MIPRO 2021, September 27 - October 01, 2021, Opatija, Croatia

Recognition of symbols and topology in the image of a DC circuit diagram based on contours and skeletons

Matej Štorga, Mirko Randić
Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia

Abstract – **Scope of research presented in this paper refers to the off-line recognition and interpretation of the content that conveys images of printed or hand-drawn direct current circuit diagrams. Photographing learning materials with different quality imprints as well as under different lighting exposures may introduce a lot of noise in the image. Besides the standard noise caused by shadows and unwanted spots, common noises that introduce ambiguities into the process of symbol and circuit topology recognition are breaks on connecting lines and symbols. That is why we have designed algorithm in such a way to ensure correct recognition even in conditions of noise. Due to the mostly regular geometric shapes and relationships that define symbols and the interconnection of symbols in an electrical circuit, and the clear topology related to the existence of nodes and branches, we decided to develop a recognition method based mostly on the structural relations. To recognize symbols and topology, we use suitable geometric primitives, contours, and skeletons that can be detected in the image. The recognition method based on the structural patterns that can handle unwanted breaks and disconnections in the image of electrical circuit diagram represents the contribution of this paper.**

Keywords – image processing; symbol recognition; electrical circuit diagram; circuit topology; OpenCV; contour; skeleton

I. INTRODUCTION

Result of the development of an Android application [1] that allows a user to take a photo of printed or hand-drawn electrical circuit diagram and then to start a process of automated recognition and interpretation of the circuit depicted on image is described in this paper. Automatic diagram recognition process, which means recognizing and interpreting the content of image showing printed or hand-drawn diagram of electrical circuit, is a challenging task. Such diagrams or schemes are network-like diagrams consisting of isolated, non-overlapping symbols that are linked together by connectors. The structure of these diagrams can vary considerably in terms of topology and the number and types of used symbols. Furthermore, there exist various annotations and adornments associated to parts of a diagram. Most common are parameter values associated to symbols. Generally, machine analysis of an image showing a circuit diagram involves extraction of information conveyed by the diagram in a multiphase process, using various algorithms and methods in the field of image processing.

The recognition algorithm described in this paper has been developed gradually. In the first step, we have designed an algorithm that was successful in recognizing symbols and topologies, but it was limited to diagrams that in the binary image had one main, outer contour in continuity without disconnections. Such an example of the diagram is shown in Figure 2. The longest whiteness in the binary image was first detected and further recognition was based on that. The advantage of such approach is that other whites that are not part of the diagram core (noises, parameter values, other text) do not affect the symbol and topology recognition. In practice, common symbols like voltage sources that often appear in diagrams introduce disconnections. Also, disconnections occur due to poor photo quality or inaccurate drawing. Disconnecting symbols or broken connecting lines are not uncommon in the images. Therefore, the diagrams captured in images are regularly composed of several contours, so that the restriction in terms of one main contour is not realistic.

In the second step we have redesigned the algorithm in such a way as to achieve a certain type of error tolerance i.e., to ensure correct recognition even in conditions of breaks on lines and symbols. These are situations in which the recognition of topology and symbols is considerably difficult. We still assume that the greatest whiteness in the binary image is part of the scheme. Then, relying on the usual geometric shapes of the symbols in DC electric circuits, we find other whites that also represent parts of scheme and are not parts of annotations and adornments. Such an algorithm has proven successful in recognizing realistic, hand-drawn or printed schemes. It is also tolerant of symbol rotation and scaling.

