Asian Journal of Information Technology 15 (7): 1146-1150, 2016
ISSN: 1682-3915

A Novel Method for Circuit Recognition Through Image Processing Techniques

Shriram K. Vasudevan, Karthik Venkatachalam, S. Anandaram and Anish J. Menon
Department of Computer Science and Engineering, Amrita School of Engineering,
Amrita Vishwa Vidyapeetham University, Coimbatore, India

Abstract: Before building any electronics product or equipment it is well known that, the person building it will finalize the circuit after lot of simulation with simulation tools or relevant coding with electronics coding packages. One needs to be conversant with simulation tools for electronics or coding or both. It is not only complex, it is also time consuming. We have come up with an innovative and effective solution for the above said problem. Here, with our proposed solution, one can draw the circuit that is being thought of in a paper. The image can be fed into our product (application) which will generate net list for the drawn circuit. This will make the life easier and save a lot of time. One can test a lot of combinations of the components of the circuit without having necessity to spend time with simulation or coding. In the past, the inventions related to this area have identified only RLC components. We have enhanced the quality and results with identification of more components, using machine learning and image processing techniques. Usage of machine learning has not only fetched good results but also it increased the chances of identifying more components with ease and accuracy.

Key words: RLC, accurracy, netlist, image, India

INTRODUCTION

Human computer interaction is one area which has a lot of boom and attention these days. If a machine could recognize what a human writes, it would be a value add and could lead to more innovations and inventions. Considering this aspect, we have worked on creating an application for recognizing the electronic components present in a circuit drawn by the user by hand in a paper. Then, the Netlist for the identified components would be generated and can be used further. The circuit recognition which has been identified as a problem is very thought-provoking. The generated Netlist can be used as input for simulators to generate the circuit. This would make the life easier as well with reducing the time needed to simulate or to write code to test the circuit being thought of. Many a times, engineers spend a lot of time in using a drag and drop tool, i.e., a simulator or a language for electronics to code to test their circuit. It always is time consuming and many may not be comfortable with the tools as well. Also, it could cost at least few hundred dollars to buy certain simulators. Our solution is very feasible towards generating the Netlist for the drawn circuit faster. Many existing solutions have been analyzed and improvisation has been done to good extent without compromising the quality of the output. Complete details have been presented below with appropriate explanation.

MATERIALS AND METHODS

Existing solutions: There are few interesting works which have been carried out in the same area. We have analyzed prominent findings and summarized below. Sridar and Subramanian (2013) has created an application with using image processing techniques to identify components. It was one of the best in this sector with accuracy being claimed around 80% in identification of RLC components. One has no restriction in drawing the shape of the components as the point of identification of a component was through the unit mentioned. For example, resistors were identified through Ohm symbol. Coming to the shortfalls, only RLCs could be detected which is not suffice considering the number of electronic components. Image should be clear, i.e., an image from white paper is better for this application. Lighting conditions are also expected to be perfect for this application to respond precisely. Camera to capture the image drawn is expected to be of good resolution, i.e., 10 mega pixels. There is another major shortfall in this app. If the unit is mentioned wrongly in the circuit drawn, i.e., Ohm as volts, the system would fails and there is no intelligence to find the component through the shape and structure. Also, if there is no unit mentioned it will be a misread as such there is no component or if there is only unit mentioned, it would be read as a component in place which is not the expected behavior.

Corresponding Author: Shriram K. Vasudevan, Department of Computer Science and Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham University, Coimbatore, India

Ravi Palakodety and coauthors used finite state machines to replace the hand drawn components by computer generated images of the identified components. Since, the images are replaced by system generated one, the look and feel of the components are really excellent. The output will be in the form of SPICE Netlist which is also appreciable. Coming to the areas of concern, the circuit has to be drawn exactly in 8×8 grid with components being in the center of any individual cell. Values and units of the drawn components are expected to be written at lower right corner side of the corresponding grid.

