

Hand-drawn Digital Logic Circuit Component Recognition using SVM

Mayuri D. Patare
Post Graduate Student
Jawaharlal Nehru Engineering College,
Aurangabad, Maharashtra, India.

Madhuri S. Joshi, PhD
Professor of CSE Department,
Jawaharlal Nehru Engineering College,
Aurangabad, Maharashtra, India.

ABSTRACT

Hand-drawn circuit diagrams are widely used in engineering fields, especially for the early design phases because sketch is a convenient tool to understand rough ideas of system development. Using such hand-drawn diagrams designers can focus more on the critical issues rather than on the intricate details. Electronic system design is one of the application areas in this context. To speed up the development process of such systems, hand-drawn circuits design need be transformed into the computers. The problem is that although it seems very easy for humans to recognize sketches, it is really a great challenge for the computers. This paper proposed a method to recognize components in hand drawn digital logic circuit diagram. This system used region based segmentation method to segment circuit sketch, and classified each component using Support Vector Machine that uses Fourier descriptor as the feature vector. An average of 83% circuit recognition accuracy is achieved.

Keywords

Circuit Sketch Recognition, Support Vector Machine (SVM), Fourier descriptor.

1. INTRODUCTION

The machine recognition of hand drawn symbols in engineering diagrams has been in research field from last few years. The automatic recognition of hand-sketched electronic circuit diagrams is one of its research fields, whose solution has many useful applications which include simulating the electronic characteristics of the circuit, beautification of circuits for layout rendering and human-computer interface for circuit input.

Digital logic circuit is an electronic circuit used in computers to perform logical operations on its two or more input signals. These circuits are made up of various logic gates. AND, OR, NOT are the basic logic gates, which can be used as building blocks to describe the behavior of circuit. While manufacturing any digital system, the first step is to design its internal digital circuitry and while designing this; it is a common practice for the engineers to spend a considerable amount of time laying down initial concepts using pencil and paper. Typically, it requires additional time to transform the work into electronic media in the form of technical drawings. This paper proposed a method for circuit sketch recognition that has fast response time, high accuracy and easy extensibility to new component recognition. A sketch recognition program would save engineers much time redrawing these circuits in technical software. The main objective of this work is check SVM classification accuracy for digital logic circuit components.

While handwritten text recognition and voice recognition are already commonly addressed as the research topics for last

few years, very less effort has been devoted to hand-drawn sketch recognition. Various pattern matching algorithms have been used for such application. Several recent studies have reported that the SVM (support vector machines) is generally capable of delivering higher performance in terms of classification accuracy than the other data classification algorithms [1]. SVM have been used for electrical circuit component recognition. So this method aims to apply it and check its accuracy for Digital logic circuit component recognition.

A difficult task in sketch recognition is to keep a good balance between the drawing freedom and the complexity of recognition. Generally, the more freely a system can endure, the more difficult sketch recognition will be [2]. Different users have their different drawing styles. This is one of the challenges for circuit sketch recognizer. So to achieve scale invariance, rotational invariance and tolerance of different drawing types, Fourier descriptor [3] of the boundaries of electronic component has been used as feature vectors for SVM classification for recognizing the component.

2. FEATURE EXTRACTION

Shape is the most important feature for recognition of objects in an image. For recognizing any object from an image, its shape related properties has to be extracted. There are two classes of techniques for shape feature extraction namely-region based techniques and boundary based techniques. Region based techniques uses whole region of an object while boundary based techniques uses only points on the boundary for feature extraction. Boundary based techniques do not involve much computation and are efficient than region based techniques. So the proposed method uses one of the popular boundary based shape extraction technique called as Fourier Descriptor (FD).

