Multimedia Tools and Applications (2020) 79:31353–31373
https://doi.org/10.1007/s11042-020-09570-6

Offline hand-drawn circuit component recognition using texture and shape-based features

Soham Roy[1] · Archan Bhattacharya[1] · Navonil Sarkar[1] · Samir Malakar[2] · Ram Sarkar[3]

Received: 13 May 2020 / Revised: 1 August 2020 / Accepted: 6 August 2020 /
Published online: 20 August 2020

Abstract

Circuit diagram is the very foundation of electrical and electronic sciences. A circuit diagram consists of various symbols called circuit components that specify the functionality of that circuit. Every day-to-day gadgets that we use are made up with a number of electrical/electronic circuits to play out their particular tasks. Till date circuit designers have to physically enter all data from the hand-drawn circuits into computers, and this procedure requires some investment in terms of time and carries mistakes with high likelihood. To this end, in this paper, we propose a method that relaxes this constraint by introducing a method for recognition of hand-drawn electrical and electronic circuit components, with both analog and digital components included. In the proposed method, the pre-processed images of circuit components are used for training and testing a recognition model using a feature set consisting of a texture based feature descriptor, called histogram of oriented gradients (HOG), and shape based features that include centroid distance, tangent angle, and chain code histogram. In addition, the texture based feature, being large in number compared to others is optimized using a feature selection algorithm called ReliefF. Classification of components is done by using sequential minimal optimization (SMO) classifier. The proposed method has been evaluated on a dataset of 20 different circuit components with 150 samples in each class. The experimental outcome shows that the proposed approach provides average 93.83% accuracy on the present database. We also compare our method with some of the state-of-the-art methods and we see that our method outperforms these methods.

Keywords Hand-drawn circuit components · Texture based feature · Shape based feature · Sequential minimal optimization · Feature selection

✉ Samir Malakar
malakarsamir@gmail.com

Extended author information available on the last page of the article

1 Introduction

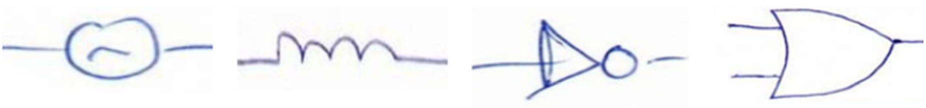
A circuit diagram is a graphical representation of an electrical/electronic circuit that contains symbols of circuit component in order to represent the electrical components and connectivity between them designed for a specific purpose. Concerned people can easily recognize these electrical components using their knowledge and experience. Thereafter, it needs manual intervention to enter the hand-drawn components into computer through simulating software like Multisim and CircuitMaker to perform the related processes like simulation and studying circuit parameters. As the complexity of the diagrams increases, this process becomes more and more cumbersome and hectic. Thus in order to analyze the hand-drawn circuits automatically, both component detection and recognition are very much essential. It helps to solve practical problems of layout and simulation through analysis, design and implementation of circuit.

In the present scope of the work, we consider 20 most commonly used electrical and electronics circuit components for the purpose of automatic recognition from its optically scanned version with image processing and machine learning approaches. The components are drawn by engineering students, faculty members and research scholars. In this context, it is to be noted that the recognition of circuit components is not an easy task as several analog and digital components have almost similar structure and shape. Complexity gets added when we consider the hand-drawn circuits because of the variation in drawing style of different individuals. Besides, erratic drawing style makes the problem more challenging to the researchers, since sometimes people follow random pen-up and pen-down during drawing; thereby resulting in unrecognizable circuit components even for the experienced persons of this field. Moreover, low quality of paper and ink, or the natural noise due to aging or even noise appearing during image acquisition are the typical problems of this domain. Fig. 1 shows some sample circuit symbols taken from the database prepared under the present work.

2 Literature survey

Technological advancements have huge impact in every field of our life. Researchers are continuously striving towards developing automated systems; thereby increasing human-computer interaction. The research problem we consider here is no different as evident from the works found in the literature (though few in number) as described below.

Dewangan and Dhole [12] use a strategy utilizing K-nearest neighbor (KNN) classifier to build a system that directly reads the electrical circuit components from a hand-drawn circuit images. Different shape and geometrical features like area, length of major axis and minor axis, centroid position and orientation of the circuit components are considered. A good recognition accuracy (90%) is observed by considering only 10 analog circuit components.



(a) AC source (b) Inductor (c) NOT gate (d) OR gate

Fig. 1 Some sample hand-drawn circuit components. (a) AC source, (b) Inductor, (c) NOT gate, (d) OR gate

In their work, many major circuit components like digital gates and transformers are not included and the number of circuit components used per class is not specified.

In another work, Feng et al. [14] rely on a two dimensional dynamic programming (2D-DP) technique allowing symbol hypothesis generation, which can correctly segment and recognize interspersed symbols. In addition, as discriminative classifiers usually have limited capability to reject outliers, some domain specific knowledge is included to circumvent those errors due to untrained patterns corresponding to erroneous segmentation hypotheses. With a point level online measurement, the experiment shows that the proposed approach is able to achieve an accuracy of more than 90%. However, very few components (only 9 classes) are used and the components are drawn with digital pen on digital surface.

A system of offline circuit recognition and simulation using digital image processing is proposed by Angadi and Naika [2]. The model consists of four stages viz., pre-processing, segmentation of components, support vector machine (SVM) based circuit component classification and simulation stages i.e. the authors proposed a complete framework of circuit components analysis. In this work, different shape based features like average component height, inclination, entropy of the components are used. However, no information regarding number of components used, accuracy and size of dataset are available.

In another work, proposed by Dinesh [13], author try to recognize hand-drawn electronic components (analog only) using histogram of oriented gradients (HOG) based features and subsequently used SVM classifier. He tests his method on a dataset that contains 2000 isolated circuit component images and the proposed method yields 96.90% recognition rate for a 10-class problem. In this work, he use only some of electrical components while completely ignores all the digital components as well as important analog components like ammeter and voltmeter.

An artificial neural network (ANN) based model is used by Rabbani et al. [24] to make a system that can directly read the electrical symbols from a hand-drawn circuit image. The recognition process involves two steps: first step is shape based feature extraction and the second one is classification using ANN that uses a back propagation algorithm. The ANN is trained and tested with different hand-drawn electrical circuit component images. The results show that their proposal is viable but, the accuracy obtained is much lower and the dataset used is very small in size.

Naika et al. [18] propose another method for recognizing hand-drawn electronic circuit diagrams. This method first detects and classifies each component present in the hand-drawn circuit diagram. For the purpose of component recognition, they use a feature vector that is constructed by combining local binary pattern (LBP) and statistical features based on pixel density and, SVM is used as classifier. The test is done in a dataset of 1000 components in total, giving recognition accuracy above 92% but the only a few components (9 in number) are considered.

Liu and Xiao [20] use topology based segmentation method to segment circuit sketch, and classify each component using the Fourier descriptors as feature vector and SVM classifier to achieve a trainable electronic sketched circuit recognizer that has fast response time, high accuracy and easy extensibility to new components. An accuracy of over 90% is achieved for each component. However, only a few (5 in number) components are tested and the dataset is also very small (55 samples per class).

Approach considered by Sala [26] consists of the use of four hidden Markov models (HMMs) per symbol class to perform classification. Each symbol is segmented into a linear sequence of image features with the conditional probability between consecutive features in the sequence being modelled by a HMM. For testing, several HMMs with different number of states are trained per symbol class (around 60 per class) and model selection is performed using a validation set to decide for the best number of states for the HMMs of each class. The average accuracy achieved is around 90% but the dataset used is quite small and all the components are drawn by the same individual. Only 12 different components are taken into consideration.

