# Identification of Annotations for Circuit Symbols in Electrical Diagrams of Document Images

Paramita De
Computer Science & Engineering
Haldia Institute of Technology
India
Email: paramitade13@gmail.com

Sekhar Mandal
Computer Science & Technology
BESU Shibpur
India
Email: sekhar@cs.becs.ac.in

Partha Bhowmick
Computer Science & Engineering
IIT Kharagpur
India
Email: bhowmick@gmail.com

***Abstract*—Though annotations are integral part of symbols in drawings, due attention is yet to be given for their identification, interpretation, and storage. A reconstructed drawing from the vector format is thus deficient in the complete description of the original, and hence requires costly and time-consuming manual intervention. This paper presents a method for segmentation and identification of annotations associated with electrical symbols in a circuit diagram, which may be used with the vectorizer to make it complete. 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 particular symbol. Morphological operations are used in identification phase. Finally, an efficient OCR is used to get the numerical values of the symbols along with their units. The performance of the algorithm is tested on a variety of images with ample variations in annotation. Some of the results are presented in this paper to demonstrate its efficiency.**

***Index Terms*—Electrical circuit, electrical symbols, text annotation, OCR, vectorization.**

## I. Introduction

Electrical engineers, architects, and mechanical engineers use different standard symbols in their drawings to convey useful information. Systems that can convert images of aforesaid drawings into vector form are in high demand today. A vectorized representation has many advantages like reduced storage space and ease of maintenance. Such a representation can readily be used for editing, browsing, indexing, and filing of document images [7], [8], [9]. Exclusive systems have been designed, therefore, over the last few decades, e.g., image and diagram extraction [3], [4], logo detection [1], [6], etc. An overview of these, along with their performance evaluation, may be seen in [8].

The symbols present in an electrical drawing are usually oriented horizontally or vertically, and associated with some annotations that mainly represent the values of the electrical components along with their units. The annotations that are associated with an active component (transistor, diode, etc.) in a diagram represent the type of the component. In an architectural drawing, an annotation is often used to represent the length of a line segment. Hence, annotations make an integral part of an engineering drawing and have immense importance in attaining its completeness.

Electrical circuit symbol recognition is a part of vectorization of electrical drawings. Unfortunately, annotations are not identified, interpreted, and stored along with the vectorized drawings. The reconstructed drawing from vector format is thus deficient in information in comparison with the original drawing. This motivates us to identify the annotations present in an electrical drawing. To the best of our knowledge, there is no literature till date for identification of annotations associated with electrical symbols.

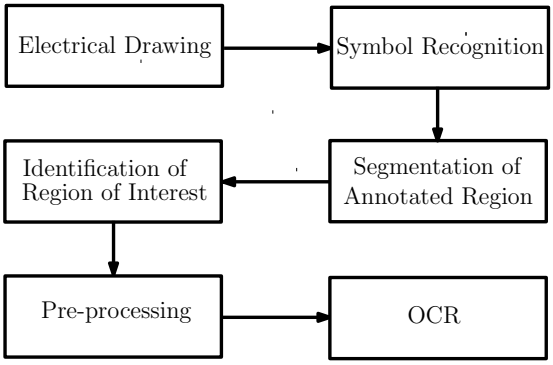


Fig. 1. Schematic flow of annotation identification.

The proposed method first segments the text region present around each electrical symbol. Next, the segmented text regions are processed using *morphological operations* to identify the annotation associated with that symbol. After necessary preprocessing, an efficient *classifier-tree-based OCR* is used to get the numerical values of the symbols along with their units. Figure 1 shows the schematic flow of the proposed method.

The organization of this paper is as follows. Section II provides the description of selecting or segmenting the interested text area. Section III describes the procedure for annotation identification. Preprocessing and character recognition are described in Section IV and Section V respectively. Section VI details the experimental results, and the conclusion is given in Section VII.

## II. Segmentation

All the electrical symbols are first recognized from the scanned image of the electrical circuit diagrams using the technique proposed in [2]. For each symbol, its type, orien-

978-0-7695-5100-5/13 $31.00 © 2013 IEEE
DOI 10.1109/ICSIP.2014.53

tation, and the coordinates of the bounding box are used by us for annotation identification. As annotations in electrical circuits are found to have wide variations, we have made some observations based on the premise that the width and the height of a symbol are given by the respective horizontal and vertical dimensions of its bounding box. The observations are as follows.

