Open Access Original Article

Hand-Drawn Electronic Component Detection and Simulation Using Deep Learning and Flask

Anmol Arun [1] , Sejal S. Patil [1] , Gayathri M. Poduval [1] , Amar Shah [1] , Rupali P. Patil [1]

1. Electronics and Telecommunication Engineering, K. J. Somaiya School of Engineering, Mumbai, IND

Corresponding author: Anmol Arun, a.arun@somaiya.edu

Received 03/11/2025
Review began 04/16/2025
Review ended 08/09/2025
Published 08/11/2025

© Copyright 2025
Arun et al. This is an open access article distributed under the terms of the Creative Commons Attribution License CC-BY 4.0., which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

DOI:
https://doi.org/10.7759/s44389-025-03676-4

Abstract

This paper presents a key innovation - the creation of a highly user-friendly web application that allows users to easily upload their hand-drawn circuit diagrams and instantly receive LTspice-compatible .asc files. This level of automation and simplicity marks a significant breakthrough in electronic design automation. A system is designed to automate the simulation of hand-drawn circuit diagrams, bridging the gap between manual circuit design and simulation readiness. Leveraging a custom convolutional neural network (CNN) model trained on a specialized custom dataset, the system accurately recognizes over 25 distinct electronic components from hand-drawn schematics. The detected components are translated into LTspice Netlist (.asc) files, enabling seamless circuit simulation. The methodology involved understanding the problem, creating the custom dataset, developing the custom CNN model, and integrating the backend with a user-friendly web interface using Flask. The proposed custom deep learning model outperformed existing solutions with a training accuracy of 99% and a test accuracy of 96%. This approach simplifies the traditionally complex simulation process, allowing educators and students to focus on circuit design principles while enhancing accessibility and enriching electronics education.

Categories: AI applications, Computer Vision, Deep Learning
Keywords: cnn, deep learning, ltspice, simulation, electronic component detection

Introduction

Circuit simulation tools have long been a fundamental part of electronics education, helping students and professionals analyze and understand circuit behavior. Traditional software such as LTspice, Multisim, and MATLAB offer powerful capabilities but often come with a steep learning curve, making them challenging for beginners. Translating conceptual circuit designs into simulation-ready formats can be particularly daunting for students in the early stages of learning.

With advancements in artificial intelligence and machine learning, there is an opportunity to simplify this process by automating the conversion of hand-drawn circuit diagrams into digital formats. Such automation reduces manual effort and fosters a more interactive, accessible learning experience. However, a key challenge lies in accurately identifying and classifying the diverse range of electronic components in hand-drawn schematics. Variability in size, orientation, and symbol style make recognition difficult for traditional image processing techniques, which rely on predefined templates and rules. Additionally, bridging the gap between recognized components and the structured syntax required for simulation software adds another layer of complexity.

To address these challenges, we propose an automated system that leverages convolutional neural networks (CNNs) for component recognition and integrates with LTspice for circuit simulation. Our approach involves training a custom CNN model to accurately identify over 25 distinct electronic components, even in the presence of inconsistencies. These recognized components are then converted into LTspice Netlists, streamlining the simulation process. The system is deployed as a web-based platform where users can upload hand-drawn circuit images and instantly generate simulation-ready files. By minimizing the complexity of traditional simulation tools, this solution enhances the learning experience, making circuit design more intuitive and accessible for students and educators alike.

Related work

Recent advancements in hand-drawn electric circuit diagram recognition have leveraged a variety of machine learning and computer vision techniques to automate interpretation and enhance accuracy. Deep learning methods like Faster R-CNN, especially with region proposal networks, have demonstrated strong performance in detecting circuit components with minimal preprocessing, achieving a 75.9% mean average precision across 12 classes [1]. The Inception V2 Faster R-CNN architecture further refined this approach, achieving high success rates in identifying series and parallel RLC circuits [2]. Real-time methodologies like YOLOv5 have improved schematic generation, with 98.2% accuracy in component detection and 80% in schematic reconstruction [3]. For efficient yet simpler applications, ORB (Oriented

FAST and Rotated BRIEF) has outperformed SIFT in hand-drawn electronic component recognition due to its speed and accuracy, though it faces scalability challenges for larger datasets [4].