Though not directly related to circuit components recognition, we come across some works that performs sketch symbol recognition (e.g., Deufemia et al. [11]), identifying hand-drawn graphics components from the textual parts (e.g., Avola et al. [3, 4]) in an online handwritten document using machine learning based approaches. The authors use discriminative features like entropy, band ratio, X scan, intersection and projection and SVM classifier [3] while in [4] they use original features like curvature and linearity along with the 6 methods of the previous paper [3] along with extreme machine learning (EML) classifier to perform classification of drawing symbols and texts. In another work, Deufemia et al. [11] propose a two-stage clustering based approach for labeling different types sketched symbols. In the first stage, the authors use latent-dynamic conditional random field (LDCRF) to analyze the features of unsegmented stroke sequences based on spatio-temporal information of the strokes, and then select the stokes of symbol part(s) using the contextual information. In the later stage, they group the previously labeled stokes into symbol labels by using a distance-based clustering algorithm that uses the geometric relationships among strokes.

3 Motivation and contribution

The above discussion reflects the fact that recognition of individual circuit components digitally with appreciable efficiency from offline hand-drawn diagrams is potentially a stepping-stone to paramount advancement in the digital analysis domain. Few notable advances have been made in this area previously, some of the earlier ones being constrained to being online [3, 4], which undeniably is a hassle in terms of cost and effort. A number of machine learning [12] and digital-image processing [2] based approaches have also set foot on this problem, contriving a promising outcome, but on the downside, compromising on variety and thus considering less complexity of components. Endeavour has been made in the direction of capitalizing on the utility of texture-based features such as HOG of digital image and subsequently feed that into a training model which is SVM classifier [13]. The accuracy obtained is highly encouraging, but the model does not incorporate a handful number of basic and important circuit components and the length of the feature dimension is high, thus leaving huge scope for improvement. Therefore, the need of framing a model for recognition of offline hand-drawn circuit components which takes into consideration a variety of important circuit symbols with notable complexity, and returns an outcome with considerable accuracy, is indeed noteworthy and important.

Considering the issues mentioned above we design a model to recognize hand-drawn circuit components satisfactorily. In a nutshell, the contributions of the present work are as follows:

- Use of texture as well as shape information to recognize the components.
- Optimal selection of the texture feature using a feature selection algorithm.

- Preparation of an in-house dataset containing frequently used analog and digital circuit components.
- Our model is tested on this dataset and it gives satisfactory result as compared to some of the state-of-the-art methods evaluated on the present dataset.

4 Proposed method

As already mentioned, our aim in this work is to recognize the analog and digital circuit components, which we frequently encounter. First we consider 5 standard texture based features namely HOG, gray-level co-occurrence matrix (GLCM), gray-level run length matrix (GLRLM), uniform local binary pattern (ULBP) and robust local binary pattern (RLBP) and select the best one out of these on the basis of experimental outcome. Then three shape-based features namely, centroid distance, tangent angle and chain code histogram (CCH) are considered to estimate the shape information of the components. As the feature length of the texture based feature is too large compared to the shape based features, they may lessen the overall effect of the shape-based feature. Besides, there may have some redundant and non-informative features in the HOG feature vector. Therefore, we select a certain number of useful texture based features by eliminating the redundant and non-informative ones by using a standard feature selection algorithm called ReliefF [17]. Finally, we combine all the features and proceed towards classification with a suitable classifier. The said steps of the proposed method are pictorially depicted in Fig. 2, and we describe all the steps in the following subsections one-by-one.

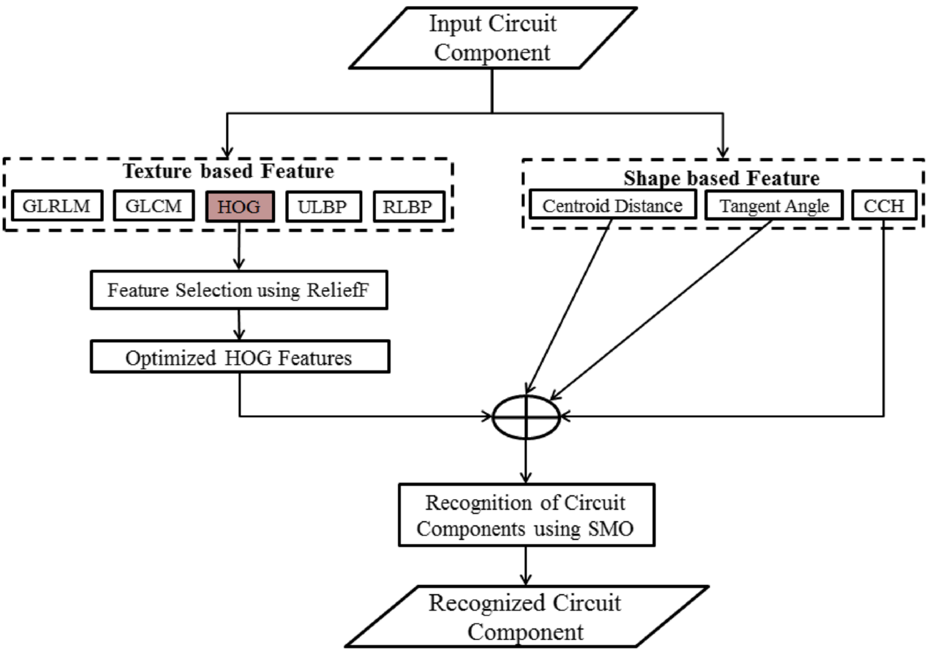


Fig. 2 Overall steps of our proposed circuit component recognition method. The highlighted texture based feature (i.e. HOG) has been selected experimentally

4.1 Database preparation

To evaluate any model, a realistic dataset is required as without proper data, performance of the underlying system could not be ensured. But to the best of our knowledge, no hand-drawn circuit component dataset on which we can test our model is publicly available. So we prepare an in-house dataset containing 20 different circuit components which are listed in Table 1. We collect 150 samples for each circuit component that are drawn by different individuals who happen to be engineering students, faculty members and research scholars in a pre-formatted datasheet similar to the works [7, 21]. In total, we make a dataset containing 3000 sample images of 20 different circuit components (150 per circuit component). Samples are collected in a pre-formatted datasheet and one such filled-in datasheet is shown in Fig. 3. Next, these datasheets are scanned using HP flatbed scanner with 300 dpi resolution and the drawn components are extracted automatically from the datasheets. The data are binarized with an appropriate technique as reported in [16] and noise pixels are removed using the method described in [6]. Some sample images are provided the link "https://github.com/Archan462/Codes".

4.2 Texture based feature extraction

Texture based features refer to the information carried by the spatial distribution of intensity levels in a given image. In our work, we initially use five texture based feature vectors namely HOG, GLCM, GLRLM, ULBP and RLBP but later we experimentally select the feature descriptor which works best for the said problem.

HOG [10] is a feature descriptor used in computer vision and image processing for the purpose of object detection and recognition. The local object appearance within an image can be described by the distribution of intensity gradients or edge directions [5]. The image is divided into small regions called cells and for the pixels in a cell, a histogram of gradient directions is determined. The total descriptor is the concatenation of all such histograms across the image. In our case, we experimentally determine the dimension of the circuit component images to extract the HOG feature descriptor. Figure 4(a) and Fig. 4(b) show samples circuit component image and its corresponding cells of HOG features respectively.

