Circuitrix

Autograding of Circuits

Jeya Vikranth Jeyakumar
Tzu-Wei Vivi Chuang

Table of Contents

Introduction

Comparison with Prior Works

Implementation

Challenges

Results

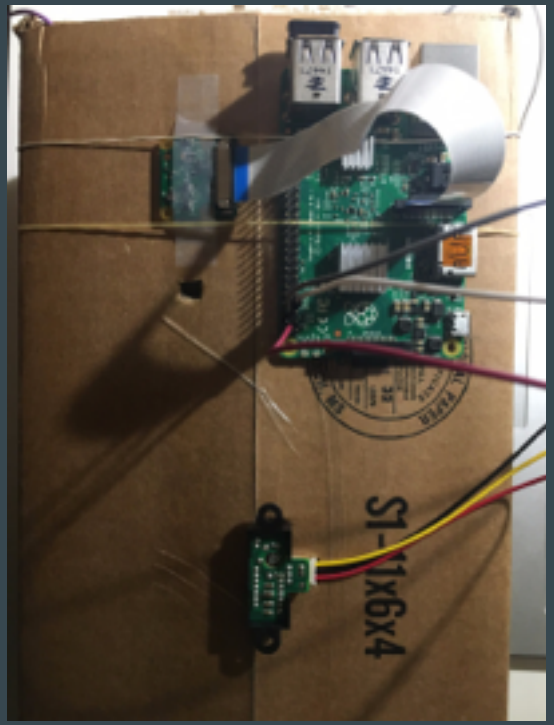
Conclusions

References

Questions

Introduction

- Grading circuits
  - Time consuming
  - Prone to errors
- Drawing schematics in SPICE
  - Difficult
  - Exasperating
- Circuitrix
  - An autonomous grading device
  - Recognize elements in hand-drawn circuit diagram
  - Regenerate a circuit schematic in SPICE

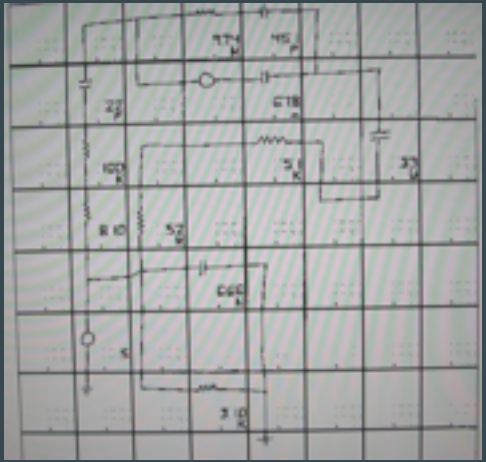


Comparison with Prior Works

Recognition of Hand-Drawn Circuits	Our system
Drawing Rules	
Draw on predefined grid paper	Draw on predefined area on paper
Text must follow separate grid	Text must be drawn relatively similar in size and clearly separated from the components
Appropriate borders must be crossed	No restriction
Component must avoid designated text area	No restriction
Entire component must fit in one grid block	Components should be drawn relatively similar in size

Comparison with Prior Works

Components Recognition	
Continuous / Gaps	Contour size / Width / Height
Text Recognition	
8*6 block	Machine Learning OCR



Components

Raspberry Pi Setup

- Headless: First time boot required monitor
- Wifi Connection: Varying IP addresses
- No keyboard: Install onscreen keyboard

