# KANTIPUR ENGINEERING COLLEGE

**(Affiliated to Tribhuvan University)**

**Dhapakhel, Lalitpur**

**[Subject Code: CT707 ]**

**A MAJOR PROJECT MID-TERM REPORT ON**

# HAND DRAWN CIRCUIT TO DIGITAL CIRCUIT


**Submitted by:**

| | |
|---|---|
| **Aayush Shrestha** | **[KAN076BCT004]** |
| **Anuj Shakya** | **[KAN076BCT014]** |
| **Jasmine Bajracharya** | **[KAN076BCT033]** |
| **Ojashwi Neupane** | **[KAN076BCT051]** |


**A MAJOR PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR IN COMPUTER ENGINEERING**

**Submitted to:**

**Department of Computer and Electronics Engineering**

**August, 2023**

# HAND DRAWN CIRCUIT TO DIGITAL CIRCUIT


**Submitted by:**

| | |
|---|---|
| **Aayush Shrestha** | **[KAN076BCT004]** |
| **Anuj Shakya** | **[KAN076BCT014]** |
| **Jasmine Bajracharya** | **[KAN076BCT033]** |
| **Ojashwi Neupane** | **[KAN076BCT051]** |


**A MAJOR PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR IN COMPUTER ENGINEERING**


**Submitted to:**

**Department of Computer and Electronics Engineering**

**Kantipur Engineering College**

**Dhapakhel, Lalitpur**



**August, 2023**

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

AC: Alternating Current

ANN: Artificial Neural Networks

API: Application Programming Interface

CAD: Computer Aided Design

DC: Direct Current

EDA: Electronic Design Automation

IC: Integrated Circuits

SPICE: Simulation Program with Integrated Circuit Emphasis

YOLO: You Only Look Once

# CHAPTER 1
# INTRODUCTION

## 1.1 Background

In today's technological era, computers have become indispensable in all branches of engineering, including Electrical and Electronics Engineering. They play a vital role in addressing issues such as circuit drawing, simulation, and coding. Despite the widespread use of computers, engineers and students in this field often begin circuit design by sketching the circuit diagram on paper. Therefore, manual sketches remain common and convenient in courses related to circuit design, circuit analysis, electronics, and other applied subjects. These hand-drawn circuit diagrams serve as a foundation for performing operations and calculations. Such diagrams can encompass a variety of circuit components, each serving different functions within the circuit. Based on the problem at hand, engineers select the appropriate components to design the circuit. Due to their familiarity with circuit components, electrical and electronic engineers can easily recognize them when examining a hand-drawn circuit diagram. However, the task of automatically recognizing and analyzing a hand-drawn circuit diagram, including its components, poses significant challenges for specialized software in the field of electrical and electronic engineering. For this purpose, there are some studies conducted for computers to recognize and classify hand-drawn electronic circuit diagrams and electronic circuit components.

## 1.2 Problem Statement

Hand-drawn circuit layouts can be difficult to read, prone to mistakes, and time-consuming to convert into correct diagrams that fully express design intent. It can be difficult to precisely reproduce hand-drawn circuits, which might result in errors when the circuit is replicated for manufacturing or sharing with others.The automation and optimization methods available in digital circuit design are not applicable to hand-drawn circuits. Working with hand-drawn circuits makes it difficult to integrate automated error checking, design rule checking, and simulation-driven design methods, which could lead to design faults going undetected.The hand-drawn circuit to digital circuit (netlist)

application addresses key challenges faced by engineering students and researchers in the circuit design process. By automating the conversion of hand-drawn circuit diagrams into digital netlists, the application streamlines the design workflow, improves accuracy, and enhances productivity. It eliminates the need for manual circuit entry, reducing errors and inconsistencies.It empowers users to leverage circuit analysis and simulation tools, facilitating in-depth exploration and optimization of circuit designs. Moreover, the application supports scalability, reproducibility, and documentation, ensuring the seamless transition from design to manufacturing or research experiments. Hence, this tool revolutionizes the circuit design experience for engineering students and researchers, providing a time-saving, accurate, and collaborative solution for converting hand-drawn circuits into digital circuits.

## 1.3 Objectives

I To be capable of accurately recognizing various circuit components, including resistors, capacitors, transistors and 10 other components.

II To convert the hand drawn circuit to digital circuit.

## 1.4 Area of Application

1. The converted digital circuits from hand-drawn circuits can be imported into Electronic Design Automation(EDA) tools or circuit simulation software which makes it easier for simulation, testing and analysis of circuits.
2. As the circuit diagrams are widely used in electronics and electrical engineering concepts, by converting the hand-drawn circuits to digital circuits, interactive learning materials can be created by help of which can enhance understanding of working mechanisms.

## 1.5 Project Features

The following are the features of the project.

1. Identify and interpret symbols and components used in the hand-drawn circuit.

2. Ensure accurate representation and connection of wiring in the digital circuit.
3. Analyze the circuit topology and determine component arrangement and interconnections.
4. Replicate the intended signal flow through the hand-drawn circuit in the digital circuit through netlist generation.
5. Utilize schematic drawing software for creating the digital circuit.
6. Validate and test the digital circuit for accuracy and expected functionality.

## 1.6 Feasibility Study

Before implementation of project design, the feasibility analysis of the project must be done to move any further. The feasibility analysis of the project gives an idea on how the project with perform and it's impact in the real world scenario. So, it is of utmost importance.

### 1.6.1 Economic Feasibility

Our system is economically achievable as a result of the development of several tools, libraries, and frameworks. Since all the software required to construct it is free and readily available online, this project is incredibly cost-effective. Only time and effort are needed to create a worthwhile, genuinely passive system. The project doesn't come at a substantial cost. From an economic standpoint, the project appears successful in this sense.

