

# A Public Ground-Truth Dataset for Handwritten Circuit Diagram Images

Felix Thoma<sup>1,2</sup>, Johannes Bayer<sup>1,2</sup>, and Yakun Li<sup>1</sup>

<sup>1</sup> Smart Data & Knowledge Services Department, DFKI GmbH  
Kaiserslautern, Germany

{felix.thoma,johannes.bayer,yakun.li}@dfki.de

<sup>2</sup> Computer Science Department, TU Kaiserslautern  
Kaiserslautern, Germany

**Abstract.** The development of digitization methods for line drawings – especially in the area of electrical engineering – relies on the availability of publicly available training and evaluation data. This paper presents such an image set along with annotations. The dataset consists of 1152 images of 144 circuits by 12 drafters and 48 563 annotations. Each of these images depicts an electrical circuit diagram, taken by consumer grade cameras under varying lighting conditions and perspectives. A variety of different pencil types and surface materials has been used. For each image, all individual electrical components are annotated with bounding boxes and one out of 45 class labels. In order to simplify a graph extraction process, different helper symbols like junction points and crossovers are introduced, while texts are annotated as well. The geometric and taxonomic problems arising from this task as well as the classes themselves and statistics of their appearances are stated. The performance of a standard Faster RCNN on the dataset is provided as an object detection baseline.

**Keywords:** Circuit Diagram · Ground Truth · Line Drawing

## 1 Introduction

The sketch of a circuit diagram is a common and important step in the digital system design process. Even though the sketch gives the designer a lot of freedom to revise, it is still time consuming to transform the sketch into a formal digital representation for existing tools to simulate the designed circuit. There are mainly two approaches which have been applied to address the transformation from sketch into a useful format to be used by existing simulation tools. The first approach allows the user to draw the diagram in an on-line fashion on a screen and tries to recognize the sketch by interpreting the pen strokes of user, like shown in [7], [2] and [3]. The second, off-line approach assumes that the diagrams are drawn with a pen on paper, from which an image is captured afterwards. This approach relies on machine learning and computer vision algorithms,

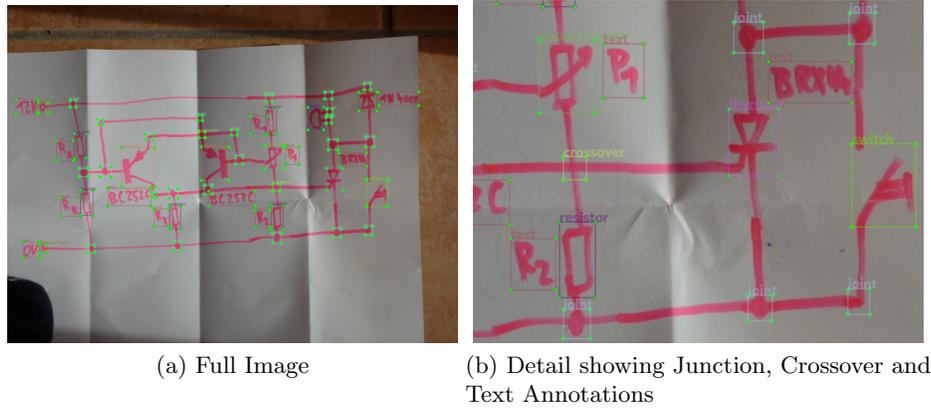


Fig. 1: Dataset Sample with Annotations

like shown in [4], [8], [9] and [1], to process the images. It is important to collect large and high quality training data for the machine learning and computer vision approach, especially when deep learning algorithms are applied.

However, most of this literature[4][8][9][1] exposes only little of the data it relies on publicly and to the best of the knowledge of this paper’s authors, there is no publicly available comparable dataset.

The dataset described in this paper mainly addresses this issue and will be made publicly available<sup>1</sup> under the MIT Licence<sup>2</sup>. Previous datasets described in literature will be summarized in section 2. Afterwards, the characteristics of the hand-drawn circuit diagrams images contained in the dataset will be stated in section 3. In the end, section 6 will present basic recognition results using a Faster-RCNN [10] on the presented dataset.

