

Automatika

Journal for Control, Measurement, Electronics, Computing and Communications



ISSN: (Print) (Online) Journal homepage: <https://www.tandfonline.com/loi/taut20>

An automated system for electrical power symbol placement in electrical plan drawing

Bayram Akgul & Hakan Kutucu

To cite this article: Bayram Akgul & Hakan Kutucu (2022) An automated system for electrical power symbol placement in electrical plan drawing, *Automatika*, 63:1, 78-89, DOI: [10.1080/00051144.2021.2008620](https://doi.org/10.1080/00051144.2021.2008620)

To link to this article: <https://doi.org/10.1080/00051144.2021.2008620>



© 2021 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group



View supplementary material [↗](#)



Published online: 29 Nov 2021.



Submit your article to this journal [↗](#)



Article views: 1358



View related articles [↗](#)



View Crossmark data [↗](#)



An automated system for electrical power symbol placement in electrical plan drawing

Bayram Akgul ^a and Hakan Kutucu ^b

^aDepartment of Computer Technologies, Bartin University, Bartin, Turkey; ^bDepartment of Computer Engineering, Karabuk University, Karabuk, Turkey

ABSTRACT

An electrical plan drawing—sometimes called a wiring diagram or electrical drawing—consists of lines and symbols. Electrical plan drawings are prepared on 2D architectural floor plans using Computer-Aided Design and/or Drafting (CAD) programs. The placement/drawing of electrical power symbols—such as sockets, lights, and switches—is the first step of an electrical plan drawing. For this purpose, a smart system has been developed in this study to automatically draw/place electrical power symbols in appropriate locations. The system is based on the detection and classification/recognition of furnishing (decorative) symbols in the floor plans. We have created a furnishing symbol dataset drawing on dozens of architectural plan drawings that contain symbols of the most commonly used tools in floor plans, such as furniture, appliances, plumbing, doors, and windows. We used a Deep Convolutional Neural Network (D-CNN) with transfer learning—Inception-v3 model—to classify furnishing symbols. We tested the model on 20 real floor plans and achieved a very satisfactory accuracy of 97.05% in furnishing symbol classification. The symbol drawing step, which is the first step of drawing the electrical plan, was automated using the work developed, thus achieving the aim of saving time and labour. Experimental studies show the effectiveness of the proposed automated system.

ARTICLE HISTORY

Received 4 July 2021
Accepted 12 November 2021

KEYWORDS

Computer-aided design; electrical plan drawing; deep learning; transfer learning; automated design

1. Introduction

Electrical plan drawings are schematic displays of an installation project that must be applied to provide the electricity requirement for construction. In earlier years, electrical plan drawings were drawn on paper by hand. With the development of computers and Computer-Aided Design and/or Drafting (CAD) programs, plan drawings moved to an electronic environment. Nowadays, with the abilities of computers, the design, modelling, calculation, and testing processes can be achieved more efficiently and effectively [1]. Although rules for electrical plan drawings vary by country, the basic process is to deliver grid power to electrical materials. Electricity providers in every country obligate users to prepare an installation plan for a building in order to use electricity from the grid line.

Electrical plan drawings are generally prepared using 2D CAD software based on an architectural floor plan prepared by an architect [2]. Many commercial CAD programs (AutoCAD Electrical, ElectricalOm, ProfiCAD, EPlan Electric, Designspark Electrical, Electra, Simaris, etc.) are used in the industry to prepare electrical plans or diagrams. Most can be used manually to create electrical plan drawings. Then, technical analysis, assessment, costing, and calculation printouts are

done automatically. Even if most calculations are done automatically, drawings are still prepared manually.

When preparing an electrical plan, placing/drawing the electrical power symbols is the first step. This time-consuming process is done manually according to the furnishing symbols determined by the architects in accordance with the intended use of the building.

Furnishing in architectural plans is a general term for drawing symbols representing all furniture, electrical appliances, plumbing fixtures, windows, and doors. Furnishing symbols—sometimes called decorative symbols—are important because they show whether the rooms drawn in architectural plans are suitable in terms of size for the use of furniture (table, bed, etc.) or appliances (tv, washing machine, etc.). In particular, the ability of electrical project designers to prepare the required drawings for electrical calculations depends on the placement of furnishing symbols in the architectural project. A sample floor plan with furnishing symbols is shown in Figure 1. While drawing the electrical plan for this floor plan, it is easy to determine where to place any given electrical power symbols by looking at the furnishing symbols.

In this study, we used machine learning techniques to intelligently place electrical power symbols, such

CONTACT Bayram Akgul bayramakgul@bartin.edu.tr Department of Computer Technologies, Bartin University, Bartin 74100, Turkey

Supplemental data for this article can be accessed here. <https://doi.org/10.1080/00051144.2021.2008620>

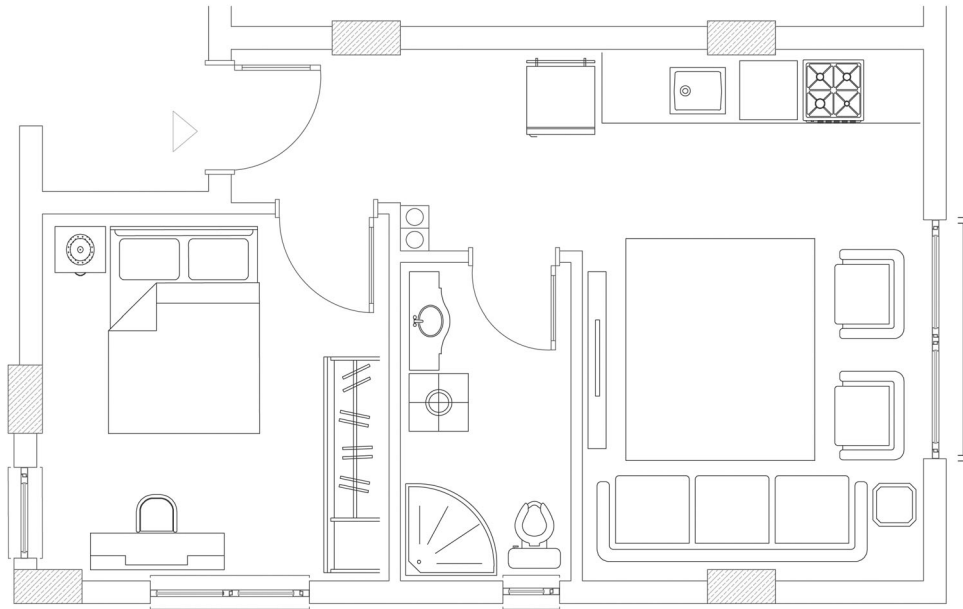


Figure 1. A sample apartment floor plan with furnishing symbols.

as sockets, lights, and switches, while drawing electrical plans for residential buildings. A Deep Convolutional Neural Network (D-CNN) with transfer learning was used for the classification/recognition of furnishing symbols. We have created a suggestion system for placing electrical power symbols corresponding to the furnishing symbols after classifying furnishings, based on expert experiences. As a result, an automated system was developed to place the electrical power symbols on electrical plan drawings.