A. Related work

Diagram recognition or capturing information content from diagrams has been an active research area for decades. Over the years, a lot of research has been done and many papers have been published, even on the topic of electrical circuit diagram recognition and interpretation. Various problems have been solved, from vectorization of images of circuits, possibilities of content search and symbol spotting to interpretation of circuits and automated analysis problems solving. In [2] an application of image processing techniques to recognition of hand-drawn circuit diagrams is presented. A diagram is observed in a way that it comprises of nodes, connections, and components. The scanned image of a diagram is preprocessed to remove noise, convert to binary form and to obtain a clean, connected representation using thinned lines. Nodes and components are segmented and classified. The authors report node recognition accuracy of 82% and a component recognition accuracy of 86%. However, the method does not handle the scale and orientation changes in the images. Another method for recognizing hand-drawn electronic circuit diagram is presented in [3]. Component recognition is based on feature vector constructed by combining Local Binary Pattern (LBP) and statistical features based on pixel density. For component classification the authors use support vector machine (SVM) classifier. For establishing a sequence of components, a finite state machine has been used. An automatic understanding system for comprehending electronic circuit diagrams in printed manuals or freehand drawings of fair quality is described in [4]. It is assumed that a circuit diagram contains three categories of essential components: circuit symbols, characters, and connection lines. The authors have described understanding of circuit diagrams as a hierarchical process that includes a series of preprocessing, extracting, synthesizing, recognizing, and interpreting phases. After running preprocessing operations, each object in a binary image is thinned down to a skeleton of a unit thickness. The component extractor traverses a circuit diagram by following the skeletons of objects. Each object in a circuit diagram is composed of a set of picture segments which are delimited with feature points. In [5] the authors use annotations related to symbols in recognition process. They state that although annotations are an integral part of symbols in drawings, due attention is yet to be given for their identification, interpretation, and storage. The proposed method first separates the text region around an intended circuit symbol, and then identifies the annotation part of the segmented text corresponding to that symbol.

B. *Concepts*

- *Electrical circuit diagram* (*scheme*) is a formal drawing that consists of interconnected symbols. The word diagram will be used interchangeably with the word image in the reminder of this paper, because we are focusing on diagrams represented as rasterized images.
- *Symbols* in diagram have well-defined pixel patterns (geometry) and they represent active and passive electrical elements or devices in real circuit. Connectors are undirected lines that represent conductors which electrically connect the elements. In DC circuits, active elements are voltage or current sources, while passive elements are resistors and instruments (voltmeters or ammeters). All symbols are two-terminal. There are loop and loop-free symbols. Some of them can also be skeleton disconnecting symbols. In an electrical circuit drawing, annotations are mainly used to denote the value and measurement unit of parameters assigned to symbols.
- *Morphological operations* alter the input image using a structural element giving an output image of equal size. Operations used in this paper are blurring, erosion, dilation and skeletonization.
- *Contour* is a closed, curved line, delimiting the whiteness from the binary image, and is described by points lying on the border with white pixels.
- *Skeletonization* is an iterative process of thinning whites of a binary image until all the whites are one pixel wide. The name for thinned whites is *skeleton*. Endpoints and branching points are skeleton points that are relevant to our algorithm.

Ready-made libraries such as OpenCV [6] can be used to find contours and skeletons, as well as for all morphological operations mentioned in this paper.

II. RECOGNITION PROCESS

A. *Input image preprocessing*

Input image that represents electrical circuit diagram is a raster image where standard image preprocessing operations can be applied (Figure 1). These operations are:

- Cutting off unnecessary parts of the image.
- Blurring the image reduces noise that can be present in the image in the form of small white dots.
- Binarization - by applying the appropriate threshold in the image only black and white pixels remain. The whites in Figure 2 provide useful information.
- Thickening and thinning of whites - in the binary image sometimes remain noise in the form of white dots. It is possible to remove them by thinning. Also, by thickening we can reconnect whites that are broken after binarization. It is important to emphasize that the algorithm described in this paper will work with certain noises and disconnections in the binary image.