Fourier Descriptors is obtained by applying Discrete Fourier Transform (DFT) on a shape signature function derived from shape boundary coordinates. Then, the Fourier coefficients are used for shape features [4]. The Fourier Descriptor (FD) has been successfully applied to many shape representation applications. In the FD methods, the Fourier transformed boundary is used as a shape feature. There are various characteristics those have made FD one of the popular shape descriptor. Meaningful perception, simple derivation, robustness to noise and less matching cost due to frequency domain shape representation are some of its properties. In the proposed system, features of basic logic gates in digital logic circuits are to be extracted. All of these logic components are having closed regions and are quite difficult to distinguish using region based shape extraction techniques. So, for more accurate results boundary coordinates of these components are used for feature extraction. Another main advantage of FD is that, it provides facility of scale, rotation and translation

invariance of object while extracting the features. Once the features are extracted, in order to make the shape scale invariant, Fourier descriptors are divided by the magnitude of the 1st Fourier descriptor. Rotation invariance can be achieved by applying normalization. In order to achieve translation invariance, the centroid is subtracted from the boundary coordinates of the shape, the 0-th and 1st Fourier descriptors do not provide any information so those are eliminated.

Shape signatures require intensive computational cost during similarity calculation, due to hard normalization of rotation invariance. As the result, these shape signatures can be conveniently used in image retrieval only after processed by Fourier descriptors (FD) [5]. Acquired Fourier coefficients represent the shape in a frequency domain and they form various frequency components. The higher frequency component contains information about finer details of the shape and lower frequency components contain information about general features of the shape which is useful for shape representation. Very high frequency components do not provide any information so those are ignored for ease of feature vector normalization and Fourier descriptor dimensionality reduction. Only lower frequency descriptors carry sufficient information about shape, so although different users draw same object with different styles, the subset of low frequency components of all those objects are going to be same.

In the proposed system, certain steps have been followed for extracting the features of various logic components. First, the shape signature is selected, which is one dimensional function derived from shape boundary coordinates to represent two dimensional boundaries. This method uses complex coordinates as the shape signature, because comparatively it is more accurate and efficient than other shape signatures. Next, Discrete Fourier Transform (DFT) is applied on a shape signature function to get Fourier coefficients. These Fourier coefficients are called as Fourier descriptor, which is used as feature vector for SVM classification.

Suppose extracted coordinates of the given shape boundary points are (x_i, y_i) . Then complex coordinate function can be derived by considering each boundary coordinate pair (x_i, y_i) as a complex number on a complex plane. This shape signature function is also called as position function and it can be written as

$$z_i = x_i + 1i * y_i \quad (1)$$

Once complex coordinate function has been obtained, discrete Fourier transform is applied on it to get the Fourier coefficients a_n which is given by following equation.

$$a_n = \frac{1}{N} \sum_{i=0}^{N-1} z(i) e^{-j2\pi i n / N} \quad n = 0, 1 \dots N - 1 \quad (2)$$

Feature extraction is very important part of any recognition system as this has great impact on recognition rate. This stage analyzes segmented component, selects the set of features that can exclusively identify the component segment and pass necessary input to classification stage.

3. CLASSIFICATION

A Support Vector Machine (SVM) has been used for classifying the circuit components. A special property of SVM is that it simultaneously minimizes the empirical classification error and maximizes the geometric margin. So it is also called Maximum Margin Classifier [1]. SVM is originally derived from statistical learning theory by Vapnik and Chervnenkis.

The statistical learning theory provides a framework for studying the problem of gaining knowledge, making predictions, making decisions from a set of data. SVMs were originally developed to solve the classification problem, but recently they have been extended to solve regression problems.

SVM is a supervised learning algorithm used for data classification. It is called so because it is aware of input and outputs while performing classification. SVM is given training input data, and then it analyzes training input data, learns from it and prepares a model to classify test data. In this learning stage SVM stores certain properties of classifying objects which will help for classification. Outputs are nothing but possible classes in which input object is to be classified. Objects in the same class are having same properties. When the test input data is fed to SVM for classification, SVM analyzes properties of that object and assigns it to a suitable class.