Automated netlist generation has seen significant advancements as well. Img2Sim, an artificial neural network (ANN)-based approach, converts images into simulation-ready netlists with a detection accuracy of 98% and 90% accuracy in netlist generation, highlighting its utility in automated circuit design [5]. Approaches combining histogram of oriented gradients (HOG) with support vector machines (SVMs) have demonstrated impressive segmentation and classification rates of 87.7% and 92%, respectively, across multiple classes [6,7]. Finite state machine-based models utilizing local binary pattern features achieved exceptional performance with 99% accuracy in component detection and 85% in circuit recognition [8]. However, methods such as those employing morphological operations and run length smoothing algorithms still face challenges in detecting oversized or closely spaced components, achieving an overall accuracy of 91.28% while missing 8.72% of components [9,10]. These advancements and ongoing refinements are paving the way for more robust and efficient automated systems in recognizing and analyzing hand-drawn circuit diagrams [11].

Materials And Methods

The dataset [12] utilized in this study was initially sourced from a publicly available collection on Kaggle, containing hand-drawn images of 12 different types of electronic components. To increase the dataset's comprehensiveness and enhance its applicability to real-world scenarios, the dataset was further augmented with six additional component categories: switches, operational amplifiers (OP-AMPs), bipolar junction transistors (BJTs), metal-oxide-semiconductor field-effect transistors (MOSFETs), junction field-effect transistors (JFETs), and connecting wires. This expansion brought the total number of component classes to 25, with each class consisting of 200 annotated images. As a result, the complete dataset comprised 4,800 images intended for training purposes. All images were standardized to a resolution of 120 × 120 pixels and stored in the uncompressed .bmp format to maintain uniformity and preserve detail critical for accurate classification. Figure *1* showcases sample images from the custom dataset, highlighting the variety in hand-drawn electronic component representations.

