Electric circuit component and connection detection:

1. Introduction

In this project, I made an electric circuit analyzer, which uses scratch and component recognition and then combines the information in order to get the results.

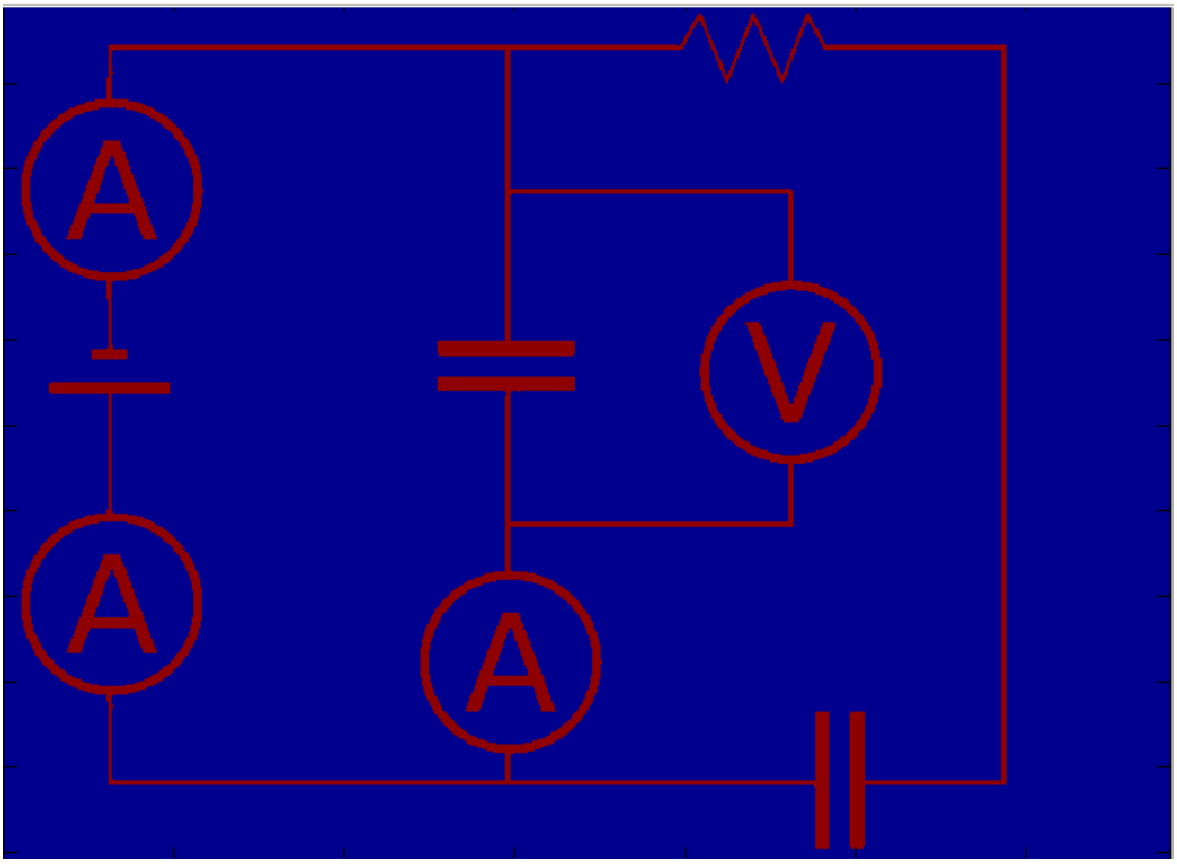
The idea is to detect any component of the electrical circuit and find the relation between any two components by knowing who's "touching" who.

Given a photo of electrical circuit, the program will align it, find any components in the circuit, find all connections between lines, and analyze the circuit. After that, an interface will appear to the user, which gives the ability to choose any component, and view it graphically.

2. Approach and methods

I. Finding edges of the circuit
II. Rotating the electrical circuit.
III. Finding all of circuit's component.
IV. Finding all the lines junctions.
V. Find the path between any components.