The working of SVM is based on finding the optimal hyperplane that separates given data points. Now, what is hyperplane? Consider some training data points which are linearly separable and laying on two dimensional spaces. As those data points are linearly separable, a single 'line' is sufficient to separate those data points. Suppose those data points are laying on higher dimensional space, a line separating them is called 'hyperplane'. There can be number of such hyperplanes but SVM always strives to find an optimal hyperplane. Optimal hyperplane is the one that correctly separates the data points from one class to the other and passes through them as far as possible. It is commonly believed that points belonging to the two data classes often lie in such a way that there is always some distance between them. Optimal hyperplane maximizes that distance (margin) by constructing two parallel hyperplanes as shown in figure 1. Points on those separating hyperplanes are called 'Support Vectors'. Thus the optimal separating hyperplane maximizes the margin (street width) of the training data where margin means the minimal distance from the separating hyperplane to the closest data points.

An important and unique feature of this approach is that the solution is based only on those data points, which are at the margin. It is not necessary to do this transformation and to determine the separating hyperplane in the possibly very-high dimensional feature space, instead a kernel representation can be used, where the solution is written as a weighted sum of the values of certain kernel function evaluated at the support vectors. Figure 1 depicts the working of SVM.

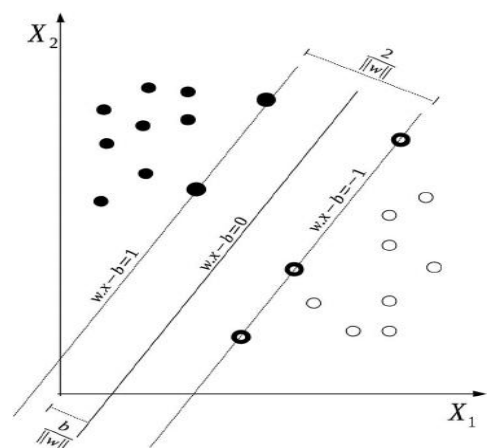


Figure 1: SVM classification for linear training data

Consider given N training data points of the form $\{x_i, y_i\}$ and $i = 1 \dots N$, where x_i is the input training point of the dimension D and y_i is the class to which it belongs. Each of the training input point is in one of the two classes $y_i = -1$ or $+1$. Assume data is linearly separable means that a single line can separate two classes on graph of x_1 Vs x_2 , where $D = 2$ and a hyperplane on the graph of x_1, x_2, \dots, x_D for $D > 2$. this Training data can be viewed by means of the dividing (or separating) hyperplanes as shown in figure1, which takes

$$w \cdot x + b = 0 \quad (3)$$

Where b is scalar and w is p-dimensional weighting vector. The vector w points perpendicular (normal) to the separating hyperplane. $\frac{b}{\|w\|}$ is the perpendicular distance from the hyperplane to the origin. For SVM implementation the variables b and w are selected such that given training data can be described by

$$w \cdot x + b = 1 \quad \text{for } y_i = +1 \quad (4)$$

$$w \cdot x + b = -1 \quad \text{for } y_i = -1 \quad (5)$$

For successful classification the margin of hyperplane need to be increased to get support vectors. This is done by adding the offset parameter b to the separating hyperplane. The hyperplane is forced to pass through the origin without addition of b and restricting the solution. So, to get maximum margin hyperplane two parallel hyperplanes are constructed which can be described by equations:

$$w \cdot x + b = 1 \quad (6)$$

$$w \cdot x + b = -1 \quad (7)$$

If the training data are linearly separable, these hyperplanes are selected such that there are no points between them and then try to maximize their distance. By geometry, the distance found between the hyperplane is $\frac{2}{\|w\|}$. Data points along the parallel hyperplanes are called Support Vectors (SVs). SVM is also used to classify data which is not linearly separable. For this purpose, it uses kernel trick. The basic idea of kernel-methods is to first preprocess the data by some non-linear mapping Φ and then to apply the same linear algorithm as before but in the image space of Φ [6]. Suppose non-linear data is to be classified. Kernel function helps to transform low dimension feature space into high dimension space to classify non-linear data efficiently with linear separation. Various kernel functions are available based on type of data which is to be classified. In this approach linear kernel has been used for SVM and binary SVM is extended to multiclass recognition problem using one-versus-rest approach.