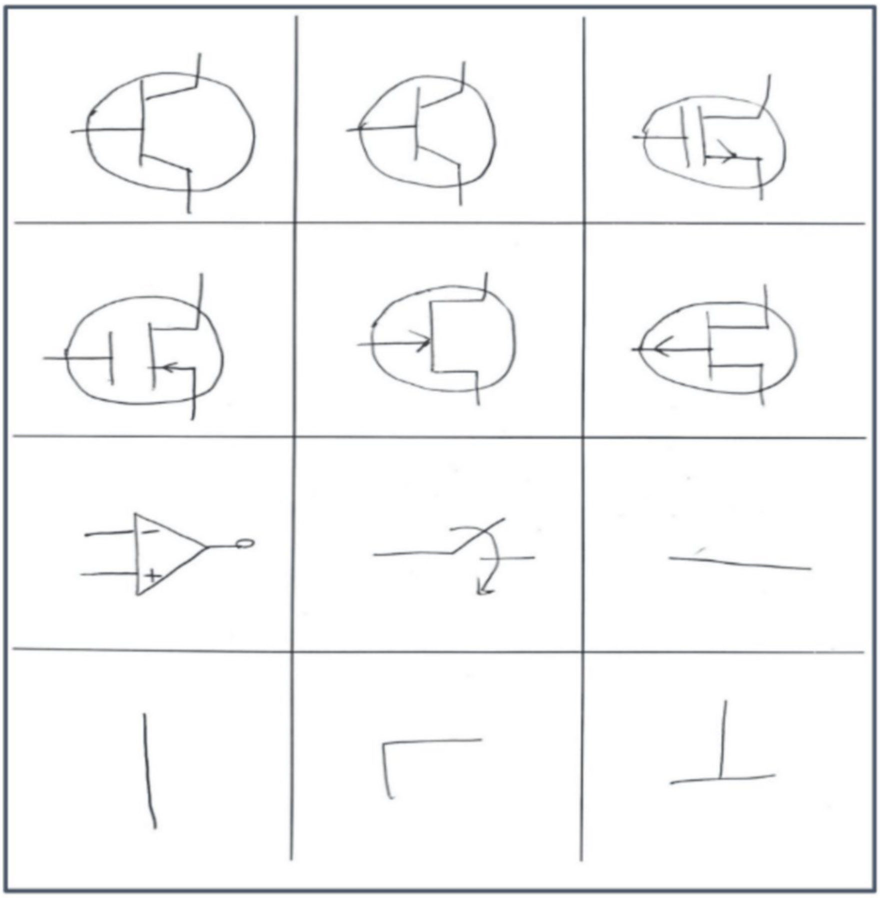


FIGURE 1: Sample custom-made dataset

To ensure optimal model performance, a thorough preprocessing pipeline was applied to the collected data. First, color inversion was performed on all images to achieve consistent contrast, thereby improving the visibility of component boundaries and enhancing feature extraction during model training. Data augmentation was employed to introduce rotational variance and improve generalization. Each image was rotated by 90°, 180°, and 270°, effectively, quadrupling the original dataset size and enabling the model to learn invariant representations across different orientations. In addition, to center the focus on the component structure while removing irrelevant background noise, all images were cropped to 60 × 60 pixels. This helped concentrate the learning on the region of interest, further boosting the classifier's robustness. Collectively, these preprocessing steps contributed to creating a highly diverse and noise-resistant dataset that supports the model's adaptability to various real-world input styles.

The entire system pipeline - from data preprocessing to model inference and feedback generation - was designed to be efficient and end-user friendly. This ensures that users can interact with the system seamlessly, using simple hand-drawn images to identify and classify electronic components accurately. The end-to-end pipeline supports the practical deployment of the system in educational, prototyping, or diagnostic environments, where quick recognition and classification of hand-sketched components can enhance productivity and accessibility.

Custom CNN model architecture

A custom CNN was developed specifically for the classification task using grayscale input images resized to 64 × 64 pixels. The model architecture begins with an input layer designed to accept the specified image dimensions. The first convolutional layer utilizes 32 filters of size 3 × 3 with 'same' padding and ReLU activation to extract low-level features such as edges and corners. This is followed by a dropout layer with a rate of 0.2 to prevent overfitting and a max-pooling layer that reduces the spatial dimensions by a factor of two. The second convolutional layer, with 64 filters and ReLU activation, captures more abstract patterns. This layer is also followed by dropout (0.3) and max-pooling.

The third convolutional layer retains the use of 64 filters and is followed by a dropout rate of 0.4 and max-pooling, helping the network learn increasingly complex features while maintaining generalization. A fourth convolutional layer with 128 filters and ReLU activation extracts high-level semantic representations relevant for classification. This is succeeded by another dropout layer with a rate of 0.3. The resulting feature maps are flattened into a one-dimensional vector, which is then passed into two fully connected (dense) layers. The first dense layer contains 128 neurons activated by ReLU, followed by a dropout layer (0.3) to further mitigate overfitting. The second dense layer comprises 64 neurons with ReLU activation, further refining the learned features. The final output layer consists of 16 neurons, each representing a unique class, and uses the SoftMax activation function to produce a probability distribution over the predicted classes.

This CNN architecture effectively combines convolutional layers for hierarchical feature extraction, dropout layers for regularization, and dense layers for classification, making it well suited for multi-class classification tasks involving hand-drawn images. The use of SoftMax in the output layer ensures that the network provides interpretable probabilistic outputs. However, it is important to note that the current output layer is configured for 16 classes; for full alignment with the 25-class dataset, the output layer should be extended accordingly. Overall, the model is compact yet capable, balancing computational efficiency with high classification performance-making it ideal for integration into real-time educational or simulation tools for electronic circuit design.

Results

Research Paper	Model Used	Training Accuracy	Test Accuracy
Hand-Drawn Electric Circuit Diagrams Recognition Using Deep Learning [2]	Faster R-CNN	-	75.9% (MAP)
From Image to Simulation: An ANN-Based Automatic Circuit Netlist Generator (Img2Sim) [12]	Artificial Neural Network (ANN)	-	98%
Hand-Drawn Electrical Circuit Recognition Using Object Detection and Node Recognition [13]	YOLOv5 , HOG + SVM, ANN, HMM	-	98.20%
Circuit Component Detection in Offline Handdrawn Electrical/Electronic Circuit Diagram [14]	Morphological 'Close' Operator, RLSA	-	91.28%
Segmentation and Recognition of Electronic Components in Hand-Drawn Circuit Diagrams [15]	HOG + SVM	-	92%
Proposed Methodology	**Multi-Layer CNN Model**	**99%**	**96%**

