] proposed a syntactic model for architectural floor plan interpretation. A stochastic image grammar over an And-Or graph was inferred to represent the hierarchical, structural, and semantic relations between floor plan elements, thus comprising architectural knowledge. Grammar was augmented with three different probabilistic models, learned from a training set, to account for the frequency of that relations. Then, a parser with a pruning strategy was used for the plan recognition. Walls and doors were detected using Mace's rule-based method [9], windows were extracted using a bag of patches approach, and rooms were assembled by joining each element with incident neighbors. Despite its recognition results, this work introduced a learnable model for interpreting a plan; however, rule-based methods were still needed to detect the structural objects. To overcome the last issue and push learning-based algorithms to become style-independent, the group later proposed a machine learning procedure [52] to study and recognize floor plan elements, thus avoiding the need for complex ad-hoc rules for each notation.

In [52], de las Heras *et al.* presented a style-invariant, automatic method that uses a Support Vector Machine Bag of Visual Words (SVM-BOVM) to detect the pixel boundaries of the structural elements. BOVW is a technique that describes an image as a set of visual words or topics created by clustering similar low-level image features extracted from training data [127]. With such a technique, an improved pipeline previously designed by the authors [128, 129] was proposed, which consisted of two steps, a statistical pixel-level patch-based segmentation, and structural recognition. The image patches were classified into three types (doors, walls, and windows) using the BOVW model. In addition, the pipeline recognizes room boundaries in the floor plan by finding closed regions surrounded by vectors in a planar graph of structural entities. Even though these works achieved a remarkable advance in architectural floor plan analysis, as learning-based methods were formerly introduced, the models were still tuned to each particular graphical style in the CVC-FP dataset [59], using different parameters for each wall type. Thus, they cannot be generalized to any scenario.

A similar approach based on the SVM-BOVW model was proposed by de las Heras *et al.* [130], but using an unsupervised segmentation as a preliminary pipeline step to avoid expensive and time-consuming image labeling [11]. In this work, a template-matching technique is done by finding parallel and closer lines to seek the wall-segment candidates; those were also ranked considering a score based on assumptions regarding the plan style. Finally, a patch-based SVM-BOVW learns the candidate's appearance and refines the initial segmentation. Although the method can be applied to several un-labeled styles, the walls must fall into strict assumptions. Furthermore, the semantical relationship between segments is ignored, as only the drawing style is considered to query the elements. For instance, if walls and furniture have a similar line notation, both are segmented, independent of their semantic representation.

An unsupervised statistical approach was also presented by de las Heras *et al.* [131]. In that work, they introduced an attributed graph grammar that represents the floor plan layout by incorporating structural and semantical relations to the building objects learned stochastically from annotated data. The stochastic model embedded in the grammar allows inferring contextual relations between architectural elements, adapting the methodology to the variability while analyzing different plans. Their parsing method relies on their previous SVM-based pipeline to recognize walls and doors [130], considering the standard rule-based Hough transform method. Although this contribution summarizes the techniques proposed by the group to assemble a complete floor from a style-invariant model, it relies on complex learning rules, and assumes a particular format for wall recognition.

After SVM-based models, different algorithms have been proposed in recent years to improve performance, simplify the analysis, and generalize recognition to more plan styles and formats. Mewada *et al.* [51], for instance, introduced a framework based on the α-shape algorithm [132] to extract room shapes from binarized images, calculating and classifying their properties, such as room's width, length, area, and type, using a linear regression model. Other works have also presented learning-based models for classification; however, rule-based algorithms were still needed to recognize the objects. Guo and Peng [133], for example, segmented walls considering their color gradient, eliminating noise by adjusting a threshold. They used a pre-trained VGG-16 network [134] (Figure 5) to extract features of the floor plan, inspired by transfer learning, whose goal is to extract information from related tasks to assist in solving new ones that lack valid training sets. Later, the wall shapes were classified with a multi-layer perceptron into rectangle, square, L-shape, and irregular classes. Another recent example is the work from Park and Kim [135], which ensembled a 3D model of the building using rule-based methods to recognize the horizontal and vertical walls; and considered the learning-based TensorFlow object detection API to detect the wall junctions, openings (door/window), and rooms. The results from junctions and walls were used later to assemble a graph representation of the plan layout employing five generation rules, allowing to vectorize the elements and reconstruct their 3D representation.

A Positive Unlabeled (PU) learning-based approach was presented by Evangelou *et al.* [136] to retrieve walls similar to a manual query by the user, exploring object recognition from unlabelled plans as a means to avoid the expensive annotation task. In PU learning, a binary classifier learns in a semi-supervised way from positive or unlabelled data points, where the assumption is that the unlabeled data can contain both positive and negative examples. It is typically used when labeled data is not available, has many outliners, or the training dataset contains a large number of false negatives [137]. In the context of the proposed method, the query serves as the positive example of the particular wall template to be matched, whereas the filtered candidate Regions of Interest (ROIs) of each floor plan are unlabelled. Despite being a single object retrieval model, this SVM-based PU improves the performance concerning the BOVW [52].

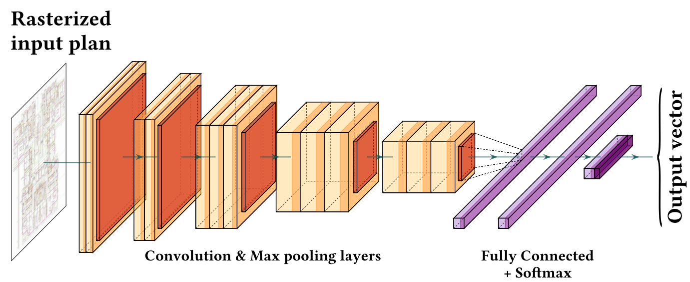
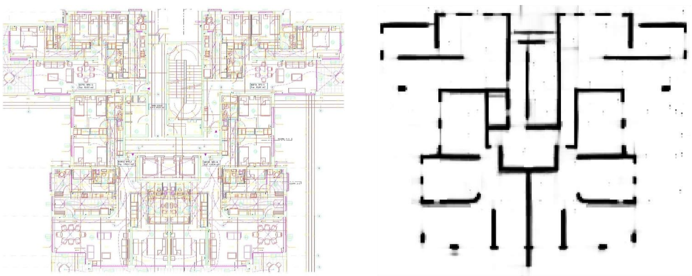


Figure 5: VGG model architecture, which extracts features from a rasterized floor plan image and outputs a vector that can be used to predict or classify several elements.

Fuzzy rule-based systems (FRBS) have also been studied within floor plan analysis. Fuzzy logic is an intelligent controller that simulates human behavior by incorporating *If–then rules* into the system, thus including human experience and knowledge [138]. Leon-Garza *et al.* [127] introduced two Type-1 FRBS models that use fuzzy logic and similarity of image patches to add context information, an approach inspired by the BOVW [139] and the patch-based segmentation process proposed by de las Heras *et al.* [128]. One model used only pixel-level information (color intensity) and the other pixel-level and context information to segment floor plans for wall retrieving. An interval Type-2 FRBS model was also presented by Leon-Garza *et al.* [140], which does not need a pre-process step to remove noise from the image, and outperformed Type-I models in terms of the Intersection over Union (IoU), a standard metric for segmentation problems [128, 141]. Although FRBS models are simple to implement, have low computational cost, are transparent, explainable, and modifiable by end-users (architects or engineers) [142], they still suffer common issues from other floor plan analysis models. In this case, they are hard to generalize to other styles after learning and rely upon low-level pixel information to compute features, such as the color intensity.

While a wide variety of learning algorithms have been presented in recent years within floor plan analysis research, those that have achieved state-of-the-art results come with the development of Deep Learning (DL) technology, especially neural networks [81]. In this way, the role of learning-based methodologies has expanded as graphic separation can be omitted from raw plan images, and rule-based recognition rules were abridged, as models were trained to infer them directly from a broad variety of styles [10]. Among DL models, object segmentation is one of the tasks that has leaded research in computer vision recently [72] and can be formulated as a classification (semantic) or partition problem (instance). Semantic segmentation performs pixel-level labeling with a set of object categories for all image pixels, such as wall, window, or room, by identifying the spatial feature of the object and reflecting it in the results. Meanwhile, the instance segmentation extends the classification scope further by detecting and delineating each object of interest in the image [143]. As an example, Figure 6 illustrates the segmented walls of a floor plan image, where it can be noticed that results are subject to noise and other artifacts, making the recovery of, for instance, the polygon or the precise contour shape, a non-trivial task.



(a) Rasterized floor plan. (b) Segmented walls.

Figure 6: Example of segmented walls from a floor plan image.

In the following subsection, the proposed deep learning models are revised, explaining their approaches in floor plan analysis to recognize, classify, and vectorize structural objects and rooms. Table 3 resumes all learning-based works, considering the datasets used and the four categories of tasks, such as (1) *Graphics separation*, (2) *Object recognition*, (3) *Vectorization*, and (4) *Structural modeling*.

3.3.2 Deep-learning models

Among deep learning techniques, the convolutional neural networks (CNN) have been widely employed within floor plan analysis to automatically extract advanced features, enhancing the recognition of several structural objects [80]. CNNs are a standard supervised learning algorithm, generally used in computer vision due to their intrinsic relationship with two-dimensional tensor processing, such as the pixel matrix of an image [152]. CNNs have a topology composed of convolutional layers, non-linear processing units, and sampling layers. The first one applies a convolution operator on the input through a kernel matrix (also known as filters), transforming the data so that certain features become more dominant in the output. The kernel matrices, commonly used in image processing, can be manually defined to perform different tasks such as edge detection, blurring, or contrast change; however, those trained in a CNN model extract more abstract non-trivial features. The convolutional layers' output is later assigned to a non-linear processing unit (activation function), which helps in the abstraction capacity while learning and provides non-linearity in the feature space, generating diverse activation patterns for different responses, facilitating the learning of semantic differences between the data. The activation function output is usually followed by a sampling layer (subsampling or oversampling), summarizing the results, and keeping the input invariant to geometric distortions [22].

CNNs have had a significant boom in detection, segmentation, classification, generation, and image recovery tasks [153]. For such reason, they have been widely used throughout related work to exploit new features hard to capture considering manual rules, as exemplified in Figure 7. Although CNNs have proved to be powerful in image classification and segmentation, they have two main disadvantages. First, there is a lack of interpretability of how the model works for end-users [154], and training requires a lot of labeled data for the models to be capable of generalizing correctly [127]. Thus, the development of such procedures led the research community to create new, large-scale datasets, which started to be publicly published after the first works tackled CNNs (2017), as shown in Table 1.

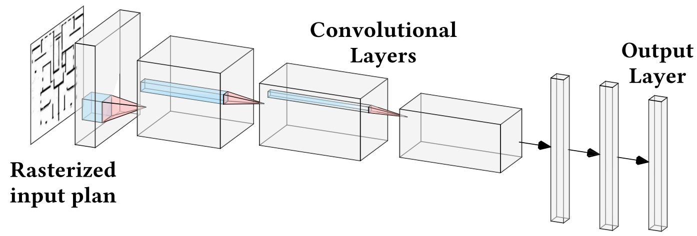


Figure 7: Generic CNN-based model that automatically retrieves features from a rasterized plan, for example, to segment walls or classify its objects.

Within deep learning, models can be discriminative or generative-based. Discriminative models (section 3.3.2.1) learn the conditional probability distribution of the classes (e.g., wall or background), that is, the decision boundary, to make predictions on the unseen data in tasks such as classification, regression, or segmentation. Therefore, their ultimate objective is to separate one class from another. Conversely, generative models (section 3.3.2.2) learn the joint probability distribution, that is, the distribution of the individual classes in a dataset, to return a probability for a given example. Generative learning algorithms tend to model the underlying patterns or distribution of the data points, and, unlike discriminative, these are also capable of generating new data points. Figure 8 illustrates the explored deep learning models within floor plan research, which are detailed in the following paragraphs.