### A. Observation

- An annotation is placed on the left, or on the right, or on both the sides for a vertically oriented symbol (Figure 2(b, c)); for a horizontally oriented symbol, it is on the top or on the bottom of the symbol. Further, the annotated text lies very close to the corresponding symbol (Figure 2(a, d)).
- For a horizontally oriented symbol, the annotation covers the entire horizontal span of the symbol, and the width of the bounding box enveloping the annotated text region is greater then the width of the corresponding symbol (Figure 2(a, e)). But for a vertically oriented symbol, the width of the bounding box enveloping the annotated text region is significantly greater than the height of the symbol (Figure 2(b)).
- Annotated texts may appear on both sides of a symbol, but the region of interest (comprising numerical values and units) generally appears on one side of the symbol (Figure 2(c, d, e)).

### B. Steps for segmentation

Based on the aforesaid observations, text annotation is segmented for each symbol, $S_i$. Let the bounding box of $S_i$ be defined by $x = x'_i$, $x = x''_i$, $y = y'_i$, and $y = y''_i$, as the respective left, right, bottom, and top boundary lines. Then the following steps are executed for segmentation of annotations.

- **Step 1:** The width and the height of $S_i$ are computed as $w_i = x''_i - x'_i$ and $h_i = y''_i - y'_i$.
- **Step 2:** For each horizontally oriented $S_i$, a window of size (width$\times$height) $= 2w_i \times 2h_i$ is placed on the top and another on the bottom side of $S_i$. The text regions lying within these two windows, if any, are segmented as annotations.
- **Step 3:** A window of size $2h_i \times h_i$ is first placed on the left and then on the right side of a vertically oriented symbol, and the text regions lying within these windows, if any, are segmented.

An example is shown in Figure 3. The segmented output for a particular symbol may contain some text region, which is not a part of the annotation of that symbol (Figure 4). Such unwanted components are removed in the identification phase, as described next.

## III. Identification

The segmented output of annotated text region usually consists of symbol name, equality symbol (=), value along with the unit of measurement, and possibly some unwanted text. The goal of the identification phase is to detect the region that contains the value and the unit of the symbol. The steps are as follows.

- **Step 1:** The equality symbol is detected first. To detect it, connected component analysis is done. Height and width of each component are computed. The median width $\widetilde{w}$ of the components is also determined. A search is performed to detect two short line segments of equal length with complete vertical overlap. This search identifies the equality symbol in the annotated text region. The text region at the right side of the equality symbol provides the value and the unit of the corresponding circuit symbol (Figure 2(a, b)). If there is no text region at right side of the equality symbol, then the text line below it provides the value and the unit of the circuit symbol (Figure 2(f)).
- **Step 2:** If the segmented region consists of only the name of the symbol and its value along with the unit (Figure 2(c, d) and Figure 4(a)), then morphological closing operation is performed on the segmented region. The size of the structuring element is $\widetilde{w} \times 1$. It is observed that the gap between two words in the annotated text region is greater than $\widetilde{w}$. The output is a set of word blobs, as shown in Figure 5(b). For this particular case, the name of the symbol appears in one side (i.e., top or left side) and the value along with the unit appears on the other side. A search is performed on both the sides of the symbol to count the number of word blobs on each side. The text region that contains more than one blob is considered as the region of interest. If the region of interest consists an unwanted text component (Figure 4(b)), then the output of the closing operation contains more than two word blobs (Figure 4(c)). Small components out of these are treated as noise and removed. After removing these unwanted components, a search is performed to detect two consecutive word blobs with almost complete horizontal overlapping. The left one between these two signifies the value and the right one, the unit (Figure 4(d) and Figure 5(c, d)).

## IV. Preprocessing

The region of interest may contain some touching and some broken characters. Hence, before applying OCR, preprocessing is done to resolve such problems.

### A. Touching Character Problem

In some cases, two characters get touched with each other. For example, in Fig 6(a), the two characters 'k' and 'Ω' representing the unit (kilo-ohm) of a resistor are in touch with each other. To identify the touching characters, the median width ($\widetilde{w}$) of the components in the region of interest is computed. If the width of a component is greater than $1.5 \times \widetilde{w}$, then it is assumed that the two characters are mutually touching and hence forming a larger component. The height $h$ of each component is also computed. To separate the touching characters, the following steps are executed on each component, $C$.

