# A recognition system for symbols of electronic components in hand-written circuit diagrams.

**Pablo Sala**
Department of Computer Science
University of Toronto
Toronto, Canada
psala@cs.toronto.edu

## Abstract

In this work we propose a method for recognizing symbols of electronic components in hand-written circuit diagrams. We do not deal with the problem of segmentation of component symbols from the circuit diagram, but we focus on the problem of recognition of individual symbols. Our approach consists in 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 modeled by a HMM. To test our approach, several HMMs with different number of states were trained per symbol class and model selection was performed using a validation set to decide for the best number of states for the HMMs of each class. The classification performance of the selected HMMs was measured on a set of images not used during training, obtaining a very low error rate, suggesting that this approach has promising potential.

## 1 Introduction and Previous Work

The machine recognition of hand-written symbols in engineering diagrams has been a focus of research for the last thirty years. The automatic acquisition of hand-sketched electronic circuit diagrams is an instance of this type of problems whose solution has many useful applications which include automatic input of circuit diagrams for circuit analysis purposes, beautification of circuits for layout rendering, and human-computer interface for circuit input.

Several approaches have been tried for the task of recognizing hand-sketched diagrams. Edwards and Chandran [1] uses a statistical classifier on the distribution of 35 image properties extracted from the thinned image of the electronic component symbol. Valois *et al.* [2] perform component symbol matching invariant to rotation, translation and changes in scale. They decompose each stroke into line and arc primitives and model the topological and structural relations between these basic features.

General sketch recognition has been approached by Sezgin and Davis [3] viewing sketching as an incremental and interactive process when the order of strokes is available, such as

when a tablet PC or PDA is used to input the sketch. They use a HMM-based technique to model the ordering of the strokes. However, this approach fails if the user is not consistent in the ordering of the strokes when drawing objects of the same class. Müller *et al.* [4] use a HMM based classifier to recognize hand-drawn pictograms using a rotation-invariant feature extraction technique. The pictogram is inscribed in a circle which is divided into a fixed number $T$ of overlaping stripes along its radius with each stripe composed of a fixed number $N$ of blocks. A vector of dimension $N$ is created per strip containing the percentage of foreground pixels of the pictogram contained in each block. The model consists of the sequence of vectors ordered by the position of the corresponding stripe in the circle. A linear HMM is trained to recognize the sequence and this model is further extended with modified initial states and exit distributions to support rotational invariance.

In this paper we present a method for electronic component symbol recognition that is scale-, translation- and rotation-invariant, and independent of the order in which the symbol strokes are drawn, hence it can be applied to circuit diagrams that are already on print without the need of having to be acquired using any special device. We assume that the image of the circuit diagram has already been segmented into individual component symbols, and we only focus on the recognition part of each particular symbol obtained in the segmentation stage. A possible way to perform the segmentation of the circuit into components is suggested in [1]. The sketch of each symbol class is modeled as a stochastical linear sequence of image features and we train four HMMs per class to learn the probability distribution of the feature sequence given the class. (Each of the four HMMs corresponds to a different *pose* of the symbol.) Classification of a new exemplar is done by segmenting the symbol into its sequence of basic features and computing, for each symbol class HMM, the likelihood of observing this sequence given the HMM. The symbol is assumed to belong to the class of the model for which the observation is most likely.

## 2 Approach

Each symbol is segmented into a sequence of basic image features. These features are the noisy observations used to train the HMM. The segmentation of a symbol into basic features is done by parsing the image of the symbol in a straight direction at a given orientation. We refer to this as the *parsing direction*. To achieve rotational invariant recognition, a canonical orientation has to be used for each class of symbols. In our experiments we used the orientation along which the set of foreground pixels of the symbol extends the most, i.e., the principal component. This direction is found using principal component analysis of the coordinates of the set of foreground pixels.

For each class, four HMMs are trained to learn the distribution of the sequence of features in that class. As shown in Figure 1, a symbol can appear in the circuit diagram in one of four possible symmetrical poses, depending of the segmentation process and drawing style of the user. A separate HMM is trained for each of these four poses. The set of symbols instances of each class used for training should consistently have a unique pose. The other three poses are generated by applying a 180 degree rotation and/or a symmetry operation to this one.

The classification of a new symbol sketch is performed by computing the likelihood of observing the feature sequence $f_1, \ldots, f_T$ of that symbol for the HMM of each class $H_k$. The class of the HMM for which the observation is most likely is predicted as the class of the symbol, i.e., $\arg\max_k p(f_1, \ldots, f_T | H_k)$.