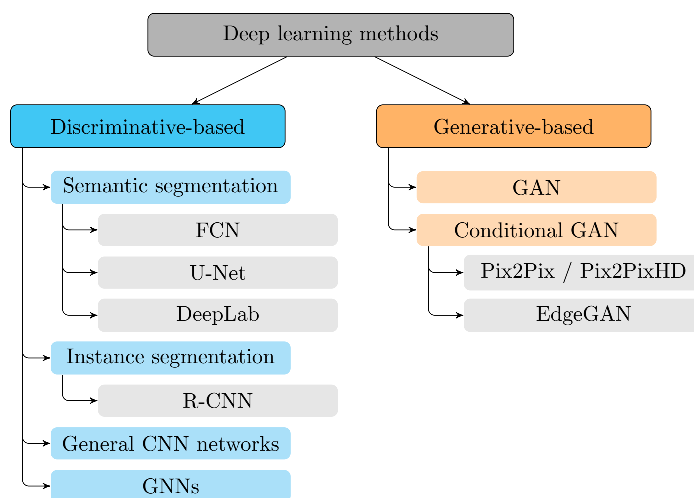


Figure 8: Deep learning methods explored within floor plan analysis research.

3.3.2.1 Discriminative-based models

Among discriminative-based models, the semantic segmentation FCN [141], U-Net [155], DeepLab [156], and instance segmentation model R-CNN [157] have been used. FCNs or Fully Convolutional Networks are composed of two main sections: encoder (contraction) and decoder (expansion). The encoder section is used

Table 3: Learning-based research, sorted by year, considering its tasks and datasets used.

Reference (year)	Dataset (number of used plans)[a]	Strategy	Aug.[b]	G. Sep.[c]	Object recognition				Vect.[f]	Mod.[g]
					Wall	Door/W.[d]	Room	OCR/Dim.[e]		
[126] (2011)	CVC-FP (25)	And-Or graph, predefined rule	–	–	✓	✓	✓	–	–	✓
[128] (2011)	CVC-FP (90)	BOVW	–	✓	✓	–	–	–	–	–
[129] (2013)	CVC-FP (100)	SVM-BOVM	–	✓	✓	–	–	–	–	–
[52] (2014)	CVC-FP (122)	SVM-BOVM	–	✓	✓	✓	✓	–	–	✓
[130] (2014)	CVC-FP (122)	SVM-BOVM	✓	✓	✓	–	–	–	–	–
[131] (2015)	CVC-FP (122)	Stochastic attributed graph grammar	–	✓	✓	✓	✓	–	–	✓
[65] (2017)	R-FP (500), CVC-FP (122)	FCN-2s, Faster R-CNN	–	–	✓	✓	–	✓	–	–
[1] (2017)	R2V (770/100)	CNN, modified ResNet-152	✓	–	✓	✓	✓	–	✓	–
[144] (2018)	LIFULL (1635/500/500)	FCN	–	–	✓	✓	✓	–	–	✓
[145] (2018)	Defined in paper (100/15)	Pix2PixHD	–	–	–	–	✓	–	–	–
[3] (2018)	EAIS (255/35/35)	U-Net + PixelDCL	–	–	✓	✓	–	–	–	–
[146] (2018)	Defined in paper (135)	Faster R-CNN	✓	–	–	✓	–	–	–	–
[133] (2018)	Defined in paper (800/200)	Predefined rule, VGG-16, MLP	–	✓	✓	–	–	–	–	–
[147] (2018)	LIFULL (20140/2000)	MCS, multi-task VGG-16	–	–	–	–	✓	–	–	✓
[49] (2019)	R2V (715/100), R3D (179/53)	VGG, RCF, DeepLabV3+, PSPNet	–	–	✓	✓	✓	–	–	–
[69] (2019)	CubiCasa5K (4200/400/400)	Modified ResNet-152	✓	–	✓	✓	✓	–	✓	–
[28, 148] (2020)	EAIS (247/25/47), R-FP (500)	DeepLabV3+	✓	–	✓	✓	–	–	✓	✓
[77] (2020)	BTI (700)	U-Net + PixelDCL, Faster R-CNN	✓	✓	✓	✓	–	–	✓	–
[88] (2020)	PFP (1514/40)	U-Net, ResNet, Transformers	✓	✓	–	–	–	–	✓	–
[51] (2020)	CVC-FP (90)	α-shape, linear regression	–	✓	–	–	✓	–	–	–
[72] (2020)	Korea LH (2400/1030)	DeepLabV3+	✓	–	✓	✓	✓	–	–	–
[8] (2020)	CubiCasa5K (480/60)	FCN-2s, DeepLabV3+	–	–	✓	–	–	–	–	–
[27] (2020)	CVC-FP (122)	Mask-RCNN	✓	–	✓	✓	✓	–	✓	–
[149] (2020)	Defined in paper (3500/500/1000)	YOLOv3	–	–	–	✓	✓	–	–	–
[50] (2020)	R2V (815), R3D (232)	GAN	–	–	✓	✓	✓	–	–	–
[10, 89] (2021)	EAIS (400/50), CVC-FP (122)	Pix2Pix, multi-task DL	✓	–	✓	–	–	–	✓	–
[79] (2021)	ZSCVFP (8800/2000)	EdgeGAN, GNN	–	–	✓	–	–	–	✓	✓
[83] (2021)	RUB (74)	GAT GNN	–	–	–	✓	–	–	–	✓
[93] (2021)	CubiCasa5K (200/200), Defined in paper (7)	GNN	✓	✓	✓	✓	✓	–	✓	✓
[80] (2021)	RFP (5600/1400), R3D (232), CubiCasa5K (5000)	YOLOv4, DeepLabV3+, FCN	–	–	✓	✓	✓	✓	✓	–
[150] (2021)	CubiCasa5K (5000)	Cascade Mask-RCNN	–	–	✓	–	✓	–	–	–
[151] (2021)	LIFULL (3800/500/500)	DeepLabV3+	–	–	✓	✓	✓	–	–	✓
[136] (2021)	CVC-FP (122), R3D (215)	Bagging SVM PU-Learning	–	–	✓	–	–	–	–	–
[127] (2021)	Defined in paper (–)	Type-I FRBS	–	✓	✓	–	–	–	–	–
[140] (2021)	Defined in paper (–)	Interval Type-2 FRBS	–	–	✓	–	–	–	–	–
[37] (2021)	ROBIN/REDA (5610)	Predefined ruke, Faster R-CNN, YOLO	✓	✓	✓	✓	–	–	–	–
[135] (2021)	Defined in paper (30/30)	Predefined rule, TensorFlow Object detection API	–	–	✓	✓	✓	–	✓	✓
[81] (2021)	RuralHomeData (700/100), R2V (770/100), CubiCasa5K (800/100)	VGG-16, U-Net, SSD	–	–	✓	✓	✓	✓	–	✓

[a] Format: (total), (train/test), (train/val/test) [b] The dataset considered data augmentation [c] Graphics separation [d] Door/Window/Furniture/Others
[e] OCR or Dimensions were recognized [f] Vectorization [g] Modeling (Graph, other)

to capture the context of the image. It comprises several convolutional and max-pooling layers, which reduce the input image size by subsampling with kernel stride, capturing finer grain structures from the input image as they have a smaller receptive field [144]. In opposition, the decoder section comprises many feature channels that enable precise localization through the transposed convolutions, propagating context information to higher resolution layers, giving the segmented output from the generated classification feature maps.

Similar to FCNs, in U-Net (Figure 9), the decoder also combines the feature and spatial information through a sequence of up-convolutions and concatenations with high-resolution features obtained from the encoder, improving localization and reconstruction of the segmented output image while keeping the underlying structure. Therefore, the expansive path is symmetric to the contracting part, yielding an u-shaped architecture [155]. Likewise, DeepLab is a semantic segmentation model which employs a pre-trained CNN to get encoded feature maps from the input and a decoder to reconstruct the segmented output image. Among their different versions, DeepLabV3+ has achieved state-of-the-art results, famous for its stacked atrous (i.e., dilated) convolutions, enlarging the kernel's field-of-view to extract long-distance features. Finally, the instance segmentation R-CNN is a family of models which produces a set of bounding boxes for each object in the image, named regions of interests (ROIs), where the position and category (e.g., wall) are inferred using neural networks.

Concerning the discriminative semantic segmentation problem in floor plan analysis, Dodge *et al.* [65] were the first to propose an FCN-2s model to segment walls and Faster R-CNN to detect objects such as doors, among other five classes. They also implemented OCR to recognize the room size and place furniture scaled to the scene. The wall segmentation experiments conducted in Dodge's work demonstrated the superiority of a CNN-based approach compared with some traditional patch-based models that use standard shallow classifiers like support vector machines [69]; while also proving that CNNs can handle various drawing styles. Yamasaki *et al.* [144] also presented a fully convolutional end-to-end FCN network to label pixels of 12 different object classes. For this purpose, a semantic segmentation was performed, taking as input the images of apartment floor plans, in which spatial relations between elements and room boundaries were ignored; the classified pixels formed a graph to model the structure and measure the structural similarity for apartment retrieval.

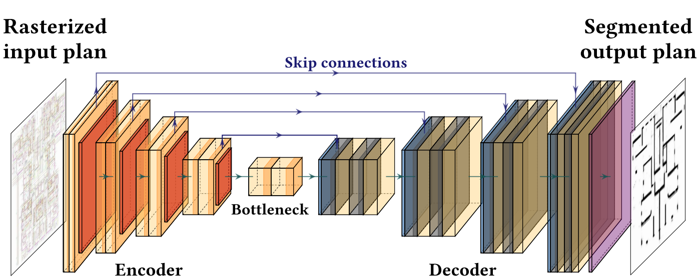


Figure 9: A U-Net model which segments the walls from a rasterized floor plan image. Layer legend: *(yellow)* convolutional block, *(orange)* max-pool, *(blue)* up-sampling, and *(purple)* softmax.

A U-Net approach was introduced by Yang *et al.* [3], where the authors also employed the pixel deconvolutional layers PixelDCL [158] to avoid checkerboard artifacts while segmenting walls and doors. This work was extended by Surikov *et al.* [77], who detected objects with the Faster R-CNN model and proposed statistical methods to vectorize walls, doors, and windows. Morphological operations were used to remove border defects, component filtration to remove connected objects, and the Ramer-Douglas-Peucker algorithm to extract and simplify the room contours. Egiazarian *et al.* [88] obtained the line primitives from floor plan drawings, using U-Net for pre-processing (eliminate background, imperfections, and fill missing parts); then, the resulting images were splitted into patches to independently estimate the line and curve primitives with a feed-forward Artificial Neural Network (ANN). Each patch is encoded with a ResNet-based feature estimator [159] and decoded using Transformer blocks [160] that allow varying the number of output primitives per patch. Predicted primitives were later refined and aligned to the raster image through an optimization procedure. Finally, Lu *et al.* [81] adopted a joint deep neural network approach to extract elements and text simultaneously from an architectural floor plan image, splitted into patches to overcome information loss of wall lines if downsampled. A VGG-16 encoder was considered to get a common feature map and extract latent features of the input image. Then, a U-Net model was used to predict the mask and class of architectural elements, and a pre-trained fast Single Shot Detector (SSD) [161] was considered to retrieve the bounding boxes of room types' text. Predicted per-pixel classes were optimized to remove boundary noise and assign unlabeled adjacent ones, for example, in pixels belonging to a wall that was blurred or partitioned into smaller but connected elements. These classes then fed a mixed-integer quadratic programming algorithm to designate a rectangle for each room beside its type recognized by OCR, leading to the assemble of a room layout graph and the 3D reconstruction of the building.