The paper is organized as follows: Section 2 provides general information about CAD, architectural floor plans, and furnishing symbols. Related work is summarized in Section 3. Section 4 presents dataset preparation, the proposed D-CNN model for the classification of furnishing symbols, and experimental results. Section 5 describes the automated placement of appropriate electrical power symbols in accordance with the classified furnishing symbols. Finally, in Section 6, we give the conclusion and future work.

2. CAD and floor plans

CAD and its applications are the most popular topics in design-related fields such as engineering and architecture. The foundations of CAD date back to the late 1950s and early 1960s [3]. In today's construction sector, many CAD programs, either 2D or 3D, are used for engineering and architecture services. AutoCAD, Revit, Archicad, SolidWorks, SketchUp, and StaCad are the most popular. CAD systems are used to increase the designer's productivity, improve the quality of the design, improve communications through documentation, and create a database for manufacturing [4].

Files in CAD programs are in a vector format. In this format, drawing objects consist of mathematical expressions. For example, a line segment consists of two points, while an arc consists of a central point, a starting angle, and a sweep angle. The vector file format includes basic geometric shapes such as lines, arcs, circles, ellipses, polygons, and blocks that can be used by grouping them in accordance with one's needs. Moreover, these geometric objects have various formatting properties, such as layer, colour, line-type, line-weight, etc. Layer usage is one of the most important factors of a CAD program. Architectural drawings are usually made in layers, which are similar to overlapping sketches.

2.1. Furnishing symbols

Furnishing symbols are drawn as blocks in vectorial drawing environments. A block is a collection of objects that is combined into a single object. The use of blocks is another important factor in CAD programs. In vectorial drawing environments, objects are combined so that multiple geometric objects can be reused in different places. Architectural plan designers store repeating objects in different plans in a library created in block form. Thus, they reuse these blocks in any plans without redrawing them.

Although furnishing blocks differ according to architect and architecture, they are usually prepared in the form of a top-down view of the material they represent. Figure 2 illustrates some sample furnishing symbols.

Hundreds of different symbols can be adopted for each class of furnishing used by architects. These

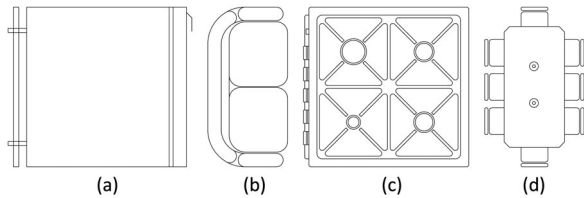


Figure 2. Some furnishing symbols: (a) Refrigerator (b) Sofa (c) Stove-oven (d) Dining Table.

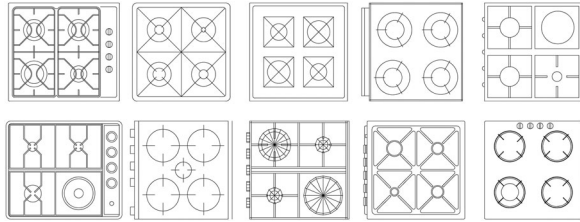


Figure 3. Different stove-oven symbols used by different architects.

symbols are usually specific to an architect and culture. Figure 3 shows ten different stove-oven symbols.

3. Related work

Architectural drawings are made in the form of floor plans. Floor plans include walls, rooms, doors, windows, and other decorative symbols. Many studies have been conducted analyzing architectural floor plans. These studies mostly aimed to obtain 3D building models from 2D architectural floor plans based on a pattern recognition approach [5–8].

Ah-Soon and Tombre [9] presented a study to find door and window symbols in scanned architectural floor plans using computer vision and pattern recognition techniques. Dosch et al. [10] described a complete system for symbol recognition on architectural floor plans to reconstruct 3D buildings by using computer vision tools such as segmentation, vectorization, and feature detection. Guo et al. [11] used an example-driven symbol recognition approach to analyze symbols from engineering drawings in their work. This study can be applied to many fields. The study has been tested in architectural floor plans and decorative symbols have been successfully recognized. Heras et al. [12] presented a database of scanned real floor plans (CVC-FP) and the ground-truthing tool (SGT tool) to generate its structural ground-truth (GT). They also present a benchmark on wall segmentation and room detection tasks in their paper.

There are many research on isolated symbol recognition. However, it was not observed the same for symbol recognition in context. Most symbol spotting techniques work on scanned floor plans. Rezvanifar et al. [13] offered a wide literature survey on symbol spotting (not recognition) in architectural drawing images.

Symbol spotting refers to the retrieval of graphical symbols embedded in larger images or documents. Additionally, Rezvanifar et al. [14] propose a deep learning based framework for symbol spotting on architectural floor plans.

Although there are a few other datasets for floor plans in the literature, these also consist of rasterized images. GREC'2003 [15], GREC'2011 [16], and SESYD [17] datasets contain either scanned or synthetic architectural and electrical documents for symbol spotting, recognition, and retrieval. The FPLAN-POLY [18] dataset contains 42 real floor plans vectorized from scanned images.

To the best of our knowledge, there is only one study about the automatic generating of electrical plans from architectural data. The paper proposes a procedural approach to automatically generate the residential electrical plans by using a stack-based finite state machine model [19]. This system receives 2D architectural data such as wall location as input and yields a customized electrical plan as output.

In practice, CAD applications (especially in the design-related industry) do not use rasterized floor plans, instead, they use vector plans. Therefore, none of these datasets are appropriate for our study. That's why we created our own dataset. In our study, we use vector (DXF/DWG) files as input to automatically draw the electrical power symbols. Additionally, the generated output is also in vector format.

4. Furnishing symbols classification

Furnishing symbol classification is the backbone of this paper, because all other processes in this study are carried out based on the recognition of furnishing symbols. Furnishing symbol classification is an image classification problem, or a computer vision problem that takes an image as input and categorizes it into a prescribed class. In the past, many classification approaches, such as artificial neural networks, expert systems, and fuzzy sets, have been widely used for this task. However, accuracy values for each are comparatively lower than for deep learning [20]. Deep learning is currently the best solution for image classification.

Unlike traditional machine learning and image processing techniques, deep learning carries out the learning process using raw data. Deep learning obtains the necessary features using representation learning carried out over different layers. Representation learning includes a group of methods allowing models to be supplied with raw data (such as the pixel values of an image) and automatically explore representations for detection or classification [21]. DNNs have been reached human-equivalent accuracy in image classification and object detection [22].