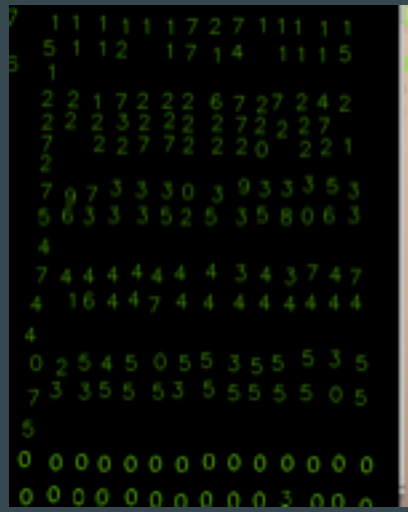
Image Processing

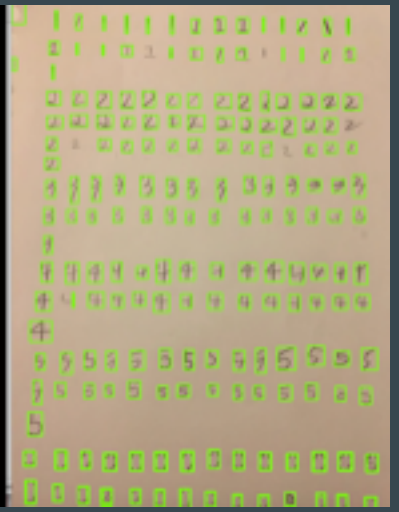
- First approach: Machine learning
  - Haar cascade classifier wasn't robust
  - Amount of data
  - Time
- Second approach: Morphological elements + edges detection

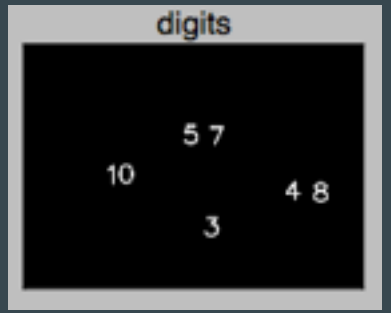
```
Training until now has taken 0 days 22 hours 27 minutes 10 seconds.

===== TRAINING 5-stage =====
<BEGIN
POS count : consumed   1000 : 1000
NEG count : acceptanceRatio    600 : 0.169396
Precalculation time: 43
+----+---------+----------+
|  N |    HR   |    FA    |
+----+---------+----------+
|   1|        1|         1|
+----+---------+----------+
|   2|        1|         1|
+----+---------+----------+
|   3|        1|         1|
+----+---------+----------+
|   4|        1|         1|
+----+---------+----------+
|   5|        1| 0.763333|
+----+---------+----------+
|   6|        1| 0.731667|
+----+---------+----------+
|   7|        1| 0.596667|
+----+---------+----------+
|   8|        1| 0.463333|
+----+---------+----------+
END>
Training until now has taken 1 days 3 hours 40 minutes 34 seconds.
```

Components

- Finding numbers
  - Optical Character Recognition
  - Filter filter filter!!
    - Parameters adjusting to eliminate noise
  - Redraw digits onto a blank background
  - Contour search







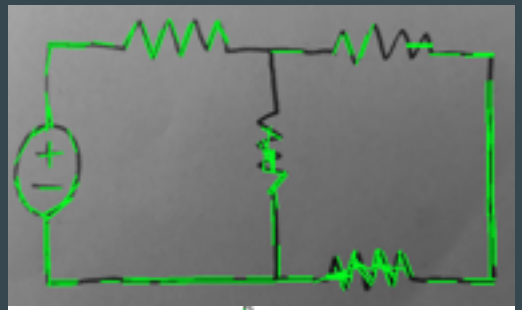
Components

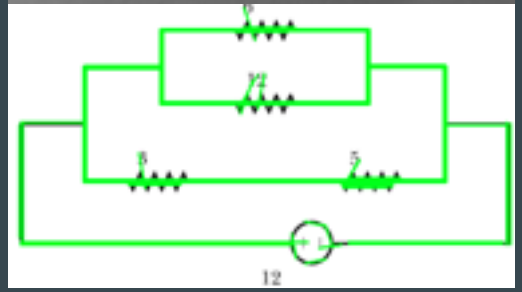
Finding lines

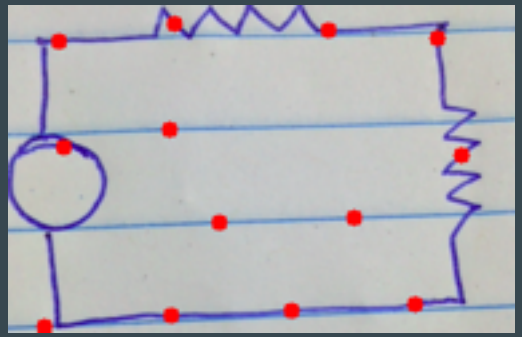
- Edges detection not enough for hand drawn circuits
- Structuring elements to extract horizontal and vertical lines