(a) (b) (c) (d)

Figure 1: Four possible symmetrical poses of a symbol in a circuit diagram.

### 2.1 The features

The basic features we used were selected to make the segmentation of the symbol independent to scale and translation, as well as changes in drawing styles. They try to capture general aspects of the geometrical properties of the image of the symbol. The symbol is asumed to be tightly contained in a rectangular box $R$ and is oriented such that its parsing direction, as explained earlier, extends along the direction of the longest sides of $R$. The rectangle defines a system of cartesian coordinates with the $x$-axis oriented along the direction of its longest sides. An analysis of the geometry of the symbol image is performed by computing measurements on the foreground pixels with same $x$ value, as illustrated in Figure 2.

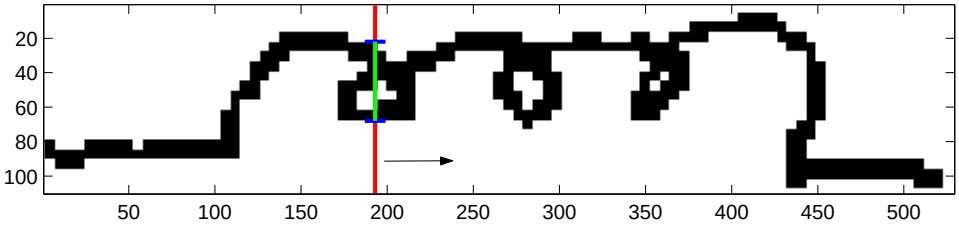



Figure 2: Symbol parsing.

Let $W$ be the length of the shortest sides of $R$. At each $x$, the following values of the image are computed: the relative *width* $w(x) = \frac{y_H(x)-y_L(x)+1}{W}$, in terms of the y-coordinates of the highest ($y_H(x)$) and lowest ($y_L(x)$) vertical foreground pixels at position $x$; the relative *center* $c(x) = \frac{y_H(x)+y_L(x)}{2W}$, i.e., the relative location of the midpoint between $y_H(x)$ and $y_L(x)$; and the *number of intersects* $i(x)$ between the image and the line $L(x)$ parallel to the $y$-axis that goes through $x$, i.e., the number of connected components of foreground pixels along $L(x)$.

Based on these measurements, the stroke type $ST(x)$ at each point $x$ is defined as nostroke iff $w(x) = 0$; thin iff $0 \leq w(x) \leq \theta$, and wide iff $\theta < w(x)$, given a threshold $\theta$. (In our experiments we used $\theta = 0.35$.) The function $ST$ is smoothed to reduce the effects of noise by discarding values that are not consistently supported by a minimum number (2 in our experiments) of closest neighbors, and in that case taking on the most frequent value among its closest neighbors.

Let $B$ be the length of the longest sides of $R$. The interval $[0, B]$ is initially divided into the minimum length sequence of subintervals $O = \{[a_1, b_1), [a_1, b_2), \ldots, [a_{n_1}, b_{n-1}), [a_n, b_n]\}$, ($b_i = a_{i+1}$), such that each subinterval $[a_i, b_i]$ corresponds to contiguous values of $x$ that have the same stroke type $ST(x)$. For the subintervals that correspond to strokes of type thin and wide, the *slope* $S(x)$ of a function $f(x)$ is computed and smoothed. In the case of thin strokes $f = c$, and in case of

wide strokes $f = w$. The slope is computed for a subinterval $[a, b)$ using only the value of $f(x)$ for $x \in [a, b]$: A function $\widetilde{f}(x) : [2a - b, 2b - a) \rightarrow \mathbb{R}$ is created by mirroring $f(x)$ on both extremes of the interval and scaling it up by $W$, i.e.:

$$\widetilde{f}(x) = \begin{cases} f(2a - x)W & , 2a - b \leq x < a \\ f(x)W & , a \leq x < b \\ f(2b - x)W & , b \leq x < 2b - a \end{cases}$$

and the slope $S(x)$ of scaled $f(x)$ is computed as the derivative of $\widetilde{f}(x)$ in the interval $[2a - b, 2b - a]$ by convolving it with the negative of the derivative of a 0 mean Gaussian (of $\sigma = 2$ in our experiments). Given a threshold $\delta$ the direction of the slope is computed as

$$D(x) = \begin{cases} 0 & , |S(x)| < \delta \\ \text{sign}(S(x)) & , \text{otherwise} \end{cases}$$

and later smoothed in the same way it was done with $ST$. We used $\delta = 0.2$ and a minimum support of 3 for this step in our experiments.