Classical machine learning techniques use manually extracted features for classification, although CNNs use

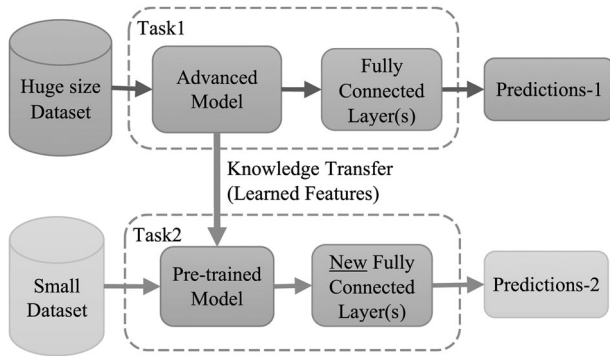


Figure 4. Transfer learning approach.

features automatically extracted from raw images via convolutional and pooling layers. For learning, automatically extracted features strongly depend on the training dataset size. If the amount of training data is not large enough, this may lead to overfitting. To avoid this, deep CNNs with transfer learning are used [23].

4.1. Transfer learning

Transfer learning (TL) is an approach that aims to apply the knowledge gained from one task to another, similar task during the machine learning process.

The most common problem in CNNs is that researchers usually cannot find enough data to create a large dataset. Transfer learning is often used when collecting training data would be expensive or difficult [24]. It also helps save time and achieve better performance with less training time. Training a deep CNN model from zero requires large amounts of both labelled data and computing resources. However, a pre-trained CNN model with transfer learning allows a short-cut in the training process. Experiments show that transfer learning gives better accuracy on CNNs than random weight distribution [25]. Figure 4 shows the transfer learning approach.

There are three type of transfer learning based on the availability and the amount of labelled data in the source and target domains: inductive, transductive, and unsupervised transfer learning [26]. We used inductive transfer learning in this paper. In inductive transfer learning, a model developed for the source task is used as a starting point for the target task. For example, using the information of a model to classify cats and dogs for vehicle classification.

To transfer the parameters of a CNN, first, the network is trained on the source task with a large amount of labelled data, such as that in ImageNet. Then, the convolution and pooling layers are frozen for new tasks. The last fully connected layer and the classifier are removed. To use this pre-trained model for new tasks, one or more new fully connected layers (adaptation layers) and a new classifier that matches the target dataset

are added. Finally, the network is trained to update the weights of new fully connected layers.

The ImageNet is a large image database (over 14 million labelled high-resolution images belonging to approximately 22,000 categories) designed for object detection and classification research [27].

Nowadays, transfer learning is used to classify images in many areas. In particular, learning gained from the ImageNet dataset is commonly used. Many advanced pre-trained models, such as Inception-v3 [28], ResNet [29,30], MobileNet [31], and their improved versions, are available for researchers on different platforms.

4.2. Dataset preparation

To create a dataset, 1020 furnishing symbol blocks for 23 classes were collected from dozens of real-world architectural floor plan drawings. The class names are given in Table 2. These symbols were then saved in a DXF file to be used and/or expanded publicly by researchers. A DXF (drawing interchange or exchange format) file is an either binary or text-based representation of a drawing file used to interchange drawing data between CAD programs. The DXF format specifications are determined by the company Autodesk [32].

Since the symbols in the CAD files are in vector format, they were converted to image format at certain resolutions to be used in a CNN. To investigate which resolution would give better results, it was necessary to try different resolution scenarios. Enlarging or shrinking scales in vector data does not change the thickness of edge lines, but it is not possible to say the same for bitmap images. When scale changes in a bitmap image, the image thickens or refines at the edges. Indeed, if the bitmap image is overly minimized, some lines of 1-pixel thickness may then be lost in the minimized image. To prevent this, furnishing symbol blocks were brought to the desired scales in vector format and then converted into bitmap images. To do this, we created an application to convert DXF blocks into bitmap images automatically based on specified parameters such as size, rotation angle, line-weight, line-type, line-colour, and background-colour.

While geometric shapes can be similar for any furnishing symbol, architects may create their own style using line-colour, line-weight, or line-type. To facilitate the work of the neural network, only geometrical furnishing shapes were used, with a standard colour and a standard line-weight. Thus, regardless of which colour or line thickness was used for furnishing, the system automatically applied white for line-colour and only one pixel for line-weight; in this way, each furnishing symbol was converted to the bitmap image.

In deep learning, large datasets are important for success, although they increase the cost of computation

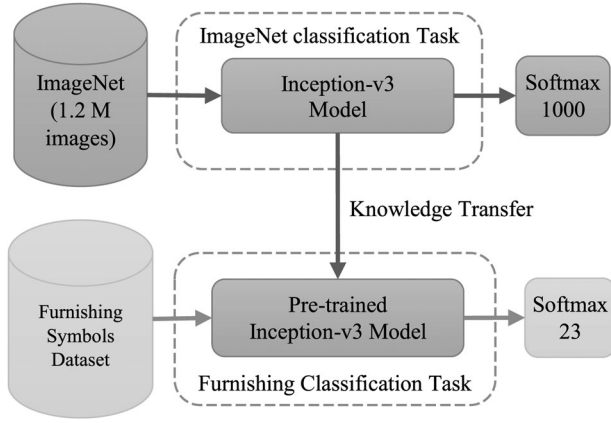


Figure 5. The proposed model for the classification of furnishing symbols.

and resources. Not only the size of the dataset but also the diversity of data is important for success. In some cases, if a dataset is not large enough, data should be increased through data augmentation methods. A dataset can be increased by resizing, rotating, mirroring, or applying distortions [33–35].

The 1020 collected symbols were increased to 24480 symbols using data augmentation. To do this, first, each block symbol was converted into three separate images with resolutions of 112×112 , 224×224 and 299×299 pixels. Then, each image was transformed into eight other images by rotation and mirroring. In addition to the original image, 90° , 180° , and 270° angles were used to perform mirroring and rotation. This was also an advantage for the prepared system, because furnishing symbols may be placed on the floor plan drawing at different angles. The dataset is renamed FDS, short for “Furnishing Dataset”. Of this dataset, 25% was reserved for later use in the testing process.

4.3. Proposed method for furnishing symbols classification

Since we do not have a large dataset, we used a pre-trained Inception-v3 model with transfer learning to perform classification. Inception-v3 is the third version of GoogLeNet, a CNN based on the Inception architecture [28]. In this study, transfer learning transfers the knowledge gained from the ImageNet classification to the furnishing symbol classification. The proposed model is shown in Figure 5.

We used the Microsoft ML.NET Image Classification API to create a classifier for FDS. The Image Classification API uses TensorFlow, an open-source machine learning platform. During the training process, the reserved training dataset was split 80:20 into training and validation sets, respectively. The mini-batch size was set to 10, and the learning rate was set to 0.01. Thus the training was completed at 32 epochs.

