

## Research Article

# SKETRACK: Stroke-Based Recognition of Online Hand-Drawn Sketches of Arrow-Connected Diagrams and Digital Logic Circuit Diagrams

Oğuz Altun and Orhan Nooruldeen 

*Department of Computer Engineering, Faculty of Electrical and Electronic Engineering, Yildiz Technical University, Istanbul, Turkey*

Correspondence should be addressed to Orhan Nooruldeen; orhan\_karadigi@yahoo.com

Received 4 July 2019; Revised 21 October 2019; Accepted 26 October 2019; Published 26 November 2019

Academic Editor: Michele Risi

Copyright © 2019 Oğuz Altun and Orhan Nooruldeen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Digitalization of handwritten documents has created a greater need for accurate online recognition of hand-drawn sketches. However, the online recognition of hand-drawn diagrams is an enduring challenge in human-computer interaction due to the complexity in extracting and recognizing the visual objects reliably from a continuous stroke stream. This paper focuses on the design and development of a new, efficient stroke-based online hand-drawn sketch recognition scheme named SKETRACK for hand-drawn arrow diagrams and digital logic circuit diagrams. The fundamental parts of this model are text separation, symbol segmentation, feature extraction, classification, and structural analysis. The proposed scheme utilizes the concepts of normalization and segmentation to isolate the text from the sketches. Then, the features are extracted to model different structural variations of the strokes that are categorized into the arrows/lines and the symbols for effective processing. The strokes are clustered using the spectral clustering algorithm based on p-distance and Euclidean distance to compute the similarity between the features and minimize the feature dimensionality by grouping similar features. Then, the symbol recognition is performed using modified support vector machine (MSVM) classifier in which a hybrid kernel function with a lion optimized tuning parameter of SVM is utilized. Structural analysis is performed with lion-based task optimization for recognizing the symbol candidates to form the final diagram representations. This proposed recognition model is suitable for simpler structures such as flowcharts, finite automata, and the logic circuit diagrams. Through the experiments, the performance of the proposed SKETRACK scheme is evaluated on three domains of databases and the results are compared with the state-of-the-art methods to validate its superior efficiency.

## 1. Introduction

Communications between humans have been performed in many modes over the generations. Sketches are the basic form of communication that has been prevalent from the ancient human civilizations since the prehistoric periods. Many remains of this communication are still found in cave arts and pictograms which are widely considered as a benchmark to study the development of civilizations. Recent years have seen many works initiated to capture and recognize the handwritten documents including the hand-drawn sketches [1–3]. Digitalization of these handwritten

documents has gained considerable research interest for historical documentation applications. These applications are merely focussed on processing the images or photographs of such handwritten documents and sketches. However, the main challenge lies in automatically recognizing these freehand contents in a machine such as computer. This challenge forms the fundamental process of digitalization [4]. Apart from historical freehand sketches and writings, today's digital era has paved the way for developing applications to follow freehand writing and sketching experience for the users. The rapid emergence of portable touch screen gadgets and smartphones has also

contributed significantly to the users to draw the hand-drawn sketches on their screens using their fingers. Recognition of these freehand sketches is still challenging due to the nonuniform sketch patterns and noncompliance with rule books. This significantly increases the complexity in recognizing the freehand sketches but has also increased the research on digitalizing the handwritten documents [5]. In this modern era, the sketches, both online and offline drawn diagrams, play a greater role in gaming, animation, architects, designers, and programmers to design their reference models. In such cases, the need for recognizing these sketches is necessary for efficient and time-saving working environments. This has increased the popularity in studying the freehand sketches in the recent years, with more focus intended towards the sketch recognition, sketch-based data retrieval, and sketch abstraction for other applications. The freehand sketches in virtual mediums can be stored in image or web documents and can be recognized effectively through suitable tools [6, 7]. However, there are more challenges in acquiring the best results when the quality of input data decreases.

The task of automatically recognizing the freehand sketches is a nontrivial problem compared to automatically recognizing the normal image and traditional sketch processing models such as computer-aided design (CAD). The major difference is the relatively large intraclass variations and interclass uncertainties in the automatic recognition procedure [8]. The sketches are composed of complex structures that are represented in abstract forms and more free-style drawing where the complete process is unconstrained. These freehand sketches are entirely different from the conventional images in which the recognition is performed based on image features or visual cues such as colour and texture [9]. Likewise, the freehand sketches do not have similar properties with textual drawing where the sketches are composed of uniform and constrained structures. These unique features of the freehand sketches make it inappropriate to use the traditional contour matching schemes for recognition [10, 11].

The offline recognition of freehand sketches is for working on images that are scanned by scan devices or pictures that are taken by cameras. Time or ordering information of traces or the points in traces is not included.

The online recognition of freehand sketches for images is drawn by gadgets like smartphones or tablets [12, 14]. Online handwriting data can be inspected based on strokes. The ordering or time information of strokes and the points in strokes is given. Our work is an example of online freehand sketch recognition. As such, the databases we use have time and ordering information of the traces and points in them.

There have been many research works focussed on some of the sketch-related fields, such as sketch-based image retrieval [15], video retrieval [16], action analysis [17], segmentation [7], and recognition [18]. The interest in modelling and developing accurate sketch recognition models has been increasing massively especially after the recent advancements in soft computing techniques. The simpler freehand sketches consisting of strokes include the arrow-based diagrams and the line-based diagrams are

highly employed for evaluating these models due to the application of these diagrams in wider domains. Even these sketches are challenging to recognize due to their varying representations by the users [18]. This work is focussed on resolving this complexity by developing a mode that can recognize the arrow-based online sketches and the simpler logic circuit sketches. This work utilizes the strokes to recognize the lines, arrows, and symbols in the sketches and then detects the relations between them. These results are merged by the structural analysis component to recreate the sketches with accurate recognition.

There are three contributions in this work. First, a feature extraction is formulated to store and encode the different structural variations of different shape strokes in the input freehand sketches. Secondly, the p-distance and Euclidean distance are utilized to reduce dimensions of the stroke features and are clustered using the spectral clustering algorithm. Third, the major contribution is the development of modified SVM (MSVM) classifier for the recognition process. The SVM is modified by utilizing the classification concept of k-nearest neighbours (KNN) algorithm for assigning the boundary limits based on which a hybrid kernel is presented and the task is optimized using the lion optimization algorithm. This modification of the SVM algorithm enhances the overall performance of the recognition of online freehand sketches. The remainder of this article is organized as follows: Section 2 presents a discussion on recent related works. Section 3 describes the sketch structure and supported formats. Section 4 explains the proposed recognition scheme of SKETRACK. The experiments and the analysis results are given in Section 5 while the conclusions are discussed in Section 6.

## 2. Related Works

Recognition of handwritten documents and freehand sketches has been a popular research domain in the recent years. Many researchers have tried to develop efficient and accurate recognition models of which some of the prominent works are discussed here. Li et al. [19] suggested the concept of utilizing star graph-based ensemble matching and unified ensemble matching with multi-SVM classification-based categorization of structured features for sketch recognition. These ensemble approaches exploit the local and global feature representations and provide accurate sketch recognition by overcoming the limitations of SVM. Li et al. [20] also developed a freehand sketch recognition approach using the multikernel feature learning concept that fuses several common features of sketches to recognize them. However, this approach has quality limitations in matching some common features due to the fault of users drawing different shapes in a similar manner. This limitation requires the application of uniform attributes to super-categorize and form subcategories that create branching processing and become ineffective. Li et al. [21] also presented another freehand sketch synthesis approach using deformable stroke model which consists of standard drawing format and varying formats of each shape and symbol. Based on these strokes' knowledge, the generative

data-driven model detects the diverse sketch objects without any training or additional alignments. However, this feature extraction model detects the sketches using perception grouping results of many similar sketches, but when the recognition is for single sketch, the results may be not promising. Likewise, the unsupervised nature of this model makes it difficult to recognize the complex structured sketches. Huang et al. [22] developed an approach for data-driven segmentation and labelling of freehand sketches for accurate recognition. This approach initially models the sketch segmentation problem using mixed integer programming to optimize the local features and global features of the connected structures in a sketch. However, the accuracy can still be improved if the semantic features are considered. Schneider and Tuytelaars [23] employed the Fisher vectors to enable the sketch classification for highly accurate recognition of freehand sketches. This data-driven approach also modified the standards to match the semantic similarity of the sketches irrespective of the humans who drew the sketch. However, the overall performance is not perfect due to the greedy nature of recognition.

Bresler et al. [24] provided an approach to detect the arrows in online sketched diagrams using the concept of relative stroke positioning for arrowhead classification. This approach is highly effective for the recognition of arrows in flowchart and finite automata sketches. However, this approach provides slightly worst results in terms of relative positioning as the fuzzy positioning principle is not used. Bresler et al. [25] also presented an online recognition model for sketched arrow-connected flowcharts and finite automata diagrams. This approach employed the concept of symbol candidate selection based on evaluation relations between them using a knowledge domain. This system is efficient in recognizing the arrow-based diagrams accurately, but the recognition time is high as the recognition process utilizes all the past knowledge before providing the result. Similarly, there is a common possibility of error when the input sketch is of different style than the past knowledge. Zheng et al. [26] proposed an approach for discovering discriminative patches of the freehand sketches for improved analysis results. The presented approach belongs to weakly supervised learning approaches and utilizes the pyramid histogram of oriented gradient to represent the discriminative patches which are further analysed using an iterative detection process for accurate discovery. However, this approach supports only qualitative analysis and does not support efficient sketch recognition.