The subintervals in $O$ corresponding to strokes of type thin are further divided into smaller subintervals such that all points in each subinterval have the same slope direction. And the subintervals of wide strokes are divided into subintervals of points with same slope direction and same number of intersects. For each interval $[a, b)$ in the final sequence, a feature is computed, consisting of four descriptive values to characterize the geometry of the image contained in the interval:

- **Curvature of $\widetilde{f}$:** a measure of the curvature of $\widetilde{f}$ in the interval is computed as the area between the values of $\widetilde{f}$ in $[a, b]$ and the secant line through $\widetilde{f}(a)$ and $\widetilde{f}(b)$, divided by $b - a$.
- **Slope of $\widetilde{f}$:** an estimate of the average slope of $\widetilde{f}$ in the interval is calculated as $\frac{\widetilde{f}(b) - \widetilde{f}(a)}{b - a}$.
- **Width:** average of the relative width function $w(x)$ in the interval.
- **Number of Intersects:** the value of $i(x)$ for all points in the interval. (In the case of a thin stroke, this value is always assumed to be 1.)

The model of a symbol class and pose is the distribution of this sequence of features for the symbols in the class having such pose. This distribution will be learned by an HMM per class and pose.

## 3 Results

We tested our idea using images of electronic symbols hand-drawn by the same individual. Figure 3 shows exemplars of the 12 different classes of symbols used in our experiments. Each HMM was trained using the Baum-Welch EM algorithm (See [5], for example). The values for the initial state distribution and the transition probabilities between states were initialized randomly. The emission probabilities of the symbol features (of dimension four) were assumed to be distributed normally and were initialized using k-means on the set of feature values of the observed feature sequences. In an attempt to reduce the problems caused by local minima, the training of each HMM was independently performed three times, and the set of converged parameters that achieved the highest log likelihood were selected for the HMM.

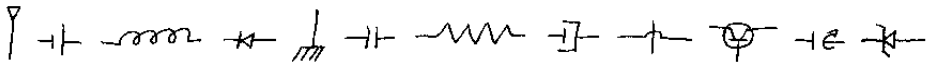


Figure 3: Exemplars of the hand-drawn symbols of each class used in the experiments.

A HMM classifier $C = \{H_{cp}\}_{c=1,\ldots,12, p=1,\ldots,4}$ for our problem consists of a set of 48 HMMs, i.e., one HMM $H_{cp}$ per symbol class $c$ and pose $p$. Classification of a symbol is done by selecting the class of the HMM that with highest probability would observe the sequence of features corresponding to the symbol. A regularization parameter of a HMM is its number of states. We decided on this attribute of each class and pose HMM by model selection implemented through an iterative selection of classifiers using a validation set composed of 15 images (not used during training) per symbol class and pose. For each class $c$ and pose $p$, a set of 15 HMMs were trained $\{H_{cp}^{2k}\}_{k=1,\ldots,15}$, each with a different even number of states, between 2 and 30. The training sets consisted of images of 45 different instances of the same symbol class.

Initially the *best performing HMM* $\widetilde{H}_{cp}$ for each class $c$ and pose $p$ was assumed to be the one with 6 states $H_{cp}^{6}$. For each class $c$ and pose $p$ at a time, 15 classifiers $\{C_{cp}^{2k}\}_{k=1,\ldots,15}$ were built, each having an HMM with a different number $n$ of states for the current class and pose, and having the corresponding best performing HMM *so far* for all other classes and poses, i.e., $C_{cp}^{n} = H_{cp}^{n} \cup \{\widetilde{H}_{c'p'}\}_{c' \neq c \vee p' \neq p}$. Each of the 15 classifiers are tested on the validation set and the classifier $C_{cp}^{\tilde{n}}$ that achieves the minimum number of misclassifications is selected. (Ties are broken by selecting the HMM with the least number of states.) The best performing HMM for class $c$ and pose $p$ is updated to be the one used in $C_{cp}^{\tilde{n}}$ for that class and pose, i.e., $\widetilde{H}_{cp} = H_{cp}^{\tilde{n}}$. This process is repeated until the best performing HMM for each class and pose does not change after a complete iteration through all symbol classes and poses. In our experiments, it took 6 iterations for this process to converge.