GLCM [22] is a feature descriptor that characterizes the texture of an image by calculating how often a pair of pixels with specific intensity values and in a specified spatial direction occur in the image. This information is stored in a matrix and the following 13 statistical measures are generally extracted from this matrix as the feature vector, *namely*, contrast, correlation, entropy, energy, homogeneity, variance, sum of variance, average, difference of

Table 1 The list of circuit components considered for recognition

Class #	Component name	Class #	Component name	Class #	Component name	Class #	Component name
1	AC source	2	Ammeter	3	AND gate	4	Capacitor
5	DC source	6	Ground	7	Inductor	8	NAND gate
9	NOR gate	10	NOT gate	11	NPN transistor	12	OR gate
13	PN junction diode	14	PNP transistor	15	Power supply	16	Resistor
17	Switch	18	Transformer	19	Voltmeter	20	Zener Diode

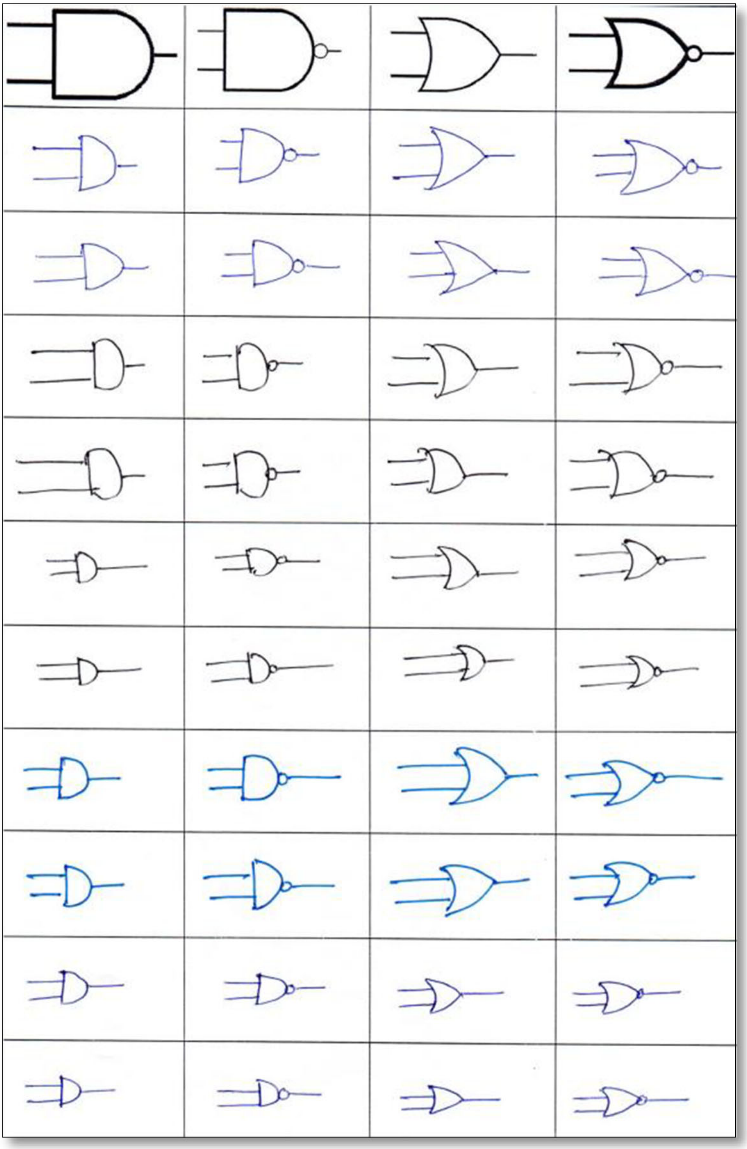
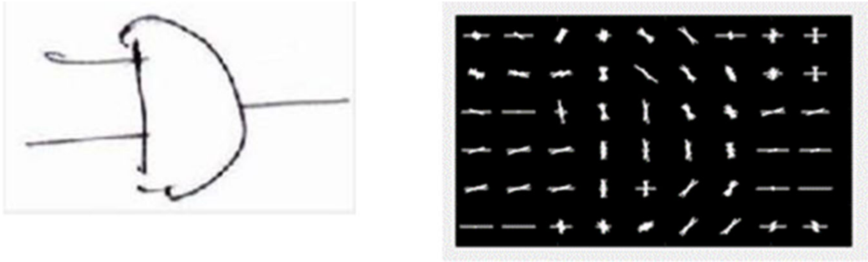


Fig. 3 A filled-in sample datasheet with circuit components drawn by different individuals

variance, Sum of entropy, difference of entropy, measure of correlation and inertia. Figure 5 represents the formation of the GLCM of the grey-level (3 intensity levels) image at the distance ‘d’ = 1 and the direction of 0°.

GLRLM [9] is a 2-D matrix from which the texture based features can be extracted on run length basis i.e. on the information provided by the consecutive number of pixels with a particular intensity value in a fixed direction. Each element of the matrix $P^{(i,j\,|\,\varnothing)}$ represents the number of elements j with intensity level i in a particular direction $\varnothing$.



(a) Original image (b) Representation of gradient directions

Fig. 4 Showing an input image with its gradient directions. (a) Original image, (b) Representation of gradient directions

LBP [23] is a feature descriptor that determines the surface texture by two complementary measures: local spatial patterns and gray scale contrast. It divides an image into several cells. A particular pixel in a cell is considered as the center pixel. Now the value of intensity of this pixel is compared with its 8 neighbors. If the value of the intensity of the neighboring pixel is greater than that of the center pixel, the number 1 is considered, else 0 is considered. The effective decimal value formed by the string of 8 '0's and'1's is considered. Then a histogram is formed of all the decimal numbers appearing across the entire image, which is considered as the feature vector. Figure 6(a) and Fig. 6(b) show a 3×3 grey-level cell and the corresponding binary numbers assigned by comparing the values of the center pixel with the neighboring ones. In our work, we use two different variants of LBP namely RLBP [1] and ULBP [8] as our texture based features.

4.3 Shape based feature extraction

Shape based features capture the information of an image by calculating its various geometrical aspects. In our work, we first skeletonize the component images before we use them for extraction of the shape based features. We use the following three features in our classification

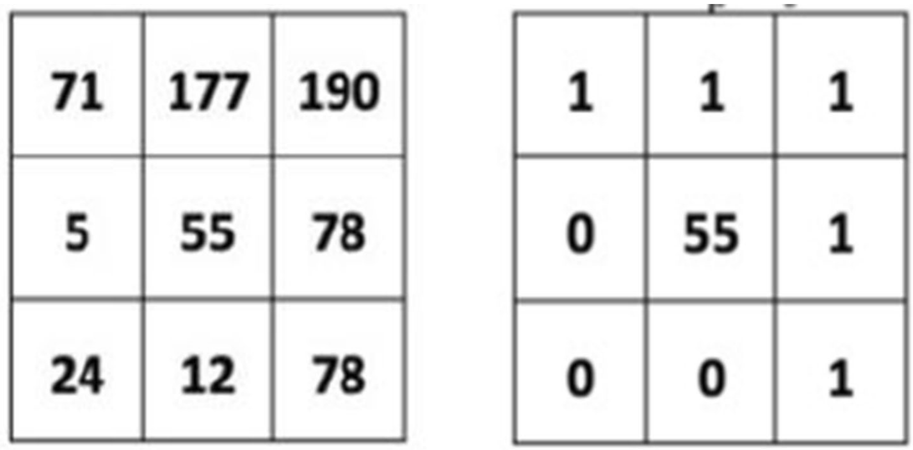
1	2	2	2
2	3	1	2
2	3	3	1
1	2	3	3

(a)

	1	2	3
1	0	3	1
2	0	2	2
3	2	0	2

(b)

Fig. 5 Showing (a) a grey level image with 3 intensity levels and (b) its corresponding GLCM



71	177	190
5	55	78
24	12	78

(a)

1	1	1
0	55	1
0	0	1

(b)

Fig. 6 Showing a cell and the corresponding binary numbers assigned by comparison. (a) indicates original gray values while (b) indicates corresponding binary values

model. The codes for shape based features are uploaded in the link "Some of the sample images are provided the link "https://github.com/Archan462/Codes".