The DeepLab semantic segmentation models have also been widely used among deep learning approaches. Jang *et al.* [28, 148] segmented walls and doors using the DeepLabV3+ model; centerline [162] and corner [163] algorithms were proposed to vectorize the walls and doors, leading to the assemble of a node-edge graph to describe their position, connectivity, and thickness obtained by a moving kernel method. Seo *et al.* [72] also used DeepLabV3+ to recognize walls, windows, doors, and room types from eight classes; data augmentation techniques were further studied to improve the model results in terms of the IoU metric. Yamada *et al.* [151] conducted semantic segmentation with the DeepLabV3+ model to recognize objects from 14 classes, which was later used to assemble a graph in a rule-based procedure for apartment retrieval. Nodes were created by extracting regions with a particular area, and edges were created between rooms adjacent to the same door or directly adjacent to each other. Finally, Zhu *et al.* [8] compared different training strategies to parse complex floor plans considering the FCN-2s and DeepLabV3+ models for wall segmentation, with VGG-16 as a backbone.

Within instance segmentation models, Faster R-CNN (Figure 10) and YOLO, as well as other anchor-based frameworks, have been used to detect the building elements, as these propose and combine numerous boxes based on the IoU to detect and classify the objects, such as walls, doors, or windows. However, if these general-purpose frameworks are used without further post-processing, the ground-truth inflated boxes and the lack of suitable annotation to describe the complex geometrical characteristic of architectural primitives lead to problems in the localization of sloped and curved walls. Thus, instance segmentation models can only replace some modules of the conventional pipeline. Anchor-free frameworks, such as CenterNet [164] and CornerNet [165], cannot solve this problem either. For such reasons, only anchor-based frameworks were explored within floor plan analysis [79].

From anchor-based frameworks, Wu *et al.* [27] used Mask-RCNN [166] to vectorize the walls by finding a rectangle proposal representing each segment's width, thickness, angle, and location. After simplifying and merging the proposals, an optimization model adjusts its vertex coordinates to resolve inconsistencies from adjacent rectangles such as overlaps and gaps conform to the topological constraints. Although the complex wall layout was described as simply connected segments, the rectangle-based modeling is able to reduce the shape complexity of the segmented regions and can represent the polygons in high accuracy while retaining the connection topology [21]. Murugan *et al.* [150] segmented walls and rooms using the Cascade Mask R-CNN model [167]; wall corners were also detected with a Keypoint Mask R-CNN to improve results after post-processing. The YOLOv3 model [168] was employed by Wang *et al.* [149] to detect doors and windows, alongside the classification of eight types of rooms with the C4.5 decision tree. C4.5 is a tree-like structure method that minimizes the measure of entropy (or impurity) by separating the dataset into smaller classes. On the other hand, Khade *et al.* [37] proposed a scale-invariant algorithm to remove doors, segment walls, and trace the outer shape of the floor plan for Content-Based Image Retrieval (CBIR). Furniture objects from 12 different classes were also detected and classified, wherein Faster R-CNN has a better performance concerning the YOLO model.

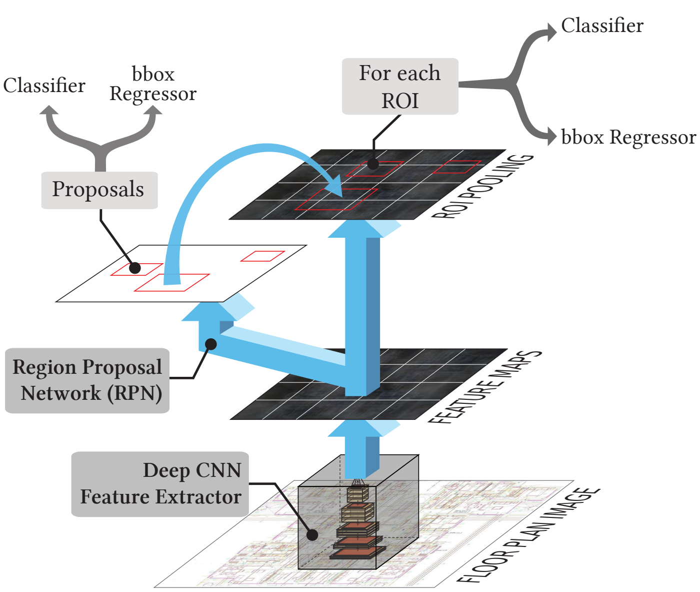


Figure 10: Instance segmentation Faster R-CNN model [94] that considers a floor plan image as input and predicts the position of the objects inside region proposals.

Recently, Lv *et al.* [80] presented a framework that combines the multi-modal information of the floor plan, such as room structure, type, symbols, text, and scale, to recognize and reconstruct its elements. The anchor-based model YOLOv4 [169] is employed to detect the ROIs alongside the text, number, and symbols containing semantic and contextual information like room types, dimensions, or areas. Twelve object classes, and the endpoints of doors, windows, and doorways, were extracted with DeepLabV3+ [156] model. In terms of the model training, the affinity field loss [170] was used to incorporate structural reasoning into semantic segmentation, despite the standard cross-entropy loss that lacks spatial discrimination ability to distinguish between similar or mixed pixels, outperforming previous works [1, 49]. Scale calculation was also implemented to retrieve the size of each object; for such an aim, dimension lines were detected by obtaining its endpoints with a modified FCN network, matched with the recognized length texts by YOLO. Finally, a room vectorization algorithm was proposed that considered room contour and wall centerline optimization, leading to the 3D reconstruction of each floor plan image.

Some works do not consider a segmentation pipeline but propose CNNs to capture spatial features to reconstruct the objects. For instance, Liu *et al.* [1] introduced a deep-learning CNN model to vectorize the plans. The pixel-wise semantic ResNet-152 network was applied to detect junction points of interior and exterior walls, considering a Manhattan assumption, that is, it only can recognize horizontal or vertical walls due to the use of a template matching technique. These detected objects fed an integer programming (IP) method to construct the vector data by finding the optimal primitive pair that correctly represented walls and openings such as doors or windows, leading to the assembly of the rooms. Despite their drawbacks, the major finding was that deep neural networks could act as an effective precursor to the final post-processing heuristics to restore the floor plan elements, including their geometry and semantics. Liu's work was further extended by Kalervo *et al.* [69], who also proposed a modified ResNet-152 model to detect wall junctions, rooms, and icons, obtaining better results as they applied a trainable module [171] for tuning the relative weights between the multi-task loss terms; similarly, these outputs were employed to vectorize the floor plan. Another example is Zeng *et al.* [49], who proposed a deep multi-task neural network to predict room-boundary objects (walls, doors, or windows) and room types. A shared VGG encoder [134] was used for feature extraction and two separate VGG decoders to perform both tasks, recognizing individual elements considering their spatial relationship and a room-boundary guided attention mechanism to enhance the pixel classification performance of the floor plan image. The results were compared against the RCF edge detection model [172], DeepLabV3+, and PSPNet [173] segmentation networks, obtaining better results.

Graph neural networks (GNN) have also been studied to model and classify the floor plan objects, describing a way to express the nodes' order and connectivity learned from the dataset structure [174, 175]. GNNs have undergone rapid development in recent years as convolution was introduced to update the latent node vector (GCN) or by studying graph operations such as aggregation or combination powered by deep neural networks [176]. Like other DL models, it extracts and compares a unique embedding vector of each entity in the target dataset to predict a result as close as possible to the label data [93]. The domain of interest of GNN varies, including nodes, edges, graphs, and subgraphs, and has been widely applied in the area, for example, to generate floor plans [35] or for architectural symbol-detection tasks [39].

Among GNN approaches, Simonsen *et al.* [82] implemented a

GNN-based model to classify the nodes of a large rasterized CAD image as door or non-door. On the other hand, Song and Yu [93] developed a framework to vectorize the floor plan objects considering a GNN for object classification. First, a pre-processing task erased texts and binarized the raster plan. The processed image is then vectorized, relying on its closed regions, and converted to a region adjacency graph according to their adjacent relationship with neighboring polygons. The graph is then fed to an inductive learning-based GNN, which compares multiple floor plan graphs and performs node classification by analyzing inherent features and the relationships, such as the distance. Despite its good performance while classifying elements, the proposed GNN approach, unlike those CNN-based, is not robust to noise and resolution changes.

3.3.2.2 Generative-based models

Ever since Goodfellow *et al.* [177] presented the generative adversarial network (GAN) in 2014, there has been tremendous development in generative models and neural style transfer [178, 179]. By providing training data in pairs, the algorithm finds the most suitable parameters in the network so that the discriminator has the least potential to distinguish the generated data from the original one [145]. GAN has sprouted many branches, including conditional GAN [180, 181], Wasserstein GAN [182], or Pix2Pix [183], and has been used successfully in image translation, style migration, denoising, superresolution and repair, image matting, semantic segmentation, and dataset expansion [184, 185].

From related work, one of the GAN applications is for recognizing structural objects. Zhang *et al.* [50] created direction-aware, learnable, and additive kernels to optimize the recognition of complex and irregular walls through the context module and convolutional blocks of a multi-task GAN-based neural network, improving accuracy and segmentation results of the objects (wall, door, window, and rooms). Despite this example, most researchers considered GANs for image style transfer, as it offers the capability to uniform the level of detail from varied types of drawings, leading to the recognition of primitives from complicated and overlapping graphics.

Recently, image style transfer models have improved remarkably with the development of GANs; among them, the deep networks such as Conditional GANs (cGANs) [180, 181], CycleGANs [186], and DiscoGANs [187] have gained a great reputation. cGANs and CycleGANs transfer images into different styles while preserving the underlying structure, whereas DiscoGANs focus primarily on their texture [89]. The cGANs model assumes that labeled pairs exist within the dataset, turning the original generation process into a conditional one. In this aspect, labeled data, such as one-hot vectors, two-dimensional images, or even three-dimensional models, provides hints to guide the training process; once it runs toward an unexpected direction, punishment will be given to correct its tendency according to the additional information [145]. Thus, cGANs learn the forward mapping, that is, $y = G(x)$, where x belongs to the input, y to the output, and G to the generative model. On the other hand, CycleGANs and DiscoGANs aim to transfer the style between domains even when their images are not paired [89], learning from a two-cycle mapping, i.e., $x = F(y') = F\big(G(x)\big)$ and $y = G(x') = G\big(F(y)\big)$, with the input x and output y unpaired. Although CycleGANS have a wider range of general-purpose applications [79] as it does not require a pixel-level annotation for the images, which can be extremely expensive, the lack of large-scale datasets imposes a difficult restriction for its usage within floor plan analysis. Therefore, only conditional GANs have been used so far.

One important milestone of GAN for image translation is Pix2Pix introduced by Isola *et al.* [183], developed from cGAN [181] using an encoder-decoder architecture for the generator, for example, the “U-Net” model. Pix2Pix was designed to become a general-purpose solution to translate an image between two domains with the same settings, corresponding, in other words, to a pixel-by-pixel mapping. For instance, Isola’s group originally employed Pix2Pix to generate: (1) a real photo from a partly-damaged one, (2) a colorful map from a black-and-white map, and (3) an image with texture and shadow from a linear sketch [145]. Based on Pix2Pix, Wang *et al.* [188] presented Pix2PixHD, expanding its capabilities to handle high-resolution image synthesis and semantic manipulation (from original 256*256 to 2048*1024) by introducing a new robust adversarial learning objective together with new multi-scale generator and discriminator architectures [79].

Concerning the recognition and generation of floor plans, Huang and Zheng [145] introduced an application of Pix2PixHD [188] to detect rooms from 8 classes and then colorize them to generate a new image. In this example, the conditional GANs lead to translating the raster plan to a segmented style using annotated pairs, classifying each pixel while preserving the image’s underlying structure. Pix2Pix was also adopted by Kim *et al.* [10, 89] to transform plans into a unified format [183]. In their study, a multi-task deep learning network transferred the style and simultaneously extracted the wall junction features (Liu *et al.* [1]), considering a Manhattan assumption. These outputs were used to assemble the wall’s vector format through a combinatorial optimization that represents a structure similar to the style-transferred plan, while satisfying the semantic constraints from the floor layout.