TABLE 1: Comparison of results with existing literature

R-CNN, Region-based Convolutional Neural Network; HOG, Histogram of Oriented Gradients; SVM, Support Vector Machine; HMM, Hidden Markov Model; RLSA, Run Length Smoothing Algorithm; YOLOv5, You Only Look Once version 5; MAP, Mean Average Precision

Table *1* presents a comparison of the proposed methodology's results with previous studies on hand-drawn circuit recognition. Various machine learning models, including Faster R-CNN, ANN, YOLOv5 with HOG and SVM, and morphological operators, were used in prior works, achieving test accuracies ranging from 75.9% (mean average precision) to 98.2% for different tasks such as component detection, simulation generation, and node recognition. While CNN-based approaches showed strong performance, with test accuracies of up to 94%, HOG+SVM achieved 92%, the proposed custom deep learning model outperformed these with a training accuracy of 99% and a test accuracy of 96%. This highlights its effectiveness in detecting hand-drawn electronic components and generating simulations compared to the existing literature.

Model training and testing

This model is designed to classify 25 different components, with 200 images of each component, resulting in 4,800 training samples. The input images, initially sized at 120 × 120 pixels, undergo preprocessing steps including color inversion, rotation augmentation (three times by 90 degrees), and cropping to 60 × 60 pixels to enhance feature diversity and reduce dimensionality. The training is conducted over 100 epochs, optimizing the model’s parameters for robust feature extraction and classification.

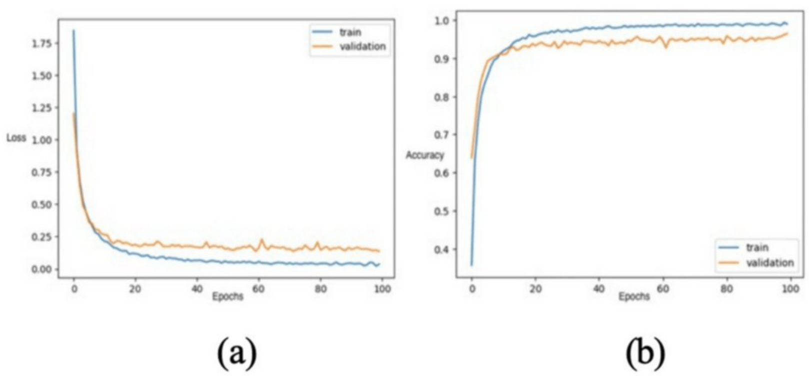


FIGURE 2: (a) Epoch versus loss (b) epochs versus accuracy

The training and testing results of the CNN model highlight its effectiveness in identifying electronic components from hand-drawn circuit diagrams. The model was trained for 100 epochs, a sufficient number to allow the network to learn complex patterns without overfitting. Figure *2* illustrates the model’s performance, where Figure *2*(a) presents the epoch versus loss curve, showing a consistent decline in both training and validation loss, indicating effective learning. Similarly, Figure 2(b) displays the epoch versus accuracy graph, where the training accuracy steadily increases, converging near 99%, while the validation accuracy stabilizes around 96%, demonstrating the model's ability to generalize well to unseen data.

The precision of the model, which measures the proportion of correctly identified components among all predicted components, was recorded at 0.963, indicating high reliability in its predictions. Similarly, the recall, which reflects the model's ability to identify all relevant components from the input data, was slightly lower at 0.961, but still demonstrates robust performance.

The accuracy metrics further validate the model's capability, with an impressive 99% accuracy on the training set. This indicates that the model has learned the training data well, successfully capturing the essential features of the circuit components. On the test set, the model achieved 96% accuracy, showcasing its ability to generalize to new, unseen data. These results affirm the CNN's ability to handle the variability and intricacies of hand-drawn circuit elements, making it a reliable tool for automating circuit recognition and simulation tasks.