Kleffmann et al. [27] developed a traceability approach for recognizing the informal hand-drawn sketches by creating the relationships between sketch elements. This is achieved by Augmented Interaction Room (AugIR), a combination model of fuzzy search and information retrieval technique of vector space model (VSM). This traceability approach achieves a precision of 92.74% and a recall of 90.04%. However, the relationships between the elements of sketches may sometimes create faults, thus causing trace link recovery and decrease in recall values. Wang et al. [28] designed and developed SketchPointNet, a

novel point-based deep network with a compact architecture for highly robust sketch recognition. This approach considerably reduces the model space and the computation complexity and ensures a high accuracy of 74.22% with minimal network parameters. Seddati et al. presented three models of DeepSketch [29–31], a deep convolutional neural network (CNN) based approach for sketch recognition with similarity search using KNN. DeepSketch [29] approach provided recognition accuracy of 75.42% which is high enough for effective similar image search applications. DeepSketch 2 [30] was an enhanced model of DeepSketch for the partial sketch recognition with an accuracy of 77.69% while DeepSketch 3 [31] was also developed with the use of different layered features of deep CNN for sketch-based image retrieval application with a sketch recognition accuracy of 79.18%.

Sarvadevabhatla and Kundu [32] proposed the use of a deep gated recurrent neural network-based framework for recognizing the sketches by controlling the deep features and weighted timestamp loss. This approach provided satisfactory results, yet the accuracy of deep feature extraction is still not perfect. Jahani-Fariman et al. [33] proposed a block sparse Bayesian learning approach called MATRACK for sketch recognition with an average accuracy of 96.6%. This approach has higher accuracy and is robust, but the only limitation is the slower recognition time due to the extended learning process. He et al. [34] proposed the use of deep visual-sequential fusion model for sketch recognition. This model captures the intermediate stroke states using the spatial and temporal features using layers of sequential networks by residual long short-term memory (R-LSTM) units. The fusion of the visual and sequential features increases the accuracy of sketch recognition. But the problem with this approach is that the semiconstrained sketches were only detected while inaccurate sketches with rough strokes were filtered before recognition to avoid accuracy degradation.

Boyacı and Sert [35] developed a feature-level fusion of CNN for sketch recognition in smartphones. This approach employed the multiple layers of CNN to extract the features of sketches using Alex-Net and VGG19 CNN architectures with a fusion operator. The abstraction of the sketches is captured, and the best fusion scheme is used in client-server application with an average accuracy of 69.175%. However, these results are second best to the Sketch-a-Net scheme by Yu et al. [36]. Sketch-a-Net employs the deep neural networks for effective recognition of sketches by exploiting the sequential ordering information and designing a deformation model to synthesize new sketches. This approach has better recognition performance due to the consideration of the unique properties of the sketches. In order to outperform Sketch-a-Net, Sert and Boyacı [37] proposed an efficient freehand sketch recognition approach using transfer learning models based on the feature-level fusion of CNN along with CNN-SVM pipeline architecture. The principal component analysis (PCA) is utilized to reduce the fused deep feature dimensions and increase the overall recognition accuracy to 73.1% on the TU-Berlin dataset [1] for smartphone applications. Zhang et al. [38] proposed a

hybrid CNN approach for sketch recognition. This hybrid CNN consists of two CNN, namely, Alex-Net and S-Net for considering the features of both appearance and shape, respectively. Compared to other models, this hybrid CNN is efficient in extracting discriminative shape features which increases the accuracy of sketch recognition and sketch-based image retrieval processes. This approach increased the recognition accuracy by 2–5%, but the drawback of this supervised hybrid CNN is the requirement of expensive labelled data and the semisupervised CNN models may reduce this limitation.

From the literature, it can be inferred that the deep learning models, especially the CNN, have been exploited most for the freehand sketch recognition. However, there are limitations and room for development in these models. Another aspectual concept is the utilization of machine learning algorithms for recognition which has been of greater importance in image processing-based sketch recognition. SVM is one such algorithm, but due to its long training time and inability to select a perfect kernel, the SVM model has provided below-par performance. In this work, a modified SVM is presented to overcome these limitations. A hybrid kernel is developed for this purpose along with the lion optimization algorithm to reduce the training time. This model of modified SVM is deemed to exceed the performance of SVM and is incorporated into the proposed SKETRACK recognition scheme.

### 3. Sketch Structures and Formats

The SKETRACK sketch recognition system has been developed to support several domains, and for achieving this objective, small modifications in retraining the classifiers are required. The domains employed in this research for evaluation of the proposed SKETRACK and the online databases used are described. The line-based diagrams and the arrow-based diagrams are utilized in which the symbols are connected by lines and arrows. The main reason for selecting these sketches is due to their uniform structures of arrows and symbols. The sketches consist of arbitrary strokes and text labels with uniform symbols and domain syntax. Three diagram domains, namely, flowcharts (FC), finite automata (FA), and digital logic circuits (DLC), are utilized in this work. FC, FA, and DLC sketches are utilized in the SKETRACK with specified features of these sketches derived for the evaluation. These sketches are employed as online input to the recognition system.

**3.1. Flowcharts.** Flowcharts are arrow-based sketches consisting of five uniform symbol classes, namely, data, decision, process, connection, and terminator. FC databases are widely available in different formats for research purposes. As annotations are mainly required for extracting the important properties of FC, the database is selected based on annotations and temporal information. The FC database utilized for evaluation is available in [25]. The database contains a total of 672 sample sketches of 28 sketch patterns drawn by 24 users. These sketches are classified into subsets

for training and testing. The database contains annotation of symbols and relations between them while the arrows with connection points and heads are also provided. The meaning of each text block is also provided.

**3.2. Finite Automata.** Finite automata are also arrow-based sketches which consist of three uniform symbol classes, namely, state denoted by a single circle, final state denoted by two concentric circles, and arrows. The FA database also contains text blocks of single-letter names for the states. The FA database is available at [24]. The database contains a total of 300 sample sketches of 12 sketch patterns drawn by 25 users which are categorized for training and testing separately.

**3.3. Digital Logic Circuits.** Digital logic circuits are line-based sketches that contain three uniform symbol classes, namely, bubble or a circle, normal gates, and concentrated curve gates (like X-OR). The symbols considered for evaluation are bubble, OR, AND, NOT, NOR, NAND, and X-OR. The text blocks in this database are just single letters naming the inputs A, B, and C and output Y. Figure 1 shows example DLC with all the symbols considered for this work and the X-NOR and complex circuit board diagrams avoided for complexity.

The DLC database is minimally available as the sketches employing the basic logic operations are quite less. The X-NOR and other complex structured sketches are avoided in this work as it convolutes the description of the SKETRACK scheme. Figures 1(c) and 1(d) show these complex sketches. The X-NOR gate is similar to the X-OR, but it includes additional symbols to visualize the NAND property and this structure convolutes the recognition in SKETRACK. Similarly, the complex structure in Figure 1(d) includes the complete circuit board structure with the logic symbols. The connector lines increase the complexity of correctly identifying logic symbols. These structures require extensive analysis which is complex and time consuming than other symbols. These symbols are planned to be recognized using highly advanced techniques in future works and hence deliberately avoided in the current research. The DLC sketches for this research are extracted from the IAMonDo Database [39]. A total of 150 sample sketches are collected based on 10 different sketch patterns by 15 users which are categorized for training and test separately.