## 2 Related Work

The current research papers for the off-line approach mentioned in section 1 mainly focus on the algorithms to recognize the sketches and their datasets are not publicly accessible. While [1] presented a comprehensive system to recognize hand drawn sketches and transform the recognition results to Verilog code, there was no description of the concrete number of images and the variety of symbols. In [9], the author proposed to use Fourier Descriptors to get feature representation of hands drawn circuit components and then use SVM to classify them. They used 10 hand drawn circuit sketches as test data, but only one circuit sketch was shown as an example. The author of [4] gave information of the number of nodes and components included in their hand drawn sketches but

<sup>1</sup> <https://git.opendfki.de/circuitgraph-public/cghd/>

<sup>2</sup> <https://opensource.org/licenses/MIT>

without the exact number of sketches. [8] mentioned that they did their evaluation on 100 hand drawn circuit diagrams, however the quantity is relatively small compared with the dataset presented in the current paper.

### 3 Images

Every image of the dataset contains a hand-drawn electrical circuit (see fig. 1). Every of the 12 drafters was instructed to draw 12 circuits, each of them 2 times, where it was allowed the drawers to alter the circuit geometrically. Each of the drafts was photographed 4 times, resulting in an overall image count of 1152. Taking multiple images of the same drawing was based on the following considerations:

- The dataset can conveniently be used for style transfer learning
- Since different images of the same drawing contain the same annotation items, they can be used for automatic verification of the labeling process
- The cost of creating a single sample is reduced
- Different real-world image captures are a realistic and rich replacement for artificial data augmentation
- Different samples of the same drawing must be carefully annotated to ensure the object detection task is well-defined (no ambiguities)
- enforcing diverse capturing conditions aim to support camera feeds

On a semantic level the user can draw a similar circuit using different symbols, for example to create a voltage source a DC source can be used, a Vcc together with GND could be used, or only the Terminals could be drawn with textual description of the meaning of it.

#### 3.1 Drawing Surfaces and Instruments

The depicted surface types include: plain paper of various colors, ruled paper, squared paper, coordinate paper, glas, whiteboard, cardboard and aluminum foil. Pens, pencils, permanent markers and whiteboard markers of different colors have been utilized for drawing. A few drawings contain multiple colors, or contain text that has been highlighted by a dedicated color. Buckling, kinking, bending, spots, transillumination, and paper cracks can be found as distortions on the samples. Some samples have been treated by corrective means, others have been created using a ruler as a drawing aid rather than free-hand sketching.

#### 3.2 Capturing

All images have been captured by consumer cameras, no artificial image distortion has been applied. Some images have been cropped and company logos have been obliterated. Camera position and spatial orientation alternations as well as steam have been used in various degrees. That way, naturally observable distortions

like (motion) blur and light reflexes of surfaces as well as ink are captured in the dataset. Despite these disturbances no major occlusions are present. More precisely, circuit symbols and their connections are recognizable by humans and the circuits are oriented right side up. Often, circuit drawings are surrounded by common (office environment) objects. The sample image sizes range between 864 by 4160 and 396 by 4032 and are stored as JPEG.

## 4 Annotations

Bounding box annotations stored in the PascalVOC[5] format have been chosen for symbols and texts as a trade-off between effort in ground truth generation and completeness in capturing the available information from the images. The downsides of this decision are the lack of rotation information of the symbols as well as the lack of wire layout capturing and will be addressed by the auxiliary classes below. The annotations have been created manually using labelIMG[12].

### 4.1 Auxiliary Classes

In order to address the lack of wire-line annotation and to simplify the extraction of a circuit graph, auxiliary classes are introduced:

- **junction** represents all connections or kink-points of wire edges. Since wire corners and junctions share the same semantic from the electrical graph point of view, they share the same label.
- **crossover** represents the optical crossing of a pair of lines that is not physically connected.
- **terminal** denotes an 'open' wire ending, i.e. a circuit input or output.
- **text** encloses a line of text, usually physical quantities or terminal or component name.

Some symbols (like potentiometers and capacitors) have outgoing lines or corners (like transistors or switches) which are considered part of their definition. Hence, these line (parts) are also included in the bounding boxes and junction annotations within them are neglected. Likewise, text annotations related to symbols are only omitted if they are part of the symbol definition (like an 'M' in the motor symbol). If they are however explaining component parts (like integrated circuit pins) or the component itself (like integrated circuit type labels), they are annotated (even if they are located inside the symbol).