- Centroid distance
- Tangent angle
- Chain code histogram (CCH)

4.3.1 Centroid distance

In general, distances of contour pixels from centroid of an object varies from one object to another that are not very similar in shape. So we consider this feature as one of the shape based features in our work. The contour of an image is generated using medial axis transformation [19]. Let the number of foreground pixels on the contour of an image is N. We take $(x(n), y(n))$ as a point on the contour, where $n \in [1, \ N]$. The centroid distance feature [21] first calculates the geometric centroid, let (g_x, g_y), of the contour image and then the distance of the point $(x(n), y(n))$ from centroid is $r(n)$ which is calculated by eq. (1).

$$r(n) = \sqrt{(x(n)-g_x)^2 + \left(y(n)-g_y\right)^2} \tag{1}$$

Figure 7 shows a toy image where contour pixels are marked with dot (.) and the centroid distance corresponding to a point $(x(n), y(n))$ on the contour.

All such distances across the entire contour may be considered as a feature vector. However, as the length of such feature vector depends upon the number of contour points (i.e., N), we choose the 7 statistical parameters *namely*, maximum, minimum, mean, standard deviation, 1st quartile, 2nd quartile and 3rd quartile of those distance values as the feature vector.

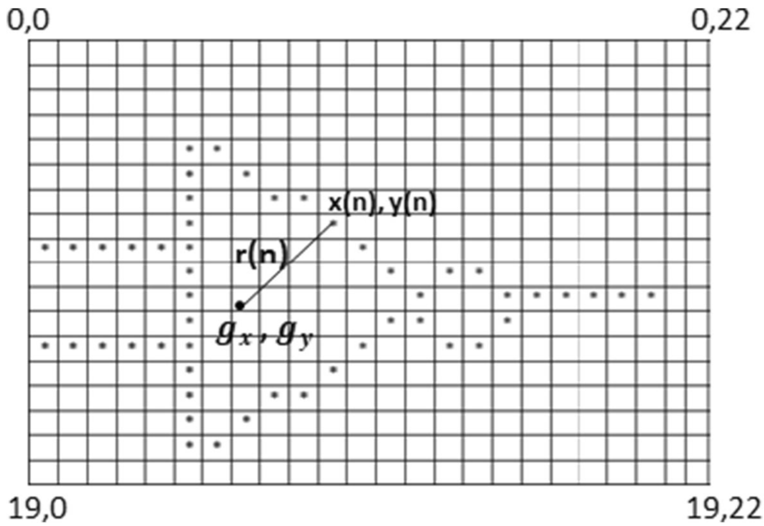


Fig. 7 Elucidates the centroid distance for a given point on the contour using toy image

To show the effectiveness of this feature, we take three component images: one from each AND gate, Capacitor and Resistor. We calculate the said centroid distance features and show them in Fig. 8. The information obtained from Fig. 8 clearly shows that the centroid distance is an effective feature to capture the shape information and thus it motivates us to choose this feature as one of our shape-based features.

But in order to capture the intricate details to distinguish very similar components like PNP transistor and NPN transistor, p-n junction diode and Zener diode, the centroid distance feature does not provide satisfactory result. So in order to gain information about those fine details, we choose two more features, *namely,* tangent angle and CCH.

4.3.2 Tangent angle

We calculate tangent angle feature for each pixel on the contour of the circuit component. Let, the number of foreground pixels on the contour of an image is N The tangent angle feature [27]

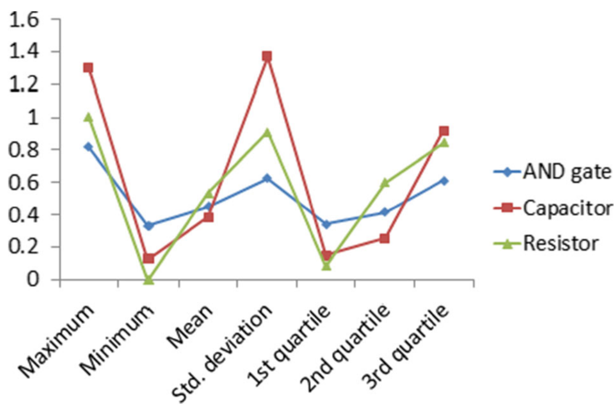


Fig. 8 Describes the variation in centroid distance features for three different hand-drawn circuit components considered here

calculates the tangential direction of a contour at given point $(x(n), y(n))$, where $n \in [1, \quad N]$, on it. It is calculated using eq. (2).

$$(n) = \arctan\left(\frac{y(n) - y(n-\delta)}{x(n) - x(n-\delta)}\right) \tag{2}$$

Here 'δ' represents a small window size which is here 1.

Fig. 9 is a hypothetical image of dimension (19 × 22) where contour pixels are marked with dot (.) and the tangent angle corresponding to a point $(x(n), y(n))$ on the contour is shown.

All such angles formed by traversing the entire circuit component image form the feature vector. Here also, the number of elements in the vector depends on the size of the image. Therefore, we choose to calculate the same 7 statistical features of the feature vector, like in case of centroid distance feature.

This feature can noticeably distinguish between very similar components like PNP and NPN transistors and so it motivates us to use this as one of our shape based features. Figure 10 shows the mentioned 7 statistical measures (i.e., the feature values) in case of a PNP and an NPN transistor. This figure clearly shows that this feature is helpful in discriminating similar shapes like PNP and NPN transistors.

4.3.3 Chain code histogram (CCH)

It is a way of encoding the boundary of a contour by a string of numbers where each number represents a particular direction in which the next point on the contour is situated. In our work, we use the 8 way connected chain code [25] where the 8 directions are represented by the digits 0–7. Figure 11 describes the convention that is followed while extracting the chain code information. Once the chain code representation of a circuit component is obtained, we find out the histogram of the digits 0–7, thus making it our feature vector.

The reason behind considering this feature as one of the shape based features in our work is illustrated in Fig. 12. We compare the feature values that are extracted using two very similar

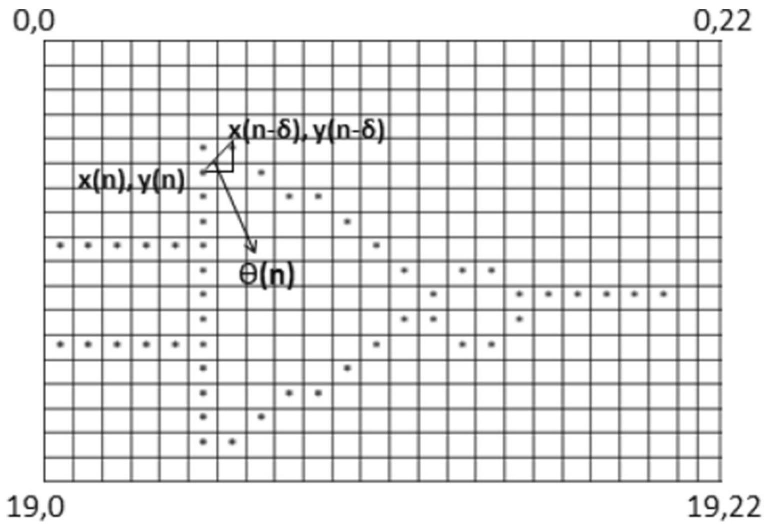


Fig. 9 Illustrates the tangent angle of a given point on the contour using a hypothetical image