4. DATASET AND TRAINING PROCESS

In this system, dataset containing training images for three types of logic gates, AND, OR and NOT. For each of these gates 60 training images has been drawn with different styles. Each training image is pre-processed and resized to 100x100 pixels resolution. For testing, some hand-drawn digital logic circuit sketches have been obtained from the students and the faculty members of Jawaharlal Nehru Engineering college of Aurangabad.

5. RECOGNITION PROCESS

Recognition procedure is divided into four parts: Pre-processing, Segmentation, Classification of each component and Overlay recognized components on the original circuit images with proper orientation and size.

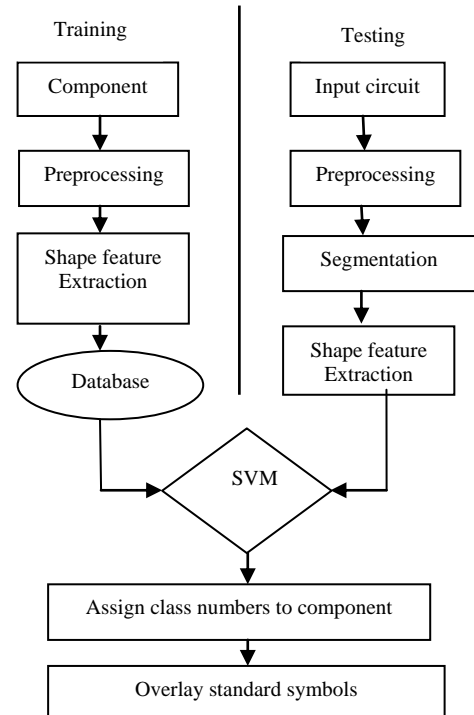


Figure 2: Recognition Process

5.1 Pre-processing

Pre-processing is a series of operations performed on the scanned input image [7]. The aim of pre-processing is to enhance some image features for further processing and to suppress unwanted distortions. Initially scanned hand-drawn circuit image is given as input at this stage which is then converted into gray scale (monochrome conversion). Binarization process converts a gray scale image into a binary image using locally adaptive threshold technique [8]. Then small regions of noise are filtered out. Training images as well as testing circuits are pre-processed at this stage.

5.2 Segmentation

Segmentation is the process of breaking up the circuit image into pieces that are small enough to be detected [9]. The aim of segmentation is to change the representation of an image into something that is more meaningful and easier to analyze. In circuit recognition segmentation tends to separate components and connections from a circuit image as test input. All the components in logic circuit diagram have a closed region so they can be easily segmented by using their region property. First, small connected components are removed those have fewer than certain number of pixels (whose size is less than our standard component size). Then set of properties for rest of the connected regions are measured so that they can be distinguished. Finally, using centre point and bounding box property of those components, bounding boxes are drawn around them. Bounding box shows all segmented components.

5.3 Classification

The aim of image classification is to analyze numerical properties of various image features and organize data into categories. Proposed system aims to classify various circuit components (logic gates). SVM is trained using 60 training images for each component meaning their features are extracted using Fourier descriptors and descriptor which carries sufficient information about the shape of those components are given input to SVM as the feature vector.

After segmentation phase each component in testing circuit is covered by bounding box and their extracted features are given input to the SVM. SVM matches these features with those stored in the database and classify the components accordingly. The class number of recognized component is displayed by the system.

5.4 Display recognized components on the original image

Finally standard symbols are displayed on recognized components. The correct position for placing those symbols is decided by the coordinates of the bounding box. The class number of recognized component is also displayed. Following Figure shows the sample run of this recognition system.