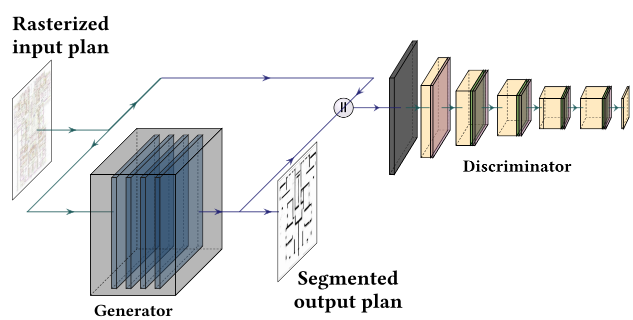


Figure 11: Pix2Pix model that translates the rasterized floor plan image style into a segmented format.

Finally, Dong *et al.* [79] developed an edge extraction GAN, named EdgeGAN, to detect walls based on Pix2Pix. EdgeGAN projects the floor plans into a Primitive Feature Map (PFM); each channel contains some lines representing one category of primitives, leading to the vectorization of walls in an end-to-end manner. Two inspection modules were also proposed to check the connectivity and consistency of PFM based on the Subspace Connective Graph (SCG). The first module contains four criteria that correspond to the sufficient conditions of a fully connected graph. The second module classifies the category of all subspaces via one single graph neural network, which should be consistent with the text annotations in the original floor plan.

4 Challenges and opportunities

Automatic floor plan analysis has seen remarkable progress over the last few years with the help of DL models. Several novel ideas have been proposed (such as recognizing the wall joints for vectorizing the plan, the use of generative image-to-image networks to convert plans to a unified format, or the application of custom losses to incorporate structural reasoning into the segmentation phase), which lead improving the recognition metrics concerning rule-based models. However, despite the progress, there are still challenges to be addressed. Thus, in the following paragraphs, we present and discuss some of these challenges, which can guide future development in the field:

Standardization of result analysis. Although each reviewed article contemplated the evaluation of its methodologies, the lack of a standard procedure makes it difficult to compare with other similar works. Several metrics have been used even to check the results of the same tasks, and many use custom ones that fit their specific purposes. Table 4 groups the typical metrics used throughout reviewed works; typically, segmentation results were evaluated in terms of the intersection over union (IoU) [141], pixel/class accuracy, and the Jaccard Index (JI) proposed by de Las Heras *et al.* [52]. By contrast, works that detected objects (e.g., walls, doors, windows) used the mean average precision (mAP), the recall & precision, the match score (*MS*), detection rate (DR), and recognition accuracy (RA) [189], or considered a confusion matrix.

Table 4: Common metrics used to evaluate results among reviewed articles.

Evaluation metric	Article
Intersection over Union (IoU)	[3, 8, 28, 49, 50, 65, 69, 72, 77, 80–82, 88, 127, 140, 151, 190]
Pixel/Class Accuracy	[1, 3, 49–52, 65, 69, 79–82, 93, 133, 135, 144, 147, 149–151]
Jaccard Index (JI)	[2, 11, 65, 129, 130, 136]
Mean Average Precision (mAP)	[37, 77, 81]
Precision	[27, 49, 82, 146, 147, 149, 150, 190]
Recall	[1, 27, 49, 69, 82, 129, 130, 135, 146, 147, 150, 190]
Match Score (MS)	[10, 27, 52, 86, 114, 117, 131]
Detection Rate (DR)	[9, 10, 86, 114, 117, 131]
Recognition Accuracy (RA)	[10, 86, 114, 117, 131]
Confusion Matrix	[2, 28, 79, 145, 147, 149]

Besides the multiple evaluation metrics used, there is no shared annotation for complicated floor plan datasets, which are fundamental barriers for learning-based approaches to compare each other [10]. Private datasets, popular in the last few years, further complicate this issue [140]. Thus, there is an urgent need to standardize how analysis is performed on each task. A common metric, which also requires a standard representation of the plan annotation, allows comparing the models and choosing the one with better results for a particular plan style and task.

New public datasets. Like in many other computer vision tasks, datasets play an essential role within automatic floor plan analysis. These documents define the geometrical, topological, and semantical information of plan objects in a highly correlated fashion, following strict restrictions such as usability, layout, and regulation [2, 3]. New datasets can provide researchers with more possible styles for the models to handle, especially if future learning-based methodologies are toward a style-independent trend.

Another major problem regarding datasets is that most current public ones consider only houses or apartments (Figure 2); however, the construction and design scope is extensive. For instance, many architectural offices process multi-unit plans, offices, and public buildings like universities and hospitals. These examples are more complex than currently available datasets; it is just a massive scale difference. Incorporating such samples can expand the scope of floor plan analysis, enabling the process of different sources among the industry. Figure 12 illustrates a sample of a multi-unit raster floor plan; unlike those presented in the reviewed datasets (Table 1), this plan has complex walls, more furniture, and new semantics. For example, some walls separate two rooms of different apartments, which together constitute the perimeter of the building. This new level of complexity is not explicit, but it is only apparent when processing the plan as a whole.

Figure 12: Example of a rasterized multi-unit floor plan [21].

Non-supervised DL models. Annotating floor plans is difficult and expensive. There is no standard notation, and some examples offer ambiguous situations that are difficult even for experts [9, 52]. For example, consider a plan in black-and-white color where windows and beams have the same annotation; in such a case, these objects are only differentiable concerning a structural perspective. Hence, non-supervised models allow the analysis without annotating the floor plans. Currently, few works have proposed unsupervised methods [11, 130, 131]; nevertheless, they fall into strict assumptions or rely on complex learning rules. In this sense, DL can help to learn these relationships and structural reasoning for recognizing new complex objects for upcoming plan styles and can enable the analysis of unexploited datasets.

Combination of rule-based and learning-based methods. While learning-based algorithms have revolutionized the plan recognition area, they still have problems in solving tasks that require a deterministic response or fine detail. Therefore, the combination of rule-based models and learning can offer the best of both worlds. The former solves fine details that are difficult to capture by a DL model because they are infrequent or require a high refinement level, such as polygon resolution, the detailing of certain sections of complex geometry, or the recognition of custom objects. The latter allows solving common problems that require specific rules such as segmentation or vectorization. Both mechanisms are not mutually exclusive and can be leveraged.

Trending applications within the industry. As architectural floor plans are one of the key products in architectural firms or structural engineering offices, the algorithms that can analyze and process them in batches have many applications, as they allow to automatize pipelines in recognition, vectorization, modeling, or searching in large databases.

One of the most active research areas belongs to BIM and 3D reconstruction, as these technologies help to improve productivity and reduce costs in different stages of the building lifecycle, especially in the early ones, requiring less paperwork to visualize or edit the projects. Also, in recent years, governments and private companies have started a more data-driven approach because models are composed of several elements that contain information about their properties and relationships with others, facilitating interdisciplinary work [191]. Despite benefits, BIM and 3D models are costly and time-consuming to produce [13], particularly if the only available documentation is 2D scanned images of their paper floor plans [127]. Therefore, one critical short-term research challenge in the renovation scope is to devise effective and reliable methods and tools to reconstruct the digital models of existing buildings [2]. In BIM and 3D reconstruction areas, algorithms have been developed since the early 2000s to recognize and vectorize building shapes from walls, beams, and slabs [106]. Zhao *et al.* [15] recently implemented a framework to assemble a BIM representation from CAD files, employing revised deep learning techniques such as Faster R-CNN and YOLO. In this aspect, the accuracy and generalization ability to process plans in different styles are critical aspects of research.

Indoor data models, maps, and spatial information is another widely studied area of application, with a globally growing market that is predicted to expand from $2.6bn in 2017 to $43bn by 2025 [82]. For such reason, research has been conducted on generating indoor spatial information from various data such as LiDAR (Light Detection and Ranging), BIM, and 2D floor plans [28]. As rasterized floor plans are more accessible compared to other sources but discard semantic and topological metadata [8], the revised algorithms of this review paper can be employed to re-assemble this representation by spotting, retrieving, and vectorizing the founding components of indoor maps automatically, like doors, walls, corridors, and furniture, or by detecting the usage of rooms, avoiding time-consuming human labor. Naturally, automatic procedures come with several challenges, mainly caused by the diversity of floor plans and the flexibility of preferences in visual styles, symbols, and topology [27].

Building search retrieval is another example of an application with growing interest. Because of the increasing demand for apartment search, the emergence of online platforms has made this task easier. Nevertheless, most only provide information regarding location, monthly rent, or room size, but little information on plan structure [144], which turns the searching process into a tedious task [37]. For such reasons, research community has developed several tools to simplify this process. Examples include the use of graphs to query similar floor plans among large databases [12, 37, 38], the search based on hand-made sketches [41–43], the use of natural language to describe the plan layout [24], the audio feedback mechanisms to help visually-impaired people navigate floor plans [45], the development of VR experiences for customers to explore real estates [25], or the use of AI networks to valuate them [44]. Due to the massive amount of data and the variety of styles, reviewed machine learning solutions, like CNN and GNN, have been used extensively to describe, extract, and query the meaningful features of the floor plans, allowing developers to create accessible and easy-to-use tools to customers, enhancing the overall experience.

Structural analysis is another area where floor plan research algorithms can be applied. New machine learning models can be trained to automatically ensemble a structural floor plan from an architectural image, predicting new walls and computing its members' thickness, length, and displacement [22, 23]. For such reason, there is a considerable need for processed datasets that consider a wide range of architectural styles and layouts to train these algorithms. By this means, discriminative deep-learning models, for example R-CNN, can be employed to transform rasterized plans into a rectangle-based representation [21] to compute features, avoiding expensive manual labeling. These upcoming solutions can simplify the decision processes, reduce costs, and improve productivity, while also adding value to the already manufactured plans, which can now be employed to develop data-driven models and improve the production lines of the structural engineering offices.

5 Conclusions

This study reviewed related work within architectural floor plan analysis that used rasterized images to automatically retrieve objects like walls, doors, windows, and rooms. In the following, the major findings related to each research question are summarized.

Concerning the revised methodologies, authors have traditionally considered rule-based methods that exploited low-level heuristics to retrieve the desired objects in the plans, generally by solving four common tasks: (1) *Graphics separation*, which removes undesirable elements from plans, (2) *Object recognition*, which recognizes the building elements from the image, (3) *Vectorization*, a process that transforms the objects to a vector form, and (4) *Structural modeling*, which transforms the floor objects to a mathematical model. Most methods that employ manual rules to solve these tasks are restricted to the datasets researchers used, as plans can have different styles, semantics, layouts, and inner-correlations, limiting the range these rules can handle.

Since the plans are complex, diverse in style, but difficult to access and produce, rule-based algorithms kept a limited development until learning-based algorithms were introduced. Unlike the previous ones, these methodologies automatically learn the relationship between the floor plan elements, exploiting low and high-level features directly from the training data, composed of dozens of validated plans from a wide variety of styles. Thus, the thresholds, limitations, and rules embedded in these documents are inferred. From the learning approach, deep learning, especially those related to neural networks, has reached the analysis to the state-of-the-art results in terms of accuracy and other metrics such as the intersection over union. These models can compute complex and non-trivial features from the plan data, which are challenging to reproduce manually, and usually avoids graphic separation as the raw plan images can be used without further pre-processing.

Even though remarkable results have been achieved in the last few years, floor plan analysis is still considered an open task within computer vision. For instance, rule-based algorithms rely on particular plan styles, being hard to generalize for other formats. On the other hand, learning-based models trained on various datasets might have great adaptability, but their outputs are usually blurry as they perform pixel-level segmentation or can have significant differences if an object is missing. Also, learning models require a high number of plans to train and generalize the results; and this can be extremely expensive or unnecessary if only a couple of plans have to be processed. Thus, future work is needed to achieve an accurate and style-independent recognition.

Despite the difficulties and limitations, current methods have multiple applications within the construction and design industry, improving productivity, simplifying the design processes, and reducing costs.

Acknowledgments

This work was funded by the project Fondecyt Regular 2021 Nº 1211484 and the Department of Computer Science, Universidad de Chile.