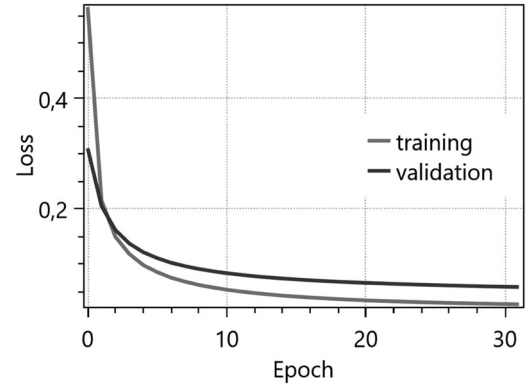
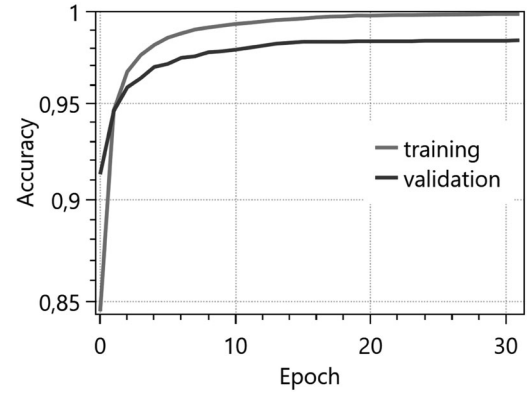


Figure 6. Accuracy and loss values per epoch on the training and validation sets.

Table 1. Final accuracy and loss values for FDS dataset.

Training		Validation		Test
Accuracy	Loss	Accuracy	Loss	Accuracy
0.9988	0.0261	0.9842	0.0577	0.9864

4.4. Experimental results on classification

Table 1 shows the final accuracy and loss values for the training and validation sets. Figure 6 show the accuracy and loss values per epoch for the training and validation sets.

The confusion matrix for the test data is given in Figure 7. The confusion matrix shows that the furnishing symbols are well separated from each other.

The number of block symbols in each class is not equal. Since the data is imbalanced, it is more appropriate to use the *F1-score* than accuracy. The *F1-score* is a common criterion used to rate the success of a classifier. To calculate the *F1-score*, we first need to calculate precision and recall values. Precision is the ratio of true positives (TP) as opposed to false positives (FP) and recall is the ratio of true positives as compared to false negatives (FN). The mathematical representation of precision and recall are shown in Equations (1) and (2), respectively. The *F1-score* is the harmonic mean of precision and recall, as shown in Equation (3).

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

Table 2. f1-scores for each class.

No	Class Name	TP	FP	FN	Precision	Recall	F-1 Score
1	BED-DOUBLE	372	14	18	0.964	0.954	0.959
2	BED-SINGLE	262	17	8	0.939	0.970	0.954
3	DISHWASHER	177	0	3	1.000	0.983	0.992
4	DOOR-DOUBLE	228	1	0	0.996	1.000	0.998
5	DOOR-SINGLE	407	0	1	1.000	0.998	0.999
6	DOOR-WINDOWED	114	0	0	1.000	1.000	1.000
7	REFRIGERATOR	114	1	0	0.991	1.000	0.996
8	SHOWER	356	0	4	1.000	0.989	0.994
9	SINK	228	0	0	1.000	1.000	1.000
10	SOFA-CORNER	95	1	7	0.990	0.931	0.960
11	SOFA-ONE	311	4	1	0.987	0.997	0.992
12	SOFA-THREE	245	12	7	0.953	0.972	0.963
13	SOFA-TWO	205	6	11	0.972	0.949	0.960
14	STOVE-OVEN	216	0	0	1.000	1.000	1.000
15	TABLE-DINNER	658	9	8	0.987	0.988	0.987
16	TABLE-STUDY	175	4	5	0.978	0.972	0.975
17	TELEVISION	77	2	1	0.975	0.987	0.981
18	TOILET	398	5	4	0.988	0.990	0.989
19	WARDROBE	324	2	0	0.994	1.000	0.997
20	WASHBASIN	384	4	0	0.990	1.000	0.995
21	WASHBASIN-CABINET	211	0	5	1.000	0.977	0.988
22	WASHINGMACHINE	180	1	0	0.994	1.000	0.997
23	WINDOW	300	0	0	1.000	1.000	1.000

		Actual Classes																						
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Predicted Classes	1	372	7	0	0	0	0	0	2	0	1	1	0	0	0	3	0	0	0	0	0	0	0	0
	2	13	262	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0
	3	0	0	177	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	228	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	407	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	114	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	7	0	0	1	0	0	0	114	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	356	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	228	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	10	1	0	0	0	0	0	0	0	0	95	0	0	0	0	0	0	0	0	0	0	0	0	0
	11	1	0	0	0	0	0	0	0	0	0	311	0	0	0	0	0	0	0	3	0	0	0	0
	12	0	0	0	0	0	0	0	0	0	1	0	245	10	0	1	0	0	0	0	0	0	0	0
	13	0	0	0	0	0	0	0	0	0	2	0	4	205	0	0	0	0	0	0	0	0	0	0
	14	0	0	0	0	0	0	0	0	0	0	0	0	0	216	0	0	0	0	0	0	0	0	0
	15	2	0	0	0	0	0	0	0	0	2	0	0	0	0	658	4	0	1	0	0	0	0	0
	16	0	1	0	0	0	0	0	0	0	0	0	0	0	0	2	175	1	0	0	0	0	0	0
	17	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	77	0	0	0	0	0	0
	18	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	398	0	0	3	0	0
	19	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	324	0	0	0	0
	20	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	384	2	0	0
	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	211	0	0
	22	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	180	0
	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	300

Figure 7. Confusion matrix of classification results.

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (3)$$

$$F_1^{macro} = \frac{1}{n} \sum_{class=1}^n F_1(class) \quad (4)$$

Unlike binary classification, in multi-class classification, the *F1-score* is calculated separately for each class. The calculated *F1-scores* for each class are given in the last column of Table 2. The *F1-macro-score* is computed using the arithmetic mean of each class's *F1-scores*, as shown in Equation (4) [36]. Thus, the *F1-macro-score* was computed as 0.9859.

We tested the model on 20 real floor plans. A total of 2139 block symbols (furnishing blocks) were detected in these floor plans. These block symbols are originally in vector format, and so must be converted into bitmap images in order to use them in the model. We created three separate bitmaps with different resolutions (112 × 112, 224 × 224 and 299 × 299) for each block. All conversions are automatically made by our application. In this way, we forwarded separately three images for each block to the created model. If the prediction for at least two of the three images was the same, it was considered the model's prediction. This also allowed us to conduct a triple check. Table 3 shows the number of symbols detected and the accuracy rate of the classification on each floor plan. As seen in the table, the

Table 3. Test results on 20 real floor plans.

File Name (floor plan)	Number of symbols detected	The number of correctly classified symbols	Accuracy Rate
M001	125	121	96.80%
M002	125	125	100.00%
M003	151	142	94.04%
M004	64	64	100.00%
M005	131	128	97.71%
M006	122	120	98.36%
M007	62	62	100.00%
M008	117	107	91.45%
M009	152	149	98.03%
M010	114	110	96.49%
M011	90	88	97.78%
M012	52	51	98.08%
M013	174	168	96.55%
M014	130	126	96.92%
M015	119	115	96.64%
M016	49	48	97.96%
M017	67	65	97.01%
M018	125	123	98.40%
M019	38	36	94.74%
M020	132	128	96.97%
Total:	2139	2076	97.05%

proposed model achieved 97.05% accuracy on real floor plans when classifying furnishing symbols.