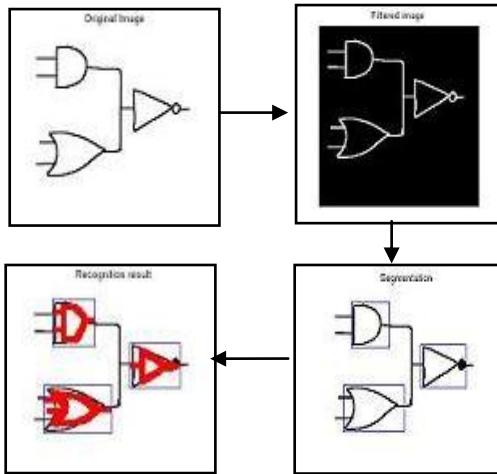


Figure 3: Sample run of the system

6. EXPERIMENTAL RESULT

The proposed system has been implemented in MATLAB R2013a. To check the accuracy of this recognition system, various experimentations has been carried out. Total of 60 training images and 10 test circuit images have been used for performance measurement purpose. Classification accuracy of SVM has been observed for individual component as well as connected circuit.

For connected circuit testing, 10 scanned hand-drawn circuits' images are used. Every circuit is containing different number of logic components. Recognition program is executed by giving one circuit image as input at a time and outputs are observed. System performance is measured based on number of correctly recognized components in individual circuits. Following table shows the Performance measurement and Confusion matrix of SVM for recognizing various digital logic circuits. True positives (TP) refer to the positive tuples that were correctly labelled by the classifier, while true negatives (TN) are the negative tuples that were correctly labelled by the classifier. False positives (FP) are the positive tuples that were incorrectly labelled. Similarly false negatives

(FN) are the negative tuples that were incorrectly labelled [10]. For experimentation purpose, AND and NOT gate are assumed as true positives and OR gate as true negatives.

The sensitivity and specificity measures can be used for correct classification. Sensitivity is also referred to as the *true positive (recognition) rate* (that is, the proportion of positive tuples that are correctly identified), while specificity is the *true negative rate* (that is, the proportion of negative tuples

that are correctly identified) [10]. These measures are defined as

$$\text{Sensitivity} = \frac{t_{\text{pos}}}{\text{pos}} \quad (8)$$

$$\text{Specificity} = \frac{t_{\text{neg}}}{\text{neg}} \quad (9)$$

The accuracy is a function of sensitivity and specificity:

$$\text{Accuracy} = \text{Sensitivity} \frac{\text{pos}}{\text{pos} + \text{neg}} + \text{Specificity} \frac{\text{neg}}{\text{pos} + \text{neg}} \quad (10)$$

Following table shows the results of component classification based on these three measures.

Table 1. Confusion Matrix for SVM

Testing circuits	Training samples	TP	TN	FP	FN
1	180	5	0	0	1
2	180	3	1	1	0
3	180	2	1	0	1
4	180	3	0	0	1
5	180	3	1	1	0
6	180	4	2	0	0
7	180	4	2	1	0
8	180	3	2	4	0
9	180	4	1	1	1
10	180	4	2	0	0

Table 2. Performance measurement of SVM

Testing circuit-ID	Training samples	Sensitivity (%)	Specificity (%)	Accuracy (%)
1	180	100	0	83.33
2	180	75	100	80
3	180	100	50	75
4	180	100	0	75
5	180	75	100	80
6	180	100	100	100
7	180	80	100	85.71
8	180	75	100	83.33
9	180	80	50	71.42
10	180	100	100	100
Average Accuracy = 83.379				

An average of 83.38% recognition accuracy is achieved for 10 connected circuits. Apart from this individual component recognition accuracy is also verified. For each component, 20 test images have been drawn with different scale, style and orientation. Each of the components is having 60 training images already stored in database. For each experiment run, numbers of training images are increased from 10 to 60 and accuracy is computed accordingly. The influence of number of training images on the accuracy of component recognition is shown in the following graph.