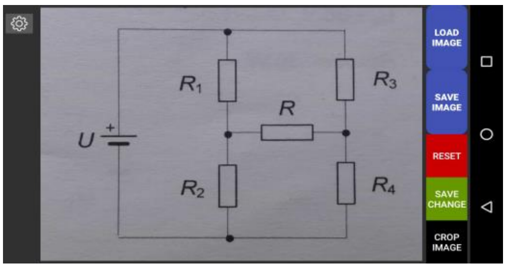

Figure 1. Input image on the smartphone screen

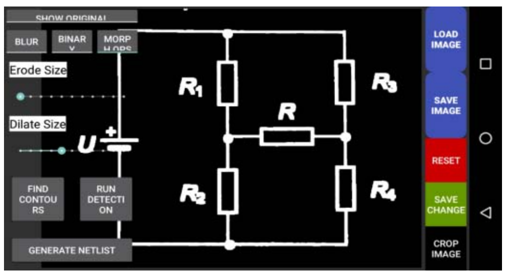

Figure 2. Binarized image

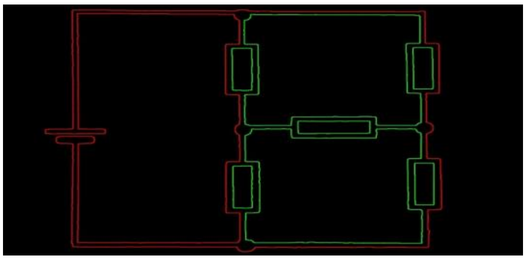

Figure 3. Parent contour (red) and children contours (green), which are each other neighbors too.

B. *Detection of contours and skeleton endpoints*

On the binary image obtained in the previous step, we detect contours. For optimal operation of the algorithm, contours need to be approximated to reduce the number of points describing them. Contours are organized hierarchically so that they have their neighbors, children, and parents (Figure 3). Root contours are contours without parents. These relations between contours are later used in the process of recognizing symbols of the scheme. The assumption of the algorithm is that the largest contour will always be part of the scheme and the largest contour is taken as the starting one. The photographed scheme often consists of several whites or contours, so it is necessary to find all those contours that make up the scheme.

In order to be able to find out whether the contour belongs to the scheme, we perform skeletonization over the binary image (Figure 4). To obtain the skeleton, the Zhang-Suen algorithm is used, which will preserve the connections between pixels and endpoints in the process of whiteness thinning. The skeleton in a binary image is programmatically described by a matrix that has values of 1 in the places where it exists and 0 otherwise. If we apply a 3x3 mask that has a value of 10 in the middle and otherwise 1 over the image matrix, then we get a new matrix where pixels with a value of 11 represent endpoints, pixels with a value of 12 represent a point on a straight line and pixels with values of 13 and 14 represent branches on a skeleton. Pixels with values less than 11 are ignored (Figure 4). Once the end and branch points are found, they need to be assigned to the contours that encompass these points. If some endpoint is encompassed by several contours, then this endpoint will be assigned to the lowest child in the contour hierarchy.

C. *Breaks and disconnections*

If the largest contour does not encompass the whole scheme, then there exist disconnections i.e., other contours that encompass other parts of the scheme. We must add these contours to the list of contours along with the largest. Then, by an iterative procedure, we find the disconnections next to the other contours until we find all the contours that encompass all parts of the scheme. To find disconnections we use the end points of the skeleton.

We distinguish 3 types of disconnections:

• voltage sources – inherently introduce disconnections in circuit's skeleton (Figure 5). If we have a series of voltage sources, then it is certain that the circuit will consist of more than one contour.

• disconnections between a symbol and connecting line (Figure 6)

• breaks on connecting lines (Figure 7)

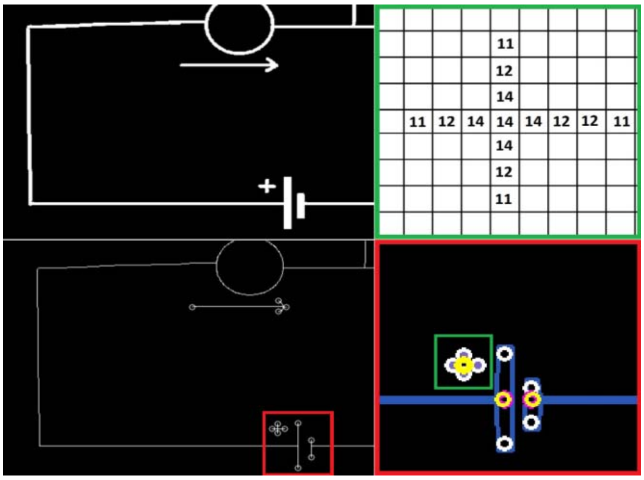

Figure 4. Binarized image (top left) and skeleton with endpoints (bottom left). The green box (top right) shows the matrix notation of the + sign (values 12 are neighbors of the endpoints (11), and 14 are the branch points). The red box (bottom right) shows the contours (blue), the end points (white) and the branch points (pink, yellow).