### 4.2 Symbol Classes

The dataset contains both IEE/ANSI as well as IEC/DIN symbols. Individual circuit drawings are allowed to contain both variants as well as a mix of analogue and digital symbols. Some drafters also used non standard-conform symbols, for example a speaker explicitly containing a coil to show its exact implementation.

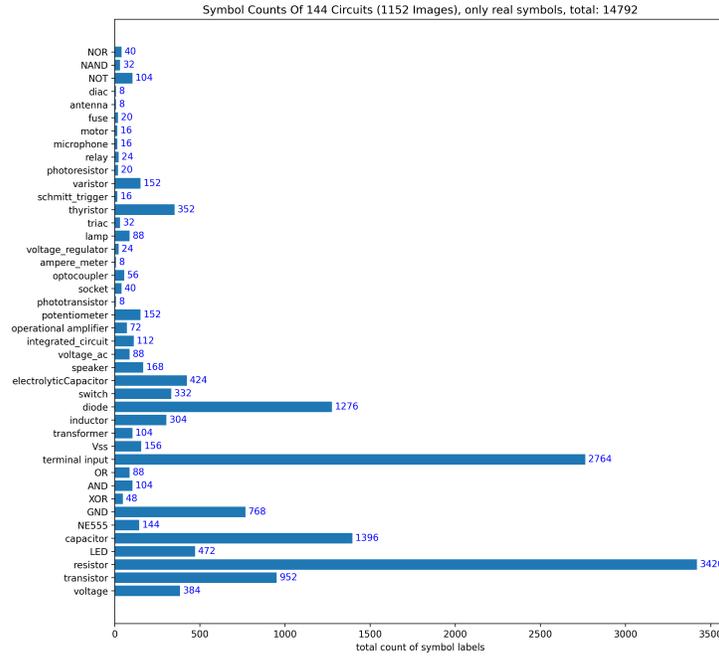


Fig. 2: Total count of label instances in the full dataset. Junctions, crossover and text are not considered, due to their dominance in the dataset.

The classes have been chosen to represent *electrically* connectable components. For example, if there is a full-wave rectifier depicted as four diodes, each of the diodes is annotated individually. Likewise, a relay label is used as an annotation rather than a switch and an inductor (or the two inductors of a transformer), since the components are coupled *inductively*. However, if the individual parts are separated so that other components are in between, an individual labeling is used as a makeshift. To simplify annotation, some of the electrical taxonomy is neglected. For example, all bipolar transistors, Mosfets and IGBTs are labeled *transistors*.

### 4.3 Geometry

Generally, the bounding box annotations have been chosen to completely capture the symbol of interest, while allowing for a margin that includes the surrounding space. Often, there is an overlap between tightly placed symbols. The task of fully capturing all parts of the symbol was not well-defined in the context of curvy straight lines connected by rounded edge corners.

## 5 Statistics

The dataset has a total amount of 48 563 annotations among 45 classes. The junction class occurs 18 229 times, the text class 14 231 times and the crossover class 1 396 times. All other electrical symbol counts are listed in detail in fig. 2. The frequency of symbol occurrences per circuit are displayed in fig. 3.