The model demonstrates excellent performance metrics, achieving a training accuracy of 99% and a test accuracy of 96%, indicating strong generalization to unseen data. Precision and recall scores of 0.963 and 0.961, respectively, reflect the model's effectiveness in correctly identifying positive samples and minimizing false negatives. This high level of accuracy and consistency suggests the model is well suited for the classification task, effectively leveraging the pre-processed dataset and augmented features.

Flask model deployment

This Flask-based web application is designed to process image uploads and generate LTspice netlist files based on detected electronic components in the images. Figure 3 illustrates the model deployment and output, showcasing the user interface where an image of a hand-drawn circuit is uploaded. The system processes the image to identify electronic components such as resistors and inductors, allowing users to input values before generating the corresponding netlist. The detected circuit is then converted into an LTspice-compatible schematic, which users can download for further simulation.

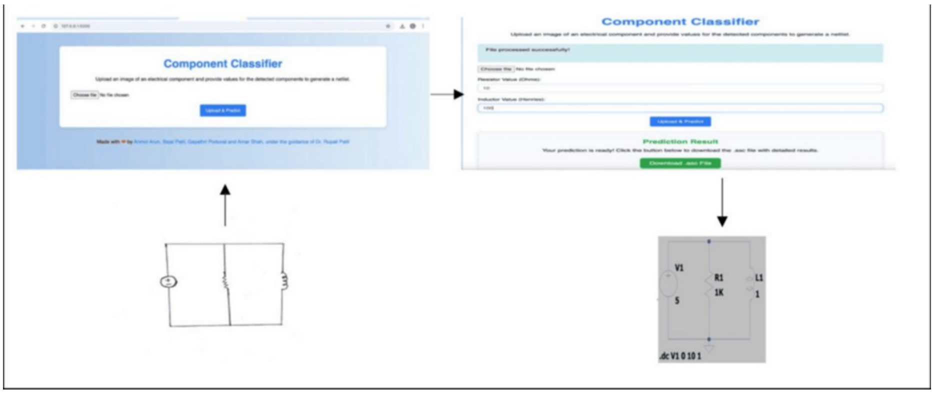


FIGURE 3: Model deployment and output

The overall backend and frontend integration of the application is depicted in Figure 4. The workflow begins with a user uploading an image through the Flask-based web interface. The server validates the file and processes the image to extract components using a pre-trained deep learning model. The model identifies bounding boxes, component classes, and their associated values. These detections are mapped to predefined circuit elements based on probability thresholds.

Once component detection is complete, the system generates an LTspice netlist, a textual representation of the circuit. The netlist follows SPICE syntax, assigning unique identifiers to circuit elements such as resistors, inductors, capacitors, and voltage sources. This generated file is then saved and made available for user download. Error handling mechanisms and flash messages ensure a seamless user experience.

By integrating image processing, machine learning, and circuit simulation, this application automates the conversion of hand-drawn circuits into digital formats, significantly simplifying the process for engineers and students working with circuit design and simulation.