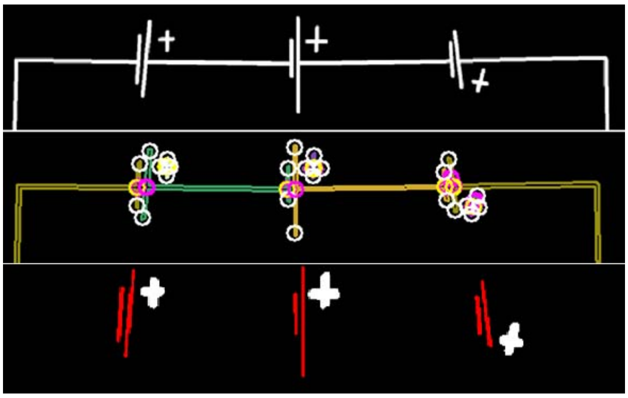

Figure 5. Voltage sources as disconnections (upper) with associated contours and end points (middle). Recognized pluses (down, white) and candidate lines for voltage sources (down, red).

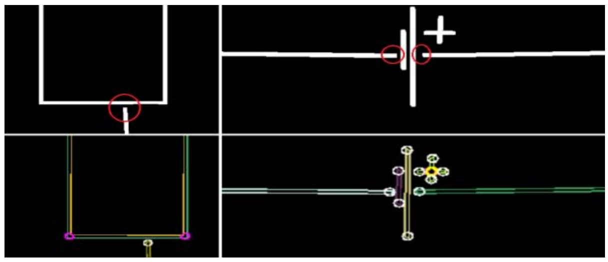

Figure 6. Disconnections between symbols and connecting line (top) with corresponding contours and end points (bottom)

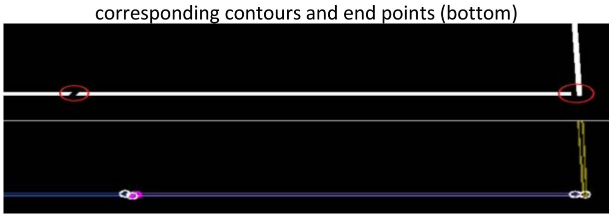

Figure 7. Breaks on connecting line (top) with corresponding contours and end points (bottom)

D. *Recognition candidate lines for voltage sources*

An ideally drawn voltage source symbol consists of two unequally long, parallel, straight lines and a plus sign located at one end of the longer line. In the skeleton image, we look for pairs of end points that are connected by as straight as possible skeleton line. Such lines are candidate-lines for the voltage source (Figure 5).

Algorithm 1. Recognition of end point pairs that form candidate lines for voltage sources.

Input: contour set $\boldsymbol{C}$, skeleton $\boldsymbol{S}$
Output: list of voltage source symbol line-candidates $\boldsymbol{L}$

for each contour $\boldsymbol{c}$ from $\boldsymbol{C}$ **do**
  for each endpoint $\boldsymbol{ep_1}$ from $\boldsymbol{c}$ endpoints **do**
    passing through $\boldsymbol{S}$
    calculate associative set $\boldsymbol{N}(\boldsymbol{point}, \boldsymbol{distance})$ of distances to nearest points to point $\boldsymbol{ep_1}$;
    for each $\boldsymbol{point}$ from $\boldsymbol{N}$ **do**
      calculate $\boldsymbol{match}$ for $\boldsymbol{ep_1}, \boldsymbol{point}, \boldsymbol{distance}$;
      if $\boldsymbol{match}$ greater than $\boldsymbol{bestmatch}$ **then**
        $\boldsymbol{ep_2} = \boldsymbol{point}$;
      end if
    end for
    add line-candidate $(\boldsymbol{ep_1}, \boldsymbol{ep_2})$ to $\boldsymbol{L}$
  end for
end for