### 1.6.2 Technical Feasibility

The software needed to implement a project can be downloaded from a wide variety of online resources. Technically speaking, the project is feasible as the necessary software is easily available. We were able to learn the information we required for the project through a variety of online sources, including classes.All the libraries and data are accessible online for free because this project does not require any licensing costs. It is technically possible if one has the necessary information and resources.

### 1.6.3 Operational Feasibility

Our project intends to digitize hand-drawn circuits with a primary focus on circuit simulation to verify the required output from the circuits. The project is intended for a sizable group of learners and students who desire to learn about circuits and their simulation. Consequently, our project is operationally feasible.

### 1.6.4 Schedule Feasibility

The workload of the project is divided amongst the project members. The scheduling is done according to the incremental model where different modules are planned to be assigned to different group members.So, the project fulfills the schedule feasibility requirements.

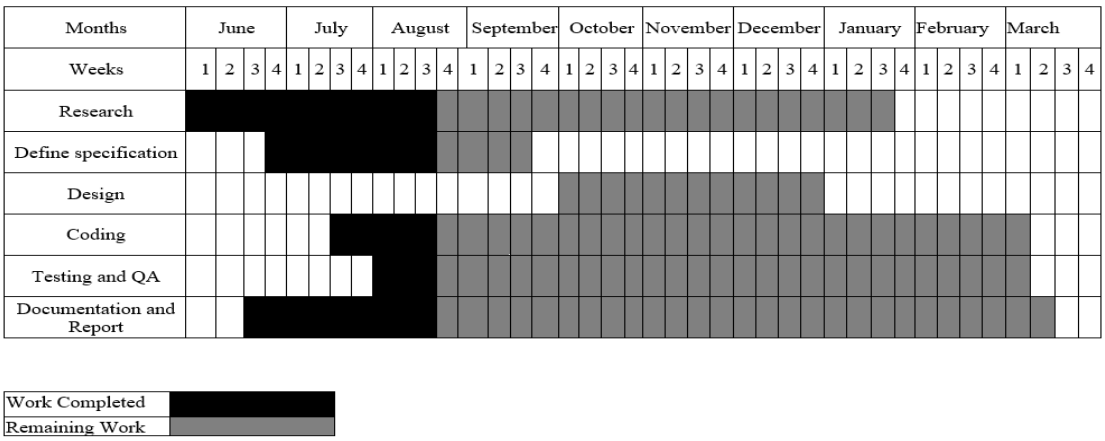


Figure 1.1: Gantt chart

## 1.7 System Requirements

### 1.7.1 Development Requirements

#### 1.7.1.1 Software Requirements

- Operating System : Windows 8 / Ubuntu/ macOS

- Browser : Any browser

#### 1.7.1.2 Hardware Requirements(Minimum)

- Processor : Dual core (x86/x64)
- RAM : 4 GB or more
- Screen Resolution: 1280*8000 screen resolution

### 1.7.2 Deployment Requirements

#### 1.7.2.1 Software Requirements

- Operating System : Windows 8 / Ubuntu/ macOS
- Browser: Any Browser

#### 1.7.2.2 Hardware Requirements

- Camera : 20 megapixels or more
- RAM:4 GB or more

# CHAPTER 2
# LITERATURE REVIEW

Regardless of whether they are documented or not, every project have helped to shape the world as it is today. Other researchers can benefit from documented projects by learning specifics about problems and how to solve them. Additionally, they boost project efficiency by removing the need to start the project from begin and specifying the starting point.

## 2.1 Related Projects

- EveryCircuit – Circuit Simulator App: EveryCircuit is a powerful circuit simulator app that offers real-time animations of voltage waveforms, current flows, and capacitor charges. It is a user-friendly circuit simulator app for beginners and experienced users.It allows interactive circuit experimentation, parameter adjustments, and features frequent component updates, smooth simulations, automated wire routing, and an analog control knob.
- ECStudio/ Electric Circuit Studio: ECStudio is a user-friendly circuit simulator app with a schematic editor, SPICE simulator, and support for DC, AC, Transient analyses. Wide range of components: resistors, capacitors, diodes, transistors, logic gates, op-amps. Real-time updates, adjustable parameters. Ideal for circuit design and experimentation.
- Smart Logic Simulator – A Circuit Simulator App: Logic Gates is a premier free circuit simulator for logic gate analysis and digital electronics learning. It offers simulations for various logic gates and electronic devices. With interactive game-like levels and a built-in circuit builder, users can enhance their understanding and skills in a fun and educational way.

## 2.2 Related Research

This paper introduces a computer vision framework that digitizes hand-drawn power converter circuit diagrams and performs automated simulations. The framework combines YOLOR, a deep learning model for component detection, with a Hough transform

algorithm for wire tracing and K-Means clustering for node identification. A netlist is generated using the detected components and nodes, enabling further analysis with spice-based circuit simulators. The framework has been successfully applied to popular non-isolated DC-DC converters. Challenges include variations in sketches, ink qualities, and inconsistent component representations. To address these challenges, a custom dataset with annotations and data augmentation techniques was created. Future work involves integrating deep learning-based OCR models for extracting component values and balancing the dataset for unbiased model performance. Thus, this framework offers an efficient and accurate solution for analyzing hand-drawn circuit diagrams, benefiting power electronics engineers and students.[1]

A new approach is proposed to address the issue of identifying electrical symbols in hand-drawn electrical diagrams. This method enables a machine to directly interpret complex circuit diagrams. The technique employs an Artificial Neural Network (ANN) to enable the machine to recognize electrical symbols in hand-drawn circuit images. The recognition process involves two steps: first, shape-based features are extracted, and then a classification procedure using ANN with a back propagation algorithm is utilized. The ANN successfully identifies and distinguishes the electrical characters present in the circuit image. Consequently, the machine can be provided with scanned circuit diagrams. The initial step involves converting the raw image into black and white pixels (binarization) and resizing them to a standard size (normalization). Next, feature extraction occurs by utilizing the image's moments. These extracted features are then used as input in the artificial neural network for training at each stage of feeding. The network calculates the error relative to the desired class through the back propagation algorithm.[2]