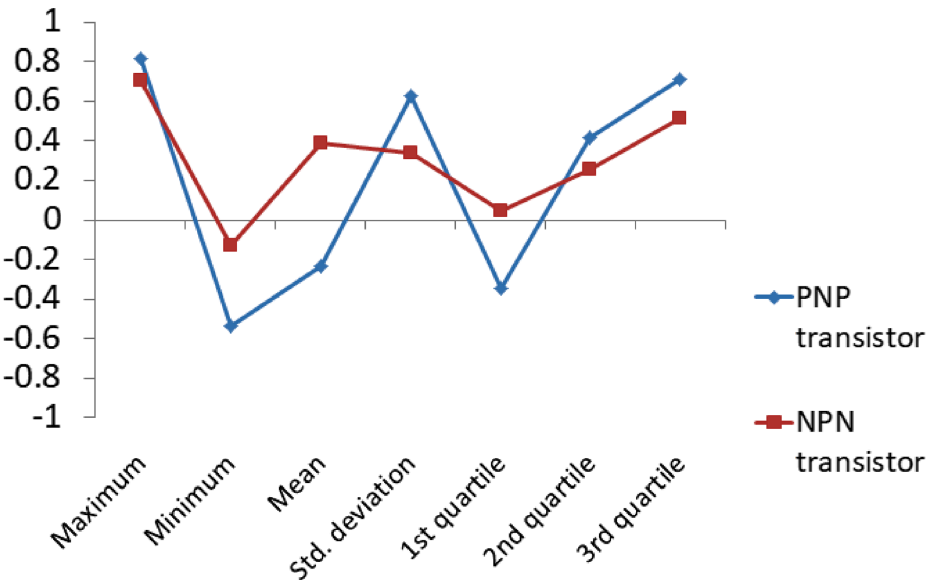


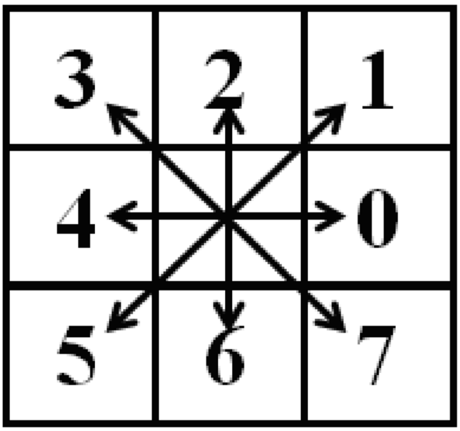
Fig. 10 Shows the variation in tangent angle features considered here for PNP and NPN transistors

looking components; like a p-n junction diode and a Zener diode. This comparison clearly establishes the benefit of using this feature for the present classification problem.

4.4 Optimization of the texture based features

Length of the texture based features that are used here, in general, is much larger as compared to the shape based features. So the chance of noisy data arising from the texture based features to affect the recognition remains high. Hence, in order to minimize the effect of noisy data, we use a feature selection algorithm called ReliefF [17]. It is an algorithm that uses a filter based approach to rank the attributes. It penalizes the predictors that give different values to neighbors of same class, and rewards the predictors that give different values to neighbors of different class. We use this feature selection algorithm on the best performing texture based feature vector. In our case it is HOG feature descriptor.

Fig. 11 Chain code representation



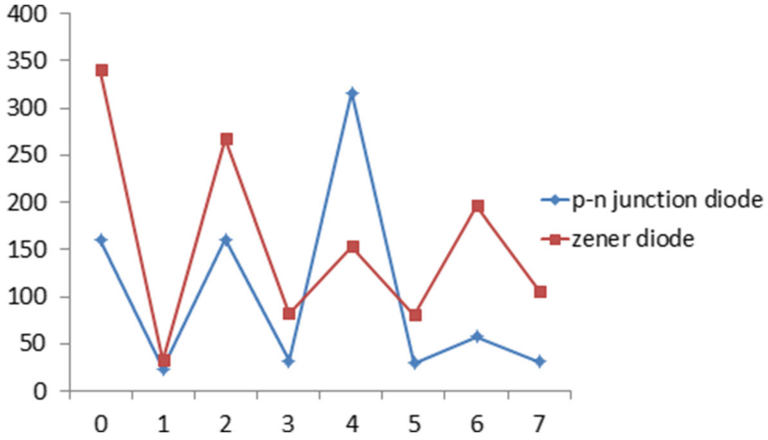


Fig. 12 Shows the variation in CCH feature values for a p-n junction diode and a Zener diode

After optimizing the texture based feature, we first combine those features with the shape based features to form the total feature set and then we proceed towards recognition of the circuit components. Initially testing is done with 5 different classifiers namely sequential minimal optimization (SMO), random forest, multilayer perceptron (MLP), KNN and Naïve Bayes .

5 Results and analysis

The most important aspect in classification and recognition is perhaps, the accuracy. Accuracy in the results obtained from a classifier gives us an idea of how effectively and precisely a model manifests itself on a selected dataset. No model can be considered acceptable if it compromises on obtaining reasonable accuracy, no matter how innovative it may seem. In our model, we attempt to capitalize on extracting the optimized features exploiting both shape and texture of a circuit component using experimentally selected classifiers. The accuracy obtained after diligent effort on optimizing the model is quite encouraging. In this section, we explore each step of our work in detail and analyze the areas in which we face hurdles.

5.1 Selection of texture based feature and classifier

As already mentioned that in the present work, we initially choose five texture based features namely, HOG, GLCM, GLRLM, RLBP and ULBP and then select the best performing feature vector. However, while we select the best texture based feature vector, we also evaluate these features using five different classifiers namely SMO, random forest, MLP, KNN and Naïve Bayes. We conduct experiments in 5-fold cross validation scheme using Waikato Environment for Knowledge Analysis (Weka) software [15]. The results obtained are shown in Fig. 13.

From Fig. 13 we can draw the following inferences

- HOG is clearly the most efficient texture based feature vector for classification of our data.
- SMO provides us with the highest classification accuracy among all the classifiers used

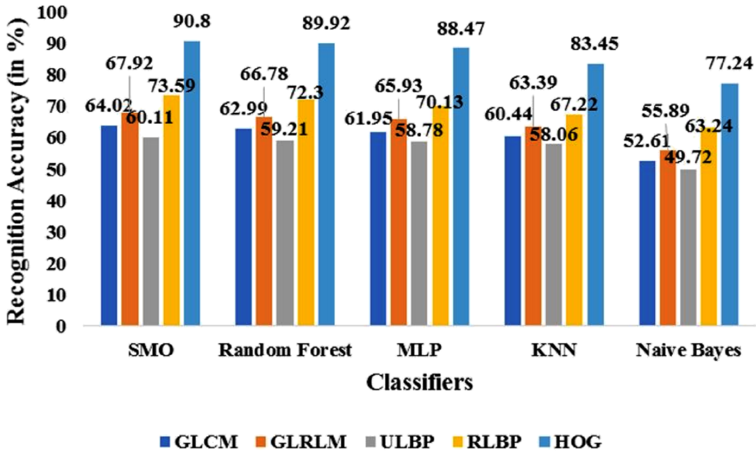


Fig. 13 Shows classification accuracies with different texture based feature descriptors and classifiers

Thus, hereon we continue the classification using SMO as our classifier and choose HOG as the texture based feature vector.

5.2 Optimizing the HOG feature descriptor

Following the selection of the best texture-based feature, we select the most significant features from the HOG feature descriptor using ReliefF feature selection algorithm. For this, we rank the features in terms of the classification ability by using ReliefF method, and then we conduct experiments to note the recognition accuracies based on the percentage of features extracted from the ranked features. The results obtained are shown in Fig. 14.

From the results shown in Fig. 14, we can clearly see that best recognition accuracy is obtained when 50% of the most significant features are taken under consideration. Thus, we keep performing experiments hereon using these optimally selected HOG features.