Edwards and Chandran (2000) built the application to recognize vast range of components through geometric features. Accuracy is found to be appreciable and above 90%. Since, based on the geometry of the drawn images the recognition is made, it becomes absolutely necessary for someone to draw the components precisely in correct shape as it could be if computer generated. Drawing circuits with very high precision by a human is difficult and it is a major constraint in the invention.

Donald Bailey and coauhtors 1997 created an early version of this invention with restriction to printed circuit images only. It was created with an intention of readers to be helped with better understanding of a circuit while walking through the book. No labels are processed in this application and only printed images get results. No hand drawings are recognized with this application. But, it is highly appreciable that someone in late 90's thought of this idea which has stood as complete base for further inventions, including our invention.

Okazaki *et al.* (1988) created an application for component recognition which was accurate about 95%. But, the only major setback was the loop structured components only were detected using the invention. But, it is a great effort very at such an early era where there were no much tools and techniques in place.

To summarize all the available inventions in this area have following deficiencies, which all have been given an attempted to be addressed through our application, we developed:

- Hand drawn circuits are to be recognized better
- Components are to be recognized based on machine learning, not just through units or geometry
- Increased support for recognizing more components other than just R L and C
- No restriction for drawing the image in a specified grid area should be imposed on the user
- No rule as in the units are to be mentioned at a specific point in the grid
- Lighting conditions can be a point of consideration. But, should not be a show stopper

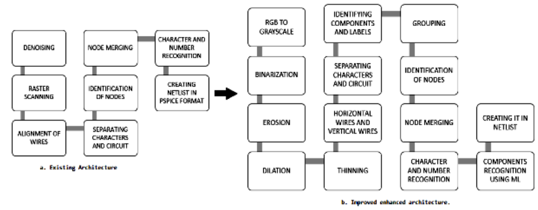

Fig. 1: Proposed architecture derived out of current architecture in existence.

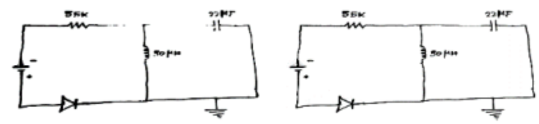

Fig. 2: De-noised image through Gaussian Filter and De-noised image through new technique

- It should not be demanded that the camera quality should be excellent for catching the image. Moderate quality camera should also be supported

MATERIALS AND METHODS

Problem statement: To create an application which would recognize the analog circuits drawn on a paper with a pen or a pencil. Then, to create Netlist for the identified components which can be fed into the simulator right away.

Proposed architecture: Figure 1 represents the changes/improvements we have carried out from the existing architecture understood from previous available literature. As one could see, there are many steps to be followed in sequence to get the final Netlist generated. Each step is explained below with appropriate reason for opting for the same. First and foremost step is to draw a circuit with free hand and take an image of it which serves as an input for our application.

Denoising: When a drawn circuit is being captured by a camera, there is every possibility that unwanted contents like shadow, dots in the surface etc. could be covered and could also be part of the final image. It is not advisable to go with this unintendend content till the end. So, it is better to remove the noise, i.e., whatever is not a component, has to be removed and we achieve that using de-noising techniques. Traditional Gaussian Filter has been deployed through OpenCV to remove noise. Usage of Gaussian Filter for de-noising did not fetch us expected results. The result obtained with Gaussian Filter technique is presented in Fig. 2. Since, not satisfactory, de-noising has been achieved through an alternate