References

[1] C. Liu, J. Wu, P. Kohli, Y. Furukawa, Raster-to-vector: revisiting floorplan transformation, in 2017 IEEE International Conference on Computer Vision, (Venice, Italy), pp. 2195–2203, IEEE, 2017, doi:10.1109/ICCV.2017.241.

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

[3] J. Yang, H. Jang, J. Kim, J. Kim, Semantic segmentation in architectural floor plans for detecting walls and doors, in 2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, (Beijing, China), pp. 1–9, IEEE, 2018, doi:10.1109/CISP-BMEI.2018.8633243.

[4] H. K. Stokes, An examination of the productivity decline in the construction industry, The Review of Economics and Statistics, vol. 63, p. 495, 1981, doi:10.2307/1935844.

[5] G. de Valence, M. Abbott, A review of the theory and measurement techniques of productivity in the construction industry, in Measuring Construction. Prices, Output and Productivity, ch. 12, p. 288, Routledge, 2015, doi:10.4324/9781315882925.

[6] M. Abdel-Wahab, B. Vogl, Trends of productivity growth in the construction industry across Europe, US and Japan, Construction Management and Economics, vol. 29, pp. 635–644, 2011, doi:10.1080/01446193.2011.573568.

[7] J. Park, Y.-B. Kwon, Main wall recognition of architectural drawings using dimension extension line, in Graphics Recognition. Recent Advances and Perspectives. Lecture Notes in Computer Science, vol 3088, pp. 116–127, Springer, Berlin, Heidelberg, 2003, doi:10.1007/978-3-540-25977-0_11.

[8] R. Zhu, J. Shen, X. Deng, M. Walldén, F. Ino, Training strategies for CNN-based models to parse complex floor plans, in Proceedings of the 2020 9th International Conference on Software and Computer Applications, (New York, NY, USA), pp. 11–16, ACM, 2020, doi:10.1145/3384544.3384566.

[9] 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, (New York, NY, USA), pp. 167–174, ACM Press, 2010, doi:10.1145/1815330.1815352.

[10] S. Kim, S. Park, H. Kim, K. Yu, Deep floor plan analysis for complicated drawings based on style transfer, Journal of Computing in Civil Engineering, vol. 35, p. 04020066, 2021, doi:10.1061/(ASCE)CP.1943-5487.0000942.

[11] L.-P. de las Heras, D. Fernandez, E. Valveny, J. Llados, G. Sánchez, Unsupervised wall detector in architectural floor plans, in 2013 12th International Conference on Document Analysis and Recognition, (Washington, DC, USA), pp. 1245–1249, IEEE, 2013, doi:10.1109/ICDAR.2013.252.

[12] D. Sharma, N. Gupta, C. Chattopadhyay, S. Mehta, DANIEL: a deep architecture for automatic analysis and retrieval of building floor plans, in 2017 14th IAPR International Conference on Document Analysis and Recognition, (Kyoto, Japan), pp. 420–425, IEEE, 2017, doi:10.1109/ICDAR.2017.76.

[13] L. Gimenez, J.-L. Hippolyte, S. Robert, F. Suard, K. Zreik, Review: reconstruction of 3D building information models from 2D scanned plans, Journal of Building Engineering, vol. 2, pp. 24–35, 2015, doi:10.1016/j.jobe.2015.04.002.

[14] Q. Lu, S. Lee, A semi-automatic approach to detect structural components from CAD drawings for constructing as-is BIM objects, in Computing in Civil Engineering 2017, (Reston, VA), pp. 84–91, American Society of Civil Engineers, 2017, doi:10.1061/9780784480823.011.

[15] Y. Zhao, X. Deng, H. Lai, Reconstructing BIM from 2D structural drawings for existing buildings, Automation in Construction, vol. 128, p. 103750, 2021, doi:10.1016/j.autcon.2021.103750.

[16] X. Yin, P. Wonka, A. Razdan, Generating 3D building models from architectural drawings: a survey, IEEE Computer Graphics and Applications, vol. 29, pp. 20–30, 2009, doi:10.1109/MCG.2009.9.

[17] Q. Wen, R.-G. Zhu, Automatic generation of 3D building models based on line segment vectorization, Mathematical Problems in Engineering, vol. 2020, pp. 1–16, 2020, doi:10.1155/2020/8360706.

[18] V. Machairas, A. Tsangrassoulis, K. Axarli, Algorithms for optimization of building design: a review, Renewable and Sustainable Energy Reviews, vol. 31, pp. 101–112, 2014, doi:10.1016/j.rser.2013.11.036.

[19] M. Zawidzki, J. Szklarski, Multi-objective optimization of the floor plan of a single story family house considering position and orientation, Advances in Engineering Software, vol. 141, p. 102766, 2020, doi:10.1016/j.advengsoft.2019.102766.

[20] A.-T. Nguyen, S. Reiter, P. Rigo, A review on simulation-based optimization methods applied to building performance analysis, Applied Energy, vol. 113, pp. 1043–1058, 2014, doi:10.1016/j.apenergy.2013.08.061.

[21] P. N. Pizarro, L. M. Massone, Structural design of reinforced concrete buildings based on deep neural networks, Engineering Structures, vol. 241, p. 112377, 2021, doi:10.1016/j.engstruct.2021.112377.

[22] P. N. Pizarro, L. M. Massone, F. R. Rojas, R. O. Ruiz, Use of convolutional networks in the conceptual structural design of shear wall buildings layout, Engineering Structures, vol. 239, p. 112311, 2021, doi:10.1016/j.engstruct.2021.112311.

[23] X. Lu, W. Liao, Y. Zhang, Y. Huang, Intelligent structural design of shear wall residence using physics-enhanced generative adversarial networks, Earthquake Engineering & Structural Dynamics, 2022, doi:10.1002/eqe.3632.

[24] S. Goyal, C. Chattopadhyay, G. Bhatnagar, Knowledge-driven description synthesis for floor plan interpretation, International Journal on Document Analysis and Recognition, vol. 24, pp. 19–32, 2021, doi:10.1007/s10032-021-00367-3.

[25] G. Gerstweiler, L. Furlan, M. Timofeev, H. Kaufmann, Extraction of structural and semantic data from 2D floor plans for interactive and immersive VR real estate exploration, Technologies, vol. 6, p. 101, 2018, doi:10.3390/technologies6040101.

[26] E. Lewandowicz, P. Lisowski, Methodology to generate navigation models in building, Journal of Civil Engineering and Management, vol. 24, pp. 619–629, 2018, doi:10.3846/jcem.2018.6599.

[27] 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, vol. 35, pp. 1205–1231, 2020, doi:10.1080/13658816.2020.1781130.

[28] H. Jang, K. Yu, J. Yang, Indoor reconstruction from floorplan images with a deep learning approach, ISPRS International Journal of Geo-Information, vol. 9, p. 65, 2020, doi:10.3390/ijgi9020065.

[29] G. Pintore, C. Mura, F. Ganovelli, L. Fuentes-Perez, R. Pajarola, E. Gobbetti, State-of-the-art in automatic 3D reconstruction of structured indoor environments, Computer Graphics Forum, vol. 39, pp. 667–699, 2020, doi:10.1111/cgf.14021.

[30] B. P. Mathew, S. Dharan, Review on room layout estimation from a single image, International Journal of Engineering Research and Technology, vol. V9, 2020, doi:10.17577/IJERTV9IS060820.

[31] Z. Kang, J. Yang, Z. Yang, S. Cheng, A review of techniques for 3D reconstruction of indoor environments, ISPRS International Journal of Geo-Information, vol. 9, p. 330, 2020, doi:10.3390/ijgi9050330.

[32] H. Fang, F. Lafarge, C. Pan, H. Huang, Floorplan generation from 3D point clouds: a space partitioning approach, ISPRS Journal of Photogrammetry and Remote Sensing, vol. 175, pp. 44–55, 2021, doi:10.1016/j.isprsjprs.2021.02.012.

[33] C. Liu, J. Wu, Y. Furukawa, FloorNet: a unified framework for floorplan reconstruction from 3D scans, in Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science, vol 11210, pp. 203–219, Springer, Cham, 2018, doi:10.1007/978-3-030-01231-1_13.

[34] W. Wu, X.-M. Fu, R. Tang, Y. Wang, Y.-H. Qi, L. Liu, Data-driven interior plan generation for residential buildings, ACM Transactions on Graphics, vol. 38, pp. 1–12, 2019, doi:10.1145/3355089.3356556.

[35] R. Hu, Z. Huang, Y. Tang, O. Van Kaick, H. Zhang, H. Huang, Graph2Plan: learning floorplan generation from layout graphs, ACM Transactions on Graphics, vol. 39, 2020, doi:10.1145/3386569.3392391.

[36] V. Azizi, M. Usman, H. Zhou, P. Faloutsos, M. Kapadia, Graph-based generative representation learning of semantically and behaviorally augmented floorplans, The Visual Computer, 2021, doi:10.1007/s00371-021-02155-w.

[37] R. Khade, K. Jariwala, C. Chattopadhyay, U. Pal, A rotation and scale invariant approach for multi-oriented floor plan image retrieval, Pattern Recognition Letters, vol. 145, pp. 1–7, 2021, doi:10.1016/j.patrec.2021.01.020.

[38] D. Sharma, C. Chattopadhyay, High-level feature aggregation for fine-grained architectural floor plan retrieval, IET Computer Vision, vol. 12, pp. 702–709, 2018, doi:10.1049/iet-cvi.2017.0581.

[39] G. Renton, P. Héroux, B. Gaüzère, S. Adam, Graph neural network for symbol detection on document images, in 2019 International Conference on Document Analysis and Recognition Workshops, (Sydney, NSW, Australia), pp. 62–67, IEEE, 2019, doi:10.1109/ICDARW.2019.00016.

[40] A. Rezvanifar, M. Cote, A. B. Albu, Symbol spotting on digital architectural floor plans using a deep learning-based framework, in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, (Seattle, WA, USA), pp. 2419–2428, IEEE, 2020, doi:10.1109/CVPRW50498.2020.00292.

[41] A. Shio, Y. Aoki, Sketch plan: a prototype system for interpreting hand-sketched floor plans, Systems and Computers in Japan, vol. 31, pp. 10–18, 2000, doi:10.1002/(SICI)1520-684X(200006)31:6<10::AID-SCJ2>3.0.CO;2-Y.

[42] M. Weber, M. Liwicki, A. Dengel, a.SCAtch - a sketch-based retrieval for architectural floor plans, in 2010 12th International Conference on Frontiers in Handwriting Recognition, (Kolkata, India), pp. 289–294, IEEE, 2010, doi:10.1109/ICFHR.2010.122.

[43] S. Ahmed, M. Weber, M. Liwicki, C. Langenhan, A. Dengel, F. Petzold, Automatic analysis and sketch-based retrieval of architectural floor plans, Pattern Recognition Letters, vol. 35, pp. 91–100, 2013, doi:10.1016/j.patrec.2013.04.005.