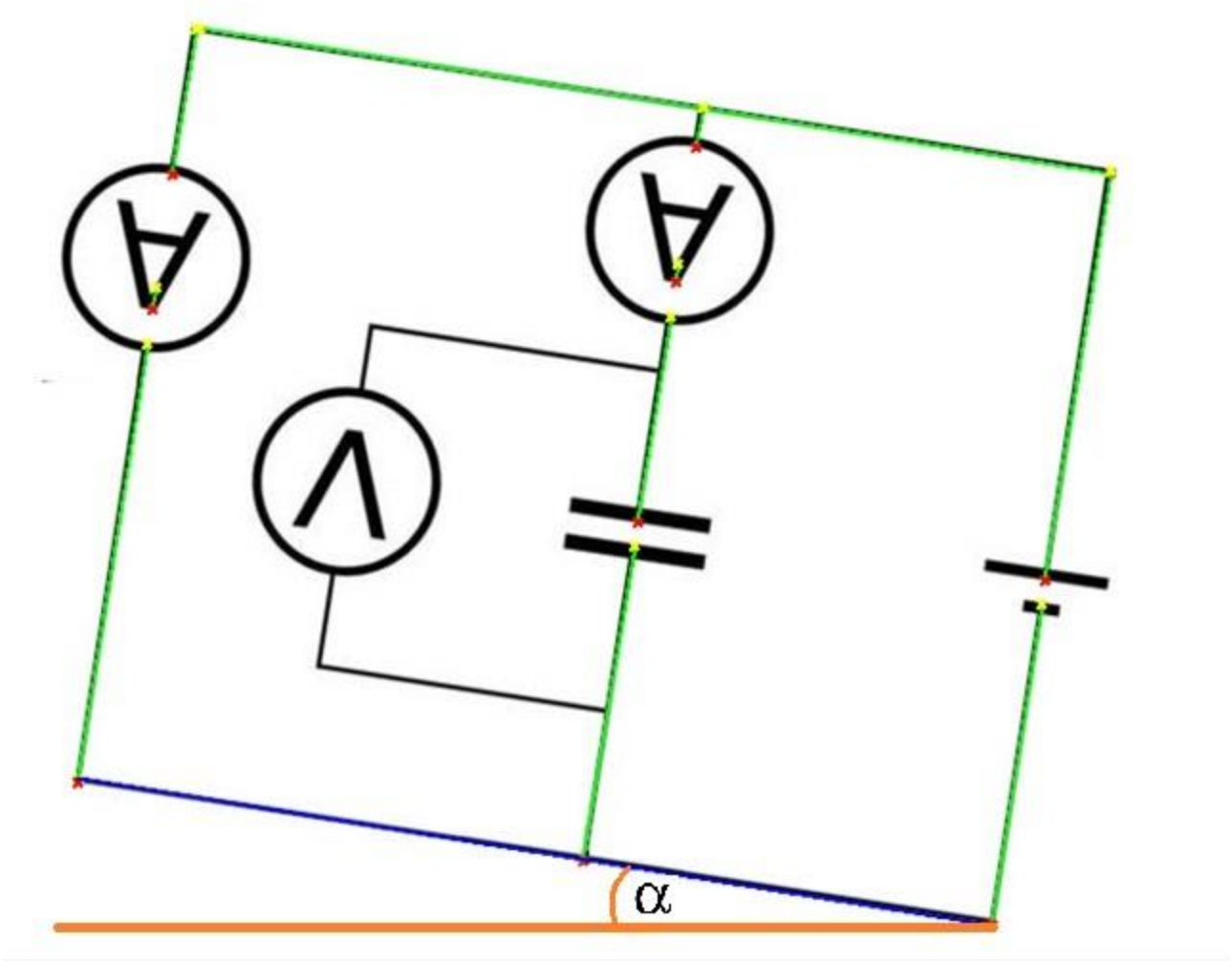
I. Finding the edges by dynamic threshold (easily done thanks to the black and white feature of the circuit).



II.

Rotating the electrical circuit:

Find all lines by Hough transform.
Search for the longest line, and aligning according to it.
The calculation is being done by inverse tangent (arctangent) of the longest line.