Table 1: Number of States per Symbol Class and Pose

| **Symbol Class** | **States per Pose** | | | |
|---|---|---|---|---|
| | **a** | **b** | **c** | **d** |
|  | 6 | 4 | 4 | 8 |
|  | 14 | 8 | 22 | 4 |
|  | 16 | 20 | 8 | 14 |
|  | 2 | 6 | 6 | 2 |
|  | 2 | 6 | 6 | 2 |
|  | 16 | 12 | 12 | 8 |
|  | 2 | 6 | 4 | 6 |
|  | 2 | 2 | 2 | 2 |
|  | 14 | 18 | 6 | 6 |
|  | 12 | 2 | 12 | 6 |
|  | 10 | 2 | 6 | 16 |
|  | 16 | 2 | 10 | 10 |

Figure 4 shows the curves of the number of misclassified symbols as a function of the number of states of the HMM for each of the classifiers tested in an iteration of the previous algorithm. The curve in row $p$ and column $c$ corresponds to the function $f(n) =$ Rate of misclassifications of $C_{cp}^{n}$. These curves were computed for the classifiers

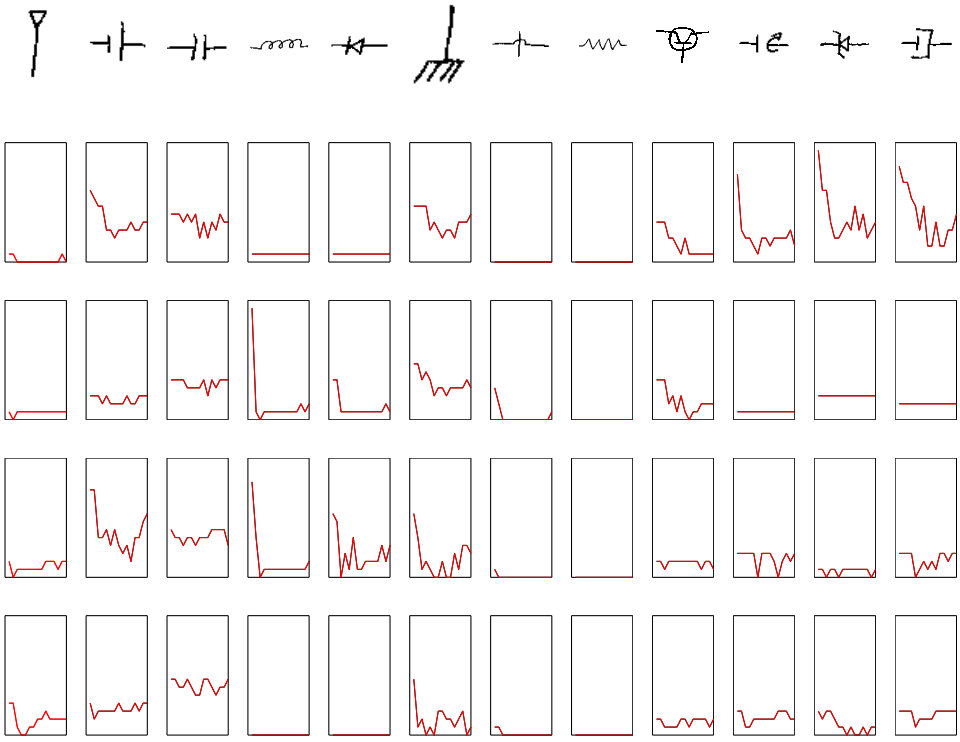
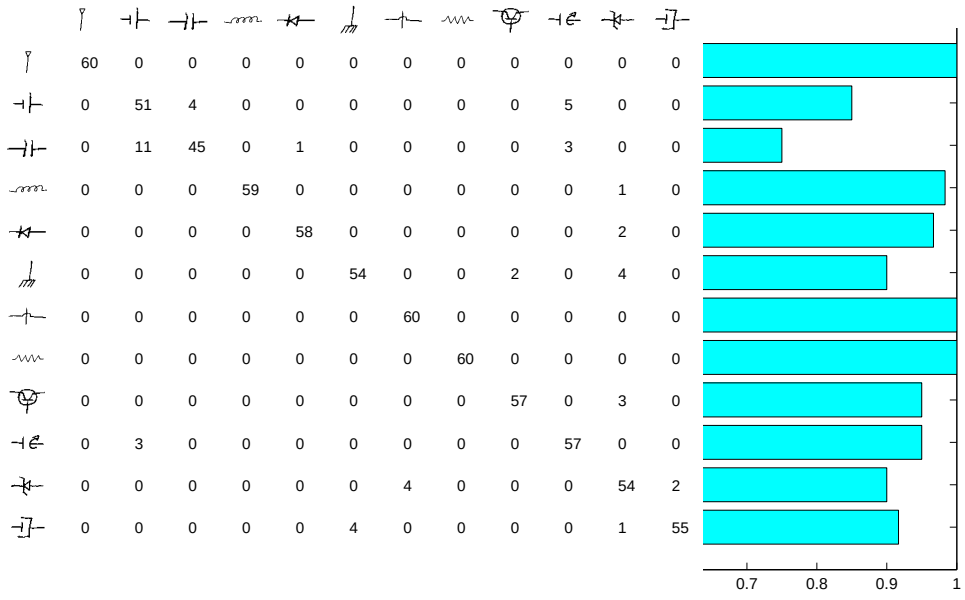