(i) The middle position and the width of $C$ is determined.

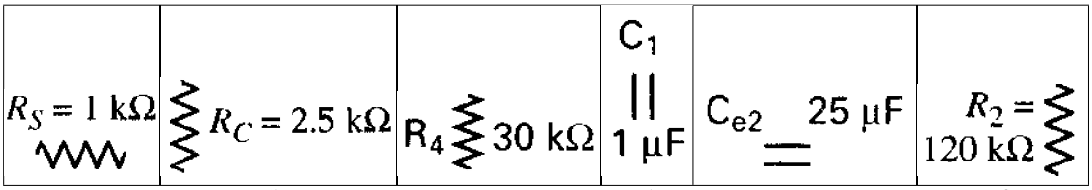


Fig. 2. Different annotation styles that are taken care of in our segmentation procedure.

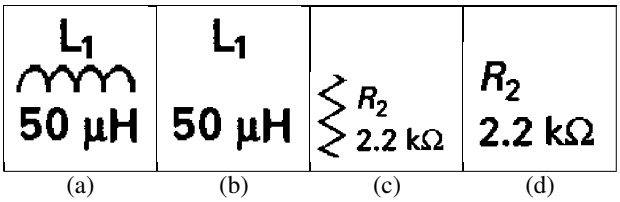


Fig. 3. Result of segmentation. Input: (a), (c). Output: (b), (d).

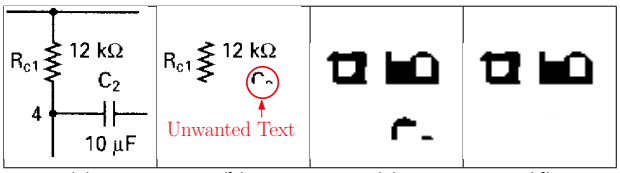


Fig. 4. Unwanted text removal. (a) Input. (b) Unwanted text crept in after segmentation. (c) Result after closing operation. (d) Result after deletion.

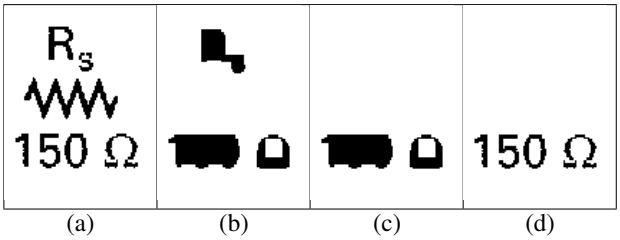


Fig. 5. Identification of text. (a) Input. (b) Output after Closing. (c) Deletion of single component. (d) Actual value and unit of the symbol.

(ii) A window $W$ of size $\frac{\widetilde{w}}{4} \times h$ is placed in the middle position of $C$.

(iii) Successive vertical scans are performed within $W$ from top to bottom of $C$; for each scan, the number of white pixels, $n_{\mathrm{w}}$, is computed, until the scan hits a black pixel (Fig. 6(a)). Then, for all those scan lines with $(h - n_{\mathrm{w}}) \leq \frac{h}{6}$, cuts are made to disconnect the touching characters (Fig. 6(b)).

### B. Broken Character Problem

The width $w$ of the each component is compared with the median width, $\widetilde{w}$. If $w \leq 0.7 \times \widetilde{w}$ (Figure 6(c)), then the horizontal distance $d_h$ between its right and left neighbors are computed. The neighbor and the component under test for which $d_h \leq 3$, are considered as a single component. The horizontal scan line between two components that satisfy the condition $d_h \leq 3$, is filled with black pixels (Figure 6(d)).

Similarly, to check a vertical break, the height $h$ of each component is compared with the median height, $\widetilde{h}$. If $h \leq 0.7 \times \widetilde{h}$, then the vertical distance $d_v$ between the component $C$ under test and its nearest vertically overlapping neighbor $C'$ is computed. If $d_v \leq 3$, then $C$ and $C'$ form a single character. The vertical scan line between $C$ and $C'$ that satisfies $d_v \leq 3$, is filled with black pixels.