5. Automated electrical power symbol placement system

To draw an electrical plan, we place the electrical power symbols automatically according to the algorithm given in Algorithm 1. Each step of this algorithm is explained in the following subSections.

5.1. Room polygon selection

Room zones on the floor plans are considered as polygons. A polygon can be created by selecting a few points or by specifying a rectangle (with two corner points). In the plan drawings, most rooms are rectangular. Zones in rectangular form can be actively detected within a locale with the help of the intersection of the wall lines closest to the left-right-top-bottom position of the cursor. Figure 8) shows how to automatically select a zone in rectangular form. Also, a rectangular zone can easily be selected by clicking two corner points (e.g. top-left and bottom-right) on the floor plan, as shown in Figure 8). If the region to be selected is not rectangular, for example, is an L-shaped room, then the boundary points of the zone can be selected with the help of individual clicks.

5.2. Detection of furnishing symbols inside a polygon area

In the present work, we need not only object classification but also object position. The aim of generic object detection is to locate and classify existing objects in an image and label them with bounding boxes [37]. If the floor plans in our study consist of raster images,

then we would have to first detect the furnishing symbols by using object detection algorithms such as Faster R-CNN [38] or YOLO [39].

There are geometrical data in vector files consisting of coordinates, angles, and similar information. The objects in the vector files are visualized at run-time by the application program. Because the floor plans in our study are in vector format, we do not need to use classic object detection algorithms. All block symbols are already in a list in vector drawings. In addition, we can already calculate the boundaries of these block objects thanks to the geometric data in the file.

All graphical objects with known coordinates inside a closed polygon were determined according to their positions, and then the blocks were grouped into a list. All block objects in the floor plan drawing were considered as potential furnishing symbols. Thus, furnishing symbols from a selected area were determined according to their positions.

5.3. Recognition of furnishing symbols

Once a zone (room) was selected and block objects in the selected zone were listed, each block in this list was treated as a furnishing symbol. These blocks were converted into individual bitmaps by ignoring the original line weights/types/colours. Then, each bitmap image was forwarded to the trained CNN model to recognize the furnishing symbols. Figure 9 shows furnishing symbol detection and recognition in the selected zone.

5.4. The suggestion of electrical power symbols based on furnishing symbols

A Look-up table, given in Table 4, was created to place electrical power symbols when drawing an electrical plan using the experience of project designers. This table can be customized according to country and/or region.

Although the washing machine, dishwasher, and oven sockets are treated as normal socket material, the power drawn from them is in fact greater than for normal sockets: 300–500 watts for normal sockets, 1500–2500 watts for these sockets. Since the power used by these sockets will be high, each is designed as a separate circuit in the electrical plan drawing and a separate circuit breaker is planned for each. Therefore, these sockets have different symbols in electrical plan drawings.

5.5. Placing electrical power symbols

When drafting an electrical plan drawing using a CAD program, the placement of the electrical power symbols in the right place is the first step. The electrical power symbols to be placed are selected from Table 4 after the furnishing symbols are recognized. Some electrical

Algorithm 1: Automatic electrical power symbols placement system**Input:** Architectural floor plan with furnishing symbols.**Output:** Architectural floor plan with electrical power symbols placed.**foreach** room in floor plan **do**

Create the polygon that will enclose the room zone. (room-polygon).

List all cad-block-object(s) within the polygon with their locations.

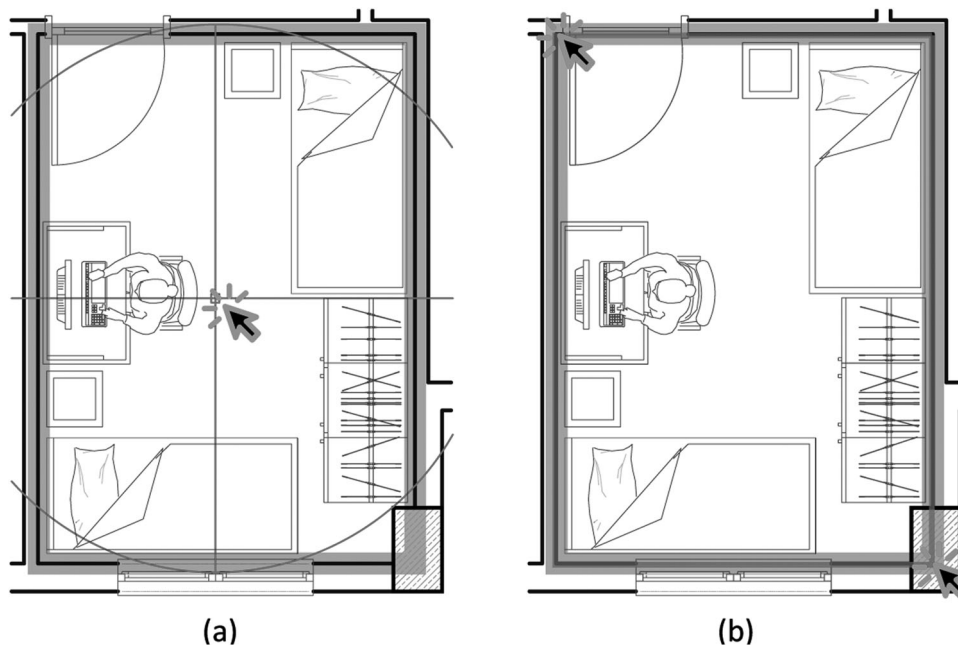
▷ Note: we know all the furnishing symbols are in block form, so each block will be considered as a furnishing symbol.

foreach cad-block-object in room-polygon **do** Convert the block object to three bitmap images with the specified resolutions (112×112 , 224×224 and 299×299) and properties (*line-weight: 1px, line-colour: white, line-type: continuous*) for the network.

Recognize the furnishing symbols in the converted images by inputting them into the trained network.

Determine the electrical power symbols for the recognized furnishing symbols using Table 4.

Place the determined electrical power symbol(s) near the wall line where the furnishing symbol is.

end**end****Figure 8.** Selection of the boundaries of a zone: (a) Automatic selection (b) Selection by clicking two corner points.

power symbols are shown in Figure 10. These symbols are used as blocks in CAD drawings.

It is necessary to detect the most suitable wall line for the electrical power symbol after recognizing the furnishing symbol. The selected electrical power symbol is placed on the wall line nearest to the geometric position where the furnishing symbol is recognized. To detect the nearest wall line and place the electrical power symbol appropriately, the following procedures are performed, in the given order [2]:

- Select the nearest line segment (wall line) to the central point (cp) of the identified furnishing symbol. This line segment will be considered the reference line ($refLine$).
- Draw a perpendicular line ($perpLine$) to the $refLine$ from the cp and find the intersection point ($intPt$).
- The angle of $perpLine$ is considered the rotation angle ($rotAng$) for the electrical power symbol.
- A specified distance ($delta$) along the $perpLine$ from the $intPt$ becomes the insertion point ($ToolPt$) for the electrical power symbol.