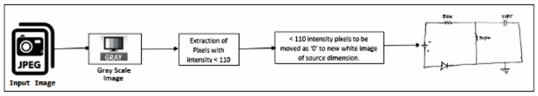

Fig. 3: Steps in proposed technique to remove noise

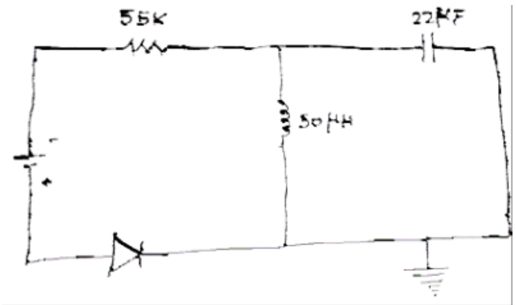

Fig. 4: Thinning wire example

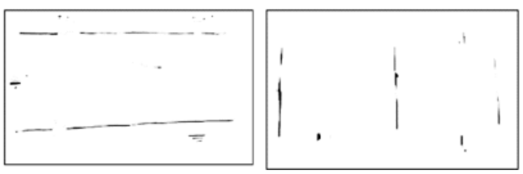

Fig. 5: Horizontal and vertical wire separation

technique. The input image has been converted to a gray scale image. Pixels with intensity < 110 (A threshold found out through repeated trial and error approach) are alone moved to a new white image with pixel intensity as '0' (Black). This results in better filtering with no impact of noise. Figure 3 has revealed the output and it is clearly visible for anyone to appreciate the second approach followed by us.

For better clarity, the followed steps for de-noising the image is presented below in Fig. 4. The next step in the sequence for the existing techniques is thinning (Okazaki *et al*., 1988). Many existing techniques support thinning. But, it is not mandatory. Thinning actually extracts the skeleton from the generated de-noised image. A sample is presented below for understanding in Fig. 5. Thinning actually resulted in a disconnected horizontal or vertical wires which affects the component identification as shown in Fig. 6. So, instead of relying on thinning, we skipped the step of thinning thereby, reducing the processing time by few microseconds. Exclusion of thinning is not a technically impacting step towards the quality of the end result.

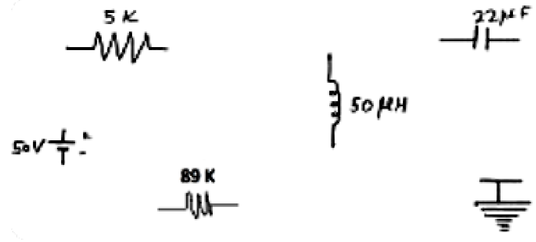

Fig. 6: Raw components identification result

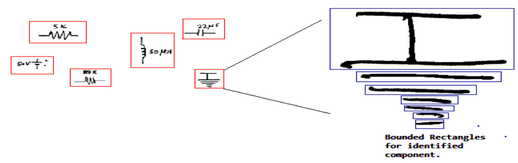

Fig. 7: Bounded rectangle identification

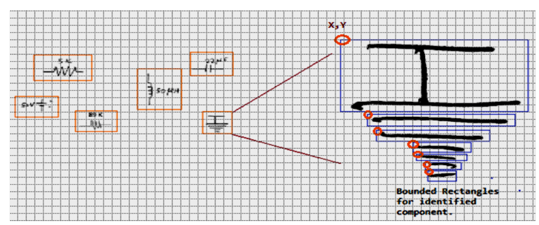

Fig. 8: X,Y point identification

Horizontal and vertical wire separation: To isolate the components from the circuit, horizontal and vertical wire separation is needed. Raster scanning is carried out on an entirety over the image to retrieve the horizontal and vertical lines. The results are presented in Fig. 7.

Grouping component: Grouping the components require a lot of care and processing. With the processing being done in previous step, one could retrieve all the raw components from the image as shown in Fig. 8.

For an instance, if you refer to ground symbol, it will be seen as many individual lines. All of them are to be grouped to get the ground as an individual component. Finding contours is the major challenge and it has been accomplished. With the available results Bounding Rectangle is found out for each sub image. The inbuilt available bounding rectangle function from OpenCV has been used to achieve this task. Figure 8 has revealed the image result containing the raw components identified.

As a component, for Ground, all the bounding rectangles identified. Grouping after this is done through identifying X and Y points at top left for each bounding