Example of Hough transform and finding the longest line

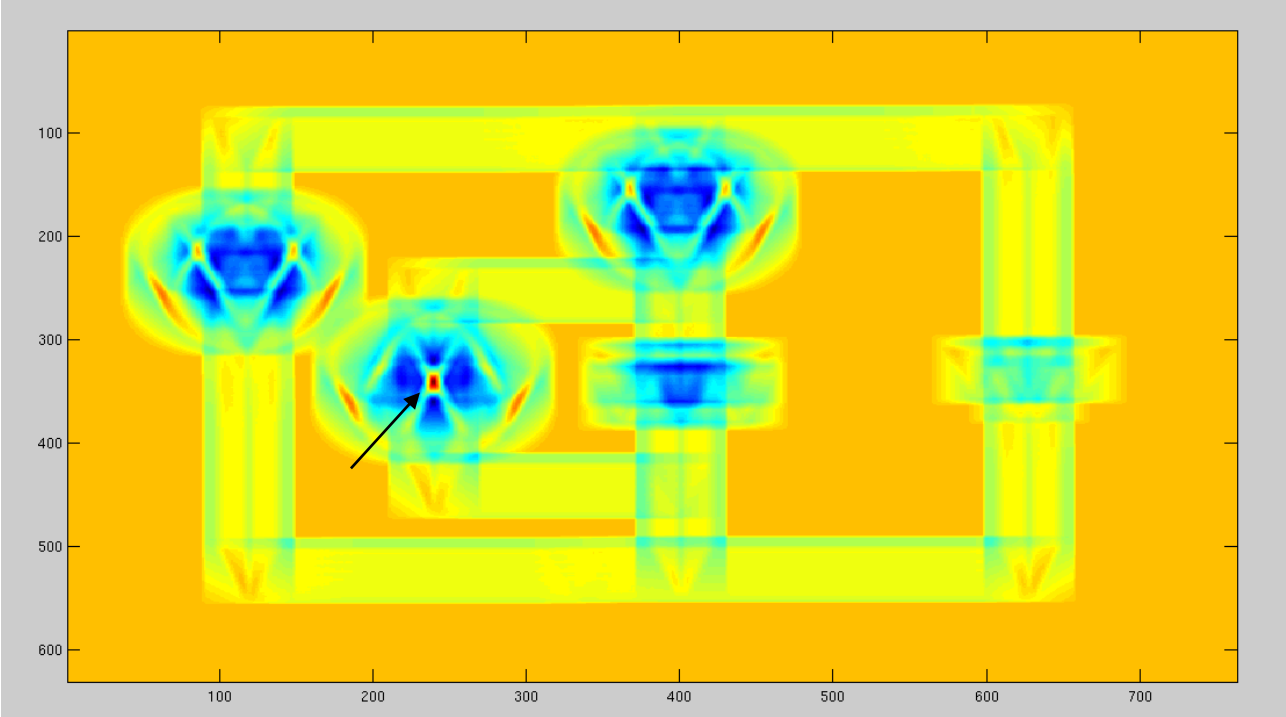
III.

Finding the circuit components:
This is being done by convolutions (Hough transform).
The detection is being done by taking 15 sizes of any components, and rotating it 4 times (according to the component).