## V. Character Recognition

An efficient OCR system is designed to recognize only the numerals `0` to `9` and a set of non-numeric characters {`F`, `H`, `V`, `k`, `M`, `m`, `p`, `n`, Ω, µ}, which are used to represent the units of electrical symbols. As our set is relatively small compared to the complete English alphabet and it contains a few Greek characters, we have built this OCR system exclusively to make the recognition fast, efficient, and robust.

A set of topological features is used for classification of characters. Based on these features, two *tree classifiers* are designed, one corresponding to non-numeric characters and another for numerals (Figure 7). At each node of the tree, a decision is taken on the basis of presence or absence of these

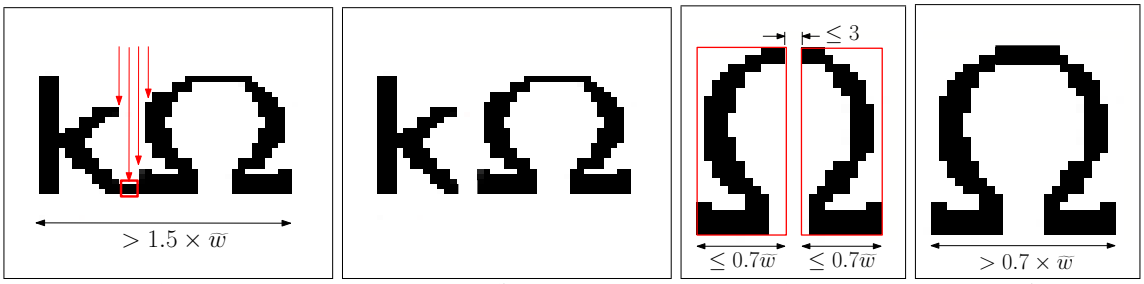


Fig. 6. Separation of touching characters and joining of broken characters. (a) Before separation. (b) After separation. (c) Before joining. (d) After joining.

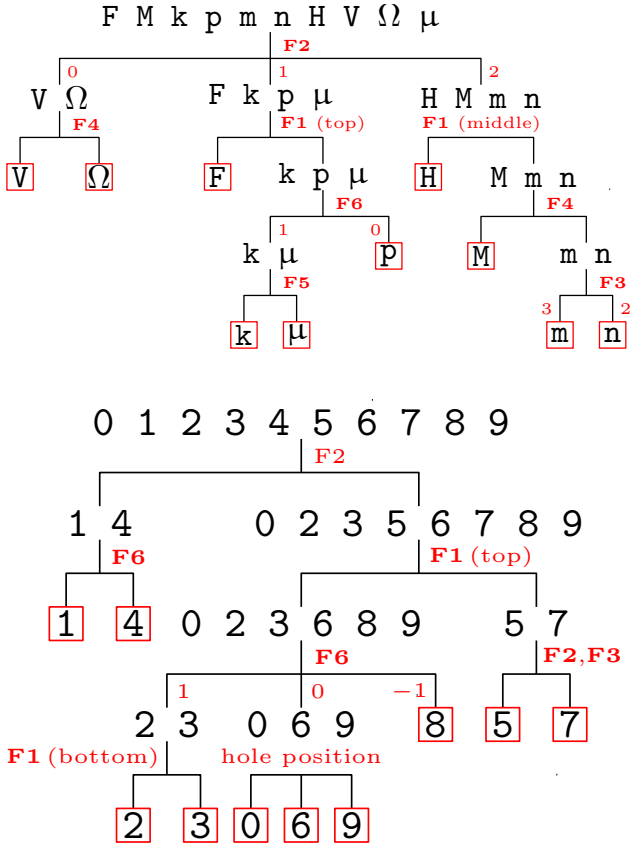


Fig. 7. Classifier trees for recognition of non-numeric characters (top) and numerals (bottom).

features, as briefed below.

(F1) **Horizontal bar:** Horizontal bar is a horizontal line segment having length nearly equal to the width of the character. Characters like F and H have horizontal bars in top and middle, respectively, which thereby make them distinguishable from k or M.

(F2) **Vertical bar:** Vertical bar is a vertical line segment of length nearly equal to the height of the character. Characters like H, M, m, and n have two vertical bars on their left and right sides, whereas characters like F, p, k, and μ have only one vertical bar on the left side. Thus, the former class is distinguishable from the latter, and each of them is again differentiable from the characters having no vertical bar, e.g., V and Ω. In some cases, a symbol is slanted, e.g., $\mu$ instead of μ; in that case, it does not have this feature, and hence the features **F3**, **F4**, and **F5** are used. For some numerals like 5 the presence of vertical bar of length half of its height is considered.