[44] P. Nkolika J., H. I. Okagbue, E. C. Obasi, A. O. Akinola, Review on the application of artificial

neural networks in real estate valuation, International Journal of Advanced Trends in Computer Science and Engineering, vol. 9, pp. 2918 – 2925, 2020, doi:10.30534/ijatcse/2020/66932020.
[45] C. Goncu, A. Madugalla, S. Marinai, K. Marriott, Accessible on-line floor plans, in Proceedings of the 24th International Conference on World Wide Web, (Republic and Canton of Geneva, Switzerland), pp. 388–398, International World Wide Web Conferences Steering Committee, 2015, doi:10.1145/2736277.2741660.
[46] C. Riedinger, M. Jordan, H. Tabia, 3D models over the centuries: from old floor plans to 3D representation, in 2014 International Conference on 3D Imaging, pp. 1–8, IEEE, 2014, doi:10.1109/IC3D.2014.7032583.
[47] H. Tabia, C. Riedinger, M. Jordan, Automatic reconstruction of heritage monuments from old architecture documents, Journal of Electronic Imaging, vol. 26, p. 011006, 2016, doi:10.1117/1.JEI.26.1.011006.
[48] W. Swaileh, D. Kotzinos, S. Ghosh, M. Jordan, N.-S. Vu, Y. Qian, Versailles-FP dataset: wall detection in ancient floor plans, in Document Analysis and Recognition – ICDAR 2021. ICDAR 2021. Lecture Notes in Computer Science, vol 12821, pp. 34–49, Springer, Cham, 2021, doi:10.1007/978-3-030-86549-8_3.
[49] Z. Zeng, X. Li, Y. K. Yu, C.-W. Fu, Deep floor plan recognition using a multi-task network with room-boundary-guided attention, 2019 IEEE/CVF International Conference on Computer Vision, pp. 9095–9103, 2019, doi:10.1109/ICCV.2019.00919.
[50] Y. Zhang, Y. He, S. Zhu, X. Di, The direction-aware, learnable, additive kernels and the adversarial network for deep floor plan recognition, arXiv, 2020, arXiv:2001.11194.
[51] H. K. Mewada, A. V. Patel, J. Chaudhari, K. Mahant, A. Vala, Automatic room information retrieval and classification from floor plan using linear regression model, International Journal on Document Analysis and Recognition, vol. 23, pp. 253–266, 2020, doi:10.1007/s10032-020-00357-x.
[52] L.-P. de las Heras, S. Ahmed, M. Liwicki, E. Valveny, G. Sánchez, Statistical segmentation and structural recognition for floor plan interpretation, International Journal on Document Analysis and Recognition, vol. 17, pp. 221–237, 2014, doi:10.1007/s10032-013-0215-2.
[53] K. Krippendorff, Content analysis: an introduction to its methodology. Sage, 2012, https://refhub.elsevier.com/S0141-0296(17)33552-6/h0105. Accessed 16 April 2022.
[54] H. Salehi, R. Burgueño, Emerging artificial intelligence methods in structural engineering, Engineering Structures, vol. 171, pp. 170–189, 2018, doi:10.1016/j.engstruct.2018.05.084.
[55] M. Rusiñol, A. Borràs, J. Lladós, Relational indexing of vectorial primitives for symbol spotting in line-drawing images, Pattern Recognition Letters, vol. 31, pp. 188–201, 2010, doi:10.1016/j.patrec.2009.10.002.
[56] M. Rusiñol, A. Borràs, J. Lladós, FPLAN-POLY dataset, 2010, http://www.cvc.uab.es/~marcal/FPLAN-POLY. Accessed 16 April 2022.
[57] M. Delalandre, E. Valveny, T. Pridmore, D. Karatzas, Generation of synthetic documents for performance evaluation of symbol recognition & spotting systems, International Journal on Document Analysis and Recognition, vol. 13, pp. 187–207, 2010, doi:10.1007/s10032-010-0120-x.
[58] M. Delalandre, E. Valveny, T. Pridmore, D. Karatzas, SESYD dataset, 2010, http://mathieu.delalandre.free.fr/projects/sesyd. Accessed 16 April 2022.
[59] 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, vol. 18, pp. 15–30, 2015, doi:10.1007/s10032-014-0236-5.
[60] L.-P. de las Heras, O. R. Terrades, S. Robles, G. Sánchez, CVC-FP dataset, 2015, http://dag.cvc.uab.es/resources/floorplans. Accessed 16 April 2022.
[61] C. Liu, A. G. Schwing, K. Kundu, R. Urtasun, S. Fidler, Rent3D: floor-plan priors for monocular layout estimation, in 2015 IEEE Conference on Computer Vision and Pattern Recognition, (Boston, MA, USA), pp. 3413–3421, IEEE, 2015, doi:10.1109/CVPR.2015.7298963.
[62] C. Liu, A. G. Schwing, K. Kundu, R. Urtasun, S. Fidler, R3D dataset, 2015, https://www.cs.toronto.edu/~fidler/projects/rent3D. Accessed 16 April 2022.
[63] H. Chu, S. Wang, R. Urtasun, S. Fidler, HouseCraft: building houses from rental ads and street views, in Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9910, pp. 500–516, Springer, Cham, 2016, doi:10.1007/978-3-319-46466-4_30.
[64] H. Chu, S. Wang, R. Urtasun, S. Fidler, SydneyHouse dataset, 2016, https://www.cs.toronto.edu/housecraft. Accessed 16 April 2022.
[65] S. Dodge, J. Xu, B. Stenger, Parsing floor plan images, in 2017 Fifteenth IAPR International Conference on Machine Vision Applications, (Nagoya, Japan), pp. 358–361, IEEE, 2017, doi:10.23919/MVA.2017.7986875.
[66] S. Dodge, J. Xu, B. Stenger, R-FP dataset, 2017, https://rit.rakuten.com/data_release. Accessed 16 April 2022.
[67] D. Sharma, N. Gupta, C. Chattopadhyay, S. Mehta, ROBIN dataset, 2017, https://github.com/gesstalt/ROBIN. Accessed 16 April 2022.
[68] C. Liu, J. Wu, P. Kohli, Y. Furukawa, R2V dataset, 2017, https://github.com/art-programmer/FloorplanTransformation. Accessed 16 April 2022.
[69] A. Kalervo, J. Ylioinas, M. Häikiö, A. Karhu, J. Kannala, CubiCasa5K: a dataset and an improved multi-task model for floorplan image analysis, in Image Analysis. SCIA 2019. Lecture Notes in Computer Science, vol 11482, pp. 28–40, Springer, Cham, 2019, doi:10.1007/978-3-030-20205-7_3.
[70] A. Kalervo, J. Ylioinas, M. Häikiö, A. Karhu, J. Kannala, CubiCasa5K dataset, 2019, https://github.com/CubiCasa/CubiCasa5k. Accessed 16 April 2022.
[71] W. Wu, X.-M. Fu, R. Tang, Y. Wang, Y.-H. Qi, L. Liu, RPLAN dataset, 2019, http://staff.ustc.edu.cn/~fuxm/projects/DeepLayout. Accessed 16 April 2022.
[72] J. Seo, H. Park, S. Choo, Inference of drawing elements and space usage on architectural drawings using semantic segmentation, Applied Sciences, vol. 10, p. 7347, 2020, doi:10.3390/app10207347.
[73] K. Land and Housing Corporation, Korea LH dataset, 2019, https://www.data.go.kr/en/data/15037046/fileData.do. Accessed 16 April 2022.
[74] S. Goyal, V. Mistry, C. Chattopadhyay, G. Bhatnagar, BRIDGE: building plan repository for image description generation, and evaluation, in 2019 International Conference on Document Analysis and Recognition, (Sydney, NSW, Australia), pp. 1071–1076, IEEE, 2019, doi:10.1109/ICDAR.2019.00174.
[75] T. Li, D. Ho, C. Li, D. Zhu, C. Wang, M. Q.-H. Meng, HouseExpo: a large-scale 2D indoor layout dataset for learning-based algorithms on mobile robots, in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, (Las Vegas, NV, USA), pp. 5839–5846, IEEE, 2020, doi:10.1109/IROS45743.2020.9341284.
[76] T. Li, D. Ho, C. Li, D. Zhu, C. Wang, M. Q.-H. Meng, HouseExpo dataset, 2020, https://github.com/TeaganLi/HouseExpo. Accessed 16 April 2022.
[77] I. Y. Surikov, M. A. Nakhatovich, S. Y. Belyaev, D. A. Savchuk, Floor plan recognition and vectorization using combination unet, faster-rcnn, statistical component analysis and ramer-douglas-peucker, in Computing Science, Communication and Security. COMS2 2020. Communications in Computer and Information Science, vol 1235, pp. 16–28, Springer, Singapore, 2020, doi:10.1007/978-981-15-6648-6_2.
[78] K. Ministry of Land and Transport, EAIS, electronic architectural administration information system, 2021, https://cloud.eais.go.kr. Accessed 16 April 2022.
[79] S. Dong, W. Wang, W. Li, K. Zou, Vectorization of floor plans based on EdgeGAN, Information, vol. 12, p. 206, 2021, doi:10.3390/info12050206.
[80] X. Lv, S. Zhao, X. Yu, B. Zhao, Residential floor plan recognition and reconstruction, in 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, (Virtual event), pp. 16712–16721, IEEE, 2021, doi:10.1109/CVPR46437.2021.01644.
[81] Z. Lu, T. Wang, J. Guo, W. Meng, J. Xiao, W. Zhang, X. Zhang, Data-driven floor plan understanding in rural residential buildings via deep recognition, Information Sciences, vol. 567, pp. 58–74, 2021, doi:10.1016/j.ins.2021.03.032.
[82] C. P. Simonsen, F. M. Thiesson, M. P. Philipsen, T. B. Moeslund, Generalizing floor plans using graph neural networks, in 2021 IEEE International Conference on Image Processing, (Anchorage, AK, USA), pp. 654–658, IEEE, 2021, doi:10.1109/ICIP42928.2021.9506514.
[83] C. P. Simonsen, F. M. Thiesson, M. P. Philipsen, T. B. Moeslund, RUB dataset, 2021, https://github.com/Chrps/MapGeneralization. Accessed 16 April 2022.
[84] J. National Institute of Informatics, LIFULL HOME'S, 2021, https://www.nii.ac.jp/dsc/idr/en/lifull. Accessed 16 April 2022.
[85] K. Tombre, B. Lamiroy, Pattern recognition methods for querying and browsing technical documentation, in Progress in Pattern Recognition, Image Analysis and Applications. CIARP 2008. Lecture Notes in Computer Science, vol 5197, pp. 504–518, Springer, Berlin, Heidelberg, 2008, doi:10.1007/978-3-540-85920-8_62.
[86] S. Ahmed, M. Liwicki, M. Weber, A. Dengel, Automatic room detection and room labeling from architectural floor plans, in 2012 10th IAPR International Workshop on Document Analysis Systems, (Gold Coast, QLD, Australia), pp. 339–343, IEEE, 2012, doi:10.1109/DAS.2012.22.
[87] J. Ravagli, Z. Ziran, S. Marinai, Text recognition and classification in floor plan images, in 2019 International Conference on Document Analysis and Recognition Workshops, (Sydney, NSW, Australia), pp. 1–6, IEEE, 2019, doi:10.1109/ICDARW.2019.00006.
[88] V. Egiazarian, O. Voynov, A. Artemov, D. Volkhonskiy, A. Safin, M. Taktasheva, D. Zorin, E. Burnaev, Deep vectorization of technical drawings, in Computer Vision – ECCV 2020. ECCV 2020. Lecture Notes in Computer Science, vol 12358, pp. 582–598, Springer, Cham, 2020, doi:10.1007/978-3-030-58601-0_35.
[89] S. Kim, S. Park, K. Yu, Application of style transfer in the vectorization process of floorplans (short paper), 10th International Conference on Geographic Information Science, vol. 114, pp. 39:1–39:6, 2018, doi:10.4230/LIPIcs.GISCIENCE.2018.39.
[90] T. Guo, H. Zhang, Y. Wen, An improved example-driven symbol recognition approach in engineering drawings, Computers & Graphics, vol. 36, pp. 835–845, 2012, doi:10.1016/j.cag.2012.06.001.
[91] K. C. Santosh, Complex and composite graphical symbol recognition and retrieval: a quick review, in Recent Trends in Image Processing and Pattern Recognition. RTIP2R 2016. Communications in Computer and Information Science, vol 709, pp. 3–15, Springer, Singapore, 2017, doi:10.1007/978-981-10-4859-3_1.
[92] E. Elyan, L. Jamieson, A. Ali-Gombe, Deep learning for symbols detection and classification in engineering drawings, Neural Networks, vol. 129, pp. 91–102, 2020, doi:10.1016/j.neunet.2020.05.025.
[93] J. Song, K. Yu, Framework for indoor elements classification via inductive learning on floor plan graphs, ISPRS International Journal of Geo-Information, vol. 10, no. 97, 2021, doi:10.3390/ijgi10020097.
[94] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, pp. 1137–1149, 2017, doi:10.1109/TPAMI.2016.2577031.
[95] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in 2016 IEEE Conference on Computer Vision and Pattern Recognition, (Las Vegas, NV, USA), pp. 779–788, IEEE, 2016, doi:10.1109/CVPR.2016.91.
[96] C. Ah-Soon, K. Tombre, Variations on the analysis of architectural drawings, in Proceedings of the Fourth International Conference on Document Analysis and Recognition, vol. 1, (Ulm, Germany), pp. 347–351, IEEE Comput. Soc, 1997, doi:10.1109/ICDAR.1997.619869.
[97] J. Rendek, G. Masini, P. Dosch, K. Tombre, The search for genericity in graphics recognition applications: design issues of the Qgar software system, in Document Analysis Systems VI. DAS 2004. Lecture Notes in Computer Science, vol 3163, pp. 366–377, Springer, Berlin, Heidelberg, 2004, doi:10.1007/978-3-540-28640-0_35.
[98] B. C. Russell, A. Torralba, K. P. Murphy, W. T. Freeman, LabelMe: a database and web-based tool for image annotation, International Journal of Computer Vision, vol. 77, pp. 157–173, 2008, doi:10.1007/s11263-007-0090-8.
[99] J. Chernef, R. Logcher, J. Connor, N. Patrikalakis, Knowledge-based interpretation of architectural drawings, Research in Engineering Design, vol. 3, pp. 195–210, 1992, doi:10.1007/BF01580842.
[100] N. O. Sönmez, A review of the use of examples for automating architectural design tasks, Computer-Aided Design, vol. 96, pp. 13–30, 2018, doi:10.1016/j.cad.2017.10.005.
[101] O. Hori, S. Tanigawa, Raster-to-vector conversion by line fitting based on contours and skeletons, in Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR '93), (Tsukuba, Japan), pp. 353–358, IEEE Comput. Soc. Press, 1993, doi:10.1109/ICDAR.1993.395716.
[102] Y. Aoki, A. Shio, H. Arai, K. Odaka, A prototype system for interpreting hand-sketched floor plans, in Proceedings of 13th International Conference on Pattern Recognition, vol. 3, (Vienna, Austria), pp. 747–751, IEEE, 1996, doi:10.1109/ICPR.1996.547268.
[103] J. Lladós, J. López-Krahe, E. Martí, A system to understand hand-drawn floor plans using subgraph isomorphism and Hough transform, Machine Vision and Applications, vol. 10, pp. 150–158, 1997, doi:10.1007/s001380050068.
[104] K. Ryall, S. Shieber, J. Marks, M. Mazer, Semi-automatic delineation of regions in floor plans, in Proceedings of 3rd International Conference on Document Analysis and Recognition, vol. 2, (Montreal, QC, Canada), pp. 964–969, IEEE Comput. Soc. Press, 1995, doi:10.1109/ICDAR.1995