Figure 4: Curves of number of misclassified symbols in function of the number of states of the HMM. The curve in row $p$ and column $c$ corresponds to the function $f(n) =$ Rate of misclassifications of $C^n_{cp}$. The portion of the x-axis of each curve bounding box corresponds to the interval [0,30] and the portion of the y-axis to the interval [0,1].

after convergence of the best performing HMMs of all classes and poses at the end of the execution of the algorithm. Table 1 shows the number of states per symbol class and pose HMM that achieved the best results.

The confusion matrix in Figure 5, contains a row per real symbol class, and a column per predicted class, showing the errors for each class made by the final converged classifier on test data (not used during training) consisting of 15 symbol images per pose and class. It can observed that the symbol classes of the battery (2nd symbol), the capacitor (3rd symbol) and the variable capacitor (10th symbol) are mutually confused to a certain extent. The three symbols are very similar in geometry: they are roughly modeled, in term of our proposed features, by a sequence of length 5 consisting of a thin centered stroke, a centered wide stroke with one intersect, a nostroke, a centerd wide stroke and a final centered thin stroke. From this, it is obvious that the features that we proposed for the symbol sequence model do a poor job to distinguish among these symbols. A larger training set would probably improve the classification rate of these three classes to some extent, since the classifier would have more examples to learn the *few small* (from the perspective of our feature model) geometrical details that separate the three classes. The low total error rate (0.0694) achieved in our experimental results on the classification of previously unseen data suggests that this approach has a promising potential.




(a) (b)

Figure 5: (a) Confusion matrix of experiments: rows are real symbol classes and columns are predicted classes, (b) Classification accuracy rate per symbol class.

## 4 Conclusions

We have presented a method for recognition of hand-sketched symbols of electronic components using HMMs. We proposed a method to model a symbol as a sequence of basic image features and learning the stochastic distribution of that sequence for each symbol class using a separate HMM. Experimental results show that our approach achieves good classification rates and has high potential to solve the problem of recognition of symbols having a geometry that can be described sequentially.

### Acknowledgments

We would like to thank Sam Roweis for his excellent Machine Learning course which he teaches so enthusiastically and have been extremely instructive to us.

## References

[1] B. Edwards and V. Chandran. Machine Recognition of Hand-Drawn Circuit Diagrams. In *Proceedings of 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing*, number 6, pages 3618–3621, Istanbul, Turkey, June 2000.

[2] J. P. Valois, M. Côté, and M. Cheriet. Online Recognition of Sketched Electrical Diagrams. In *ICDAR*, pages 460–464, 2001.

[3] Tevfik Metin Sezgin and Randall Davis. Hmm-based efficient sketch recognition. In *Proceedings of the 2005 International Conferences on Intelligent User Interfaces*, page (To appear), New York, New York, January 2005. ACM Press.

[4] S. Müller, G. Rigoll, A. Kosmala, and D. Mazurenok. Invariant recognition of hand-drawn pictograms using hmms with a rotating feature extraction. In *International Workshop on Frontiers in Handwriting Recognition*, number 6, pages 25–34, Taejon, Korea, August 1998.

[5] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE*, number 77, pages 257–286, February 1989.