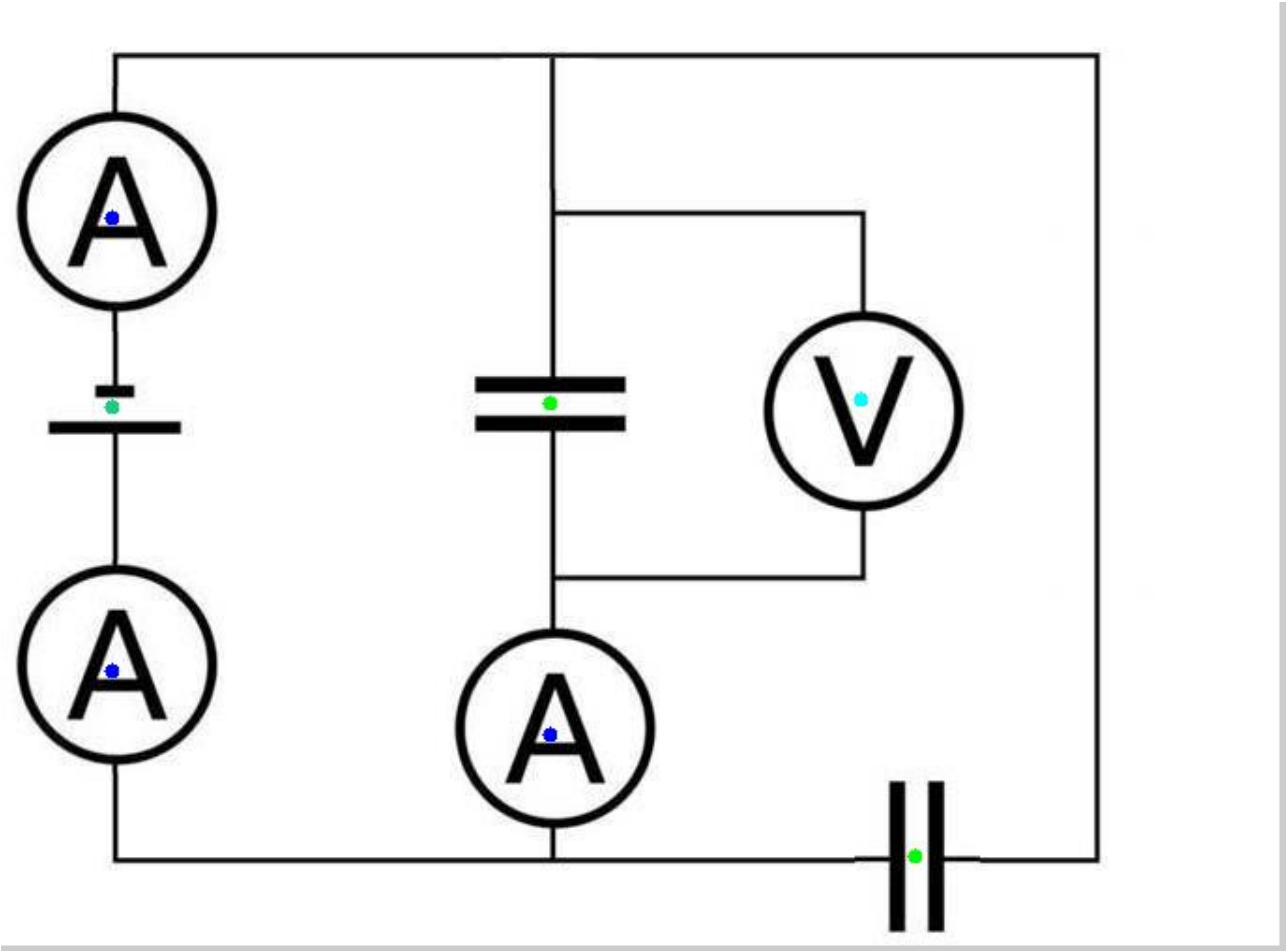
Then we make a kernel of the convolution in the shape of the specific component that we want to detect.

The kernel is being made by putting positive values in the shape, and negative values in the other locations.
Also, the kernel is being normalized to fit in any size. This makes the recognition not constant to specified size.
We are looking for results above specific threshold, and if we pass the threshold, probably the shape contributed to the positive value of the kernel convolution, and we can recognize a component.

Here is an example of Voltmeter recognition in the circuit. We can see a "jump" in the values at the center of the upside Voltmeter.