Usually, simulating circuits is done manually by creating circuit diagrams on circuit tools, which is a time consuming and tedious process. This paper addresses the need for efficient circuit simulation methods and improved engagement in engineering education. The system proposes the method of simulating circuits from hand-drawn diagrams using smartphones through an image recognition system. The primary advancement offered by this research is the creation of a circuit identification system that utilizes

deep learning capsule networks. The system undergoes preprocessing, region proposal, classification, and node detection to generate a Netlist, which is then fed into a circuit simulator for simulation. The goal of the system is to achieve higher accuracy using less training data with capsule networks and developing a comprehensive system that captures hand-drawn circuit diagrams and produces circuit simulation results. 400 samples per class is used and accuracy of 96 % for stratified 5-fold cross-validation is acquired. The paper discusses the evolution of circuit recognition techniques and highlights the advantages of capsule networks in reducing the training data requirement.[3]

This study presents a technique aimed at segmenting and recognizing electronic components in hand-drawn circuit diagrams. The process involves employing various methods for both segmentation and feature extraction. Initially, the hand-drawn circuit diagram undergoes preprocessing steps, such as converting it to grayscale, applying smoothing filters to eliminate noise, and binarizing it through thresholding. Segmentation is then carried out using morphological operations. Following the segmentation process, feature extraction is performed using the Histogram of Oriented Gradients (HOG) approach, which aids in categorizing the components. Prior to feature extraction, the components are resized to a standardized size of 256x256 pixels, resulting in a feature vector of 1764 dimensions for a block size of 32x32 pixels. Classification of the components is accomplished using the Support Vector Machine (SVM) algorithm. As a result, this paper introduces an effective method for segmenting and identifying electronic components in manually drawn circuit diagrams. The segmentation phase relies on a sequence of morphological operations, while the recognition stage employs HOG descriptors along with an SVM classifier. The reported findings reveal an accuracy of 87.75 % in segmenting 350 components from a dataset of 100 hand-drawn circuit diagrams, where 307 out of 350 images were correctly segmented. Furthermore, a classification rate of 92 % is achieved when identifying 150 query components. [4]