## 4. SKETRACK Freehand Sketch Recognition Scheme

The proposed stroke-based online freehand sketch recognition scheme (SKETRACK) employs the concepts of text segmentation, feature extraction, and symbol candidate recognition using the MSVM classifier. In SKETRACK, initially the normalization of the input ink files of the FC, FA, and DLC domains is performed to correct the edges and noises. Then, the text segmentation process is performed to separate the text blocks from the sketch symbols. Text strokes and the symbols are recognized separately. This

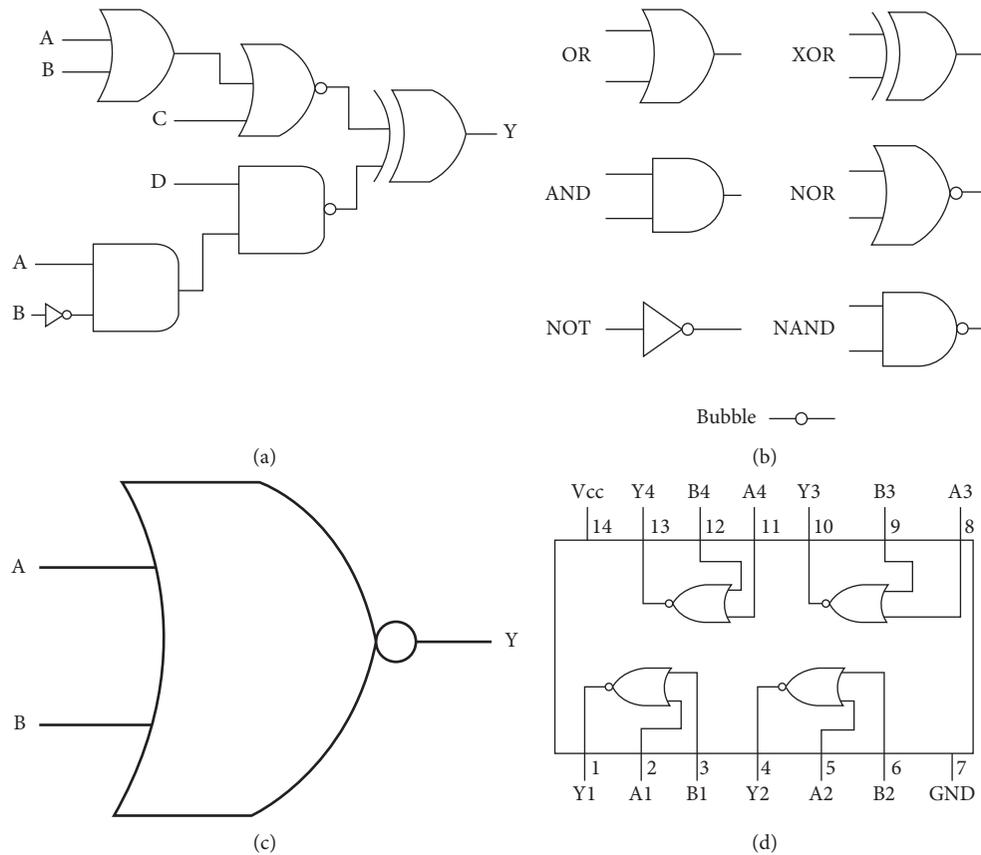


FIGURE 1: (a) Example of freehand DLC, (b) structured symbols, (c) X-NOR gate, and (d) complex structure.

process significantly minimizes the computation complexity. Then, the symbol strokes are modelled with the help of a feature extraction where the different variations of the structure are stored and the new strokes are compared with them to form its respective diagram structure. Then, the symbol strokes are clustered using spectral clustering to group similar continuous strokes to model a specified diagram part with dimensionality reduction using the p-distance and Euclidean distance. Finally, the symbol strokes are recognized using the MSVM classifier. These recognized symbols are analysed for the diagram structure, and the text strokes are replaced to its original positions to recreate the original sketch. The complete recognition pipeline of SKETRACK is given in Figure 2.

**4.1. Normalization and Text Recognition.** The input sketches in the ink files contain sketch symbols, annotation, and the relations between them. Normalization of these files is effective in minimizing the least accurate diagrams. The input files may not be accurate as the strokes might have been incomplete or corrupted from the user side or during the creation of ink files. This can be rectified by normalizing the ink files to correct the edges and noises. Also, the normalization of texts is performed along with this step to improve the text segmentation process. The texts might be in a noncanonical language or usual speech transcription which is considered as nonstandard forms. Hence, the

normalization process standardizes the texts to standard canonical forms as presented in globally recognized dictionaries and lexicons such as Oxford. Once the normalization is complete, the text strokes are separated for processing isolated from the processing of diagram strokes. Figure 3 shows the text separation process through normalization and text classifier.

The normalization includes the process of mapping the input diagram to a standardized image plane in the predefined square shape. It controls the aspect ratio of the text in the diagram through 14 basic normalization steps in the form of linear and moment normalization with different processes of the aspect ratio. This results in the filling and centring the horizontal and vertical dimensions. Then, the text features are extracted using a text extractor and the text separation process isolates the texts and shapes separately. The text separation is performed using deep learning recurrent neural networks (DLRNN) [40] as it has been found to be most efficient in classifying the text data. DLRNN utilizes the internal memory for processing the input strokes. It has greater applications in the recognition process of handwriting, speech, and objects. It forms a loop of information cycle and makes a decision using short-term memory. This concept is briefed in order to provide a higher focus on symbol recognition. DLRNN has a higher accuracy of 99.05% for shape classes and 98.94% for text classes. This process is applied to all three domain databases. DLRNN has accuracy in shape/text class of 99.56/97.7, 98.8/99.76, and

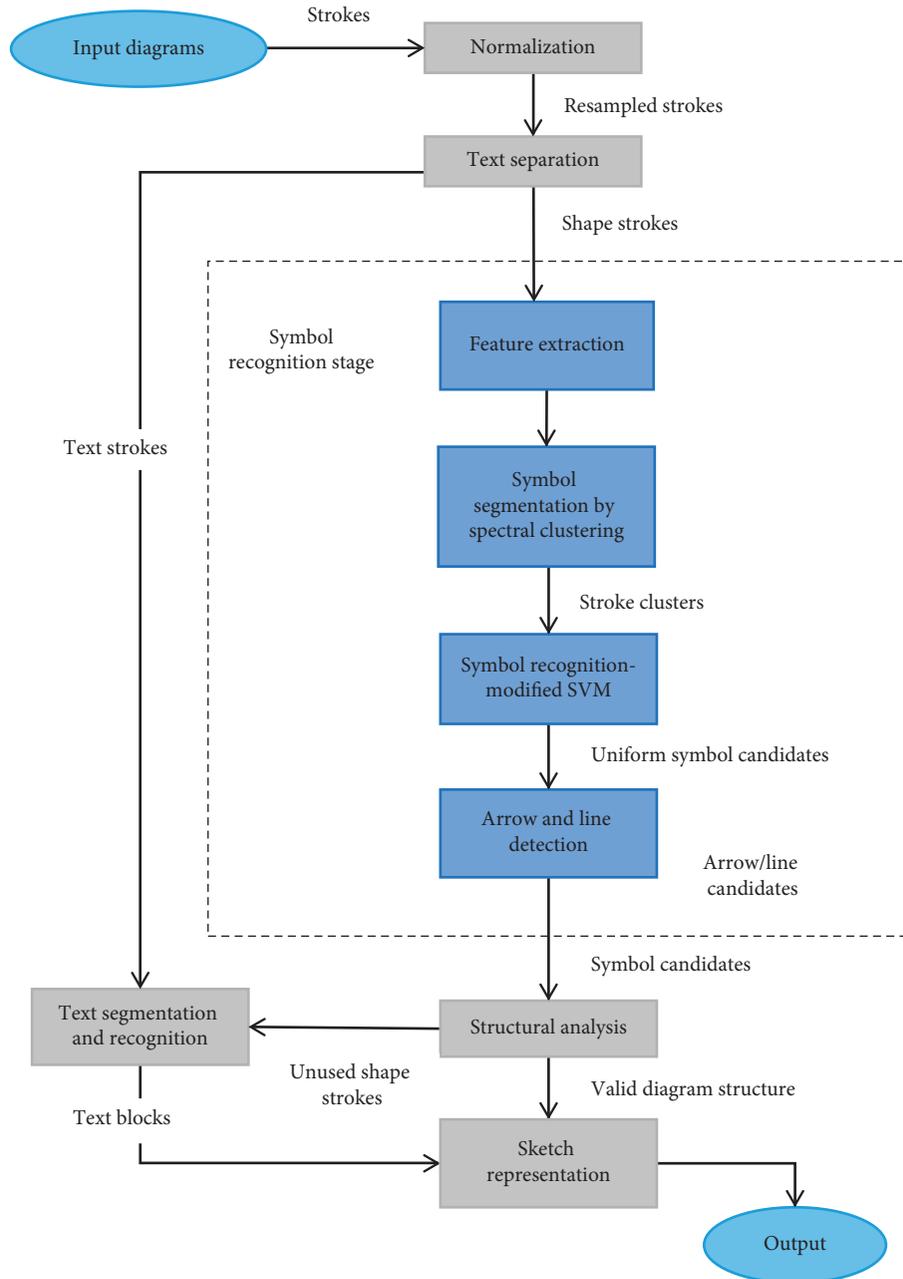


FIGURE 2: SKETRACK recognition pipeline.



FIGURE 3: Text separation process.

99.12/94.88% for FC, FA, and DLC databases, respectively. Figure 4 shows the input FC sketch and its text segmentation results. It can be seen that there is a symbol and an arrowhead detected along with the text. This error is due to the low quality of the input image and will be minimized in the uniform symbol classifier.