- Finding circuit elements
  - Clean circuit image - lines found in previous step = circuit elements

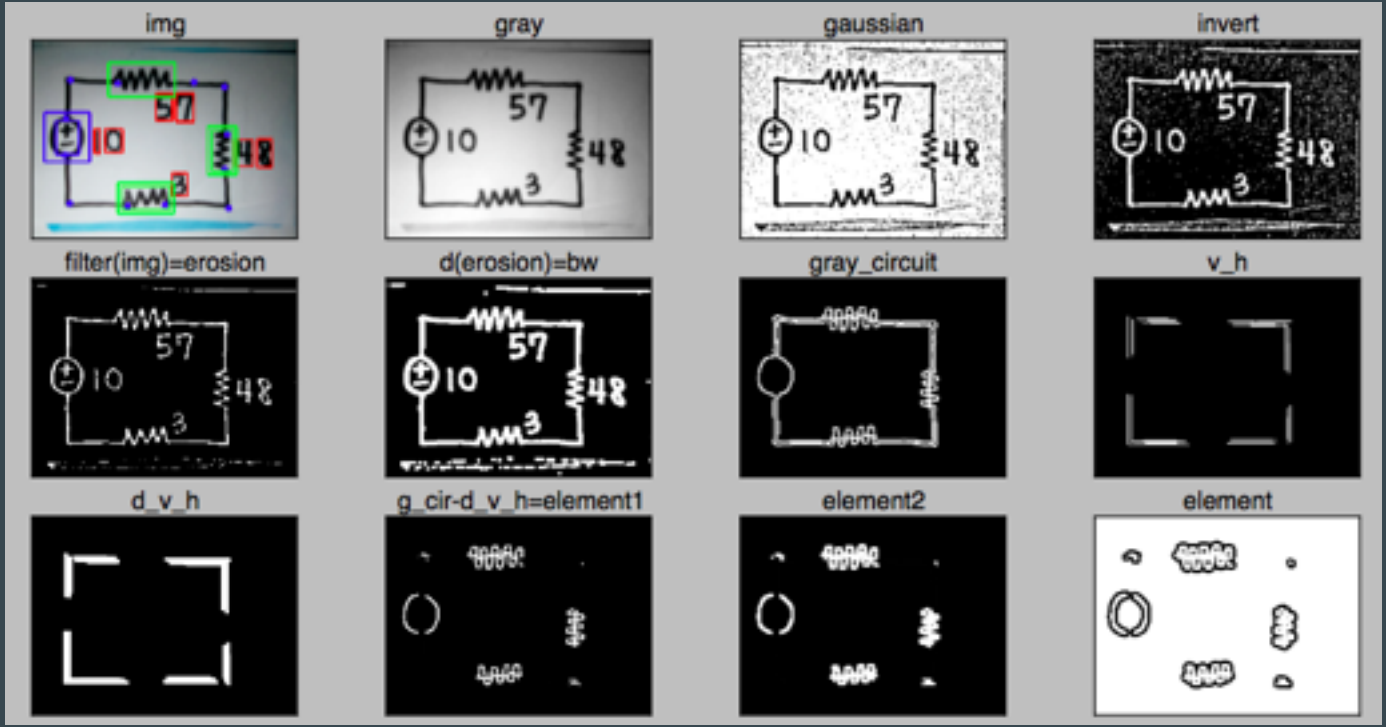
- Finding corners
  - Good feature to track on lines