# CHAPTER 3
# METHODOLOGY

## 3.1 Working Mechanism

### 3.1.1 Block Diagram of Hand drawn to digital circuit system

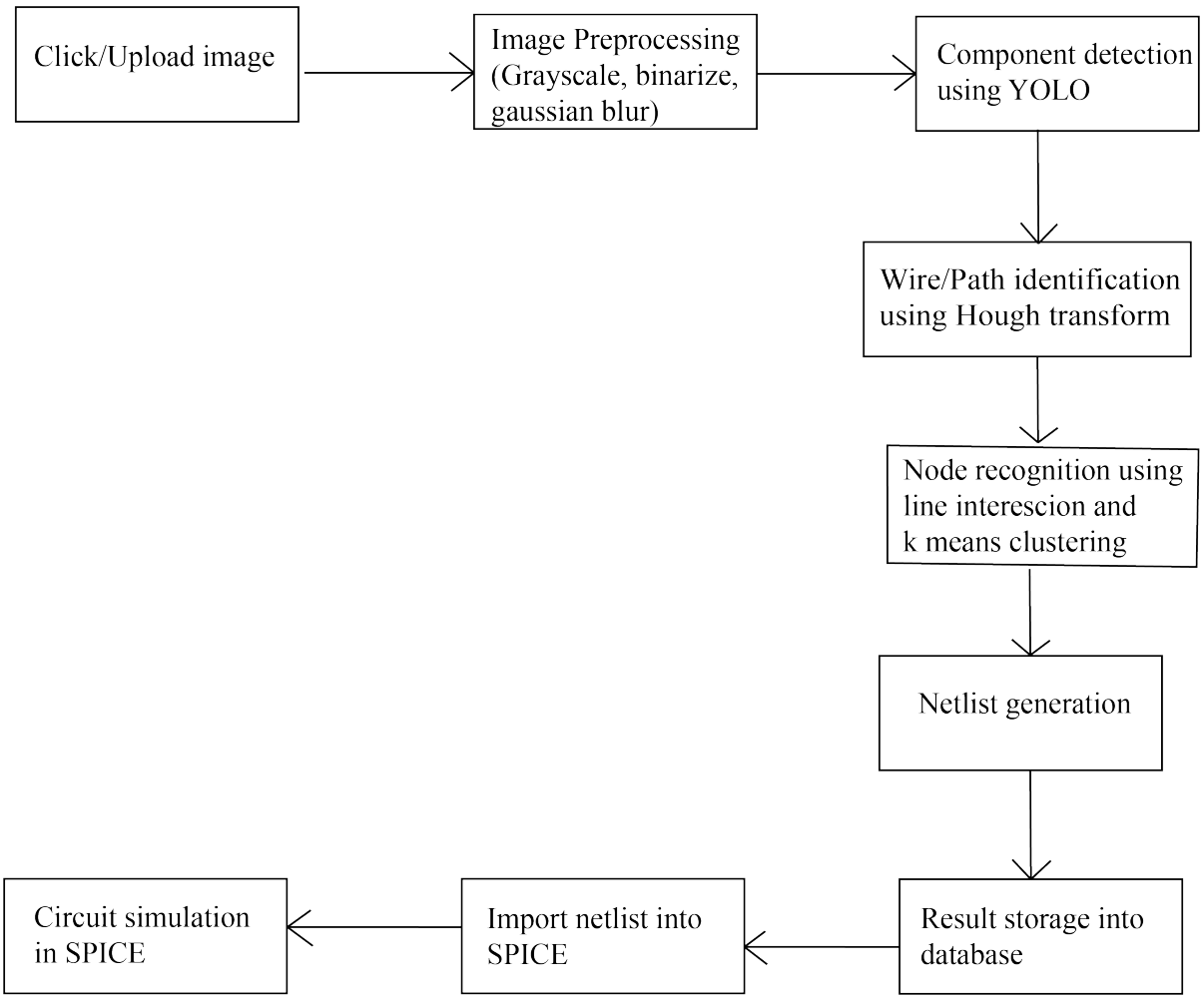


Figure 3.1: Block diagram Of Hand Drawn To Digital Circuit System

- Click/Upload image: The image of the desired circuit will be clicked and uploaded by the users directly.

- Image preprocessing: During image processing, the hand drawn circuit will be made ready for the conversion by changing the image inorder to make the extraction of features easy and more effective for a correct circuit recognition. The preprocessing of the image generally involves a number of steps like grayscale

conversion, binarization, gaussian blur for noise removal and smoothing of the image.

- Component Detection using YOLO: After the image preprocessing would have been successively completed, the components will be detected using the YOLO algorithm using the dataset.

- Wire/Path identification: The wires or path will be identified through Hough transform which will identify the paths connecting the components.

- Netlist generation: After the components, the path and the nodes will have been identified, a netlist for the circuit will be generated.

- Netlist Storage: The netlist generated will be then stored into the database for the retrieval when the circuit would have to be simulated.

- Netlist import: The netlist stored will be imported during the simulation of the circuit.

- Simulation: After the netlist would have been obtained through various steps followed, the netlist will be imported and the circuit will be simulated using an open source electronic circuit simulator SPICE.

### 3.1.2 Algorithms and methods

- YOLO: Object detection is a computer vision task that involves identifying and localizing objects within an image or video. One image can include several regions of interest pointing to different objects. YOLO, You Only Look Once is an object detection model which was introduced by Joseph Redmon and Ali Farhadi. This algorithm uses features learned by a Convolutional Neural Network (CNN) to detect an object. YOLO is a single-shot object detector which uses a single pass of the input image to make predictions about the presence and location of objects in the image at once. The YOLO algorithm divides the image into N

grids, each of which has an equal-sized SxS region. These N grids are each in-charge of finding and locating the thing they contain. As the same object might be predicted by several grids, multiple bounding boxes might be created for one object. YOLO makes use of Non Maximal Suppression to deal with this issue. The bounding boxes with the largest Intersection over Union (IoU) with the current high probability bounding box are then suppressed until the final bounding boxes are obtained, this phase is repeated.The dataset titled "Hand-drawn electric circuit schematic components" [5] is to be used for the training of the YOLO model and it consists of 13 classes which includes the images of resistors, capacitors, inductors and 10 other components. Each class contains 200 images each of every component with a resolution of 120*120 pixels.

- Hough's Transform: Hough Transform algorithm is a popular method used in image processing and computer vision for detecting geometric shapes, particularly lines and curves. It was developed as a technique for automatically detecting straight lines in images. Since then, it has been extended to detect other shapes as well. The Hough Transform works by representing lines or curves in a parameter space known as the Hough space. This space is defined by the parameters that characterize the shape, such as the slope and intercept of a line or the radius and center coordinates of a circle. The algorithm consists of the following steps:
  1. Edge Detection: The first step is typically to apply an edge detection algorithm, such as the Canny edge detector, to the input image. This helps in identifying the regions of the image where edges are present.
  2. Initialize the Hough Space: A Hough space, also known as the accumulator, is created. It is a 2D array where each cell corresponds to a specific line in the image space. The dimensions of the Hough space are determined by the range of possible values for the parameters of the lines to be detected.

     Equation for calculating ρ in a polar coordinate system:

     $$\rho = x * \cos\theta + y * \sin\theta \tag{3.1}$$

     where, $\rho$ is the shortest distance from the origin to the line, x and y are the coordinates of the edge pixel, and $\theta$ is the angle parameter.

     For each edge pixel $(x, y)$ in the edge image:

- Loop through all possible $\theta$ values.
- Calculate $\rho$ using the equation used in 3.1
- Increment the corresponding cell in the Hough accumulator.

And the range of $\rho$ $(rho)$ and $\theta$ $(theta)$ values, typically $[0, 180]$ degrees for $\theta$ and $[-d, d]$, where d is the width of the image's diagonal, for $\rho$.

3. Line Extraction: Set a threshold value to determine the minimum number of points required to estimate a line. Iterate over all cells in the Hough accumulator. If a cell value exceeds the threshold:
   - Retrieve the corresponding $\rho$ and $\theta$ values.
   - Transform the $\theta$ and $\rho$ values into the form

$$y = mx + c \tag{3.2}$$

     where m is the slope and c is the intercept.Equation for calculating the slope of a line:

$$m = \frac{y2 - y1}{x2 - x1} \tag{3.3}$$

     where $(x1, y1)$ and $(x2, y2)$ are two points on the line.
   - Classify the line segment as horizontal if the slope $(m)$ is less than or equal to deg(0)+$\epsilon$.
   - Classify the line segment as vertical if the slope $(m)$ is greater than or equal to 1 for $-$ $\epsilon$ or m $= \infty$.Where $\epsilon$ is a user-defined limit that specifies the degree of alignment and ranges is $\epsilon \in 0, 5$.
4. Output: Obtain the detected line segments, separated into horizontal and vertical lines based on the slope.