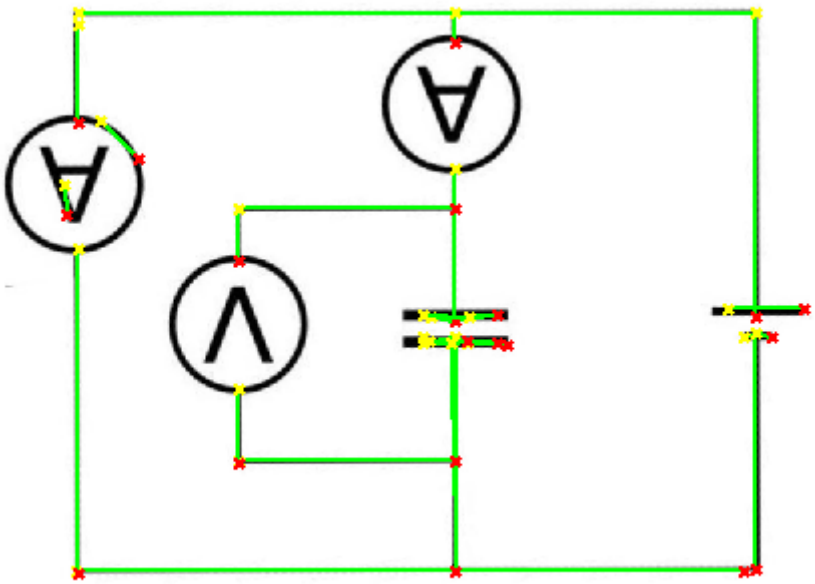


Generated output of segmentation.



IV.

We use Hough transform on the rotated image and find all of the lines.
And then we find all the junction areas.



We can see there is a lot of fake lines, and distance between two lines that might start from the same place, and even worse, there are missing junctions in the intersection of two lines.

The idea is to determine the fake lines, and create new junctions. The Hough transform alone is not enough.

The result of the calculations is a graph, an adjacency matrix of all components and lines junctions.

V.

Finding the path between any components:

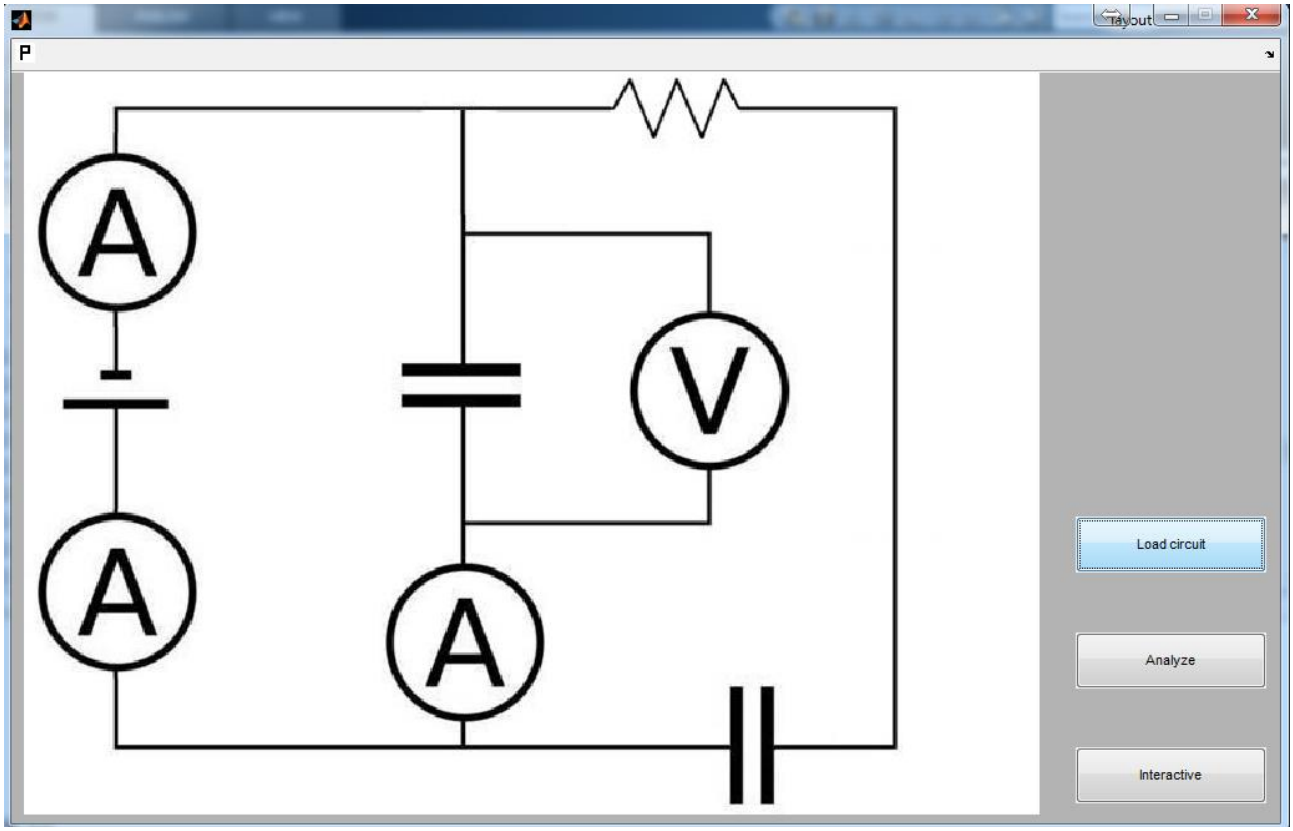
To analyze the circuit, we're going **recursively** from any component, and storing a path list in array.