In Algorithm 1 you iterate through all contours and take one endpoint at a time within a contour. For each endpoint, we first look for the nearest endpoints connected by the skeleton. Distances between endpoints are calculated by counting pixels while traversing through skeleton. Once visited, the pixels are marked so that we never visit the same pixel twice. If there are more than one non-zero and non-marked neighbor pixels the counting path will be split in two separate directions. Counting ends when endpoint pixel is reached. In addition, "air" distance between endpoints is calculated too. Distance through the skeleton divided by "air" distance gives us ratio which is used to determine how "flat" the line is. Threshold for this ratio is 1.3 and anything above that is graded with 0. Score is calculated as a sum of distance through the skeleton and quotient representing the input image height divided by ratio. In other words, the closer the ratio is to 1 and the greater the distance through the skeleton, the better the score will be. Points that represent a flatter and longer line-connector represent better candidate-lines. If more than one point is found for a point with which it forms a potential candidate line, the point with the best grade is taken and added to the list of candidate lines.

Algorithm 2. Detecting contour of a symbol next to end point

Input: endpoint $\boldsymbol{ep}$, contour set $\boldsymbol{C}$
Output: contour $\boldsymbol{bc}$ which is a best match for endpoint $\boldsymbol{ep}$

for each contour $\boldsymbol{c}$ from $\boldsymbol{C}$ **do**
  find associative set of line-candidates $\boldsymbol{L}$ and child contours $\boldsymbol{CC}$ for contour $\boldsymbol{c}$;
  for each $(\boldsymbol{ep_1}, \boldsymbol{ep_2})$ from $\boldsymbol{L}$ **do**
    calculate $\boldsymbol{match}$for $(\boldsymbol{ep_1}, \boldsymbol{ep_2})$, $\boldsymbol{ep}$, $\boldsymbol{c}$;
    if $\boldsymbol{match}$ greater than $\boldsymbol{bestmatch}$ **then**
      $\boldsymbol{bc} = \boldsymbol{c}$;
  end for
  for each $\boldsymbol{cc}$ from $\boldsymbol{CC}$ **do**
    calculate $\boldsymbol{match}$ for $\boldsymbol{cc}$, $\boldsymbol{ep}$, $\boldsymbol{c}$;
    if $\boldsymbol{match}$ greater than $\boldsymbol{bestmatch}$ **then**
      $\boldsymbol{bc} = \boldsymbol{c}$;
  end for
end for

If a contour has two candidate lines perpendicular to each other, then it is a plus sign (Figure 5). The plus sign found will help to evaluate the candidate for voltage source.

The steps described in the sections E, F, and G are iteratively performed for each contour in the scheme contour list and repeated until at least one new, root contour is added to the list. In section E, we describe how to add new contours for the first time in the list.

E. *Recognition of voltage sources*

The score of a pair of candidate lines depends on the position they have towards each other. A better grade will be given to the pair whose: total length is larger, mutual distance is smaller, the angle at which the lines passing through the end points of the candidate lines intersect is smaller, ratio of their lengths is more regular and if a plus is found next to a longer line. Pairs of lines that have a score higher than the required threshold are recognized as the source. Candidate lines that do not have a pair with which they have a satisfactory grade are rejected and their endpoints are used further in recognition.

Lines of recognized voltage sources can be encompassed by the same contour, or with two different contours. In the case when these are different contours, and one of them already belongs to the scheme contour list, then the other contour is added to that list.