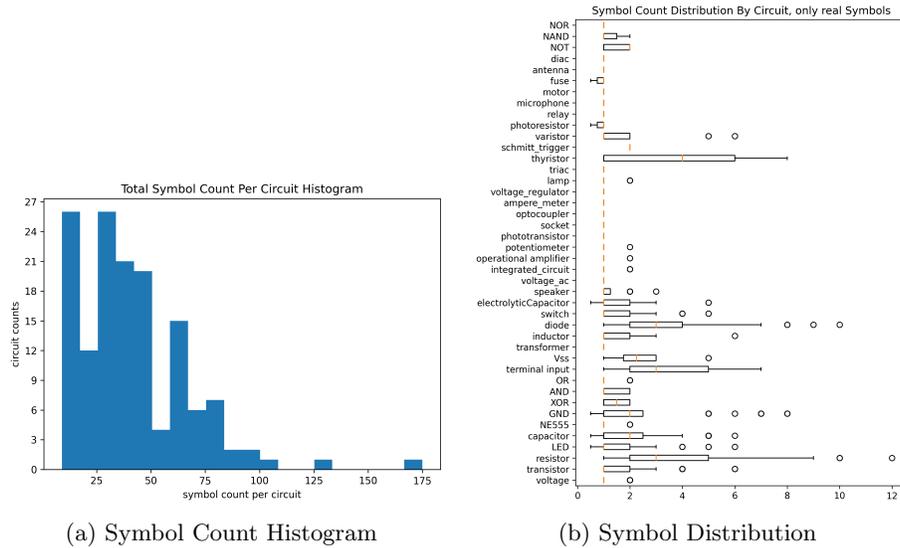


Fig. 3: a) Total symbol count per circuit. Histogram is counting the occurrences in 20 evenly spaced bins.  
 b) Distribution of symbol count per circuit that the symbol is part of. Junctions, crossover and text are not considered, due to their dominance in the dataset.

## 6 Baseline Performance

A suggested division of the dataset into subsets for training (125 circuits), validation (7 circuits) and testing (12 circuits, all from the same drafter), is provided so that images of every circuit (and therefore drawing) are exclusively assigned to one of the sets only.

In order to establish a baseline performance on the dataset, a Faster RCNN[10] with a ResNet152[6] backbone was applied using the Torchvision[11] implementation, a learning rate of 0.007 and a batch size of 8, yielding an mAP of 52%.

## Acknowledgement

The authors cordially thank Thilo Pütz, Anshu Garg, Marcus Hoffmann, Michael Kussel, Shahroz Malik, Syed Rahman, Mina Karami Zadeh, Muhammad Nabeel Asim and all other drafters who contributed to the dataset. This research was funded by the German Bundesministerium für Bildung und Forschung (Project SensAI, grant no. 01IW20007).

## References

- [1] Mohammad Abdel-Majeed et al. “Sketic: a machine learning-based digital circuit recognition platform”. In: *Turkish Journal of Electrical Engineering and Computer Sciences* 28 (2020), pp. 2030–2045.
- [2] Christine Alvarado and Randall Davis. “SketchREAD: A Multi-Domain Sketch Recognition Engine”. In: *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*. UIST ’04. Santa Fe, NM, USA: Association for Computing Machinery, 2004, 23–32. ISBN: 1581139578. DOI: 10.1145/1029632.1029637. URL: <https://doi.org/10.1145/1029632.1029637>.
- [3] Christine Alvarado et al. “LogiSketch: A Free-Sketch Digital Circuit Design and Simulation SystemLogiSketch”. In: July 2015, pp. 83–90. ISBN: 978-3-319-15593-7. DOI: 10.1007/978-3-319-15594-4\_8.
- [4] B. Edwards and V. Chandran. “Machine recognition of hand-drawn circuit diagrams”. In: *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)* 6 (2000), 3618–3621 vol.6.
- [5] M. Everingham et al. “The Pascal Visual Object Classes (VOC) Challenge”. In: *International Journal of Computer Vision* 88.2 (June 2010), pp. 303–338.
- [6] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: <http://arxiv.org/abs/1512.03385>.
- [7] Marcus Liwicki and Lars Knipping. “Recognizing and Simulating Sketched Logic Circuits”. In: *Knowledge-Based Intelligent Information and Engineering Systems*. Ed. by Rajiv Khosla, Robert J. Howlett, and Lakhmi C. Jain. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 588–594. ISBN: 978-3-540-31990-0.
- [8] Momina Moetesum et al. “Segmentation and Recognition of Electronic Components in Hand-Drawn Circuit Diagrams”. In: *EAI Endorsed Trans. Scalable Inf. Syst.* 5 (2018), e12.
- [9] Mayuri D. Patare and M. Joshi. “Hand-drawn Digital Logic Circuit Component Recognition using SVM”. In: *International Journal of Computer Applications* 143 (2016), pp. 24–28.
- [10] Shaoqing Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *CoRR* abs/1506.01497 (2015). arXiv: 1506.01497. URL: <http://arxiv.org/abs/1506.01497>.

- [11] *torchvision*. URL: <https://github.com/pytorch/vision>. (accessed: 25.04.2021).
- [12] Tzutalin. *LabelImg*. URL: <https://github.com/tzutalin/labelImg>. (accessed: 25.04.2021).