- Node Recognition:
  - The circuit's nodes are identified through the intersection points of the separated vertical and horizontal lines. The intersection point $(x, y)$ between two line segments, L1:

$$ax + by + c = 0 \tag{3.4}$$

    and L2:

$$a'x + b'y = c' \tag{3.5}$$

can be computed using:

$$x = \frac{b1c2 - b2c1}{a1b2 - a2b1} \tag{3.6}$$

and

$$y = \frac{c1a2 - c2a1}{a1b2 - a2b1} \tag{3.7}$$

Here, a = y1 − y2, b = x2 − x1, c = x1y2 − x2y1 represent the parameters of line L1. Similarly, a' = y1' − y2', b' = x'2 − x'1, c' = x'1y2' − x'2y1' represent the parameters of line L2. The points $(x1, y1)$ and $(x2, y2)$ lie on line L1, while $(x'1, y1')$ and $(x'2, y2')$ lie on line L2.

However, a large number of horizontal and vertical lines can be detected, resulting in numerous points of intersection and an excessive number of nodes. To group the closer points of intersection based on their distance, a K-Means clustering method is used.

- Use K-means clustering to identify the node.
- Store the coordinates of point of nodes

- Netlisting: An electronic circuit's connections are described in a netlist, which includes a list of components and the nodes they are connected to. A netlist can be physical, logical, instance-based, or net-based, representing different types of connections. The netlist also specifies the attributes and device types used. Networks, or "wires," link components together in a circuit. Depending on the language and its features, there may be special qualities associated with the nets. In larger designs, concepts like hierarchy, unfolding, backannotation, and inheritance are employed, with each design piece serving as a definition.

  In general, an element is declared with the following general syntax:
  $\langle K \rangle$ $\langle descriptionstring \rangle$ $\langle n1 \rangle$ $\langle n2 \rangle$ [value] [$\langle option \rangle$= $\langle value \rangle$] [...]...
  where:

  - $\langle K \rangle$ is a character, a unique identifier for each type of element (e.g. R for resistor).
  - $\langle descriptionstring \rangle$ is a string without spaces (e.g. 1).
  - $\langle n1 \rangle$, a string, is the node of the circuit to which the anode of the element is connected.

- $\langle n2 \rangle$, a string, is the node of the circuit to which the cathode of the element is connected.
- $[value]$ if supported, is the 'value' of the element, in mks (e.g. R1 1 0 500k)
- $\langle option \rangle$= $\langle value \rangle$ are the parameters of the element.

For example: The general syntax is given by : R$\langle string \rangle$ n1 n2 $\langle value \rangle$


Figure 3.2: Simple circuit diagram for representation of netlist

- n1 and n2 are the element nodes.
- value is the element resistance. It may any non-zero value (negative values are supported too).

The netlist for above circuit is represented as: R1 1 0 1k

- SPICE : A general-purpose, free and open-source analog electrical circuit simulator is called SPICE, which stands for Simulation Program with Integrated Circuit Emphasis. It is a program used to verify the accuracy of circuit designs and forecast circuit behavior in integrated circuit and board-level design. Before committing to the production of an integrated circuit, the industry-standard method to ensure circuit functionality at the transistor level is to simulate the circuit with SPICE. The SPICE simulators aid in predicting how the IC will behave under various operating circumstances, such as varying voltage and current levels, varying temperatures, and noise..The most well-known circuit simulation software, SPICE, converts a text netlist defining the circuit's components (transistors, resistors, capacitors, etc.) and their connections into equations that must be solved. To successfully simulate multiple circuits, SPICE integrates operating point solutions, transient analysis, and various small-signal analyses with the circuit components and device models required.Analyses such as AC, DC, DC transfer curve, noise, transfer function, and transient analysis are all included in SPICE. When given a text netlist as input, SPICE outputs line-printer listings.

