Designing a Sketch Based Interface for Electronic Circuit Simulation

Student: Taharka Okai, zp19374@bristol.ac.uk, **Supervisor:** Dr. Fadi Karameh, fadi.karameh@bristol.ac.uk

University of Bristol, Dept. of Engineering

1. Project Background

Circuit simulation is a well-known and useful computer-aided design (CAD) tool for designing electronic circuits. However, ease of use and speed of prototyping are two major drawbacks of circuit simulation using traditional CAD tools.

Additionally, there is an increasing trend in the use of machine learning tools to aid research and development [3]. Using recent advances in computer vision research and machine learning to develop a tool that can quickly and accurately interpret hand-drawn diagrams.

2. Project Outcome

This project aims to produce a software tool that receives hand-drawn circuit diagrams as input and generates an LT-SPICE simulation file as output, which will be paired with a compatible backend to produce a fully functional circuit simulation tool. The user will be able to:

- Draw a circuit diagram on paper/whiteboard/tablet/digital image
- Pass the sketch to the application
- Simulate/download the simulation file on their device

3. Existing Solutions

There are a number of existing solutions that are similar to the project outcome. These methods each have their own merits and drawbacks, as detailed below:

- **Programmatic**: [?]: use image processing and properties of the shapes of the circuit components to identify them.
- **Support Vector Machines**: [?]: use a Support Vector Machine (SVM) to classify the components of the circuit.
- **Object Detection**: [4, 2]: use deep learning to classify the components of the circuit.

These approaches [?] require intimate knowledge of the geometry of the components and the image processing techniques required to extract the relevant information from the image.

Additionally, only a few of these solutions [?] go on to produce the file required for simulation, and less still [?] produce a fully functional simulation tool.

Figure 1. Creating a digital image of the circuit diagram

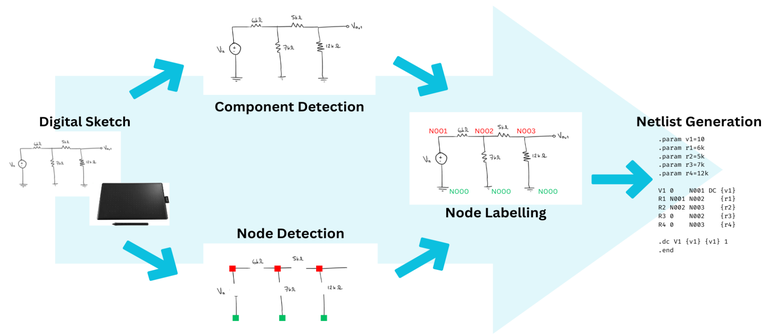


Figure 2. Creating a representation of the circuit diagram in software

4. Methodology

Firstly, a model was trained to create a representation of the circuit:

- **Data collection** – labelled images of circuit diagrams
- **Preprocessing** – Contrast Limited Adaptive Histogram Equalisation (CLAHE)
- **Training and Validation** – 80/10/10 train/test/validation split

Then, the hand-drawn circuits go through the following steps to produce a simulation file:

- **Component/Node Detection** – YOLOv8 bounding box and Hough Transform used to detect components and nodes
- **Node Labelling** – Graph representation of the circuit is generated
- **Simulation** The simulation file is passed to a Python wrapper around LT-SPICE to produce a simulation

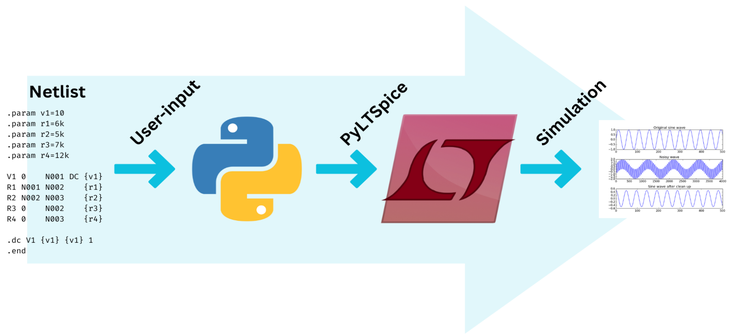
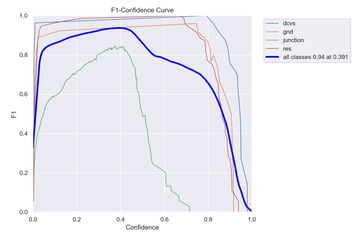
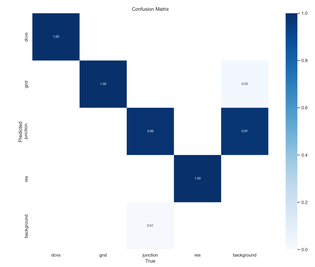


Figure 3. Simulating the digital representation of the circuit diagram



(a) F1 Score



(b) Confusion Matrix

Figure 4. Component detection accuracy as F1 Score and confusion matrix. Demonstration available at QR code at the end of this document.

5. Findings

Using deep learning techniques in the form of the latest advancements in object detection algorithms, YOLOv8 [?], a high F1-score was achieved for all classes except junctions (figure 4a, green line).

A welcome discovery was made where the detector can function directly on CAD images. This is a promising result, as it means that the detector can be trained on more complex schematics that are CAD generated in the future.

An improvement that was found is that junction classes can be replaced by node detection which is a more reliable means of detecting the intersection of wires.

6. Ongoing and Future Work

- **Ongoing Work:**
  - Interpret the circuit diagram and produce a simulation file.
  - Run the simulation using a Python interface to LT-SPICE.
  - Design a user interface for the application to allow the simulation to be displayed on the user's device.
- **Future Work:**
  - Increase the set of detectable components by training the detector on schematics including analogue voltage sources, capacitors, inductors, diodes, transistors, and operational amplifiers.

7. References

[1] A. Author, B. Author, and C. Author. Title of the paper. *Some Journal*, V(X), 2019.

[2] Mrityunjoy Dey, Shoif Md Mia, Navonil Sarkar, Archan Bhattacharya, Soham Roy, Samir Malakar, and Ram Sarkar. A two-stage cnn-based hand-drawn electrical and electronic circuit component recognition system. *Neural Computing and Applications*, 33(20):13367–13390, Oct 2021.

[3] Carlos Francisco Moreno-García, Eyad Elyan, and Chrisina Jayne. New trends on digitisation of complex engineering drawings. *Neural Computing and Applications*, 31(6):1695–1712, Jun 2019.

[4] Rohith Reddy Rachala and Mahesh Raveendranatha Panicker. Hand-drawn electrical circuit recognition using object detection and node recognition. *SN Computer Science*, 3(3):244, Apr 2022.

University of BRISTOL