The text recognition is performed after the completion of the symbol recognition and structural analysis. The isolated strokes of text during the text separation process are processed with the corresponding sketch structure analysis. The text blocks are processed in two types: arrow text labels and labels for uniform symbols. First, the uniform symbol labels

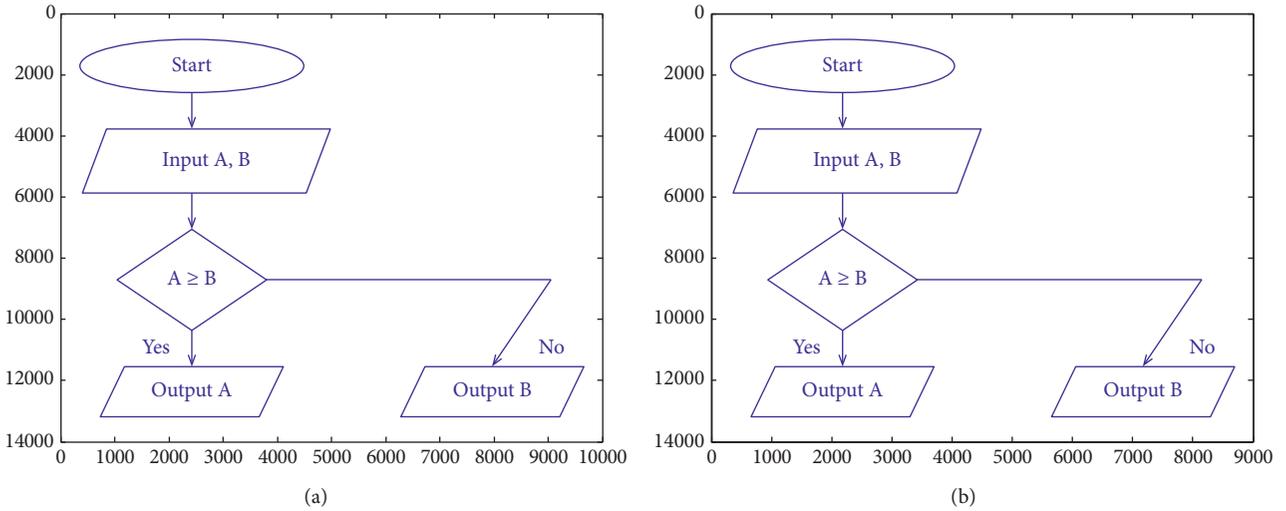


FIGURE 4: Text segmentation of FC sketch. (a) Input FC and (b) text segmented FC.

are detected followed by the arrow labels. Finally, the text blocks are assigned to the nearest arrow and symbols based on matching.

**4.2. Feature Extraction.** The feature extraction model contains the strokes of similar structures with different variations. The semantic strokes are initially fixed for each of the symbols of the FC, FA, and DLC databases based on training data. Then, the different possible variations resembling each symbol are gathered to form a cluster. Some symbols like OR and NOR in DLC may seem much similar, and hence, there occurs a possibility of misrecognition. Figure 5 shows the feature extraction process of the symbol strokes.

The sketches contain all necessary symbols, and some symbols will have been used more than once in the same sketch. In such cases, the system might recognize the repeated symbol and considers only one symbol for clustering. This will create a different sketch at the final classifier. Hence, the clustering approach is tuned to select both the symbols and group them together so that the sketch is complete when restructured. First, the diagrams with strokes are converted into binary form and denoising is applied to eliminate disturbance from the noisy pixels. Then, the strokes are determined by their colours and skeletonizing is applied to retrieve the stroke features. The matching concept is used to group these strokes when there is more instance of the same symbol. For highly effective grouping, the high dimensionality features must be reduced and an effective clustering algorithm has to be applied. A total of 29960 training samples are employed, and among them, only optimal features are selected based on this concept. The higher number of feature variables will be time consuming as well as incompetent as most of the features have little information about the symbols. Hence, the number of features must be reduced and this process is called dimensionality reduction. For the purpose of reducing the

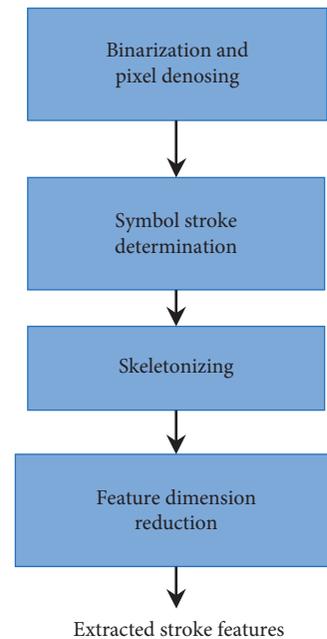


FIGURE 5: Feature extraction of symbol strokes.

feature dimension, the p-distance [41] and Euclidean distance [42] are utilized. p-distance is the proportion of the stroke at which two sequences of compared symbols are different. It is computed as follows:

$$p - \text{distance} = 1 - \frac{\text{similar features}}{\text{total features}}. \quad (1)$$

Similarly, the Euclidean distance between two features is the measuring length connecting these two features and can be computed to evaluate the dimension of the features. Euclidean distance between two features  $p$  and  $q$  with  $i$  number of variations can be computed as follows:

$$\text{Ed}(p, q) = \text{Ed}(q, p) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}, \quad (i = 1, 2, \dots, n). \quad (2)$$

Based on these parameters, the dimension of the features is minimized. Then, the clustering/segmentation of the symbol strokes is performed using spectral clustering.

**4.3. Symbol Segmentation Using Spectral Clustering.** The strokes are mapped on a graph, and the clusters are formed as said in the feature extraction model. The symbol strokes are defined on a weighted undirected graph  $G$  of quad tree  $T$ . The topology of this graph is defined to auxiliary stroke structure for navigation and trajectory generation. The vector  $V = \{v_1, v_2, \dots, v_n\}$  is the collection of all leaf nodes of the quad tree  $T$ , and  $E = \{e_{ab} \mid a, b \leq n\}$  is the collection of all connections of the leaf nodes. The incidence matrix  $I$  is estimated for the similarity between the strokes and is used on the degree of connection in this graph model. Figure 6 illustrates the steps involved in the symbol segmentation process using the spectral clustering algorithm.

For segmentation of the graph  $G$ , it must be modified into sparse graph using a threshold defined as

$$\text{Th}_{I_{a,b}} = 1.05 \frac{\max(x_{\text{map}}, y_{\text{map}}) * (2^{I_a} + 2^{I_b})}{2^{I_a} 2^{I_b}}, \quad (3)$$

where  $\text{Th}_{I_{a,b}}$  is the threshold of similarity matrix “ $T$ ” of graph  $G$  between two points  $(a, b)$ ,  $x_{\text{map}}, y_{\text{map}}$  is the mapping location, and 1.05 is the coefficient added to avoid the floating point computation error.

Now the quad tree of stroke segmentation problem is modelled into the weighted unidirectional graph segmentation problem. Spectral clustering [43] segments this graph using generalized Laplace matrix formation. Laplace matrix  $L_{a,b}^I$  is defined by the following function:

$$L_{a,b}^I = \begin{cases} 1, & a = b, \deg(v_a) \neq 0. \\ \frac{1}{\sqrt{\deg(v_a)\deg(v_b)}}, & a \neq b, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

When  $k$  number of submaps is given, then the  $k$  minimum eigenvectors  $ev_1, ev_2, \dots, ev_n$  of the  $L_{a,b}^I$  must be determined using the auxiliary matrix  $P$ :

$$P = [ev_1, ev_2, \dots, ev_n] = \begin{bmatrix} ev_{11} & \cdots & ev_{k1} \\ \vdots & \ddots & \vdots \\ ev_{1n} & \cdots & ev_{kn} \end{bmatrix}. \quad (5)$$

Each row of this auxiliary matrix  $P$  is considered as the corresponding sample of symbol strokes. The  $k$  sample clustering provides the spectral clustering result:

$$\text{stroke}_i = \{P_{ab} \mid 1 \leq a \leq k, b \in [1, n]\}. \quad (6)$$

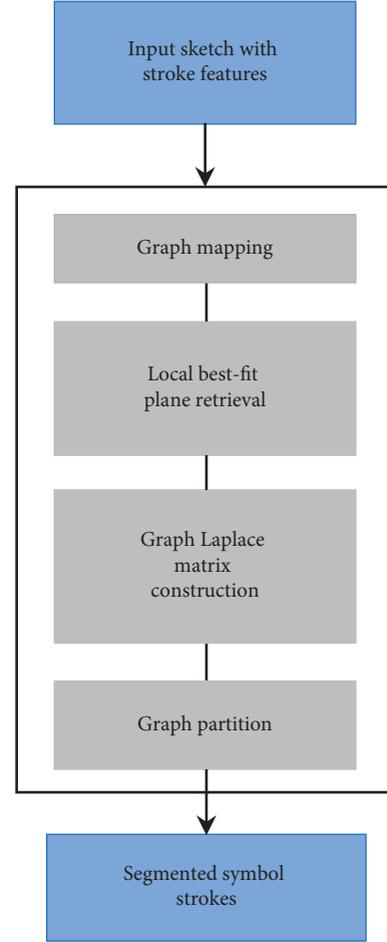


FIGURE 6: Symbol segmentation process.

To determine the value of  $k$ , a score function is estimated using any distance function. As said in the feature extraction model, the Euclidean distance is used to determine the value of  $k$ . Euclidean distance between two clusters  $A_i$  and  $B_i$  is computed using the following equation:

$$d(A, B) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}. \quad (7)$$

Thus, the value of  $k$  can be set and the graph  $G$  can be segmented effectively. This graph directly provides the stroke clusters of the symbols. These clusters will be recognized using the classification for final structure analysis.