## 3.2 UML Diagrams

### 3.2.1 Use Case Diagram

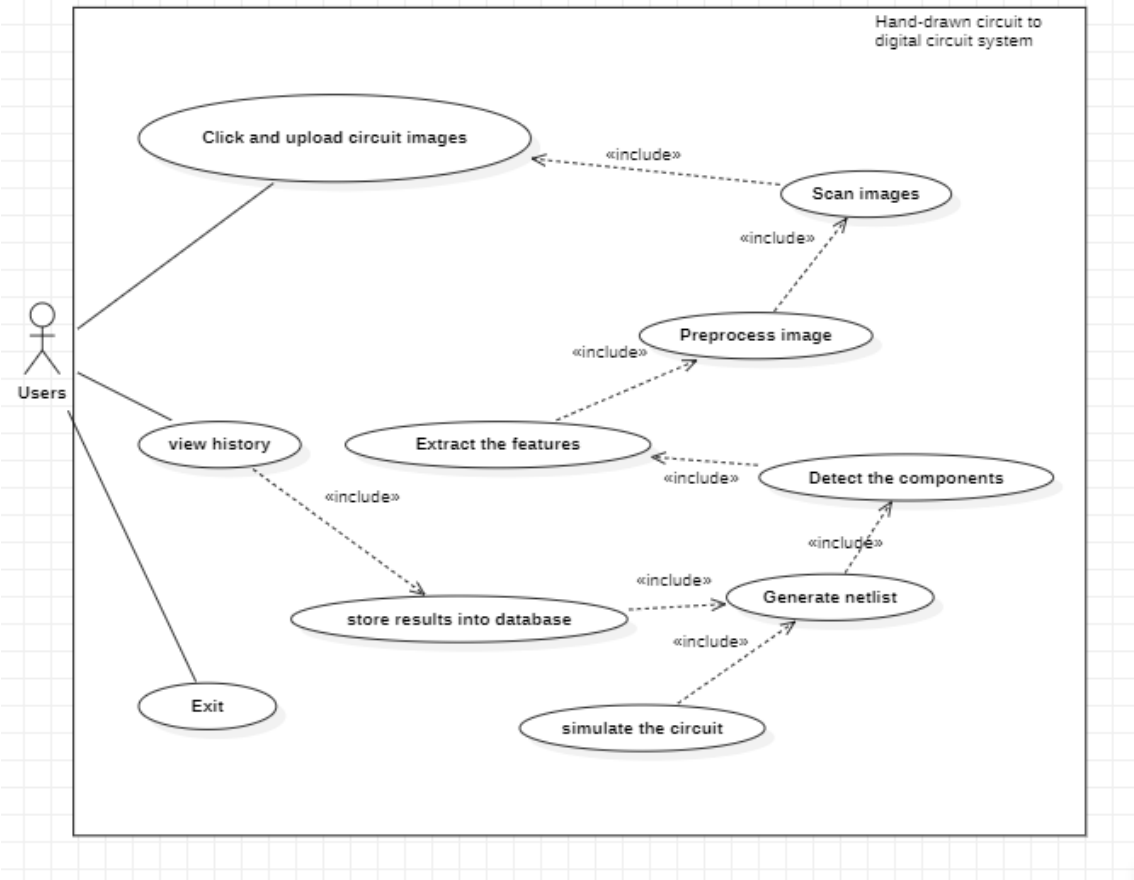


Figure 3.3: Use Case Diagram Of Hand Drawn To Digital Circuit System

### 3.2.2 DFD level 0 Diagram

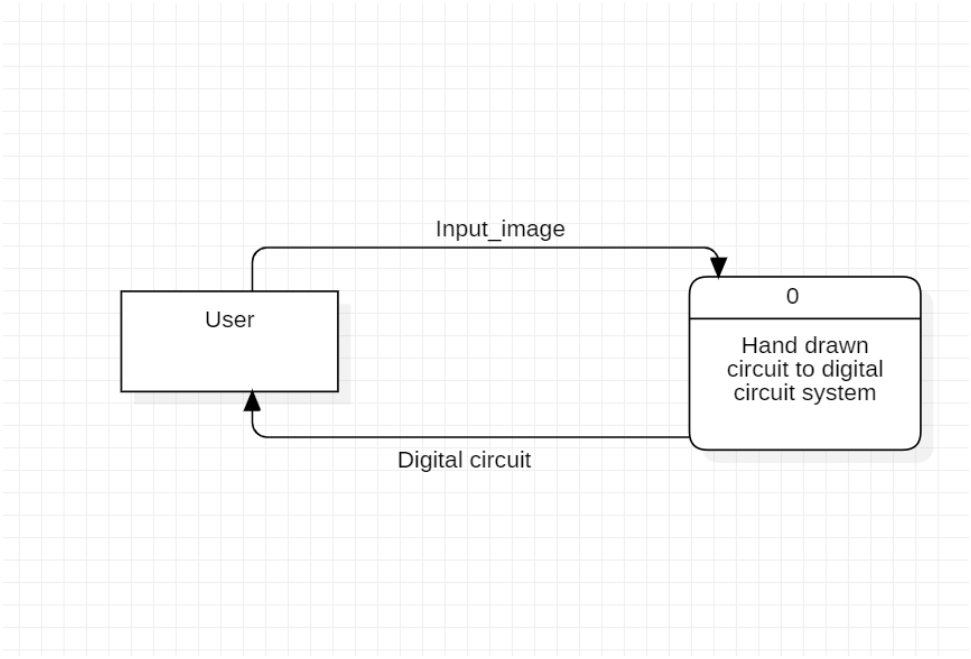


Figure 3.4: DFD level 0 diagram Of Hand Drawn To Digital Circuit System

### 3.2.3 DFD level 1 Diagram

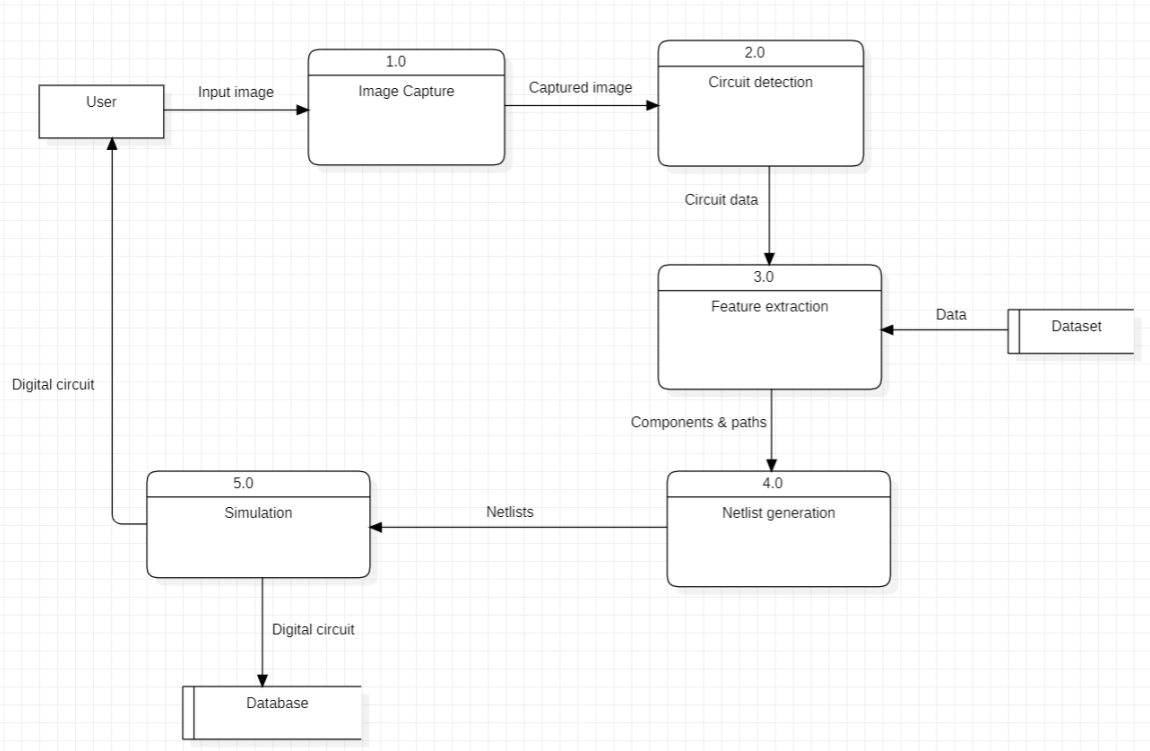


Figure 3.5: DFD level 1 diagram Of Hand Drawn To Digital Circuit System

### 3.2.4 Activity Diagram

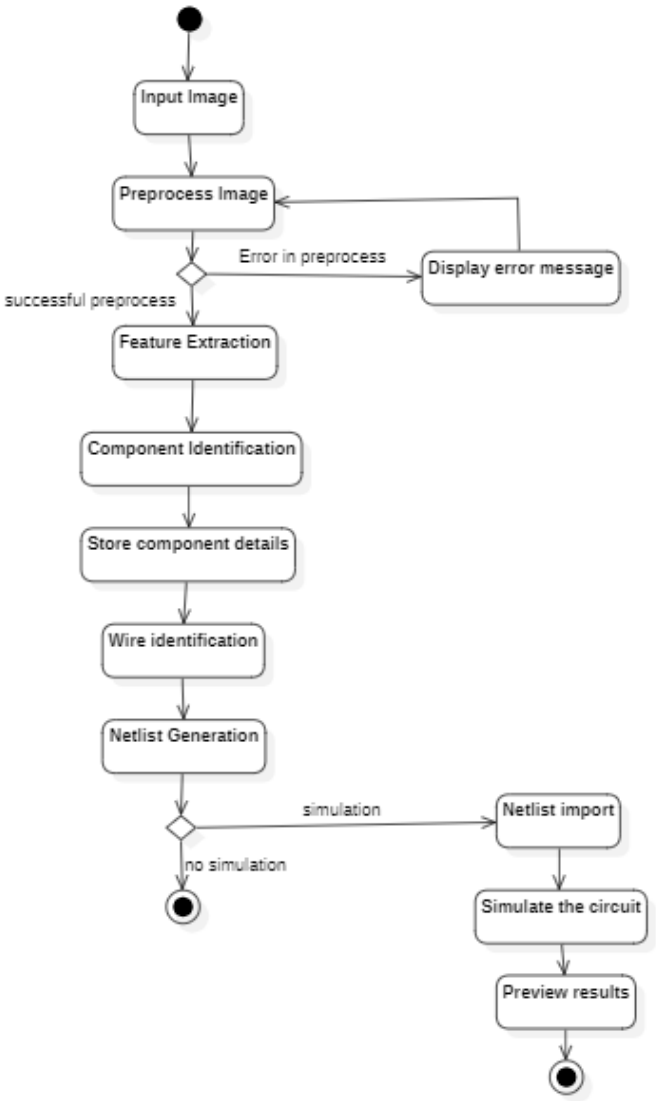


Figure 3.6: Activity Diagram of Hand Drawn To Digital Circuit System

## 3.3 Software Development Model

This project will be developed using an incremental methodology since it offers a functioning prototype at an early stage of development. The