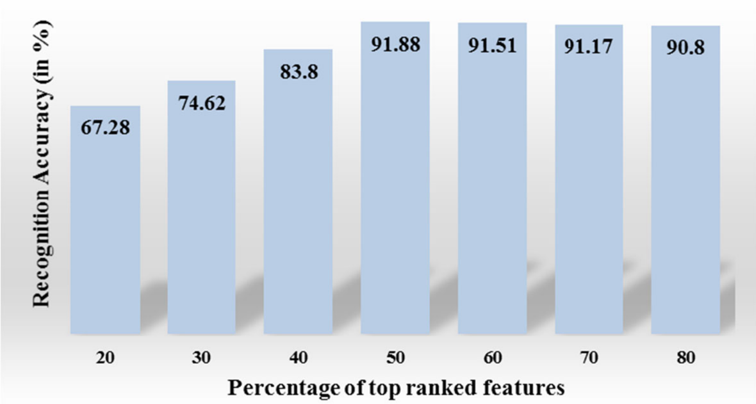


Fig. 14 Shows recognition accuracy obtained for different percentage of ranked features taken from HOG feature descriptor while using ReliefF as a feature selection method and SMO as classifier

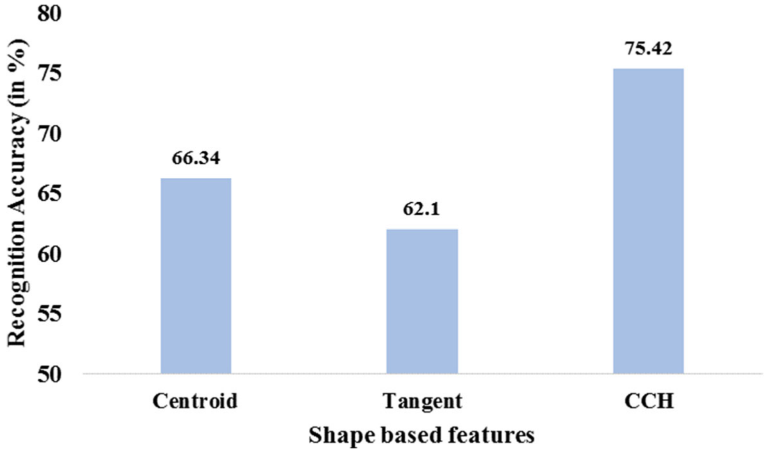


Fig. 15 Shows recognition accuracy (in %) using SMO as classifier for each shape based feature

5.3 Performances of individual shape based features

We consider three shape-based features, as mentioned earlier, in our work namely, Centroid, Tangent and CCH. The results obtained upon testing our dataset with these features alone in 5-fold cross validation using SMO as classifier are given in Fig. 15.

5.4 Final recognition

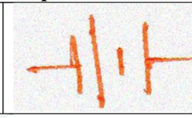
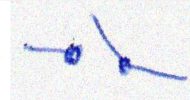
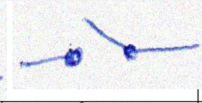
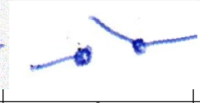
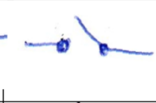
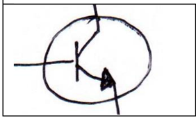
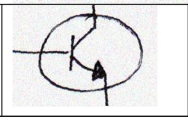
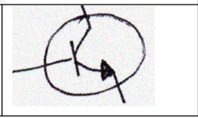
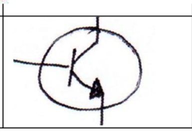
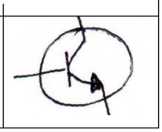
In the final experiment, we combine the optimized HOG features with the above mentioned shape based features and test it on the entire dataset using SMO as the classifier in 5-fold cross validation scheme. We obtain 93.83% average recognition accuracy on entire dataset. The confusion matrix is given in Fig. 16, from which we can infer that most of the components are classified with a relatively high degree of recognition accuracy but some of them are misclassified during the recognition process. Similar looking components like voltmeter, ammeter, and some digital gates like AND and NAND, OR and NOR show some errors while being recognized. Possible reasons for such errors are discussed in the end of the section.

In addition to this, we perform another experiment to test the effectiveness of using optimized HOG features in the final feature set. For this we combine the entire HOG features

	AC Source	Ammeter	AND	Capacitor	DC Source	Ground	Inductor	NAND	NOR	NOT	NPN Transistor	OR	PN Junction Diode	PNP Transistor	Power Supply	Resistor	Switch	Transformer	Voltmeter	Zener Diode
AC Source	136	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0
Ammeter	5	136	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	8	0
AND	0	0	131	0	0	0	0	17	0	0	0	0	0	0	0	0	0	1	1	0
Capacitor	0	0	0	145	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DC Souce	0	0	0	0	146	1	0	0	0	0	0	0	0	0	0	1	0	0	2	0
Ground	0	0	0	1	1	147	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Inductor	0	0	0	0	0	1	147	0	0	0	0	0	0	0	0	2	0	0	0	0
NAND	0	0	14	0	0	0	1	133	0	1	0	1	0	0	0	0	0	0	0	0
NOR	0	0	0	0	0	0	1	0	129	0	0	20	0	0	0	0	0	0	0	0
NOT	0	0	0	0	0	0	0	0	0	145	0	0	2	0	0	0	0	0	0	3
NPN Transistor	0	1	0	0	0	0	0	0	0	0	139	0	0	9	0	0	0	0	1	0
OR	0	0	1	0	0	0	0	1	13	0	0	135	0	0	0	0	0	0	0	0
PN JUnction Diode	0	0	0	1	0	0	0	1	0	2	0	0	141	0	0	0	1	1	0	3
PNP Transistor	1	0	0	0	0	0	0	0	0	0	8	0	0	140	0	0	0	0	1	0
Power Supply	0	0	0	0	0	1	1	0	0	0	0	0	0	2	145	0	0	0	1	0
Resistor	0	0	0	0	0	1	1	0	0	0	0	0	0	0	2	146	0	0	0	0
Switch	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	147	0	0	0
Transformer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0
Voltmeter	7	6	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	133	1
Zener Diode	0	0	0	0	0	0	0	0	0	2	0	0	3	0	0	0	0	0	1	144

Fig. 16 Shows the confusion matrix of the hand-drawn circuit component recognition procedure obtained by using HOG and some shape based features with SMO classifier

Table 2. Some sample hand-drawn circuit component images and their augmented versions.

Original samples	Augmented samples			
				
				
				

with the shape-based ones and test using 5-fold cross validation scheme. We obtain 91.79% average recognition accuracy which implies that we obtain 2.04% better recognition accuracy by using optimized HOG features (50% of the HOG feature) in the final feature set.

5.5 Performance on the augmented dataset

The present dataset contains only 150 samples per circuit component and these samples are collected in constraint condition. Therefore, to test effectiveness of our method in unconstraint conditions and with increased data samples, we first create more samples by applying some augmentation techniques on the actual data, then apply our method on the increased dataset. The augmented data are prepared by first rotating each actual circuit component image by angles -5°, -10°, $+5^{\circ}$ and $+10^{\circ}$ respectively and subsequently adding some amount of Gaussian noise to all of the actual as well as rotated circuit component images. Some examples of augmented samples along with their original versions are shown in Table 2.

The increased dataset thus prepared contains 1500 samples per class (150 original, $150 * 4 = 600$ rotated circuit images and $150 * 5 = 750$ Gaussian noise added samples). We follow the exactly same experimental setup as mentioned before on the augmented dataset. The accuracy obtained is 97.34%. The confusion matrix is given in Fig. 17. The rise in accuracy is due to the increased number of training samples per circuit component generated through data augmentation. It helps the classification model to learn about the input patterns more accurately.