.602062.
[105] P. Dosch, G. Masini, Reconstruction of the 3D structure of a building from the 2D drawings of its floors, in Proceedings of the Fifth International Conference on Document Analysis and Recognition, (Bangalore, India), pp. 487–490, IEEE, 1999, doi:10.1109/ICDAR.1999.791831.
[106] P. Dosch, K. Tombre, C. Ah-Soon, G. Masini, A complete system for the analysis of architectural drawings, International Journal on Document Analysis and Recognition, vol. 3, pp. 102–116, 2000, doi:10.1007/PL00010901.
[107] K. Tombre, C. Ah-Soon, P. Dosch, A. Habed, G. Masini, Stable, robust and off-the-shelf methods for graphics recognition, in Fourteenth International Conference on Pattern Recognition, vol. 1, (Brisbane, Australia), pp. 406–408, IEEE Comput. Soc, 1998, doi:10.1109/ICPR.1998.711167.
[108] C. Ah-Soon, A constraint network for symbol detection in architectural drawings, in Graphics Recognition Algorithms and Systems. GREC 1997. Lecture Notes in Computer Science, vol 1389, pp. 80–90, Springer, Berlin, Heidelberg, 1998, doi:10.1007/3-540-64381-8_41.
[109] C. Ah-Soon, K. Tombre, Network-based recognition of architectural symbols, in Advances in Pattern Recognition. SSPR /SPR 1998. Lecture Notes in Computer Science, vol 1451, pp. 252–261, Springer, Berlin, Heidelberg, 1998, doi:10.1007/BFb0033243.
[110] K. Tombre, Analysis of engineering drawings: state of the art and challenges, in Graphics Recognition Algorithms and Systems. GREC 1997. Lecture Notes in Computer Science, vol 1389, pp. 257–264, Springer, Berlin, Heidelberg, 1998, doi:10.1007/3-540-64381-8_54.
[111] S.-h. Or, K.-H. Wong, Y.-k. Yu, M. M.-y. Chang, Highly automatic approach to architectural floorplan image understanding & model generation, Proceedings of Vision, Modeling, and Visualization 2005, pp. 25–32, 2005, https://www.cse.cuhk.edu.hk/~shor/paper/vmv05.pdf. Accessed 16 April 2022.
[112] K. Tombre, S. Tabbone, L. Pélissier, B. Lamiroy, P. Dosch, Text/graphics separation revisited, in Document Analysis Systems V. DAS 2002. Lecture Notes in Computer Science, vol 2423, pp. 200–211, Springer, Berlin, Heidelberg, 2002, doi:10.1007/3-540-45869-7_24.
[113] R. O. Duda, P. E. Hart, Use of the Hough transformation to detect lines and curves in pictures, Communications of the ACM, vol. 15, pp. 11–15, 1972, doi:10.1145/361237.361242.
[114] S. Ahmed, M. Liwicki, M. Weber, A. Dengel, Improved automatic analysis of architectural floor plans, in 2011 International Conference on Document Analysis and Recognition, (Beijing, China), pp. 864–869, IEEE, 2011, doi:10.1109/ICDAR.2011.177.
[115] S. Ahmed, M. Weber, M. Liwicki, A. Dengel, Text/graphics segmentation in architectural floor plans, in 2011 International Conference on Document Analysis and Recognition, (Beijing, China), pp. 734–738, IEEE, 2011, doi:10.1109/ICDAR.2011.153.
[116] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Speeded-up robust features (SURF), Computer Vision and Image Understanding, vol. 110, pp. 346–359, 2008, doi:10.1016/j.cviu.2007.09.014.
[117] M. Feltes, S. Ahmed, A. Dengel, M. Liwicki, Improved contour-based corner detection for architectural floor plans, in Graphics Recognition. Current Trends and Challenges. GREC 2013. Lecture Notes in Computer Science, vol 8746, pp. 191–203, Springer, Berlin, Heidelberg, 2014, doi:10.1007/978-3-662-44854-0_15.
[118] R. Tang, Y. Wang, D. Cosker, W. Li, Automatic structural scene digitalization, PLOS ONE, vol. 12, p. e0187513, 2017, doi:10.1371/journal.pone.0187513.
[119] G. Pan, J. He, R. Fang, Automatic floor plan detection and recognition, in 2017 2nd International Conference on Image, Vision and Computing, (Chengdu, China), pp. 201–205, IEEE, 2017, doi:10.1109/ICIVC.2017.7984546.
[120] P. De, Vectorization of architectural floor plans, in 2019 Twelfth International Conference on Contemporary Computing, (Noida, India), pp. 1–5, IEEE, 2019, doi:10.1109/IC3.2019.8844930.
[121] D. Sharma, C. Chattopadhyay, G. Harit, A unified framework for semantic matching of architectural floorplans, in 2016 23rd International Conference on Pattern Recognition, (Cancún, México), pp. 2422–2427, IEEE, 2016, doi:10.1109/ICPR.2016.7899999.
[122] A. Barducci, S. Marinai, Object recognition in floor plans by graphs of white connected components, in Proceedings of the 21st International Conference on Pattern Recognition, (Tsukuba, Japan), pp. 298–301, IEEE, 2012, https://ieeexplore.ieee.org/document/6460131. Accessed 16 April 2022.
[123] U. Ramer, An iterative procedure for the polygonal approximation of plane curves, Computer Graphics and Image Processing, vol. 1, pp. 244–256, 1972, doi:10.1016/S0146-664X(72)80017-0.
[124] X. Hilaire, K. Tombre, Robust and accurate vectorization of line drawings, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, pp. 890–904, 2006, doi:10.1109/TPAMI.2006.127.
[125] M. Li, F. Lafarge, R. Marlet, Approximating shapes in images with low-complexity polygons, in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, (Seattle, WA, USA), pp. 8630–8638, IEEE, 2020, doi:10.1109/CVPR42600.2020.00866.
[126] L.-P. de las Heras, G. Sánchez, And-or graph grammar for architectural floor plan representation, learning and recognition. a semantic, structural and hierarchical model, in Pattern Recognition and Image Analysis. IbPRIA 2011. Lecture Notes in Computer Science, vol 6669, pp. 17–24, Springer, Berlin, Heidelberg, 2011, doi:10.1007/978-3-642-21257-4_3.
[127] H. Leon-Garza, H. Hagras, A. Pena-Rios, A. Conway, G. Owusu, A fuzzy rule-based system using a patch-based approach for semantic segmentation in floor plans, in 2021 IEEE International Conference on Fuzzy Systems, (Luxembourg, Luxembourg), pp. 1–6, IEEE, 2021, doi:10.1109/FUZZ45933.2021.9494427.
[128] L.-P. de las Heras, J. Mas, G. Sanchez, E. Valveny, Wall patch-based segmentation in architectural floorplans, in 2011 International Conference on Document Analysis and Recognition, (Beijing, China), pp. 1270–1274, IEEE, 2011, doi:10.1109/ICDAR.2011.256.
[129] L.-P. de las Heras, J. Mas, G. Sánchez, E. Valveny, Notation-invariant patch-based wall detector in architectural floor plans, in Graphics Recognition. New Trends and Challenges. GREC 2011. Lecture Notes in Computer Science, vol 7423, pp. 79–88, Springer, Berlin, Heidelberg, 2013, doi:10.1007/978-3-642-36824-0_8.
[130] L.-P. de las Heras, E. Valveny, G. Sánchez, Unsupervised and notation-independent wall segmentation in floor plans using a combination of statistical and structural strategies, in Graphics Recognition. Current Trends and Challenges. GREC 2013. Lecture Notes in Computer Science, vol 8746, pp. 109–121, Springer, Berlin, Heidelberg, 2014, doi:10.1007/978-3-662-44854-0_9.
[131] L.-P. de las Heras, O. R. Terrades, J. Llados, Attributed graph grammar for floor plan analysis, in 2015 13th International Conference on Document Analysis and Recognition, (Tunis, Tunisia), pp. 726–730, IEEE, 2015, doi:10.1109/ICDAR.2015.7333857.
[132] H. Ganapathy, P. Ramu, R. Muthuganapathy, Alpha shape based design space decomposition for island failure regions in reliability based design, Structural and Multidisciplinary Optimization, vol. 52, pp. 121–136, 2015, doi:10.1007/s00158-014-1224-6.
[133] X. Guo, Y. Peng, Floor plan classification based on transfer learning, in 2018 IEEE 4th International Conference on Computer and Communications, (Chengdu, China), pp. 1720–1724, IEEE, 2018, doi:10.1109/CompComm.2018.8780679.
[134] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv, 2014, arXiv:1409.1556.
[135] S. Park, H. Kim, 3DPlanNet: generating 3D models from 2D floor plan images using ensemble methods, Electronics, vol. 10, p. 2729, 2021, doi:10.3390/electronics10222729.
[136] I. Evangelou, M. Savelonas, G. Papaioannou, PU learning-based recognition of structural elements in architectural floor plans, Multimedia Tools and Applications, vol. 80, pp. 13235–13252, 2021, doi:10.1007/s11042-020-10295-9.
[137] J. Bekker, J. Davis, Learning from positive and unlabeled data: a survey, Machine Learning, vol. 109, pp. 719–760, 2020, doi:10.1007/s10994-020-05877-5.
[138] S.-J. Heo, Z. Chunwei, E. Yu, Response simulation, data cleansing and restoration of dynamic and static measurements based on deep learning algorithms, International Journal of Concrete Structures and Materials, vol. 12, p. 82, 2018, doi:10.1186/s40069-018-0316-x.
[139] X. Peng, L. Wang, X. Wang, Y. Qiao, Bag of visual words and fusion methods for action recognition: comprehensive study and good practice, Computer Vision and Image Understanding, vol. 150, pp. 109–125, 2016, doi:10.1016/j.cviu.2016.03.013.
[140] H. Leon-Garza, H. Hagras, A. Pena-Rios, A. Conway, G. Owusu, An interval type-2 fuzzy-based system to create building information management models from 2D floor plan images, in 2021 IEEE International Conference on Fuzzy Systems, (Luxembourg, Luxembourg), pp. 1–7, IEEE, 2021, doi:10.1109/FUZZ45933.2021.9494464.
[141] E. Shelhamer, J. Long, T. Darrell, Fully convolutional networks for semantic segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, pp. 640–651, 2017, doi:10.1109/TPAMI.2016.2572683.
[142] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI, Information Fusion, vol. 58, pp. 82–115, 2020, doi:10.1016/j.inffus.2019.12.012.
[143] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, D. Terzopoulos, Image segmentation using deep learning: a survey, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1–1, 2021, doi:10.1109/TPAMI.2021.3059968.
[144] T. Yamasaki, J. Zhang, Y. Takada, Apartment structure estimation using fully convolutional networks and graph model, in Proceedings of the 2018 ACM Workshop on Multimedia for Real Estate Tech, (New York, NY, USA), pp. 1–6, ACM, 2018, doi:10.1145/3210499.3210528.
[145] W. Huang, H. Zheng, Architectural drawings recognition and generation through machine learning, in Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA), (Mexico City, Mexico), pp. 156–165, 2018, doi:S0020-0255(21)00275-9/h0105.
[146] Z. Ziran, S. Marinai, Object detection in floor plan images, in Artificial Neural Networks in Pattern Recognition. ANNPR 2018. Lecture Notes in Computer Science, vol 11081, pp. 383–394, Springer, Cham, 2018, doi:10.1007/978-3-319-99978-4_30.
[147] Y. Takada, N. Inoue, T. Yamasaki, K. Aizawa, Similar floor plan retrieval featuring multi-task learning of layout type classification and room presence prediction, in 2018 IEEE International Conference on Consumer Electronics, (Las Vegas, NV, USA), pp. 1–6, IEEE, 2018, doi:10.1109/ICCE.2018.8326163.
[148] H. Jang, J. Yang, K. Yu, Automatic wall detection and building topology and property of 2D floor plan (short paper), in 10th International Conference on Geographic Information Science, (Dagstuhl, Germany), pp. 33:1–33:5, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018, doi:10.4230/LIPIcs.GIScience.2018.33.
[149] W. Wang, S. Dong, K. Zou, W.-s. LI, Room classification in floor plan recognition, in 2020 4th International Conference on Advances in Image Processing, (New York, NY, USA), pp. 48–54, ACM, 2020, doi:10.1145/3441250.3441265.
[150] G. Murugan, V. Moyal, P. Nandankar, O. Pandithurai, E. John Pimo, A novel CNN method for the accurate spatial data recovery from digital images, Materials Today: Proceedings, 2021, doi:10.1016/j.matpr.2021.05.351.
[151] M. Yamada, X. Wang, T. Yamasaki, Graph structure extraction from floor plan images and its application to similar property retrieval, in 2021 IEEE International Conference on Consumer Electronics, (Las Vegas, NV, USA), pp. 1–5, IEEE, 2021, doi:10.1109/ICCE50685.2021.9427580.
[152] I. Goodfellow, Y. Bengio, A. Courville, Deep learning. MIT Press, 2016, https://www.deeplearningbook.org. Accessed 16 April 2022.
[153] A. Khan, A. Sohail, U. Zahoora, A. S. Qureshi, A survey of the recent architectures of deep convolutional neural networks, Artificial Intelligence Review, vol. 53, pp. 5455–5516, 2020, doi:10.1007/s10462-020-09825-6.
[154] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8689, pp. 818–833, Springer, Cham, 2014, doi:10.1007/978-3-319-10590-1_53.
[155] O. Ronneberger, P. Fischer, T. Brox, U-Net: convolutional networks for biomedical image segmentation, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Lecture Notes in Computer Science, vol. 9351, pp. 234–241, 2015, doi:10.1007/978-3-319-24574-4_28.
[156] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science, vol 11211, pp. 833–851, Springer, Cham, 2018, doi:10.1007/978-3-030-01234-2_49.
[157] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in 2014 IEEE Conference on Computer Vision and Pattern Recognition, (Columbus, OH, USA), pp. 580–587, IEEE, 2014, doi:10.1109/CVPR.2014.81.
[158] H. Gao, H. Yuan, Z. Wang, S. Ji, Pixel deconvolutional networks, arXiv, 2017, arXiv:1705.06820.
[159] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in 2016 IEEE Conference on Computer Vision and Pattern Recognition, (Las Vegas, NV, USA), pp. 770–778, IEEE, 2016, doi:10.1109/CVPR.2016.90.
[160] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, arXiv, 2017, arXiv:1706.03762.
[161] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, SSD: single shot multibox detector, in Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9905, pp. 21–37, Springer, Cham, 2016, doi:10.1007/978-3-319-46448-0_2.
[162] L. Huang, G. Wan, C. Liu, An improved parallel thinning algorithm, in Seventh International Conference on Document Analysis and Recognition, vol. 1, (Edinburgh, UK), pp. 780–783, IEEE