The placement of the electrical power symbols on the appropriate line segments (walls) according to the steps above is shown in Figure 11.

As seen in Figure 12, the electrical power symbols matching the furnishing symbols found were selected and placed on the appropriate walls. The light symbol was selected according to the study in [40] and placed

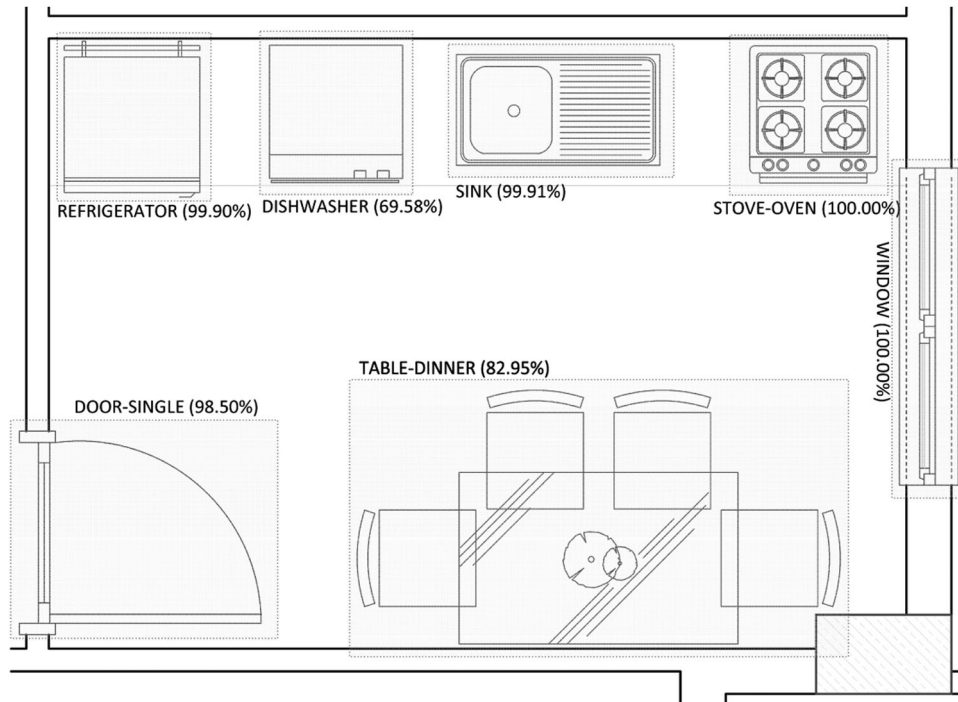


Figure 9. Furnishing symbol detection and recognition in the selected zone.

Table 4. Selection of electrical power symbol according to furnishing symbol.

Furnishing symbol	#	Electrical power symbol	Power (Watt)	Description
STOVE-OVEN	1	Oven Soc.	2500	For oven
	2	Normal Soc.	300	For stove
WASHINGMACHINE	1	Wash. Mac. Soc.	2500	
DISHWASHER	1	Dishwasher Soc.	2000	
REFRIGERATOR	1	Normal Soc.	300	
SINK	1	Normal Soc.	300	For various kitchen appliances (Kettle, blender, mixer...)
TABLE-DINNER	1	Normal Soc.	300	
TELEVISION	1	Normal Soc.	300	For TV
	2	Normal Soc.	300	For satellite receiver
	3	Antenna Soc.	-	
	4	Phone Soc.	-	
TABLE-STUDY	1	Normal Soc.	300	Computer, Table Lamp,...
	2	Phone Soc.	-	
SOFA-TWO	1	Normal Soc.	300	
SOFA-THREE	1	Normal Soc.	300	To the right side
	2	Normal Soc.	300	To the left side
SOFA-CORNER	1	Normal Soc.	300	
BED-DOUBLE	1	Normal Soc.	300	To the right side
	2	Normal Soc.	300	To the left side
BED-SINGLE	1	Normal Soc.	300	
WASHBASIN-CABINET	1	Waterproof Soc.	300	Electric shaver, hair dryer...
DOOR-SINGLE	1	Light switch	-	

in the appropriate place. The readers can refer to [40] for detailed information about selecting and placing appropriate luminaires according to the specified zone (room), based on lighting calculations.

6. Conclusions and future work

This paper carries out a study to intelligently draw electrical power symbols in appropriate places, depending on furnishing symbols, in order to automatically draw

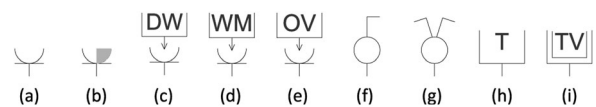


Figure 10. Some electrical power symbols: (a) Normal Socket, (b) Waterproof Socket, (c) Dishwasher Socket, (d) Washing Machine Socket, (e) Oven Socket, (f) Single Light Switch, (g) Double Light Switch, (h) Phone Socket, (i) Antenna Socket.

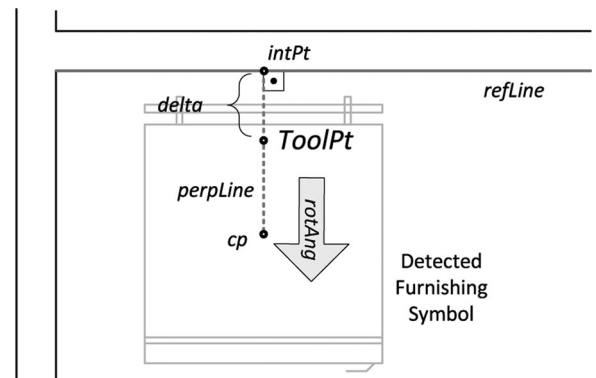


Figure 11. Placement of electrical power symbol on the appropriate wall.

electrical plans. To the best of our knowledge, deep CNN and transfer learning are used for the first time in electrical plan drawing to automatically place electrical power symbols. With this study, a new vision has been presented for the automatic placement of electrical power symbols based on the recognition of furnishing symbols. After recognizing furnishing symbols, algorithms were developed and successfully applied to

Table 5. The steps of “the automated electrical power symbols placement for electrical plan drawing”.