F. *Detecting disconnections between symbols and connecting lines*

If an end point is adjacent to a contour that represents a symbol, then it belongs to a connecting line that is not connected to the symbol. In Algorithm 2, we consider the lines of the voltage source and the contours that are children of the root contours as a potential symbol (Figure 6). First, we test the lines of the voltage sources. The closer the breakpoint is to the center of the line and the larger is the contour to which the line belongs, the better the score will be. Score is calculated as a sum of contour length (scaled by a parameter that depends on the input image height) and quotient representing the input image height divided by breakpoint-center distance. The candidate with the best grade, if one is found, is added to the contour list of the scheme together with his pair. It is important to note that only candidate lines for sources whose contours have not yet been added to the scheme contour list are considered. The children of the root contours are then tested. The grade will be higher if the distance to the contour is smaller and if the parent contour is higher in the hierarchy.

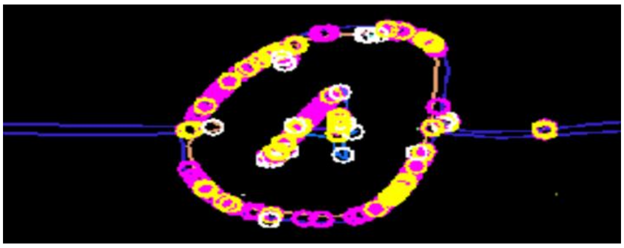

Figure 8. Hand-drawn ammeter with several misrecognized end points (white dots) that are discarded

G. *Detecting breaks on connecting lines*

In the case when a symbol is not found next to the end point, then we look if there is another end point nearby (Figure 7). The score for that pair of points is better if two points are closer to each other and contours encompassing points are larger. The grade grows if contours encompassing both points are in the scheme contour list. If contour of the point with the best score is not in the list, it is added. The best rated pair of points representing the break on the line as well as those points for which the neighbor symbol was found are not considered in the following iterations. Also, if no

endpoint is found near the current endpoint then it is a misrecognized endpoint and is not considered too. Incorrectly recognized endpoints appear most often in hand-drawn diagrams. An example is shown in Figure 8. If at least one new contour is added in steps E, F, and G in the scheme contour list, then another iteration starts with all steps (E, F and G).

H. *Recognizing symbols*

When all disconnections and contours that make up the scheme are found, the recognition of symbols begins. The symbols that are recognized are: resistor, current source, ammeter, and voltmeter. Voltage sources have already been recognized as a "skeleton disconnecting" type. Recognition of other symbols is based on the hierarchy of contours. By visiting children of all root contours, accordingly to their shape, we determine whether it is a symbol or not. An example of a root contour with its children is shown in Figure 9. If a child

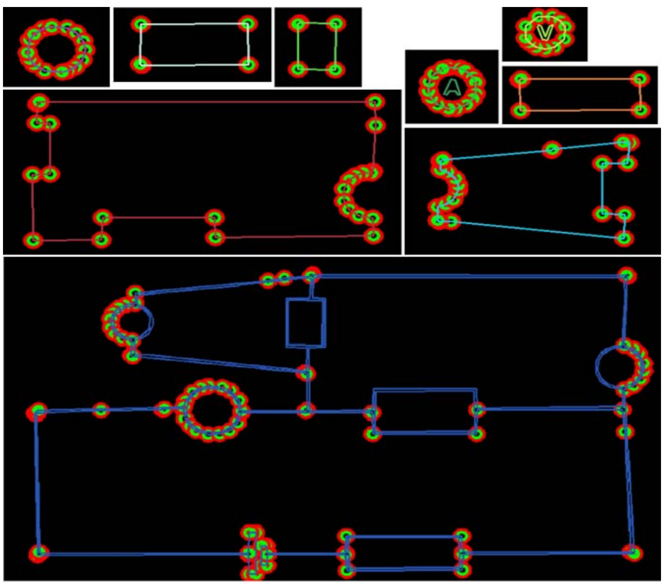

Figure 9. Root contour (bottom) with children (top)

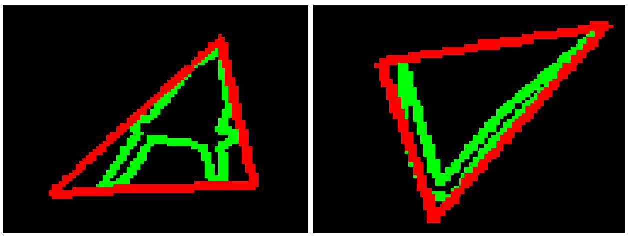

Figure 10. The smallest triangle bounding children of contours that have a circular shape

contour has rectangular or circular (elliptical) shape, then it is a symbol. If a child contour has some other shape, we neglect it. The number of points in a contour and contour area are used to classify its shape. Contours shaped like a rectangle have 4 points and they are recognized as resistors. Circular contours are recognized by comparing their shape and area with a minimum enclosing ellipse (OpenCV's *fitEllipse* function is used to get this ellipse).