To search for circuit short, we are going on the path list and searching for an intersection between two poles of components. If there is an intersection, we have a circuit short.

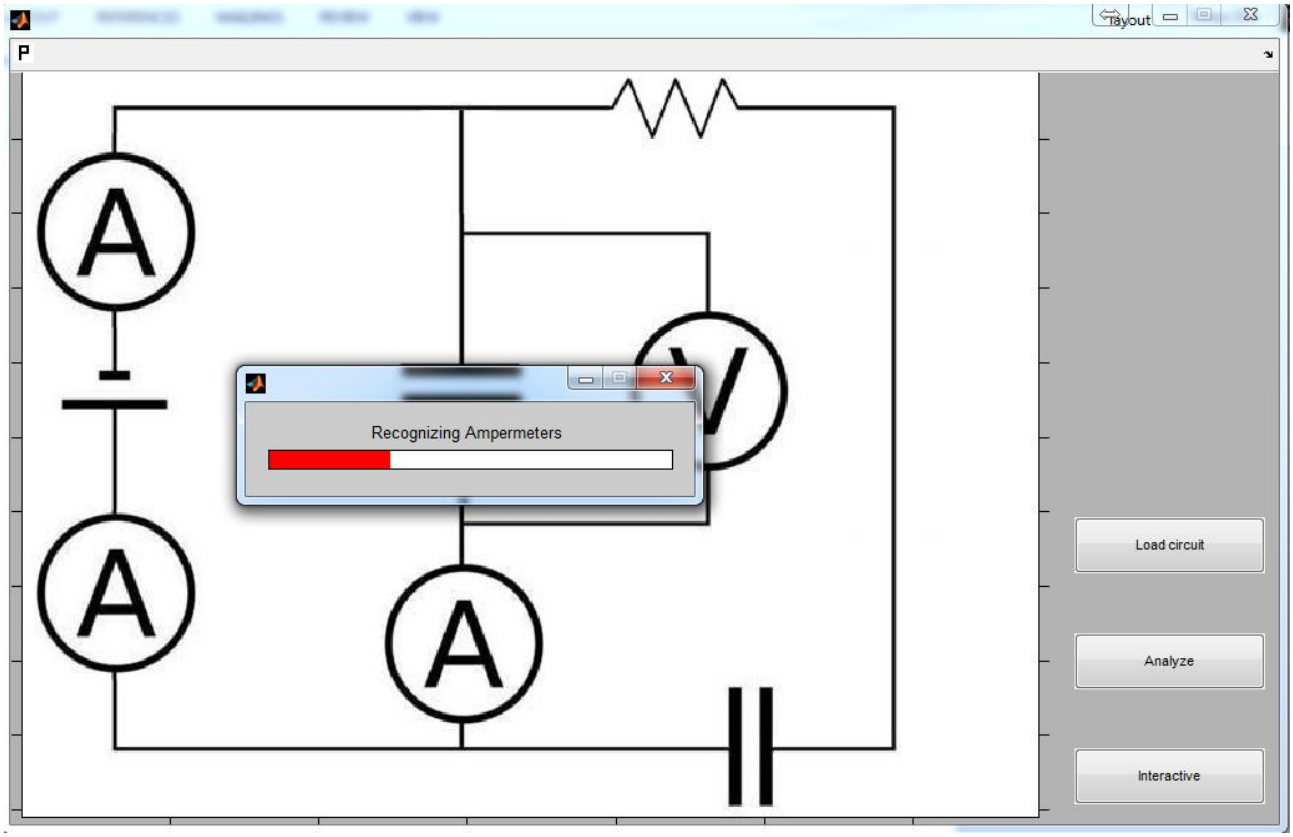
After the analysis we have an opportunity to show to the user which component is connected to another component on the real graph.