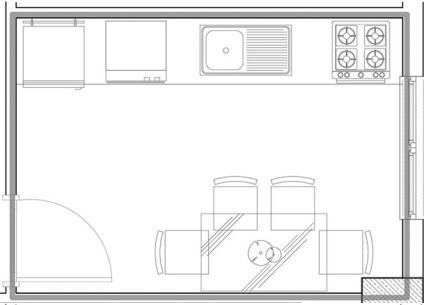
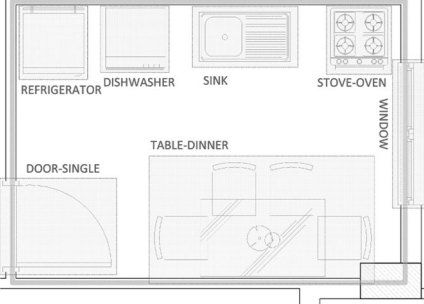
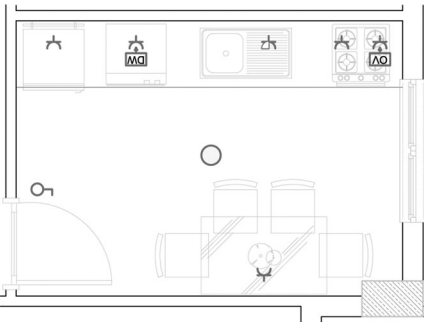
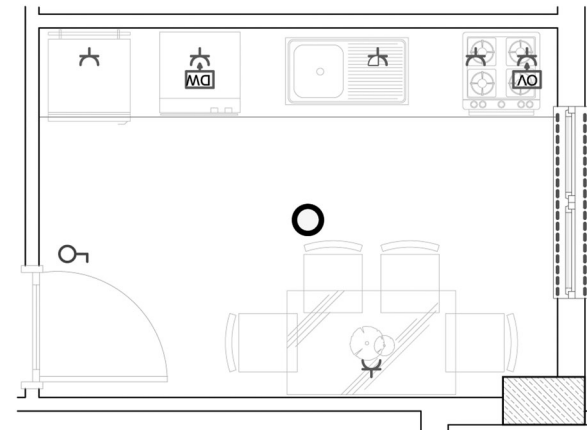
	<p>Step-1</p> <ul style="list-style-type: none"> • Rectangular zones can be selected automatically by clicking inside an area with the help of cursor crosshair reference lines (see Figure 8)). • Selection of a zone (room) is carried out by clicking two corner points (see Figure 8)). • Any polygonal room can be selected by individual clicks on the corner points of the zone. • A polygon is created that determines the boundaries of the zone (room-polygon).
	<p>Step-2</p> <ul style="list-style-type: none"> • All block objects inside the room-polygon are listed. <ul style="list-style-type: none"> * Seven blocks are listed in the left-hand figure. • Each block will be considered a furnishing symbol. • Convert each block to three bitmap images with the specified resolutions (112×112, 224×224 and 299×299) and properties (line thickness: 1 pixel, line colour: white, line type: continuous) for the CNN model. • Forward each converted image to the trained CNN model to classify/recognize the “furnishing symbols”. <ul style="list-style-type: none"> * If the prediction for at least two of three images is the same, then it will be considered as the prediction of the model.
	<p>Step-3</p> <ul style="list-style-type: none"> • Determine the electrical power symbols for the recognized furnishing symbols using Table 4. • Place the identified electrical power symbols on the nearest wall line where the furnishing symbol is. • Place the lighting fixture in the centre of the selected area. <ul style="list-style-type: none"> * The type, number, and position of the selected lighting fixture are determined according to [40].
	<p>only from among the those most commonly used for residential buildings.</p> <p>Furnishing symbol classification is basically an image classification problem. From past to present, much work has been done to classify images. Numerous different (new or pre-trained) CNN models (ResNet, MobileNet, VGG, etc.) with different fine-tuning can be tested to improve the success of classification for this objective.</p> <p>The automated placement of the electrical power symbols successfully depends on the correct recognition of the furnishing symbols. Therefore, it can be said that the success of the proposed system is the same as the success of the classification. Since we created the dataset for the proposed system ourselves, it would not be appropriate to compare it with the previous symbol recognition studies. Yet the success rate of 97.05% is a very satisfactory result.</p> <p>We used the Microsoft ML.NET Image Classification API in the .Net environment to create a classifier model for furnishing symbols. Satisfactory results were obtained according to the success of the proposed model. This study will stimulate the use of artificial intelligence in the field of electrical plan drawings.</p>

Figure 12. Placement of electrical power symbols according to recognized furnishing symbols.

automatically place electrical power symbols in appropriate locations. The steps of the study are summarized in Table 5.

The proposed system works on vector-based real-world residential architectural floor plans. The furnishing (decorative) symbols should be in block forms. The furnishing symbols used in this study were selected

The furnishing symbol dataset and test floor plans (in dxf, pdf and jpg formats) are publicly available at <https://www.kaggle.com/bayramakgl/furnishing-data-set>.

A video from the implementation of this study is available for the electronic version of this paper as supplemental material.

We aim to draw a fully automatic electrical plan drawing. In the future, we would expect to parse floor plans for rooms automatically instead of selecting them manually. This will take us one step closer to fully automated electrical plan drawings. Room types can be identified according to the furnishing symbols inside. For example, if an oven symbol is recognized in a room zone, this means this zone is most likely a kitchen. In our previous work [40], we created a lighting simulation program that makes recommendations for the number of luminaires, types, and insertion points to be placed in a three-dimensional closed area to provide a predetermined minimum illumination level in accordance with the type, size, and purpose of use. By integrating this previous work into our current study, we will be able to automatically place the correct number and type of lighting fixtures in the automatically identified rooms.

Automating the drawing process in CAD applications involves obtaining engineering calculations from the drawing. Consequently, the next step of this study is to connect electrical power symbols to distribution panels automatically and to ensure that all calculations are done in the same manner.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Bayram Akgul  <http://orcid.org/0000-0003-0885-4978>