requirements and scope of the project can be altered as necessary by studying the prototype. The rationale for the preference for this software development strategy is the flexibility offered by adopting the incremental technique. In this paradigm, the project goes through several releases or iterations prior to its official release.

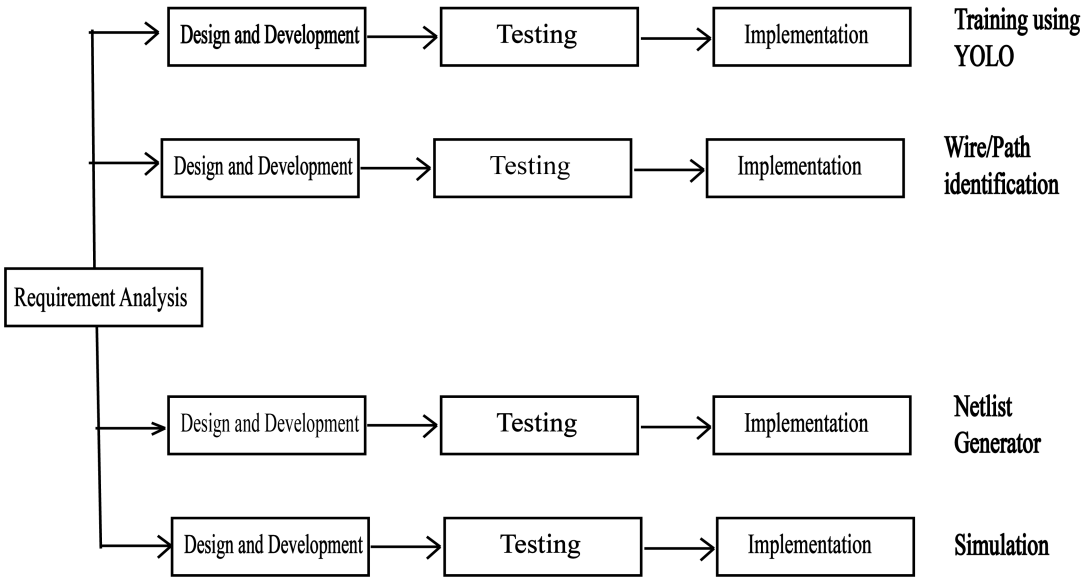


Figure 3.7: Software Model for the project(Incremental model)

# CHAPTER 4
# EPILOGUE

## 4.1 Works completed

Various papers regarding the conversion of the hand drawn circuit to the digital circuits have been studied. Studying the papers, the Hough Transform for the line detection and the node detection through the k-means clustering in the circuit diagrams has been implemented.