3. Results

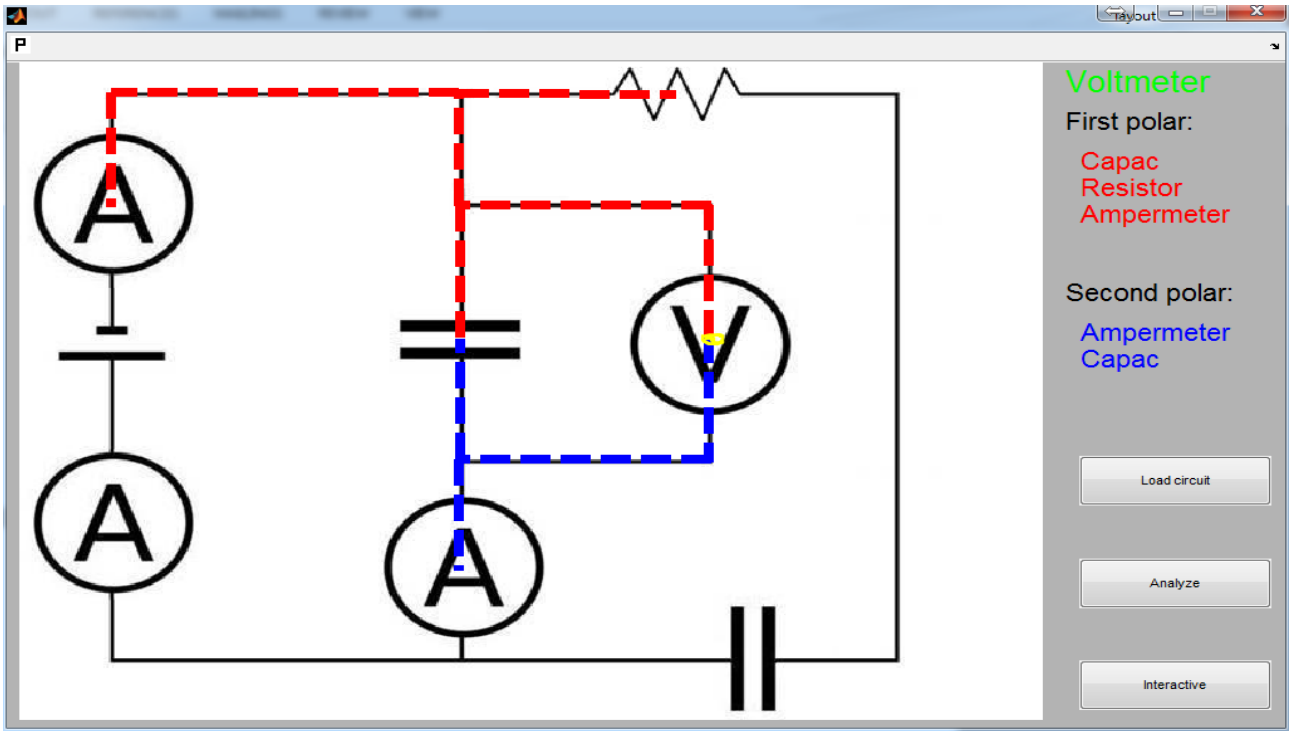
1. Loading the circuit for example:



2. Analyzing:

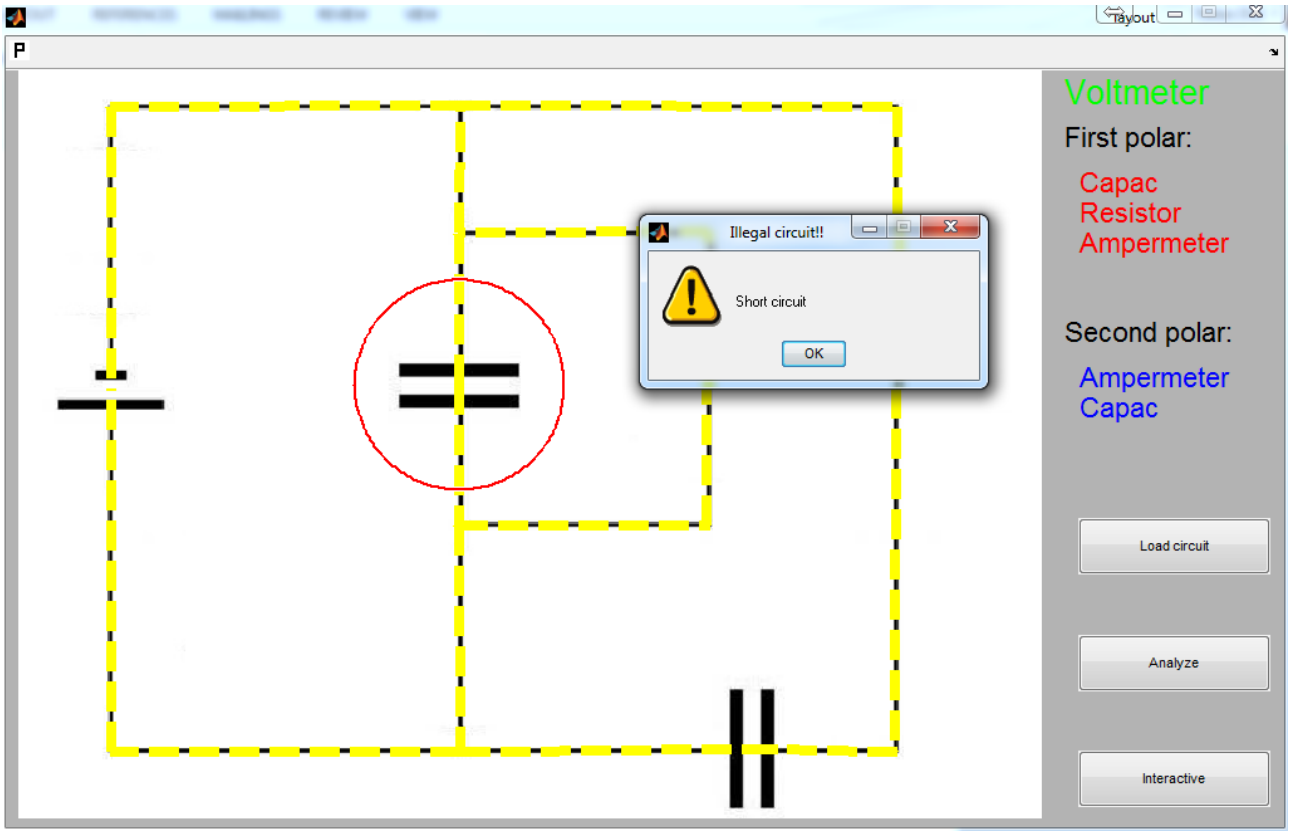


3. Clicking on Voltmeter for example:



On the right side of the screen we can see the list of all the components that are connected to it (for each polar separately), and the connectivity in the graph.

4. A circuit short example :



We can see that it has found the problematic component, and that everything can be reached from this value.

Conclusions

The running time was a bit slow, because we used a lot of size convolutions kernel to adjust to all electrical circuit size we got as input.

With the output graph table, we can extend the ability of the program to analyze a lot more problems in a field of Electrical engineering, and even attach to another API simulator to proceed analyzing.

References

1. Electric circuit
2. Template matching Hough Transform
3. Hough transform
4. Circle Hough Transform
5. Dynamic threshold