Comput. Soc, 2003, doi:10.1109/ICDAR.2003.1227768.
[163] C. Harris, M. Stephens, A combined corner and edge detector, in Procedings of the Alvey Vision Conference 1988, (Manchester, UK), pp. 23.1–23.6, Alvey Vision Club, 1988, doi:10.5244/C.2.23.
[164] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, Q. Tian, CenterNet: keypoint triplets for object detection, in 2019 IEEE/CVF International Conference on Computer Vision, (Seoul, Korea), pp. 6568–6577, IEEE, 2019, doi:10.1109/ICCV.2019.00667.
[165] H. Law, J. Deng, Cornernet: detecting objects as paired keypoints, International Journal of Computer Vision, vol. 128, pp. 642–656, 2020, doi:10.1007/s11263-019-01204-1.
[166] K. He, G. Gkioxari, P. Dollar, R. Girshick, Mask R-CNN, in 2017 IEEE International Conference on Computer Vision, (Venice, Italy), pp. 2980–2988, IEEE, 2017, doi:10.1109/ICCV.2017.322.
[167] Z. Cai, N. Vasconcelos, Cascade R-CNN: high quality object detection and instance segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, pp. 1483–1498, 2021, doi:10.1109/TPAMI.2019.2956516.
[168] J. Redmon, A. Farhadi, YOLOv3: an incremental improvement, arXiv, 2018, arXiv:1804.02767.
[169] A. Bochkovskiy, C.-Y. Wang, H.-Y. M. Liao, YOLOv4: optimal speed and accuracy of object detection, arXiv, 2020, arXiv:2004.10934.
[170] T.-W. Ke, J.-J. Hwang, Z. Liu, S. X. Yu, Adaptive affinity fields for semantic segmentation, in Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science, vol 11205, pp. 605–621, Munich, Germany: Springer, Cham, 2018, doi:10.1007/978-3-030-01246-5_36.
[171] R. Cipolla, Y. Gal, A. Kendall, Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, (Salt Lake City, UT, USA), pp. 7482–7491, IEEE, 2018, doi:10.1109/CVPR.2018.00781.
[172] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, X. Bai, Richer convolutional features for edge detection, in 2017 IEEE Conference on Computer Vision and Pattern Recognition, (Honolulu, HI, USA), pp. 5872–5881, IEEE, 2017, doi:10.1109/CVPR.2017.622.
[173] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, in 2017 IEEE Conference on Computer Vision and Pattern Recognition, (Honolulu, HI, USA), pp. 6230–6239, IEEE, 2017, doi:10.1109/CVPR.2017.660.
[174] M. Gori, G. Monfardini, F. Scarselli, A new model for learning in graph domains, in 2005 IEEE International Joint Conference on Neural Networks, vol. 2, (Montreal, QC, Canada), pp. 729–734, IEEE, 2005, doi:10.1109/IJCNN.2005.1555942.
[175] F. Scarselli, M. Gori, Ah Chung Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Transactions on Neural Networks, vol. 20, pp. 61–80, 2009, doi:10.1109/TNN.2008.2005605.
[176] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P. S. Yu, A comprehensive survey on graph neural networks, IEEE Transactions on Neural Networks and Learning Systems, vol. 32, pp. 4–24, 2021, doi:10.1109/TNNLS.2020.2978386.
[177] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, (Montreal, QC, Canada), pp. 2672–2680, MIT Press, 2014, doi:10.5555/2969033.2969125.
[178] L. Gatys, A. Ecker, M. Bethge, A neural algorithm of artistic style, Journal of Vision, vol. 16, p. 326, 2016, doi:10.1167/16.12.326.
[179] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, M. Song, Neural style transfer: a review, IEEE Transactions on Visualization and Computer Graphics, vol. 26, pp. 3365–3385, 2020, doi:10.1109/TVCG.2019.2921336.
[180] M. Mirza, S. Osindero, Conditional generative adversarial nets, in Proceedings of the 34th International Conference on Machine Learning - Volume 70, (Sydney, NSW,Australia), p. 7, 2014, arXiv:1411.1784.
[181] A. Odena, C. Olah, J. Shlens, Conditional image synthesis with auxiliary classifier GANs, arXiv, 2016, arXiv:1610.09585.
[182] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein GAN, arXiv, 2017, arXiv:1701.07875.
[183] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, Image-to-image translation with conditional adversarial networks, in 2017 IEEE Conference on Computer Vision and Pattern Recognition, (Honolulu, HI, USA), pp. 5967–5976, IEEE, 2017, doi:10.1109/CVPR.2017.632.
[184] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, A. A. Bharath, Generative adversarial networks: an overview, IEEE Signal Processing Magazine, vol. 35, pp. 53–65, 2018, doi:10.1109/MSP.2017.2765202.
[185] Y. Hong, U. Hwang, J. Yoo, S. Yoon, How generative adversarial networks and their variants work, ACM Computing Surveys, vol. 52, pp. 1–43, 2019, doi:10.1145/3301282.
[186] J.-Y. Zhu, T. Park, P. Isola, A. A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in 2017 IEEE International Conference on Computer Vision, (Venice, Italy), pp. 2242–2251, IEEE, 2017, doi:10.1109/ICCV.2017.244.
[187] T. Kim, M. Cha, H. Kim, J. K. Lee, J. Kim, Learning to discover cross-domain relations with generative adversarial networks, in Proceedings of the 34th International Conference on Machine Learning - Volume 70, (Sydney, NSW, Australia), pp. 1857–1865, JMLR.org, 2017, doi:10.5555/3305381.3305573.
[188] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, B. Catanzaro, High-resolution image synthesis and semantic manipulation with conditional GANs, in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, (Venice, Italy), pp. 8798–8807, IEEE, 2018, doi:10.1109/CVPR.2018.00917.
[189] I. Phillips, A. Chhabra, Empirical performance evaluation of graphics recognition systems, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 9, pp. 849–870, 1999, doi:10.1109/34.790427.
[190] S. Mishra, K. A. Hashmi, A. Pagani, M. Liwicki, D. Stricker, M. Z. Afzal, Towards robust object detection in floor plan images: a data augmentation approach, 2021, doi:10.3390/app112311174.
[191] J. Zak, H. Macadam, Utilization of building information modeling in infrastructure's design and construction, IOP Conference Series: Materials Science and Engineering, vol. 236, p. 012108, 2017, doi:10.1088/1757-899X/236/1/012108.