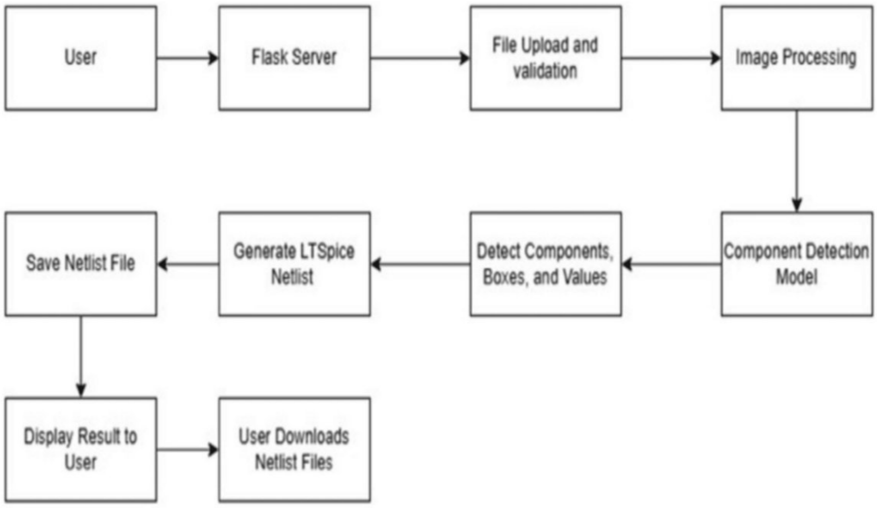


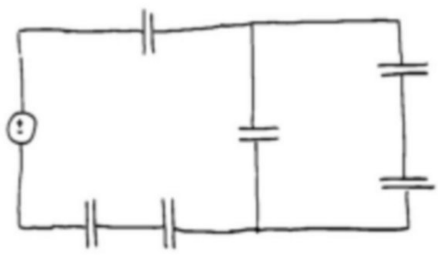
FIGURE 4: Proposed backend and frontend integration

Experimental setup

To develop and validate the proposed automated simulation system for hand-drawn circuit diagrams, a custom dataset was curated and expanded from an existing Kaggle source. The initial dataset comprising 12 component types was enhanced by incorporating six additional electronic components - switches, OP-AMPs, BJTs, MOSFETs, JFETs, and wires - resulting in a total of 25 distinct classes. Each class included 200 images, standardized to 120 × 120 pixels in .bmp format, culminating in 4,800 training samples. Preprocessing techniques such as grayscale inversion, cropping to 60 × 60 pixels, and data augmentation through rotation were applied to improve robustness and model generalization across diverse hand-drawn styles and orientations.

The system was trained using a custom-designed CNN implemented in TensorFlow. The architecture consisted of four convolutional layers with progressively increasing filter sizes, each followed by dropout and max-pooling layers to prevent overfitting and reduce dimensionality. The final classification was performed through two fully connected dense layers and a softmax output layer catering to 16 output classes, optimized using the Adam optimizer. The model was trained over 100 epochs and evaluated using metrics such as accuracy, precision, and recall, yielding 99% training accuracy and 96% test accuracy. The trained model was then deployed within a Flask-based web application that accepted user-uploaded images, processed them for component recognition, and generated corresponding LTspice-compatible netlists for circuit simulation.

Discussion



V1 N001 0 DC 10

C1 N002 0

C2 N002 N003

.end

FIGURE 5: Identified inefficiencies

As depicted in Figure 5, the issues identified in the paper includes challenges in detecting multiple capacitors and current sources in hand-drawn circuits. For capacitors, the model often fails to detect all or any of them, as seen in examples where only two out of six capacitors were identified in one case. This issue could stem from overlapping components, ambiguities in hand-drawing styles, or insufficient training data. Similarly, the model failed to detect a current source in a circuit, potentially due to underrepresentation in the dataset, visual similarities with other symbols, or limited contextual understanding. These detection inaccuracies result in incomplete or erroneous LTspice Netlists, hindering simulation functionality. Broader challenges include the need for the algorithm to generalize across diverse hand-drawing styles, accurately identify component connections, and scale to detect 25+ components without overfitting. Additionally, even when components are correctly detected, errors in mapping to valid LTspice Netlist formats can impact usability, underscoring the need for robust data and model improvements.

Using this specific CNN models when compared to using pre-trained models is because the size of the images used is larger, and this model has obtained the highest accuracy compared to others.

Conclusions

This paper presents a key innovation in the form of a highly user-friendly web application that enables users to effortlessly upload their hand-drawn circuit diagrams and quickly obtain LTspice-compatible .asc files. This level of automation and ease of use represent a significant advancement in electronic design automation, surpassing previous solutions. By automating and simplifying circuit simulation, the system empowers educators and students, enriching learning experiences and providing a solid foundation for future advancements in electronic design automation.

The development process involved creating a custom dataset and exploring optimal methods for electronic component detection. We developed a custom CNN model for this task and seamlessly integrated the backend deep learning model with a front-end web interface using Flask, resulting in an intuitive platform for circuit simulation. The system accurately recognizes over 25 hand-drawn electronic components and generates simulation-ready netlists compatible with LTspice. Achieving precision and recall scores of 96.3% and 96.1%, along with 99% training accuracy and 96% test accuracy, the system demonstrates robust and reliable performance across diverse scenarios.