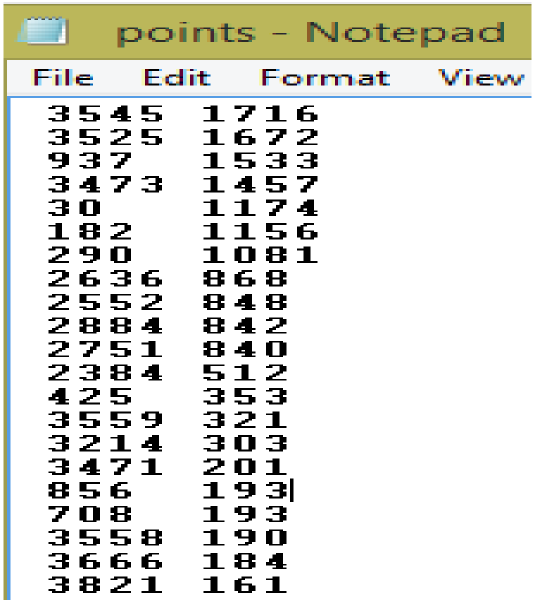

Fig. 9: X, Y values in a text file

rectangle. All these points are stored for further processing. This s revealed in Fig. 9 and 10.

The X, Y points identified is stored as numerical values in a notepad, i.e., text file which enables us to proceed with further processing. Obtained values are shown in Fig. 10 for all the contours.

The obtained numbers would help in identifying the contours towards building the components, i.e., All the related contours identified through the distance as a parameter and would be brought together to build a component. That is, a component gets constructed only with closer contours. The result for each component would be stored as an individual directory. As an instance, the ground as a component would be stored with all its identified lines in a unique directory. Same is the case for the resistors, capacitors, etc. Figure 9 would reveal the idea discussed above.

Here, all the components of the input image have been identified successfully with appreciable accuracy and negligible noise. The next subsequent task is to find the area of each sub image if the grouped components. An algorithm, i.e., a method has been devised towards achieving this task. The algorithm helps in classifying the grouped components into labels (units) and component (i.e. capacitor, resistor, etc.) and to store them in a separate folder. Bounded rectangle used in the previous step can help us in getting the numeric for area. The numbers representing area would be unsorted initially and our algorithm would sort it first. Difference between the successive values would be found. The most significant

Fig. 10: Work flow towards component building

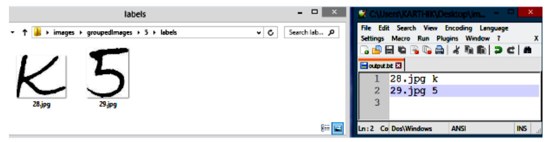

Fig. 11: Label identification and recognition

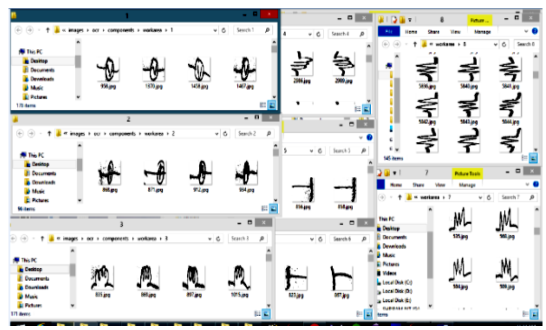

Fig. 12: Sample data set content for reference

difference between two numbers would help in differentiating the component and label. The complete illustration is presented in Fig. 10.

RESULTS AND DISCUSSION

Character recognition: MNIST database (Mixed National Institute of Standards and Technology database) is collection of handwritten digits. MNIST is one of the most commonly preferred dataset for training image processing systems. Also, there is a lot of scope found in the area of machine learning towards testing and training the system. All the symbols as Ω, µ, etc. are all hand drawn and they are also part of the database. This is all dealt with respect to the labels. Similar approach has been followed for components too. Many of the components including resistor, inductor, capacitor, diode, transistor etc. are hand drawn and made a part of the database. Identification becomes perfect through usage of both label and component. In the previous attempts, only one of those two have been done and it could as well be imperfect. We have avoided that and enhanced accuracy through this approach without compromising anything. Fig 11

The label recognition is achieved with high accuracy and a sample is presented below in Figure. 12. Similar