	AC Source	Ammeter	AND	Capacitor	DC Source	Ground	Inductor	NAND	NOR	NOT	NPN Transistor	OR	PN Junction Diode	PNP Transistor	Power Supply	Resistor	Switch	Transformer	Voltmeter	Zener Diode
AC Source	1460	10	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	29	0
Ammeter	18	1444	0	0	0	2	0	0	0	2	0	0	0	0	4	2	0	4	22	2
AND	0	0	1420	0	0	0	0	70	0	0	0	0	0	0	0	0	0	0	10	0
Capacitor	0	0	0	1492	5	0	1	0	0	0	0	0	0	2	0	0	0	0	0	0
DC Souce	0	0	0	10	1480	0	0	0	0	0	0	0	2	4	0	4	0	0	0	0
Ground	0	0	0	0	0	1496	0	0	0	2	0	0	0	0	0	0	2	0	0	0
Inductor	0	2	0	0	0	10	1486	0	0	0	0	0	0	0	0	2	0	0	0	0
NAND	0	2	120	0	0	0	0	1356	0	0	0	18	0	0	0	0	0	0	2	2
NOR	0	2	2	0	0	0	0	0	1420	0	0	76	0	0	0	0	0	0	0	0
NOT	4	0	0	0	0	0	0	0	0	1490	0	0	2	0	0	0	2	0	0	2
NPN Transistor	2	0	2	2	0	0	0	0	0	0	1468	0	0	26	0	0	0	0	0	0
OR	0	0	6	0	0	2	0	12	94	0	0	1386	0	0	0	0	0	0	0	0
PN JUnction Diode	0	0	0	4	0	4	0	2	0	9	0	0	1470	0	0	0	1	1	0	11
PNP Transistor	0	0	0	0	0	0	0	0	0	0	14	0	0	1486	0	0	0	0	0	0
Power Supply	3	1	0	0	0	1	1	0	0	0	0	0	0	4	1490	0	0	0	4	0
Resistor	0	0	0	8	2	10	2	0	0	0	0	0	2	0	2	1474	0	0	0	0
Switch	0	0	0	0	0	4	0	2	0	0	0	0	0	0	2	0	1492	0	0	0
Transformer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1500	0	0
Voltmeter	32	22	10	8	0	0	0	0	2	2	0	1	0	0	0	0	0	0	1424	0
Zener Diode	0	0	0	2	2	4	0	4	0	4	0	0	16	0	0	0	2	0	2	1464

Fig. 17 Shows the confusion matrix of the hand-drawn circuit component recognition on augmented dataset

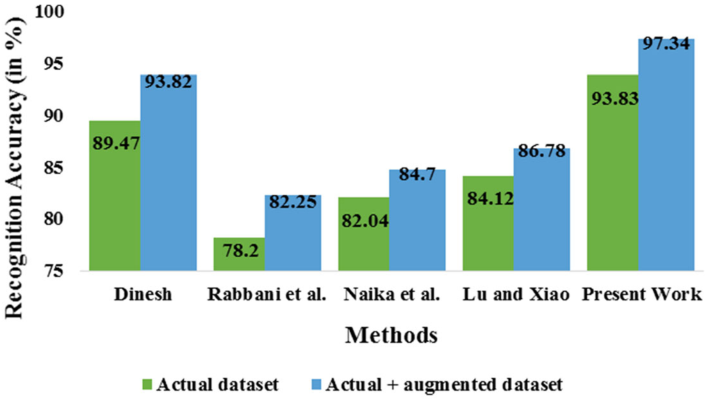


Fig. 18 Comparison of proposed hand-drawn circuit component recognition method with some state-of-the-art methods

5.6 Comparison with state-of-the-art methods

To compare the performance of our model with *state-of-the-art* methods we perform more experiments. For this reason, we implement four different methods (i.e., Dinesh [13], Rabbani et al. [24], Naika et al. [18] and Liu and Xiao [20]) and test on our original dataset as well as on augmented dataset. The comparative results are shown in Fig. 18. From these results, we can safely state that our method outperforms the state-of-the-art methods considered here for comparison.

5.7 Error analysis

The recognition of the electrical and electronic circuit components poses many challenges owing to the similar shapes of many components (like digital gates and meters). The accuracy is comparatively less for such components in contrast to unique ones like transformer and NOT gate. We observe three such cases where misclassification within classes is too high as compared to rest. These cases are listed below.

- Digital gates like AND and NAND (see Fig. 19(a-b)), OR and NOR (see Fig. 19(c-d)) are very similar to each other in shape and thus lead to misclassification yielding lower accuracy (see Fig. 20).
- Analog circuit components like ammeter, voltmeter and AC source (see Fig. 21) have similar outer boundary, i.e. circular, and differ from each other based only on the text or symbol marked inside them. So such inconsiderable difference between these three

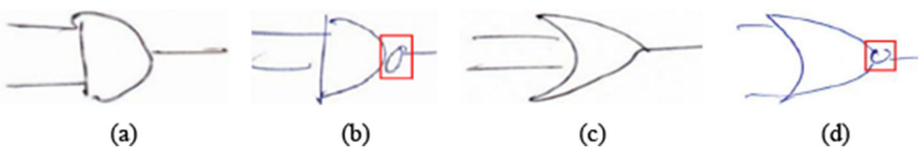


Fig. 19 (a) an AND gate, (b) a NAND gate, (c) an OR gate and (d) a NOR gate. The region of dissimilary between (a) and (b), (c) and (d) are shown with the aid of red bounding boxes.

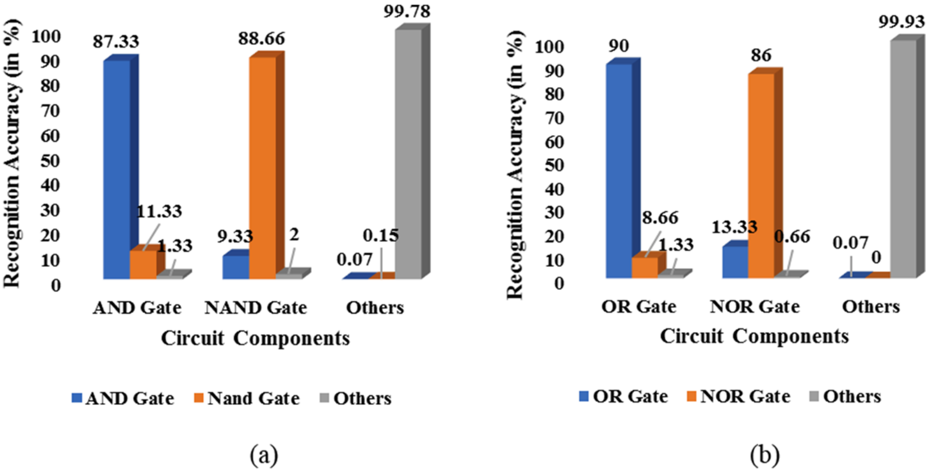


Fig. 20 Chart depicting comparison of recognition accuracy among a) AND gate /NAND gate and b) OR gate / NOR gate. In these figures, "Others" indicate any component except (a) AND or NAND gate (b) NOR or OR gate

components cause hindrance to the overall recognition process, leading to some amount of error, decreasing accuracy. The interclass misclassification rates are depicted in Fig. 22.

Scrutinizing the samples in Fig. 19, it can be inferred that the pair of components ((a-b) and (c-d)) under consideration are extremely similar in shape, the only difference between them being the presence of a bubble (as marked with a red bounding box), hence lessening the accuracy in classification. Fig. 20 compares the accuracy of recognition of individual components of similar shape.

As we can observe in Fig. 21, the components are similar in terms of the circular skeleton present in their symbols, the difference in representation being due to the presence of 'A', 'V', and '~' in the symbol. Therefore, the classifier has to rely majorly upon identifying the difference in shape and texture of these symbols that is quite difficult considering the complexity. Hence, the slightly higher error in classification is observed. Figure 22 depicts the comparison of classification of Ammeter, Voltmeter and AC Source.