**4.4. Symbol Recognition Process.** Symbol recognition is performed to classify the subsets of stroke clusters generated at the end of the segmentation process. The main process is to assign a symbol class to each cluster. The symbol recognition process consists of two steps: arrow detection and uniform symbol classifier. The arrow detection is carried out by recognizing them as connectors between the uniform symbol candidates. The uniform symbol is recognized using the MSVM classifier. Both the arrow detector and uniform

symbol recognizer provide list of symbol candidates from which the actual symbols are selected by the structural analysis process.

**4.4.1. Symbol Classifier Using MSVM.** The classifier has the function of assigning a symbol class to each of the clusters. The traditional multiclass SVM has been utilized for this objective in some of the recent research studies. However, the SVM classifier has some disadvantages when used for recognition purposes. The major limitations are the longer training time and the inefficient selection of kernel function. In order to overcome these challenges, the MSVM model is presented which utilizes a hybrid kernel function based on a Gaussian kernel and a polynomial kernel. Hybrid kernel concept has been already utilized by many authors [44], but the hybrid kernel in this work is novel function. Then, the SVM parameters are optimized by using the lion optimization algorithm. The lion optimization algorithm has dual purpose in the SKETRACK scheme. First, it is used for optimizing the SVM parameters, and then, it will be used for optimizing the max-sum problem in structural analysis. Figure 7 shows the process involved in the proposed MSVM algorithm whose parameters are optimized using lion optimization.

A dynamic feature descriptor based on the normalization is utilized with the MSVM classifier. The MSVM classifier is modelled with a hybrid kernel and is much suitable for the DLC database and any type of arrow-connected diagrams. The MSVM works on finding the optimal hyperplane that separates the stroke clusters. For deriving two hyperplanes for graph of clusters  $x_1, x_2, \dots, x_n$  where  $x$  is the graph point indicating a cluster is described as

$$\begin{aligned} w_1 \cdot x + c_1 &= 1, \\ w_2 \cdot x + c_2 &= -1, \end{aligned} \quad (8)$$

where  $c_1$  and  $c_2$  are scalar learning parameters while  $w_1$  and  $w_2$  are the p-dimensional weighting vectors perpendicular to the separating hyperplane. The values of  $c_1$ ,  $c_2$ ,  $w_1$ , and  $w_2$  are selected such that the training time is shorter.

Selecting a proper kernel function is necessary to improve the recognition performance. The commonly utilized kernels are linear kernel, polynomial kernel, radial basis function, and sigmoid tanh kernel. However, these kernel functions do not provide best learning ability to SVM. Hence, in this work, the MSVM utilizes a hybrid kernel by combining the Gaussian kernel and polynomial kernel. The Gaussian kernel is developed based on the radial basis function, and thus, constructed hybrid kernel  $K_{RP}$  is given as

$$K_{RP} = \lambda K_P + (1 - \lambda) K_R, \quad (9)$$

where  $K_P$  is a polynomial kernel which is formulated for clusters  $x_1, x_2$  with distance  $d$ , and random value  $r$  as

$$K_P(x_1, x_2) = \left( \text{poly} \sum_{i=1}^n x_{1i} * x_{2i} + r \right)^d. \quad (10)$$

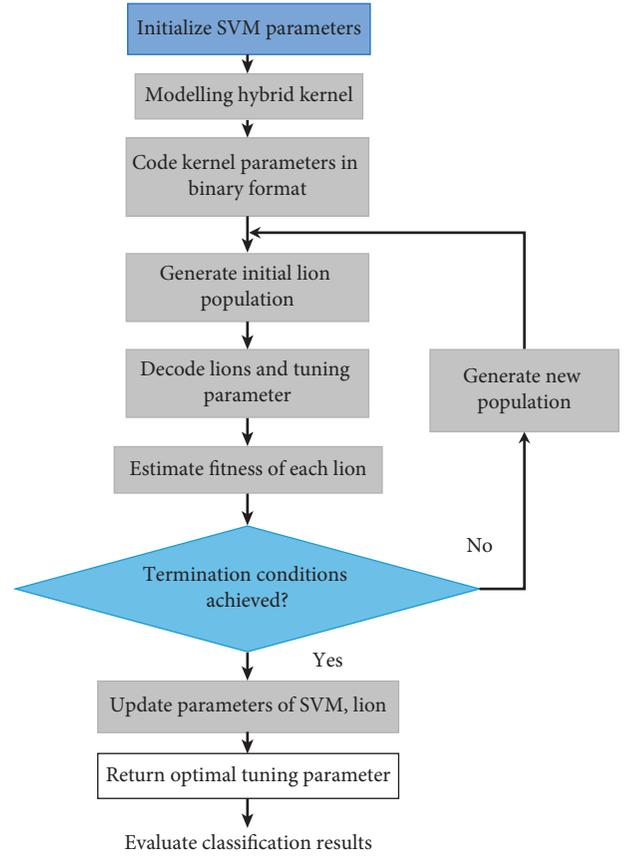


FIGURE 7: Parameter optimization process in MSVM.

Similarly, the  $K_R$  is the radial basis function which can be formulated with a Gaussian parameter  $g$  as

$$K_R(x_1, x_2) = \exp\left(-g \sum_{i=1}^n (x_{1i} - x_{2i})^2\right). \quad (11)$$

However, the radial basis function  $K_R$  is a local kernel and does not provide global solutions. In order to provide a global solution for optimal hyperplane selection with faster convergence, the hybrid kernel employs the tuning parameter  $\lambda$  which also tunes the influence the two individual kernels  $K_P$  and  $K_R$  in the hybrid function  $K_{RP}$ .

For further enhancing the performance of MSVM, the parameters are optimized using lion optimization [45]. The tuning parameter  $\lambda$  is optimized by modelling it as lions of the traditional lion optimization algorithm. The hunting behaviour of the lions is formulated for this work. The solution search operation of MSVM has to be improved, and hence, a parameter  $Q$  is introduced to adjust the speed of moving lions based on the preys' position. The positions of the lions are also updated based on the change of position by prey. If the hunting lion is odd, the value of  $Q$  is set as “-1” making the lion move in the opposite direction of the prey. Similarly, if the hunting lion is even,  $Q$  value is set as “1” and the lion continues to move in the same direction. The speed of movement and position of the lion can be formulated by

$$\begin{aligned} M_{id}(t+1) &= Q\{\omega \times M_{id}(t) + c \times \text{rand}() \times X_{id}(t), \\ X_{id}(t+1) &= X_{id}(t) + S_{id}(t+1), \end{aligned} \quad (12)$$

where  $d$  is the distance of two cluster points in problem space,  $X_{id}(t)$  is the previous position of lion based on position of prey while  $M_{id}(t)$  is the previous speed of lion,  $\omega$  is the weighting vector,  $\text{rand}()$  is the random value between  $[0,1]$ , and  $c$  is the scalar learning factor. The fitness in this work is the classification accuracy. Based on these concepts, the MSVM overcomes the limitations of SVM and provides higher uniform symbol recognition accuracy.

The major modification of the proposed MSVM from the traditional SVM is the hybrid kernel function with the optimized tuning parameter  $\lambda$ . The lion optimization selects from the range  $0 < \lambda < 2$  in our experiments. In practical cases, these values can be set between 0.25 and 1.99 such that optimal classification is obtained. Larger value of  $\lambda$  results in better classification, but it is not the same for all databases, and based on the database features, the value varies. In some cases, the larger value of  $\lambda$  will be less than 1 depending on the diagram instances, and hence, the lion optimization selects adaptive values suitable for each instance. The classification accuracy obtained by selecting different optimal values of  $\lambda$  for the three domains of diagrams is given in the Experiments section.

**4.4.2. Arrow/Line Detector.** Most existing models of arrow/line detection utilized the arrow subclasses to detect them, but the results were not satisfactory. As it is difficult to recognize the arrows based on its appearance, the special properties of the arrows are exploited in this work to differentiate them. The most special property is that it connects two symbols, and once the uniform symbols are detected by MSVM classifier, the connecting elements are analysed. Figure 8 illustrates the steps performed in the proposed arrow/line detector.

The arrow detection is performed in two steps. The first step is the detection of the shaft (line) of an arrow that connects two symbols with a sequence of strokes from the first symbol to the second symbol. This concept of detection is enough for a line connecting two symbols in a DLC also. The pairs of symbols are detected using the MSVM classifier, and the arrow shafts connecting them are detected. For this purpose, the strokes in vicinity of a symbol are found by increasing the size of bounding boxes. This will increase the search space and provide many symbol pairs. Then, it has to be determined whether the symbol stroke is closer in the search space and is properly fit with the shaft length and the open connector length, and the smallest distance between the two symbols is computed and is compared with a general distance threshold  $d_{\text{thresh}}$ .