We recognize a circular shape without a further child as a current source. If the child exists, then we are looking for the smallest bounding triangle for that contour (Figure 10) and, depending on the direction of the triangle, we conclude about the letter. If the triangle is directed upwards, then it is a letter A or an ammeter, otherwise, it is a letter V or a voltmeter.

I. *Detecting nodes in circuit*

When all symbols are found, we create a new binary image in which each whiteness represents a node - an area of equal electric potential. Such image is created by keeping only those whites in the initial binary image that represent parts of the scheme. Then by deleting all the symbols from the scheme we get separate whites (Figure 11). The remaining contours in such an image represent nodes. By skeletonizing new binary image, we obtain endpoints that are used to assign pairs of nodes to which

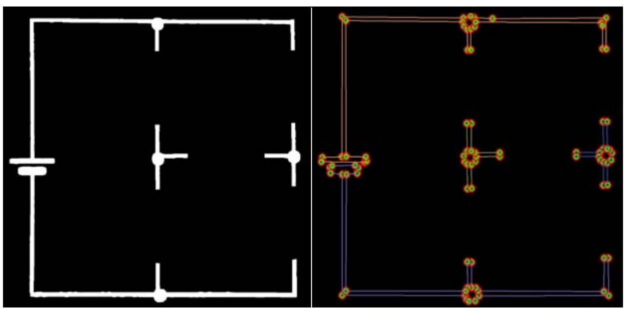

Figure 11. Binary image with deleted symbols and whites representing nodes (left) and node contours (right)

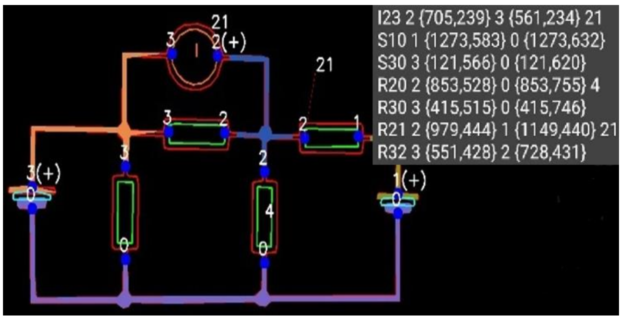

Figure 12. Example of a schema with recognized elements and nodes (topology) with a netlist record

symbols (two terminal elements) belongs.

J. *Creating netlist model*

Once all the schema symbols and nodes are recognized, it is possible to generate a model that is written in a standardized netlist format. Each line in the record represents one element in the schema and consists of one letter representing the element designation, two integers denoting the nodes to which the element is connected, and optionally a parameter describing the value of the element. The element designation can be: "R" for the resistor, "S" for the voltage source, "I" for the current source, "A" for the ammeter, and "V" for the voltmeter (Figure 12). We have expanded the record to include graphical information about the node coordinates to make it easier to redraw the schema. The last parameter in the record is optional and represents numerical element value.

K. *Assessment of recognition accuracy*

We used a set of test images to assess the accuracy of the recognition. We consider a test as successful if all symbols and the topology of the scheme are recognized. 40 images have been used for testing. 20 computer-drawn images were taken from printed books, and the other 20 images contained hand-drawn diagrams of DC circuits (Figure 13). Most of the test images have had some disconnections and irregularities we mentioned earlier in the paper. There were 18 successfully recognized computer-drawn images, while 12 were successfully recognized by hand-drawn schemas. When we considered situations where at most one symbol could be incorrectly recognized, then all computer-drawn images were accurately recognized, while hand-drawn ones were 16. The most common recognition problem occurs with a voltage source that is not correctly connected with connecting lines. In this case, it is possible that the two lines representing voltage source can be recognized as connecting lines that belong to the same node. Less commonly, the problem occurs with a resistor that is more ellipse-like than a rectangle. Such resistor can be recognized as a current source. Recognition performance could be increased by finding better parameters during the evaluation of voltage source candidate endpoints and evaluation of disconnections near symbols or adjacent endpoints.