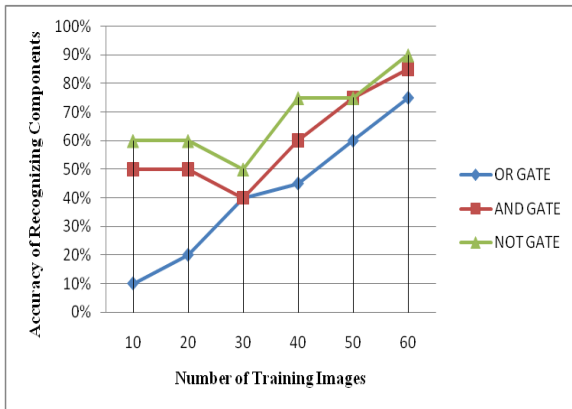


Figure 4: Individual Component Recognition Accuracy

7. CONCLUSION

This paper proposes a method for sketched digital logic circuit component recognition. The main aim of this work is to check SVM accuracy for circuit component classification. Fourier descriptor has been used for feature extraction which provides required feature vector to SVM. Around 83% component recognition accuracy is obtained with invariance to scale, rotation and translation. Classification accuracy depends on the size of training set and drawing style of training images. It is a promising approach to recognize hand-drawn sketches and produce standard circuit diagram. Accuracy can be enhanced by making system interactive. Future scopes - a) Recognized circuit diagrams can be used as input to various simulator softwares b) Other logic circuit components can also be recognized. c) Text and numerical values in the circuit diagram can be recognized. d) Output of recognized circuit can be generated by giving certain inputs.

8. REFERENCES

- [1] Durgesh k. srivastava, lekha bhambhu, "Data Classification Using Support Vector Machine", Journal of Theoretical and Applied Information Technology (JATIT), vol.12, No.1, February 2010.
- [2] Guihuan Feng, Christian Viard-Gaudin, Zhengxing Sun, "On-line Hand-drawn Electric Circuit Diagram recognition using 2D dynamic programming". Pattern Recognition, Elsevier, 42, pp.3215-3223, 2009.
- [3] G. G. Rajput and S. M. Mali. "Marathi Handwritten Numeral Recognition using Fourier Descriptors and Normalized Chain Code", International Journal of Computer Applications (IJCA), Special Issue on RTIPPR (3), pp.141-145, 2010.
- [4] Yong Hu, Zuoyong Li, "An Improved Shape signature for shape Representation and Image Retrieval", Journal of Software, vol.8, no.11, pp:2925-2929, November 2013.
- [5] Haiyu Song, Xiongfei Li, Pengjie Wang, "Adaptive Feature Selection and Extraction Approaches for Image Retrieval Based on Regions", Journal of Multimedia, vol. 5, no. 1, pp.85-92, February 2010.
- [6] Vivek verma ,Ram Niwas Giri, "Implementation Of User Define Kernel For Non-Linear Data Classification In Support Vector Machine", International Journal Of Engineering And Computer Science(IJECS),vol.4,Issue 6,pp.12568-12575,June 2015.
- [7] Manjunath Angadi, Lakshman Naika R. "Handwritten Circuit Schematic Detection and Simulation using Computer Vision Approach", International Journal of Computer Science and Mobile Computing, Vol.3 Issue.6, pp.754-761, June 2014.
- [8] Yuchi Liu, Yao Xiao, "Circuit Sketch Recognition", Department of Electrical Engineering Stanford University Stanford, CA, 2013.
- [9] Edward. B and chandran V., "Machine Recognition of Hand-drawn Circuit Diagrams", In proceeding of IEEE International conference on Acoustics, speech and Signal processing, vol.6, pp.3618-3621, June 2000.
- [10] J. Han and M. Kamber, "Data Mining: Concepts and Techniques", Morgan Kaufmann Publication, 2006.