Once the connectors are identified, they must be formed into arrows. However, for detecting the arrows, the second step is very important. The shaft/lines have no direction, and only the arrowhead defines the direction. The second step is the detection of arrowhead at the end of the second symbol. The shaft without a head is considered as a simple line while the shaft with a head is defines as arrow. The shaft detection

is performed by adding the iterative stroke sequences. The spatially closer strokes of arrowheads identified in the previous step are then analysed to find any interference with other strokes. The detected arrowhead strokes must be interfered with other symbol strokes. To validate the arrowheads, the arrowhead candidate list is generated and the arrowhead with highest confidence is selected as in [24]. The confidence value of a symbol is defined as a number assigned to each symbol in a sketch showing how much that sketch is trusted with the current symbol. Confidence of an arrowhead is computed as the product of confidence of heads' bounding box shape and distance of the head centroid to the end point. It is estimated as

$$\begin{aligned} \text{confidence} &= \text{shapeConf} \cdot \text{distanceConf}, \\ \text{shapeConf} &= \exp(-(z-h) \cdot (z+h)), \end{aligned} \quad (13)$$

$$\text{distanceConf} = \exp\left(\ln(0.5) \cdot \frac{d_{\text{centroid}}}{d_{\text{thresh}}}\right),$$

where  $z$  denotes the width and  $h$  denotes the height of the head's bounding box;  $d_{\text{centroid}}$  is the distance of the bounding box's centroid to the endpoint of connector and  $d_{\text{thresh}}$  is the general distance threshold.

The arrowhead direction can be determined by the arrowhead's association with the end point of the connector. The confidence of an arrow can be estimated as the product of confidence of connector and confidence of head's confidence. The confidence of a connector is based on the connector distance which is a sum of distance between connector's endpoints and connector points of symbols and distance between the consecutive connector strokes. This can be formulated as

$$\text{connectorConf} = \exp\left(\ln(0.5) \cdot \frac{d_{\text{connector}}}{d_{\text{thresh}}}\right), \quad (14)$$

where  $d_{\text{connector}}$  is the connector distance. Based on these confidence values, the direction of the arrowheads is determined.

**4.5. Structural Analysis.** Structural analysis is literally the final step in SKETRACK for recognizing the sketches. The symbol candidates detected by the uniform symbol classifier of MSVM and arrow detector are fed as the input to the structural analysis module. The structural analysis task is for recognizing the subset of symbol and arrow/line candidates that can be grouped to form a valid sketch of FC, FA, or DLC. For connecting these sets of symbol candidates, the relation between symbols has to be determined by score values. The score is computed upon three fundamental types of relations. The first type is the conflict relation when two symbol candidates share one or more strokes or arrows at the same connection point. The second relation is the overlap in which two symbols have overlapping boundaries. The third relation is the end point in which the arrow/line has two symbols at the end points. The first and second relations are effective and both symbol candidates are

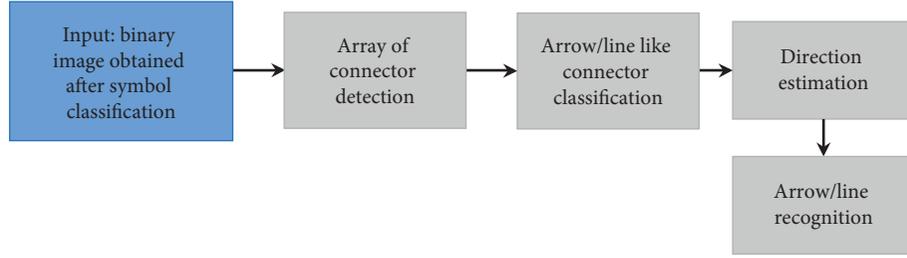


FIGURE 8: Arrow/line detection.

selected while the third relation is effective if the arrow is selected for structure solution. The conflicts and the end point relations can be determined through estimation properties while the score is necessary for computing the overlap relation. The confidence score function is modelled for the first two relations of conflict and overlap [25]. Figure 9 shows the final output of the FLC sketch recognized by the proposed SKETRACK.

For conflict relation, the scores are negative infinity, i.e.,  $s_1 = -\infty$  while for the overlap relation it is given as

$$s_2 = -S_{R \cap S} / \min(S_R, S_S), \quad (15)$$

where  $R$  and  $S$  are the bounding box of the first and second symbols while  $S_R$ ,  $S_S$ , and  $S_{R \cap S}$  are surfaces of  $R$ ,  $S$ , and  $R \cap S$ , respectively.

Once the scores are computed for the relations between the symbol candidates, the diagrams are represented. The diagrams are represented based on the confidence scores computed. First, the symbol candidates are mapped as nodes of undirected graph  $G = (V, E)$  and the labels  $K=0$  and  $K=1$  are used to select the symbol candidate. The graph cost functions can be formulated by modelling the pairwise max-sum labelling problem using the maximizing sum of unary and binary cost functions:

$$\max_k \left[ \sum_{u \in V} g_u(k_u) + \sum_{\{u,v\} \in E} g_{uv}(k_u, k_v) \right], \quad (16)$$

where  $K$  is the set of finite labels,  $g_u(k_u)$  is the unary cost,  $g_{uv}(k_u, k_v)$  is the binary cost, and  $\{u, v\}$  is the edge. By simplifying this function, the diagram structure is finally represented. However, this max-sum problem is NP-hard and hence needs to be solved using any optimization algorithm. As stated before, the lion optimization algorithm is employed to resolve this problem. First, the max-sum problem in equation (16) is modelled into optimization problem and lion optimization is applied. The hunting behaviour and the nomadic movement of lions are explored to optimally select the solution for this problem in polynomial time. Thus, the diagrams can be recognized and represented as in the original input sketches. The performance of the proposed SKETRACK will be evaluated in the experiments section.

## 5. Experiments and Results

The experiments are performed using MATLAB on the online sketches from FC, FA, and DLC databases. The

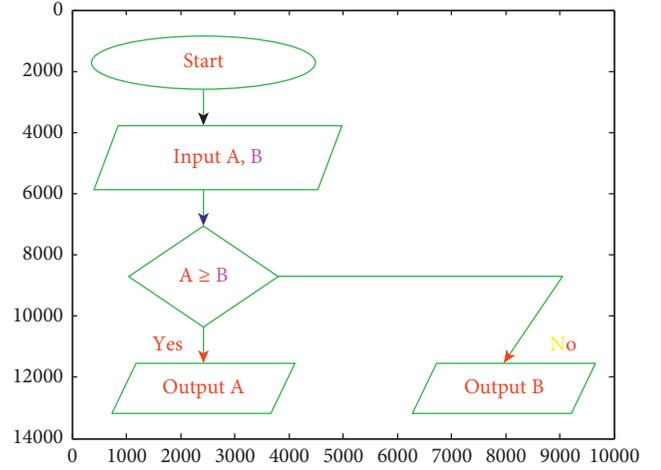


FIGURE 9: Final recognized FC sketch.

proposed system has been implemented in a computer with minimum requirements of Intel i3 processor, 4 GB RAM with Windows 10 operating system. The performance of the MSVM-based SKETRACK is evaluated and also compared with the existing methods. Also, the influence of the hybrid kernel function is also illustrated by estimating the symbol classification accuracy of the MSVM with optimal values of  $\lambda$  for FC, FA, and DLC sketches.

**5.1. Evaluation of SKETRACK Classification Accuracy with respect to  $\lambda$ .** For estimating the symbol classification accuracy of MSVM when using optimally selected  $\lambda$  value instead of fixed tuning parameter, the parameters must be set. The parameters include the scalar and weight parameters of SVM and the movement parameters of lions. The population of lions is set as  $N=50$ , the maximum iteration is set as 100, the maximum value of target location  $X_{id, \max} = 1$ , the minimum value of target location  $X_{id, \min} = 0$ , and the speed of movement  $M_{id, \max} = (X_{id, \max} - X_{id, \min}) \times 0.1$ . The scalar learning parameter  $c_1 = c_2 = 1.4962 \approx 1.5$ , and the weight vectors are selected as  $w_1 = 0.9$  and  $w_2 = 0.4$ . The initial optimal value will be selected for each database randomly, and the best value will be returned only when the classification accuracy stays constant for at least 20 iterations. Table 1 shows the optimal  $\lambda$  value obtained by lion optimization and corresponding classification accuracy for the FC, FA, and DLC databases.

TABLE 1: Classification accuracy with respect to optimal  $\lambda$ .

Database	Optimal $\lambda$	Accuracy (%)
FC	1.8542	95.87
FA	1.8496	94.99
DLC	1.9277	79.21

When the optimal values is selected for  $\lambda$ , the manual setting process for tuning of the SVM classifier is avoided and the classification accuracy is also increased due to the optimal as well as adaptive selection of  $\lambda$  value for each diagram database.

*5.2. Comparison of SKETRACK Recognition Results with Other Methods.* The performance of the MSVM-based SKETRACK is evaluated on online diagram databases. The results obtained on FC and FA diagram instances are compared with that of three state-of-the-art methods: multiclass SVM [24, 25], CNN [31], and CNN-SVM [37] while SVM [46, 47] is only compared with the proposed MSVM results for the DLC diagrams due to lack of more existing methods. The two main reasons for the less research on DLC sketches and their limited capacity are as follows: (1) the complexity of the DLC sketches and (2) the inability to adapt with the quality of input employed. The comparison results on the three online diagram databases are given in Tables 2–4.