(F3) **W-to-B Transitions:** Total number of white-to-black transitions is recorded by a horizontal scan from the left to the right boundary through the middle of the bounding box of a connected component. Depending on the number of transitions, characters are classified. Characters like m and n are well-distinguishable by this feature.

(F4) **White Pixel Count, Vertical:** A vertical scan is performed in the middle of a component from top to bottom (or bottom to top) until it hits a black pixel. For the characters like V, M, and Ω, this count is significantly large compared to others.

(F5) **White Pixel Count, Horizontal:** A number of horizontal scans are performed from left to right and top to bottom for each component. Each horizontal scan line encounters four transitions in the sequence of WB, BW, WB, and BW, where WB denotes white-to-black and BW denotes black-to-white. For each scan, the number of white pixels, $n_w$, between two consecutive BW and WB transitions, is counted. For character like V, $n_w$ decreases as the scan line moves from top to bottom; whereas, for characters like μ, $n_w$ remains unchanged.

(F6) **Euler Number:** As each character is a connected component, we measure it as one minus the number of holes [5]. A character like p has one hole, and thus distinguishable from k or μ. For two characters having equal number of holes, the hole position is considered. Numerals like 0, 6, and 9 are resolved by this.

A decimal point, if present in the value part, is recognized easily, as its height and width are significantly less than the corresponding median values, i.e., $\tilde{h}$ and $\tilde{w}$.

## VI. Experimental Results

All our experiments have been carried out in Ubuntu 12.04 LTS on a computer with Intel (R) core (TM) 2 Duo CPU T6400 2 GHz. The algorithm is implemented in C language

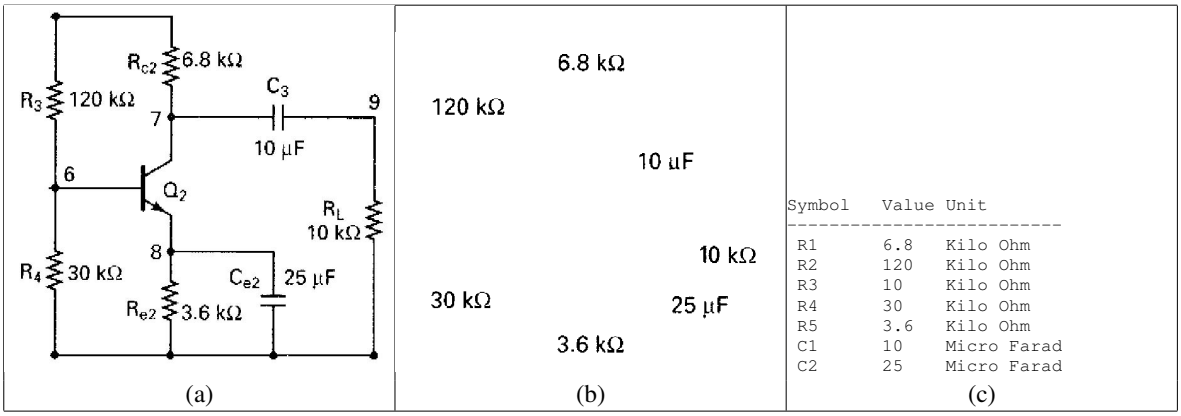


Fig. 8. Result by our algorithm. (a) Input. (b) Recognized annotation regions corresponding to value and unit. (c) Identified symbols with their values and units.