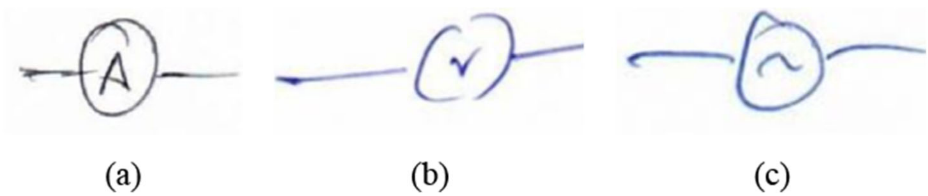


Fig. 21 Sample images of (a) ammeter and (b) voltmeter and (c) AC source

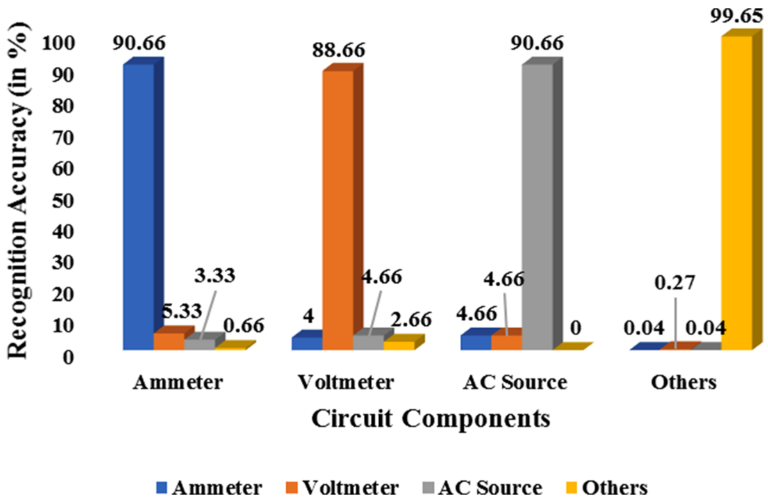


Fig. 22 Recognition accuracy comparison of ammeter, voltmeter and AC Source. In this figure "others" represent the circuit components except ammeter, voltmeter and AC Source

6 Conclusion

In the present work, we develop a model to recognize various hand-drawn analog and digital circuit components from their optically scanned versions. We prepare an in-house dataset to test our model, as no such dataset is publicly available. We use a texture based feature descriptor along with some shape based features to recognize them. Also to reduce the effect redundancy from texture based features, we use a feature selection algorithm. The overall recognition accuracy obtained is encouraging as most of the components are recognized with a high degree of accuracy, the exception being in a few very closely resembling circuit components. The noticeable error of our model i.e. inability of detecting very similar components with a higher accuracy, can be reduced by introducing novel approaches that capture the local features more accurately. Thus in future, the proposed approach can be made even better and can be commercially used for industrial as well as academic purposes.

Acknowledgements We would like to thank CMATER research laboratory of the Computer Science and Engineering Department, Jadavpur University, India for providing us the infrastructural support. This work is partially supported by the PURSE-II and UPE-II, Jadavpur University projects. Ram Sarkar is thankful to DST, Govt. of India, for the grant (EMR/2016/007213) to carry out this research.

References

1. Ahonen T, Matas J, He C, Pietikäinen M (2009) Rotation invariant image description with local binary pattern histogram fourier features. In: Scandinavian conference on image analysis. Springer, pp. 61–70
2. Angadi M, Naika RL (2014) Handwritten circuit schematic detection and simulation using computer vision approach. Int J Comput Sci Mob Comput 3:754–761
3. Avola D, Bernardi M, Cinque L, et al (2017) A machine learning approach for the online separation of handwriting from freehand drawing. In: International Conference on Image Analysis and Processing. Springer, pp. 223–232

4. Avola D, Bernardi M, Cinque L, Foresti GL, Massaroni C (2020) Online separation of handwriting from freehand drawing using extreme learning machines. Multimed Tools Appl 79:4463–4481
5. Barua S, Malakar S, Bhowmik S, et al (2017) Bangla handwritten city name recognition using gradient-based feature. In: 5th international conference on Frontiers in intelligent computing: theory and applications. Springer, Singapore, pp. 343–352
6. Bhattacharya A, Roy S, Sarkar N, et al (2020) Circuit component detection in offline hand-drawn electrical/ electronic circuit diagram. In: IEEE Calcutta conference (CALCON 2020). IEEE, Kolkata
7. Bhowmik S, Malakar S, Sarkar R, Basu S, Kundu M, Nasipuri M (2019) Off-line Bangla handwritten word recognition: a holistic approach. Neural Comput Appl 31:5783–5798
8. Chen J, Kellokumpu V, Zhao G, Pietikäinen M (2013) RLBP: robust local binary pattern. In: BMVC
9. Chu A, Sehgal CM, Greenleaf JF (1990) Use of gray value distribution of run lengths for texture analysis. Pattern Recogn Lett 11:415–419
10. Dalal N, Triggs B (2005) Histograms of Oriented Gradients for Human Detection. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE computer society conference on. IEEE, pp 886–893
11. Deufemia V, Risi M, Tortora G (2014) Sketched symbol recognition using latent-dynamic conditional random fields and distance-based clustering. Pattern Recogn 47:1159–1171
12. Dewangan A, Dhole A (2018) KNN based hand drawn electrical circuit recognition. Int J Res Appl Sci Eng Technol 6:1–6
13. Dinesh R (2018) Handwritten electronic components recognition: an approach based on hog+svm. J Theor Appl Inf Technol 96:
14. Feng G, Viard-Gaudin C, Sun Z (2009) On-line hand-drawn electric circuit diagram recognition using 2D dynamic programming. Pattern Recogn 42:3215–3223
15. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. ACM SIGKDD Explor Newsl 11:10–18
16. Jana P, Ghosh S, Bera SK, Sarkar R (2017) Handwritten document image binarization: An adaptive K-means based approach. In: 2017 IEEE Calcutta Conference (CALCON). IEEE, pp 226–230
17. Kononenko I, Robnik-Sikonja M, Pompe U (1996) ReliefF for estimation and discretization of attributes in classification, regression, and ILP problems. Artif Intell Methodol Syst Appl 31–40
18. Lakshman Naika R, Dinesh R, Prabhanjan S (2019) Handwritten Electric Circuit Diagram Recognition: An Approach Based on Finite State Machine Int J Mach Learn Comput 9:
19. Lee T-C, Kashyap RL, Chu C-N (1994) Building skeleton models via 3-D medial surface axis thinning algorithms. CVGIP Graph Model Image Process 56:462–478
20. Liu Y, Xiao Y (2013) Circuit sketch recognition. Dep Electr Eng Stanford Univ Stanford, CA
21. Malakar S, Sharma P, Singh PK, Das M, Sarkar R, Nasipuri M (2017) A holistic approach for handwritten Hindi word recognition. Int J Comput Vis Image Process 7:59–78. https://doi.org/10.4018/IJCVIP.2017010104
22. Mohanaiah P, Sathyanarayana P, GuruKumar L (2013) Image texture feature extraction using GLCM approach. Int J Sci Res Publ 3:1–5
23. Ojala T, Pietikainen M, Maenpaa T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 24:971–987
24. Rabbani M, Khoshkangini R, Nagendraswamy HS, Conti M (2016) Hand drawn optical circuit recognition. Procedia Comput Sci 84:41–48
25. Sahoo S, Nandi SK, Barua S, et al (2018) Handwritten Bangla city name recognition using shape-context feature. In: 6th international conference on Frontiers in intelligent computing: theory and applications. Springer, Singapore, pp. 451–460
26. Sala P (2004) A recognition system for symbols of electronic components in hand-written circuit diagrams. CSC 2515-machine learn Proj rep
27. Yang M, Kpalma K, Ronsin J (2012) Shape-based invariant feature extraction for object recognition. In: Advances in Reasoning-Based Image Processing Intelligent Systems. Springer, pp. 255–314

Affiliations

Soham Roy[1] • Archan Bhattacharya[1] • Navonil Sarkar[1] • Samir Malakar[2] • Ram Sarkar[3]

Soham Roy
sohamroymiller@gmail.com

Archan Bhattacharya
archanbhattacharya4@gmail.com

Navonil Sarkar
sarkar.navonil2000@gmail.com

Ram Sarkar
raamsarkar@gmail.com

[1] Department of Electronics and Telecommunication, Jadavpur University, Kolkata, India

[2] Department of Computer Science, Asutosh College, Kolkata, India

[3] Department of Computer Science and Engineering, Jadavpur University, Kolkata, India