From Table 2, it can be seen that the proposed MSVM classifier has comparatively better recognition performance than the other methods for the FC instances. The accuracy of MSVM is 95.87% which are increased by 1.11% than CNN-SVM while also outperforming SVM and CNN methods. MSVM also provides higher precision of 94.96% but slightly less values of recall and  $F$ -measure than CNN-SVM; however, overall, it provides comparatively better results than other models.

From Table 3, it is revealed that the accuracy, precision, recall, and  $F$ -measure of the proposed MSVM classifier are significantly higher than the other methods for FA diagrams. The accuracy value of MSVM for FA database is 94.99% which is increased by 1.77% than CNN-SVM. The precision value of MSVM is 94.31% which is increased by 0.36%. Likewise, the recall value is 94.13%, increased by 1.97% than CNN-SVM and  $F$ -measure value increased 1.4% than CNN-SVM for FA. It is also noted that MSVM outperformed SVM and CNN models which can be indicative that the MSVM is comparatively the better classifier.

Table 4 illustrates that the accuracy, recall and  $F$ -measure of the proposed MSVM classifier are significantly higher than the other methods for DLC. The IAMonDo database [39] contains high number of hand-drawn sketches, but the quality of around 50%–75% and this quality metric can impact the performance. The accuracy, recall, and  $F$ -measure values are increased by 5.8%, 1.6%, and 1.29%, respectively, while the precision value is reduced by 0.3% than SVM-1 for DLC. The difficult processing for recognizing the DLC sketches is reflected through these results. The important point from these results is that the DLC sketches

have been recognized with greater performance than the existing research models.

The recognition precision of the proposed SKETRACK is evaluated based on two metrics, namely, stroke labelling (SL) and the symbol segmentation and classification measure (SR1). These two metrics help in evaluating the correctness of the recognition of each unique symbol and the arrows/lines effectively. The results for the three databases are evaluated separately. Tables 5–7 show the recognition results for the online FC, FA, and DLC databases.

It can be seen that the proposed MSVM-based approach of SKETRACK has significantly good performance. The performance in some of the classes varies from other pivotal classes due to the introduction of noise and missing of temporal information. The results of the DLC database is around 5–10% less than the FA and FC databases. However, the results are promisingly better because the other existing SVM-based models for DLC have provided much worst performance. The performance improvement in the online and offline FC, FA, and DLC databases is highly efficient because of the improved classification of SKETRACK.

Table 8 lists the minimal, maximal, average, and median time required to resolve the max-sum problem using lion optimization in selecting the lambda values of the hybrid kernel while Table 9 shows the minimal, maximal, average, and median of the total time. This table helps in identifying the cost of performing the optimization task additionally. It can be seen that the time is optimal, and hence, the additional overhead is minimized.

From Tables 8 and 9, it can be seen that the time taken to solve the optimization problem is only a small fraction of the total time, and hence, the fast processing is performed. The optimization process does not create additional cost or overhead to the proposed system.

When analysing the system performance, the misrecognition is also a major criteria. In this work, although best techniques are utilized, the system fails to detect some of the sketches due to the low quality of the input. In some cases, the text blocks have been misinterpreted as symbols. In other cases, the user who drawn the diagram has utilized fast skills that are not recognizable. Some texts are not recognizable due to the conflicts in the font style.

Figures 10–12 show the examples for misrecognition results. The FA and FC sketches are recognized and the result are coloured to differentiate the misrecognized symbols. The colour green indicates correct symbol recognition, red indicates text, and blue indicates the incorrect recognition. In Figure 10, all the symbols are recognized correctly except three. The oval shape containing the start is misrecognized; the end word is partially recognized and the end oval shape is recognized as text due to conflicts in recognition. Similarly, the DLC sketch in Figure 12 has colours denoting the symbols, namely, brown for OR, violet for AND, green for NOR, yellow for NAND, blue for X-OR, pink for Bubble, and black for connector lines. But the final result has misrecognition in bubbles at NAND and NOR symbols. Also the NOT gate with bubble is completely recognized as bubble. Apart from these misrecognitions, it can be seen that the sketches have been recognized with an average of 80%

TABLE 2: Recognition performance results for FC database.

Methods	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
SVM [25]	84.76	83.45	81.6	86.5
CNN [31]	90.66	89.71	88.72	89.018
CNN-SVM [37]	94.76	94.43	<b>92.01</b>	<b>90.32</b>
Proposed MSVM	<b>95.87</b>	<b>94.96</b>	91.86	90.27

TABLE 3: Recognition performance results for FA database.

Methods	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
SVM [25]	86	83.83	85.5	83.54
CNN [31]	90.23	88.13	89.30	88.87
CNN-SVM [37]	93.22	93.95	92.16	90.85
Proposed MSVM	<b>94.99</b>	<b>94.31</b>	<b>94.13</b>	<b>92.25</b>

TABLE 4: Recognition performance results for the DLC database.

Methods	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
SVM 1 [40]	75.5	<b>74.2</b>	70.83	72.72
SVM 2 [48]	73.4	73.6	71.5	73.66
Proposed MSVM	<b>79.21</b>	73.9	<b>73.1</b>	<b>74.95</b>

TABLE 5: Recognition results for the FC database.

Class	SL (%)	SR1 (%)
Arrow	85.3	77.54
Connection	82.45	96.2
Data	86.22	92.3
Decision	92.4	75.3
Process	92.1	88.5
Terminator	71.06	90.35
Text	98.8	91.15
Total	<b>94.15</b>	<b>86.2</b>

TABLE 6: Recognition results for the FA database.

Class	SL (%)	SR1 (%)
Arrow	92.4	89.2
Initial arrow	89.7	93.65
Final state	95.0	92.8
State	85.1	92.5
Label	90.4	91.3
Total	<b>91.25</b>	<b>91.89</b>

TABLE 7: Recognition results for the DLC database.

Class	SL (%)	SR1 (%)
Line	92.0	89.24
OR	85.96	84.1
AND	78.45	81.5
NOT	79.6	79.2
NOR	82.1	77.66
NAND	81.23	72.24
X-OR	69.78	66.66
Bubble	75.77	74.58
Text	92.08	90.9
Total	<b>82.22</b>	<b>79.8</b>

TABLE 8: Optimization time.

Class	Optimization time (s)			
	Minimal	Maximal	Average	Median
FC	0.11	0.35	0.14	0.11
FA	0.12	0.19	0.14	0.12
DLC	0.15	0.26	0.17	0.15

TABLE 9: Total running time.

Class	Running time (s)			
	Minimal	Maximal	Average	Median
FC	0.19	3.66	0.83	0.72
FA	0.28	2.54	0.87	0.84
DLC	0.38	1.43	0.77	0.67

TABLE 10: Misrecognition rates for different process of SKETRACK.

Database	Number of misrecognized symbols	Text separation (%)	Segmentation (%)	Classification (%)	Structural analysis (%)
FC	359	8.92	43.5	30.33	8.9
FA	266	3.95	62.67	51.5	21.1
DLC	428	5.5	71.2	59.2	37.7

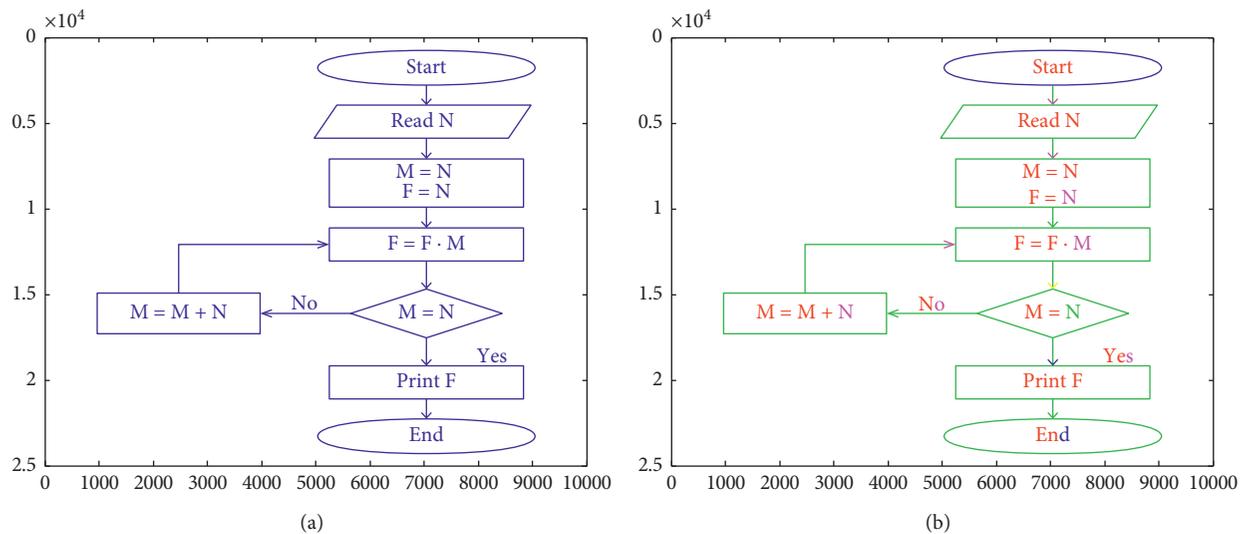


FIGURE 10: Examples of recognized results of FC sketch: (a) input sketch and (b) recognized sketch (green—correct symbol recognition, red—text, and blue—incorrect symbols).

precision in almost all databases. It must also be noted that the sketches recognized in Figures 10 and 11 have misrecognized symbols. The misrecognition is due to the incorrect strokes in which the single strokes are identified correctly while the merging of these symbols in the diagram representation of structural analysis has not been accurate. The strokes have a wider significance in influencing the overall performance. The misrecognition in DLC sketches is due to the similarity of class symbols, for example, similarity OR with NOR and X-OR. Likewise, the second curved structure of X-OR is quite complex to detect when the sketch

is viewed in two-dimensional or three-dimensional structures. Table 10 shows the misrecognition results for all three online databases in a combined inference.