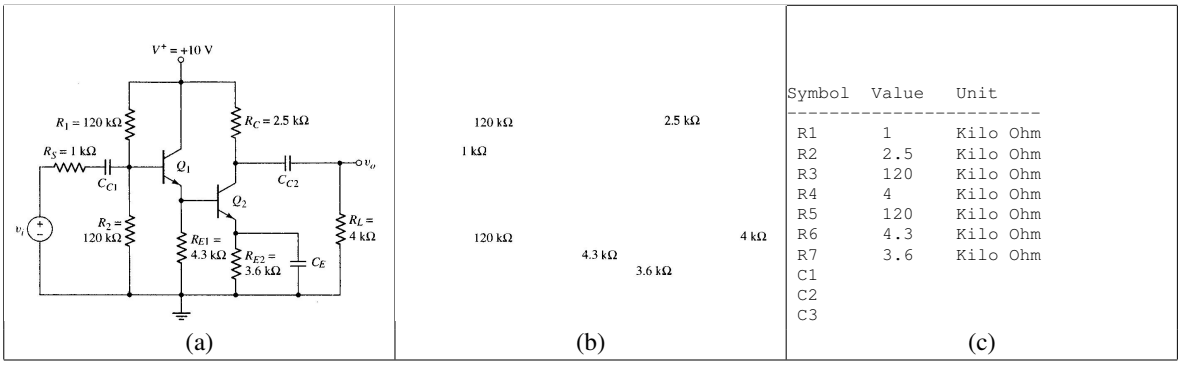


Fig. 9. Result by our algorithm on another image. (a) Input. (b) Recognized annotation regions corresponding to value and unit. (c) Identified symbols with their values and units.

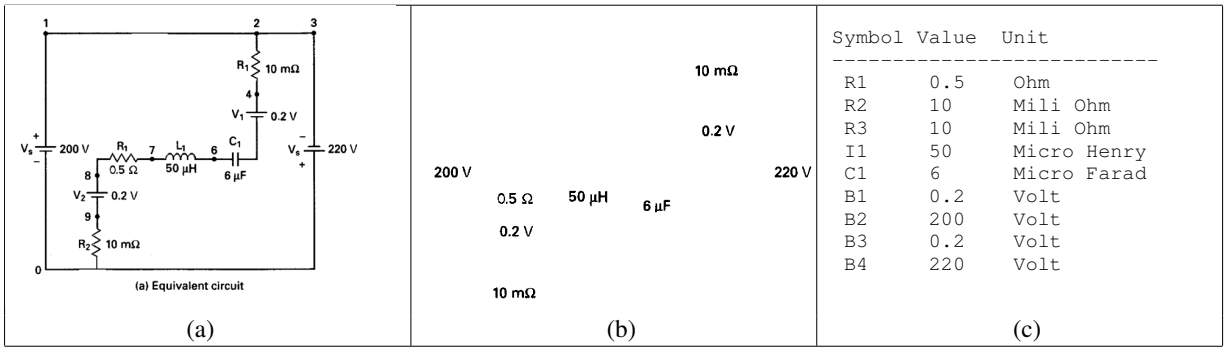


Fig. 10. Result by our algorithm on another image. (a) Input. (b) Recognized annotation regions corresponding to value and unit. (c) Identified symbols with their values and units.

and tested on various document images scanned from different books. Results for some of these images are presented in this paper.

Figure 8(a) shows an image where the annotated text for each symbol appears on both sides of that symbol. The region of interest for each symbol is extracted correctly, as shown in Figure 8(b). Note that in the OCR output, shown in Figure 8(c), the symbol names are different from those in the original image, since we report these names as per the symbol-recognition algorithm [2] (Sec. II). Here, the value and the unit are recognized by our OCR. In Figure 8(a), in the unit part of resistor $R_3$ and $R_4$, the symbol Ω is broken; after preprocessing (Sec. IV), the broken parts are repaired and correctly recognized by our algorithm.

The annotations presented in Figure 9(a) differ in the way of their representation from those in Figure 8(a). Here, the regions of interest appear on the right side of the equality symbol or below it. The proposed algorithm properly segments these regions and correctly reports the values and the units (excepting three capacitors having no values), as shown in