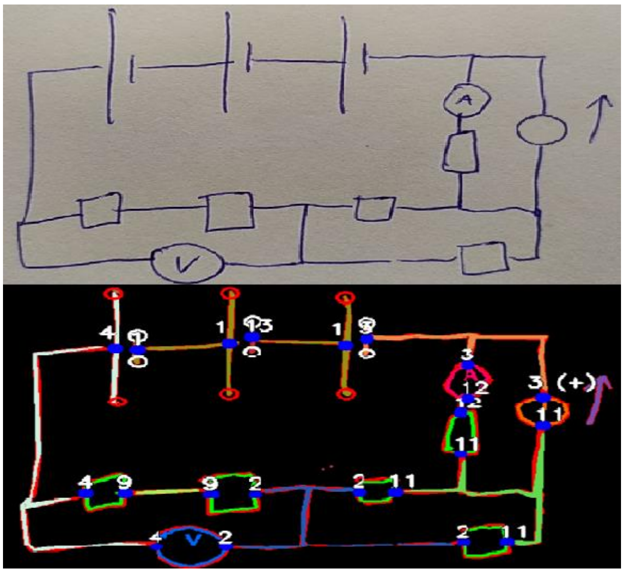

Figure 13. Example of a hand-drawn test scheme with intentionally induced disconnections and line breaks

III. CONCLUSION

The scheme recognition process begins with the preparation of a photograph leading to a binary image of the scheme. Then, the symbols of the scheme and the connection between them are recognized from the binary image by a series of developed algorithms, and finally, a formal model suitable for further analysis is built.

Building a practical and functional application for automated electrical circuit diagram recognition and interpretation is a large project that would be useful to students and professors in the process of learning and acquiring skills in electrical circuit analysis.

Unlike other approaches, our approach to recognizing and interpreting is predominantly based on information carried by skeletons and hierarchically organized contours related to electrical circuit diagram. Although it is desirable to have a high-quality photo of a printed or hand-drawn scheme, the algorithm has a certain level of noise tolerance. Symbol rotation and scaling do not affect recognition too. Structurally, circuit diagrams belong to a family we call network diagrams. We have demonstrated that automated interpretation of images containing such a type of diagram can be efficiently done by analyzing the end points of skeletons and the relationships between contours of the diagram.

LITERATURE

[1] M. Štorga, "Android application for automated recognition and analysis of photographed electrical circuit schematic diagrams", *Master's degree thesis (in Croatian)*, Faculty of Electrical Engineering and Computing, University of Zagreb, 2020.

[2] B. Edwards and V. Chandran, "Machine recognition of hand-drawn circuit diagrams," 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings, vol.6, pp. 3618-3621, Istanbul, Turkey, 2000.

[3] Lakshman Naika, R., R. Dinesh and S. Prabhanjan. "Handwritten electric circuit diagram recognition: an approach based on finite state machine." International Journal of Machine Learning and Computing 9 (2019): 374-380.

[4] Chin-Shyurng Fahn, Jhing-Fa Wang, Jau-Yien Lee, A topology-based component extractor for understanding electronic circuit diagrams, Computer Vision, Graphics, and Image Processing, Volume 44, Issue 2, 1988, Pages 119-138

[5] P. De, S. Mandal and P. Bhowmick, "Identification of annotations for circuit symbols in electrical diagrams of document images," 2014 Fifth International Conference on Signal and Image Processing, Bangalore, India, 2014, pp. 297-302

[6] OpenCV, https://opencv.org/, 12th January 2021.