Hakan Kutucu  <http://orcid.org/0000-0001-7144-7246>

References

- [1] Teo CY. Computer-aided design and simulation of low-voltage electrical distribution systems. *Comput Ind.* 1997;34(1):87–94. DOI:10.1016/s0166-3615(96)00072-3[Q4]
- [2] Orak LM, Akgul B. Fast and simple computer aided internal electrical wiring project design and integration of calculations. *Proc Technol.* 2012;1:280–285. DOI:10.1016/j.protcy.2012.02.059
- [3] Harris LVA, Meyers F. Engineering design graphics: into the 21st century. *Eng Des Graph J.* 2007;71(3):9–23.
- [4] Bi Z, Wang X. Computer aided design and manufacturing. 1st ed. New Delhi: Prentice Hall of India; 2020. DOI:10.1002/9781119667889
- [5] Domínguez B, García ÁL, Feito FR. Semiautomatic detection of floor topology from CAD architectural drawings. *CAD Comput Aided Des.* 2012;44(5):367–378. DOI:10.1016/j.cad.2011.12.009
- [6] Gerstweiler G, Furlan L, Timofeev M, et al. Extraction of structural and semantic data from 2D floor plans for interactive and immersive VR real estate exploration. *Technologies.* 2018;6(4):101. DOI:10.3390/technologies6040101
- [7] Zhu J, Zhang H, Wen Y. A new reconstruction method for 3D buildings from 2D vector floor plan. *Comput Aided Des Appl.* 2014;11:704–714. DOI:10.1080/16864360.2014.914388
- [8] Yin X, Wonka P, Razdan A. Generating 3D building models from architectural drawings: a survey. *IEEE Comput Graph Appl.* 2009;29(1):20–30. DOI:10.1109/mcg.2009.9
- [9] Ah-Soon C, Tombre K. Architectural symbol recognition using a network of constraints. *Pattern Recognit Lett.* 2001;22(2):231–248. DOI:10.1016/S0167-8655(00)00091-X
- [10] Dosch P, Tombre K, Ah-Soon C, et al. A complete system for the analysis of architectural drawings. *Int J Doc Anal Recognit.* 2000;3(2):102–116. DOI:10.1007/PL00010901
- [11] Guo T, Zhang H, Wen Y. An improved example-driven symbol recognition approach in engineering drawings. In: *Computers and graphics (Pergamon)*. Vol. 36. Elsevier Ltd; 2012. p. 835–845. DOI:10.1016/j.cag.2012.06.001
- [12] de las Heras L-P, Terrades OR, Robles S, et al. CVC-FP and SGT: a new database for structural floor plan analysis and its groundtruthing tool. *Int J Doc Anal Recognit (IJ DAR).* 2015;18(1):15–30. DOI:10.1007/s10032-014-0236-5
- [13] Rezvanifar A, Cote M, Branzan Albu A. Symbol spotting for architectural drawings: state-of-the-art and new industry-driven developments. *IPSJ Trans Comput Vis Appl.* 2019;11(1):2. DOI:10.1186/s41074-019-0055-1
- [14] Rezvanifar A, Cote M, Albu AB. Symbol spotting on digital architectural floor plans using a deep learning-based framework. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE Computer Society; 2020 Jun. p. 2419–2428. arXiv:2006.00684. DOI:10.1109/CVPRW50498.2020.00292
- [15] Valveny E, Dosch P. Symbol recognition contest: a synthesis. In: *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*. 2004. p. 368–385. (LNCS; vol. 3088). DOI:10.1007/978-3-540-25977-0_34
- [16] Valveny E, Delalandre M, Raveaux R, et al. Report on the symbol recognition and spotting contest. In: *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*. Berlin, Heidelberg: Springer; 2013. p. 198–207. (LNCS; vol. 7423). DOI:10.1007/978-3-642-36824-0_19
- [17] Delalandre M, Pridmore T, Valveny E, et al. Building synthetic graphical documents for performance evaluation. In: *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*. Berlin, Heidelberg: Springer; 2008. p. 288–298. (LNCS; vol. 5046). DOI:10.1007/978-3-540-88188-9_27
- [18] Rusiñol M, Borràs A, Lladós J. Relational indexing of vectorial primitives for symbol spotting in line-drawing images. *Pattern Recognit Lett.* 2010;31(3):188–201. DOI:10.1016/j.patrec.2009.10.002
- [19] Cote M, Rezvanifar A, Albu AB. Automatic generation of electrical plan documents from architectural data. In: *Proceedings of the ACM Symposium on Document Engineering 2020, DocEng '20*. New York

- (NY): Association for Computing Machinery; 2020. DOI:10.1145/3395027.3419598
- [20] Gavali P, Banu JS. Deep convolutional neural network for image classification on CUDA platform. In: Deep learning and parallel computing environment for bioengineering systems. 2019. p. 99–122. DOI:10.1016/b978-0-12-816718-2.00013-0
- [21] Lecun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015;521(7553):436–444. DOI:10.1038/nature14539
- [22] Yun K, Huyen A, Lu T. Deep neural networks for pattern recognition. In: Advances in pattern recognition research. 2018. p. 49–79. arXiv:1809.09645. DOI:10.1016/0893-6080(95)90002-0
- [23] Kaur T, Gandhi TK. Deep convolutional neural networks with transfer learning for automated brain image classification. In: Machine vision and applications. Vol. 31. 2020. p. 20. DOI:10.1007/s00138-020-01069-2
- [24] Weiss K, Khoshgoftaar TM, Wang DD. A survey of transfer learning. *J Big Data*. 2016;3(1). DOI:10.1186/s40537-016-0043-6
- [25] Bird JJ, Kobylarz J, Faria DR, et al. Cross-domain MLP and CNN transfer learning for biological signal processing: EEG and EMG. *IEEE Access*. 2020;8:54789–54801. DOI:10.1109/ACCESS.2020.2979074
- [26] Pan SJ, Yang Q. A survey on transfer learning. *IEEE Trans Knowl Data Eng*. 2010;22(10):1345–1359. DOI:10.1109/TKDE.2009.191
- [27] Russakovsky O, Deng J, Su H, et al. ImageNet large scale visual recognition challenge. *Int J Comput Vis*. 2015;115(3):211–252. arXiv:1409.0575. DOI:10.1007/s11263-015-0816-y
- [28] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2016. p. 2818–2826. DOI:10.1109/CVPR.2016.308
- [29] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition; Vol. 2016. 2016 Dec. p. 770–778. arXiv:1512.03385. DOI:10.1109/CVPR.2016.90.
- [30] He K, Zhang X, Ren S, et al. Identity mappings in deep residual networks. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics). 2016. p. 630–645. (LNCS; vol. 9908). arXiv:1603.05027. DOI:10.1007/978-3-319-46493-0_38.
- [31] Sandler M, Howard A, Zhu M, et al. MobileNetV2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2018. p. 4510–4520. arXiv:1801.04381. DOI:10.1109/CVPR.2018.00474
- [32] Autodesk Inc. DXF format. 2018. Available from: <http://help.autodesk.com/cloudhelp/2018/ENU/AutoCAD-DXF/files/index.htm>.
- [33] Yang H, Patras I. Mirror, mirror on the wall, tell me, is the error small? In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2015 Jun 7–12. p. 4685–4693. arXiv:1501.05152. DOI:10.1109/CVPR.2015.7299100
- [34] Xie S, Tu Z. Holistically-nested edge detection. *Int J Comput Vis*. 2017;125(1-3):3–18. arXiv:1504.06375. DOI:10.1007/s11263-017-1004-z
- [35] Gu J, Wang Z, Kuen J, et al. Recent advances in convolutional neural networks. *Pattern Recognit*. 2018;77:354–377. DOI:10.1016/j.patcog.2017.10.013
- [36] Pan W, Narasimhan H, Protopapas P, et al. Optimizing the multiclass F-measure via biconcave programming. In: Proceedings – IEEE International Conference on Data Mining, ICDM; 2017. p. 1101–1106. DOI:10.1109/ICDM.2016.184
- [37] Zhao ZQ, Zheng P, Xu ST, et al. Object detection with deep learning: a review. 2019. arXiv:1807.05511. DOI:10.1109/TNNLS.2018.2876865
- [38] Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell*. 2017;39(6):1137–1149. arXiv:1506.01497. DOI:10.1109/TPAMI.2016.2577031
- [39] Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2016. arXiv:1506.02640. DOI:10.1109/CVPR.2016.91
- [40] Akgül B, Bilici A, Kutucu H. A fast and simple computer aided lighting simulator. In: International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES 2018); Karabük; 2018. p. 442–446.