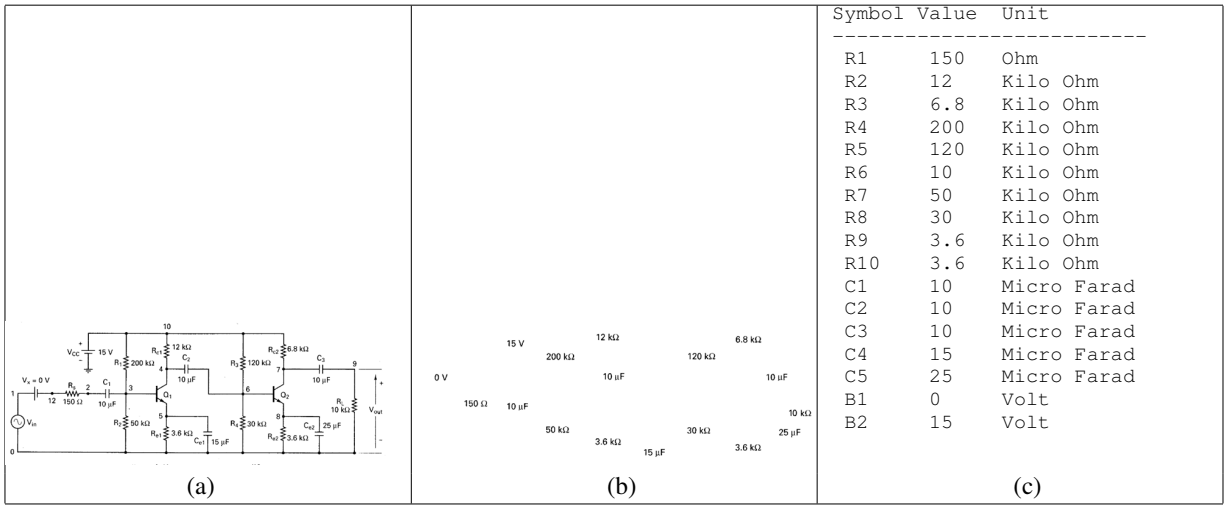


Fig. 11. Result of our algorithm. (a) Input (b) Annotations of symbols (region of interest) (c) Identified symbols with their value and unit

Figure 9(b, c). In Figure 9(a), the symbols k and Ω have touched each other. These two symbols are used to represent the units of the resistor $R1$, $R_{E1}$, and $R_{E2}$. The preprocessing successfully separates out these two touching symbols (see Figure 9(b)). In Figure 9(a), there are three capacitors. They do not have values along with units, and only their names are there; that is, there is no region of interest. Hence, in the corresponding output of segmentation (Figure 9(b)), there is no annotated text regions corresponding to these three capacitors. Figures 10 and 11 show two more sets of experimental results obtained by the proposed method.

The CPU time depends on the number of symbols as well as the number of annotations given and the style of annotation. For the images of Figure 8 and Figure 9, CPU times are 0.140 seconds and 0.237 seconds respectively.

## VII. Conclusion

The proposed segmentation and identification algorithm based on morphological operations successfully extracts the annotations linked with the symbols from the paper based electrical drawing. The classifier trees can correctly recognize the small set of characters and numerals as well. The efficacy and efficiency of the proposed method is also endorsed by testing on a variety of document images available in our database with different font sizes used for annotations. The result of proposed algorithm may include the information extracted from the annotations for a meaningful and useful vectorization.

## References

[1] J. Chen, M. K. Leung, and Y. Gao. Noisy logo recognition using line segment hausdorff distance. *Pattern Recognition*, 36(4):943–955, 2003.

[2] P. De, S. Mandal, and P. Bhowmick. Recognition of electrical symbols in document images using morphology and geometric analysis. In *IEEE Conf. Image Information Processing*, ICIIP, pp. 1–6, 2011.

[3] R. P. Futrelle et al. Extraction, layout analysis and classification of diagrams in pdf documents. In *Proc. ICDAR'03*, pp. 1007–1014, 2003.

[4] J. Li, A. Najmi, and R. Gray. Image classification by a two-dimensional hidden Markov model. *IEEE Trans. Sig. Proc.*, 48(2):517–533, feb 2000.

[5] X. Lin, J. Ji, and Y. Gu. The Euler number study of image and its application. In *IEEE Conf. Ind. Electronics & Appl.*

[6] T. D. Pham. Unconstrained logo detection in document images. *Pattern Recognition*, 36(12):3023–3025, 2003.

[7] S. Tabbone. Indexing of technical line drawing based on f-signature. In *Proc. ICDAR'01*, pp. 1220–1224, 2001.

[8] Y. Wang, I. T. Phillips, and R. M. Haralick. Document zone content classification and its performance evaluation. *Pattern Recognition*, 39(1):57–73, 2006.

[9] L. Wenyin. On-line graphics recognition: State-of-the-art. In *Proc. GREC'03*, LNCS: 4958, pp. 291–304, 2004.