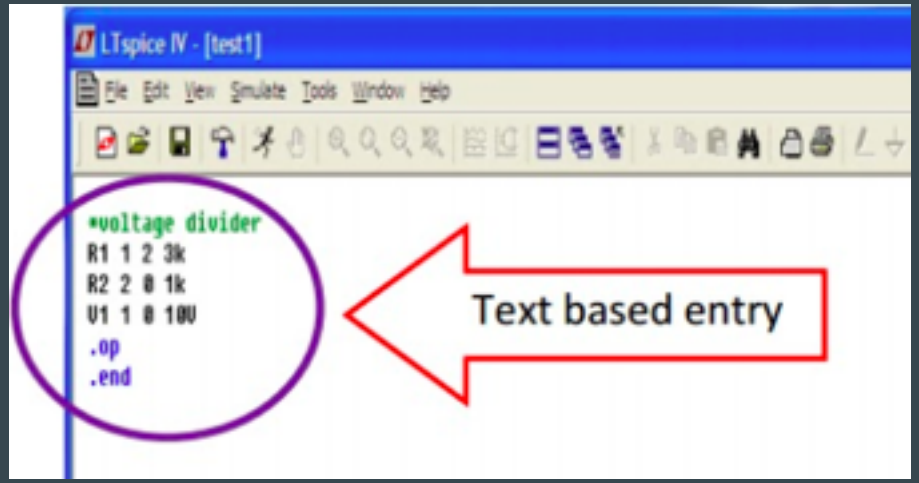


Components


img
gray
gaussian
invert
filter(img)=erosion
d(erosion)=bw
gray_circuit
v_h
d_v_h
g_cir-d_v_h=element1
element2
element

Components

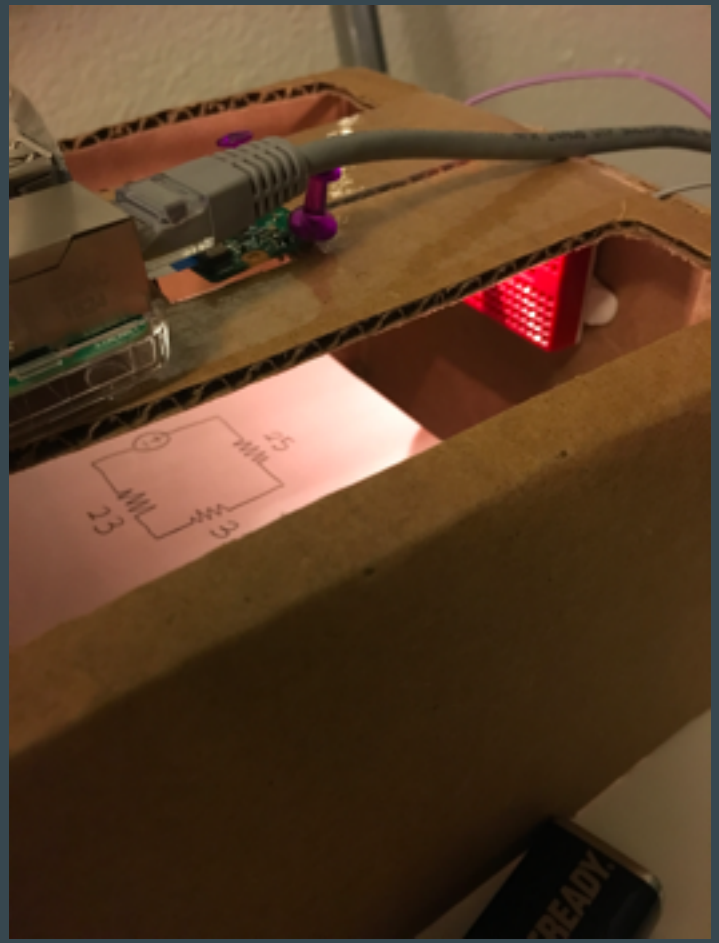
- Netlist generation
  - Conversion from matlab to python
  - Coordinates sorting 2D Array
  - SPICE not supported on Pi, send .txt to PC



Components

Hardware

- IR sensor
  - Detect color
  - Detect distance
- Suspension mechanism
- Lighting
  - Affects image processing significantly
  - Christmas light
    - Shadow
    - Draining batteries
    - Brightness
  - Cell phone flashlight



Conclusions

What was learned?

- Python
- OpenCV functions
- Hand-drawn image recognition

Future Implications

- Adding more components
- Auto simulate without intervention

References

http://sprg.massey.ac.nz/pdfs/1997_enzcon_147.pdf

http://rationale.csail.mit.edu/publications/Sezgin2006Sketch.pdf

http://web.mit.edu/6.111/www/s2005/PROJECT/Groups/3/main.html

http://smarttools.engr.ucr.edu/

The End

Thank you for listening :)