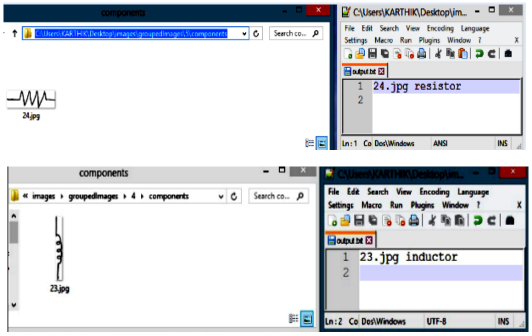

Fig. 13: Component identification inductor

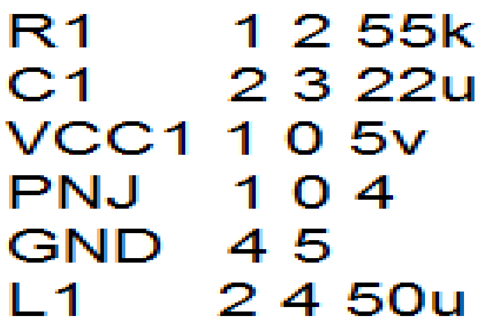

Fig. 14: Generated netlist for the taken input image

results are obtained for all the labels and perfect identification and recognition of labels are accomplished. The final screenshot below presents the component identification result. Through the collective usage of both component and labels, the final yield is assured to be perfect. Figure 13 reveals the component Identification result. Figure 13 shows the resistor being identified and the same has been given as the updated in the text file. Figure 13 shows an inductor being recognized. Similarly all the components as capacitor, voltmeter, ammeter, battery (power source), grounding symbol, transistors are all identified and recognized properly.

Thus, an uphill task of identification and recognition of the components has been achieved. The netlist generation is presented in the study.

With all the above steps being done successfully beginning with Preprocessing followed by RGB Grayscale conversion, removal of horizontal and vertical wires leading to component and label grouping. Then the grouping of relevant components and its label was carried out. Components and labels were recognized and the screenshots are presented in the previous section. Numbering the nodes is carried out properly to ensure smooth generation of netlist. The final task of getting the netlist generated is also carried out the following snapshot (Fig. 14) exposes the netlist generated for the taken input image.

CONCLUSION

The netlist for the any drawn circuit has been generated and following points are addressed too in the application created:

- Hand drawn circuits with pencil or pen is recognized
- Machine learning techniques are used to train the system and data collection was done extensivel
- Existing techniques just supported RLC. We have gone beyond the same and we could achieve appreciable accuracy
- User has been given a free hand with no restriction for drawing the image in a specified grid area
- User is also given a comfort of not imposing the rule as “units are to be mentioned at a specific point in the grid”
- We could achieve good amount of accuracy with any sort of camera ranging from 5 MP

Needless to say, there are few areas of concern which are to be treated as future scope. Lighting conditions remain a concern. Also, this application can be made an android application as the next step. Usage of augmented reality to enhance the readability and usability can be a crowing factor for the application design. Using AR with android has already been a trend and we can use the opportunity in near future (Sundaram *et al*., 2015; Vasudevan *et al*., 2015).

REFERENCES

Edwards, B. and V. Chandran, 2000. Machine recognition of hand-drawn circuit diagrams. Proceedings of the 2000 IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP'00, June 5-9, 2000, IEEE, Istanbul, Turkey, ISBN: 0-7803-6293-4, pp: 3618-3621.

Okazaki, A., T. Kondo, K. Mori, S. Tsunekawa and E. Kawamoto, 1988. An automatic circuit diagram reader with loop-structure-based symbol recognition. Pattern Anal. Mach. Intell. IEEE. Trans., 10: 331-341.

Sridar, S. and K. Subramanian, 2013. Circuit recognition using netlist. Proceedings of the 2013 IEEE Second International Conference on Image Information Processing (ICIIP), December 9-11, 2013, IEEE, Shimla, India, ISBN: 978-1-4673-6099-9, pp: 242-246.

Sundaram, V.M., S.K. Vasudevan, C. Santhosh, R.G.K.B. Kumar and G.D. Kumar, 2015. An augmented reality application with leap and android. Indian J. Sci. Technol., 8: 678-682.

Vasudevan, S.K., A. Ritesh and C. Santhosh, 2015. An innovative app with for location finding with augmented reality using cloud. Procedia Comput. Sci., 50: 585-589.