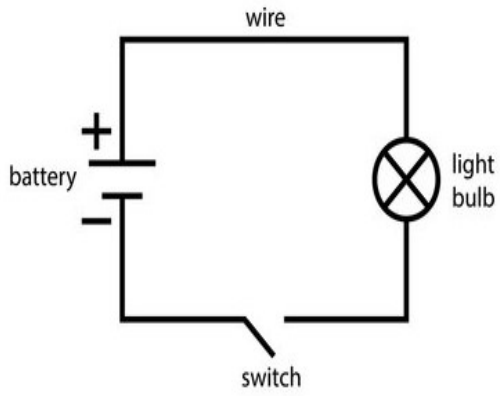


Figure 4.1: Input for line and node detection

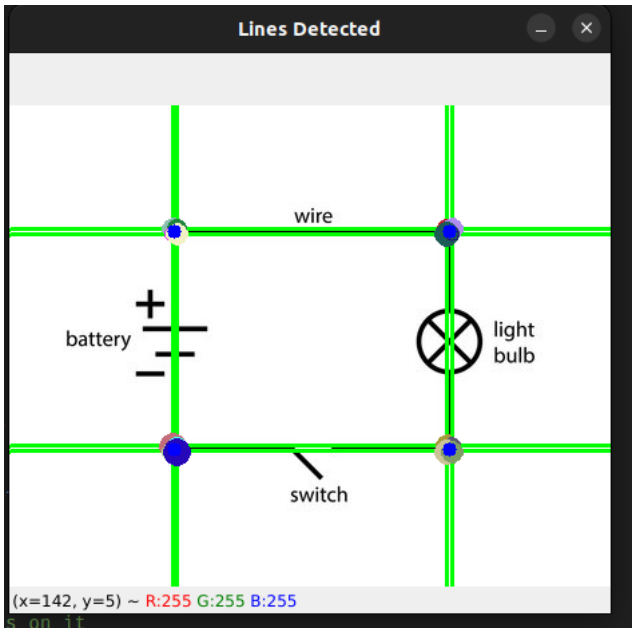


Figure 4.2: Output from the Hough Transform and node detection(Implementation)

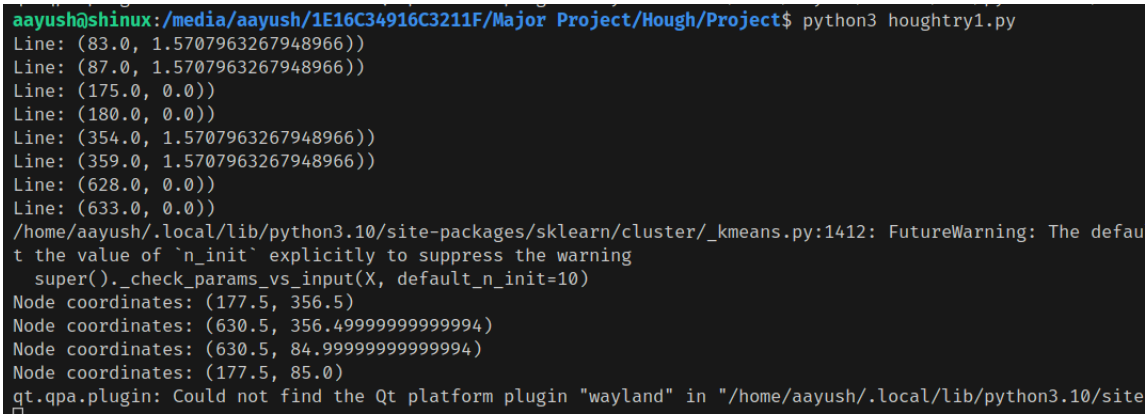


Figure 4.3: Output coordinates from the Hough Transform

## 4.2 Works Remaining

So far some of the research and the implementation of Hough transform has been completed. The YOLO training for the circuit components is ongoing at present. The efforts to improve the accuracy in the component detection through YOLO is being worked upon. The netlist generation, result storage and the circuit simulation is yet to be worked upon.

# REFERENCES


[1] B. Bohara and H. S. Krishnamoorthy, "Computer vision based framework for power converter identification and analysis," in *2022 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*. IEEE, Dec. 2022.

[2] M. Rabbani, R. Khoshkangini, H. S. Nagendraswamy, and M. Conti, "Hand drawn optical circuit recognition," *Procedia Comput. Sci.*, vol. 84, pp. 41–48, 2016.

[3] M. Alhalabi, M. Ghazal, F. Haneefa, J. Yousaf, and A. El-Baz, "Smartphone hand-written circuits solver using augmented reality and capsule deep networks for engineering education," *Educ. Sci. (Basel)*, vol. 11, no. 11, p. 661, Oct. 2021.

[4] M. Moetesum, S. Younus, M. Warsi, and I. Siddiqi, "Segmentation and recognition of electronic components in hand-drawn circuit diagrams," *ICST Transactions on Scalable Information Systems*, vol. 5, p. 154478, 04 2018.

[5] M. Gody, "Hand-drawn electric circuit schematic components," Sep 2022. [Online]. Available: https://www.kaggle.com/datasets/moodrammer/handdrawn-circuit-schematic-components