Additional Information

Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

Concept and design: Anmol Arun, Sejal S. Patil, Gayathri M. Poduval, Amar Shah, Rupali P. Patil

Acquisition, analysis, or interpretation of data: Anmol Arun, Sejal S. Patil, Gayathri M. Poduval, Amar Shah, Rupali P. Patil

Drafting of the manuscript: Anmol Arun, Sejal S. Patil, Gayathri M. Poduval, Amar Shah, Rupali P. Patil

Critical review of the manuscript for important intellectual content: Anmol Arun, Sejal S. Patil, Gayathri M. Poduval, Amar Shah, Rupali P. Patil

Supervision: Rupali P. Patil

Disclosures

Human subjects: All authors have confirmed that this study did not involve human participants or tissue. **Animal subjects:** All authors have confirmed that this study did not involve animal subjects or tissue. **Conflicts of interest:** In compliance with the ICMJE uniform disclosure form, all authors declare the following: **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work. **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work. **Other relationships:** All authors have declared that there are no other relationships or activities that could appear to have influenced the submitted work.

References

1. Günay M, Köseoğlu M: Detection of circuit components on hand-drawn circuit images by using faster R-CNN method. International Advanced Researches and Engineering Journal. 2021, 5:372-378. 10.35860/iarej.903288
2. AlMughrabi A, Hiary H: Hand-drawn electric circuit diagrams recognition using deep learning. 2024 28th International Conference on Information Technology (IT), Zabljak, Montenegro. 2024, 1-4. 10.1109/IT61232.2024.10475731
3. Pavithra S, Shreyashwini NK, Bhavana HS, Nikhitha G, Kavitha T: Hand-drawn electronic component recognition using ORB. Procedia Computer Science. 2023, 218:504-513. 10.1016/j.procs.2023.01.032

4. Mohan A, Mohan A, Indushree B, Malavikaa M, Narendra CP: Generation of netlist from a hand drawn circuit through image processing and machine learning. 2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP), Vijayawada, India. 2022, 1-4. 10.1109/AISP53593.2022.9760577
5. Lakshman Naika R, Dinesh R, Prabhanjan S: Handwritten electric circuit diagram recognition: An approach based on finite state machine. International Journal of Machine Learning and Computing. 2019, 9:374-380.
6. Ren S, He K, Girshick R, Sun J: Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017, 39:1137-1149. 10.1109/TPAMI.2016.2577031
7. Zitnick CL, Dollár P: Edge boxes: Locating object proposals from edges. Computer Vision - ECCV 2014. Fleet D, Pajdla T, Schiele B, Tuytelaars T (ed): Springer, Cham; 2014. 8693:391-405. 10.1007/978-3-319-10602-1_26
8. Girshick R, Donahue J, Darrell T, Malik J: Rich feature hierarchies for accurate object detection and semantic segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA. 2014, 580-587. 10.1109/CVPR.2014.81
9. Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y: OverFeat: Integrated recognition, localization and detection using convolutional networks. 2014, 10.48550/arXiv.1312.6229
10. Lenc K, Vedaldi A: R-CNN minus R. 2015, 10.48550/arXiv.1506.06981
11. Hand-Drawn Electric Circuit Schematic Components. Kaggle. (2022). https://www.kaggle.com/datasets/moodrammer/handdrawn-circuit-schematic-components.
12. Sertdemir AE, Besenk M, Dalyan T, Gokdel YD, Afacan E: From image to simulation: an ANN-based automatic circuit netlist generator (Img2Sim). 2022 18th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), Villasimius, Italy, 2022. 2022, 1-4. 10.1109/SMACD55068.2022.9816254
13. Rachala RR, Panicker MR: Hand-drawn electrical circuit recognition using object detection and node recognition. SN Computer Science. 2022, 3:244. 10.1007/s42979-022-01159-0
14. Bhattacharya A, Roy S, Sarkar N, Malakar S, Sarkar R: Circuit component detection in offline handdrawn electrical/electronic circuit diagram. 2020 IEEE Calcutta Conference (CALCON), Kolkata, India. 2020, 80-84. 10.1109/CALCON49167.2020.9106527
15. Moetesum M, Younus SW, Warsi MA, Siddiqi I: Segmentation and recognition of electronic components in hand-drawn circuit diagrams. EAI Endorsed Transactions on Scalable Information Systems. 2018, 5:e12.