From Table 10, it can be seen that the failure to recognize the symbols is mainly due to the ineffectiveness of the segmentation process to adapt to the low-quality sketches. The next reason is the performance of classifier. Hence, the need for improvement can be attributed to the segmentation process to avoid the rate of misrecognition. However, it must be noted that these misrecognitions in SKETRACK are only a fraction of the total system and only contribute to less than

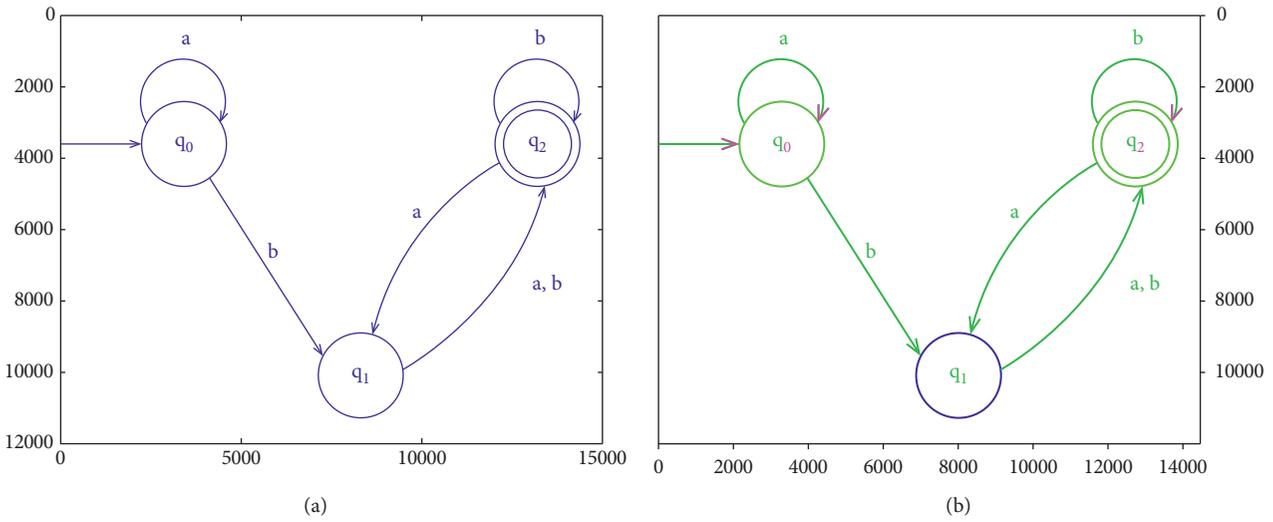


FIGURE 11: Examples of recognized results of FA sketch: (a) input sketch and (b) recognized sketch (green—correct symbol recognition, red—text, and blue—incorrect symbols).

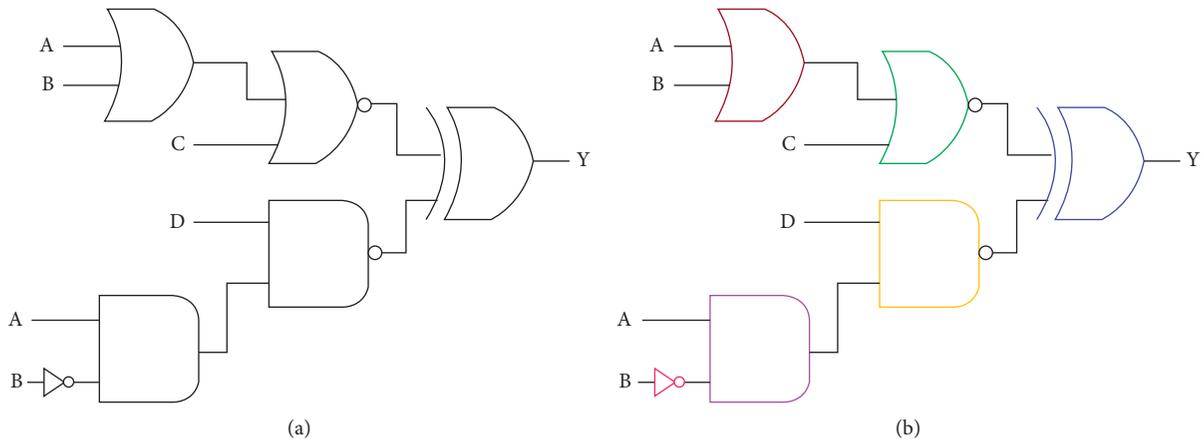


FIGURE 12: Examples of recognized results of DLC sketch: (a) input sketch and (b) recognized sketch (brown—OR, violet—AND, green—NOR, yellow—NAND, blue—X-OR, pink—bubble, and black—line).

10% of the total input sketches. Therefore, the proposed SKETRACK can be concluded to be efficient for FC and FA sketch recognition while there needs some adjustments for recognizing DLC sketches.

## 6. Conclusion

Efficient online freehand sketch recognition of FC, FA, and DLC sketches has been developed and evaluated in this paper. The proposed SKETRACK scheme consists of text segmentation, feature extraction, spectral clustering for symbol segmentation, symbol recognition using the MSVM classifier, and structural analysis. The experimental results are compared with the state-of-the-art methods to validate the performance efficiency of the sketch recognition methods. The proposed system performed better than the state-of-the-art methods. The accuracy of recognizing online FC, FA, and DLC diagrams is high using the proposed SKETRACK scheme. Although recognition performance for

few sketches is not perfect, it is mainly due to the noise and low quality of the database instances.

Although efficient, the small fraction of misrecognition has been attributed mainly to the symbol segmentation process. This will be exploited in future with more advanced techniques to further minimize the recognition errors. Also the DLC databases consist of only simple symbols while other complex symbols like converters and processors are difficult and are neglected; the inclusion of these symbols in recognition will be examined. Moreover, the proposed scheme will be utilized in other domains like electronic circuits and optical diagrams. Finally, the iterative recognition of sketches using immediate feedback system is also planned to be tested in the future.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

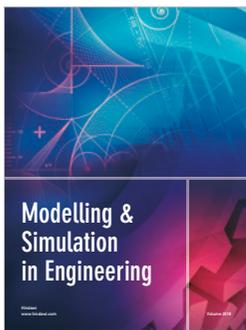
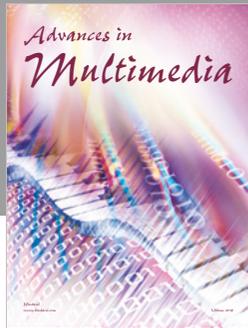
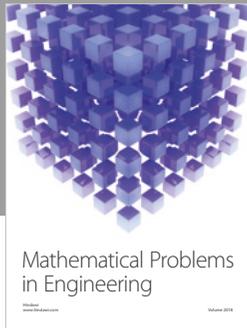
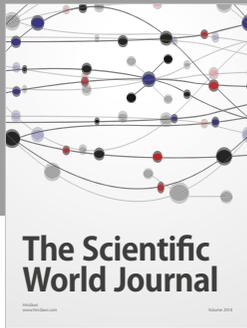
## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] M. Eitz, R. Richter, T. Boubekeur, K. Hildebrand, and M. Alexa, "How do humans sketch objects?," *ACM Transactions on Graphics*, vol. 31, no. 4, pp. 1–10, 2012.
- [2] Q. Yu, F. Liu, Y. Z. Song, T. Xiang, T. M. Hospedales, and C. C. Loy, "Sketch me that shoe," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 799–807, Las Vegas, NV, USA, June 2016.
- [3] P. Sangkloy, N. Burnell, C. Ham, and J. Hays, "The sketchy database: learning to retrieve badly drawn bunnies," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–12, 2016.
- [4] Y. Cao, C. Wang, L. Zhang, and L. Zhang, "Edgel index for large-scale sketch-based image search," in *Proceedings of the CVPR 2011*, pp. 761–768, IEEE, Colorado Springs, CO, USA, USA, June 2011.
- [5] X. Cao, H. Zhang, S. Liu, X. Guo, and L. Lin, "SYM-FISH: a symmetry-aware flip invariant sketch histogram shape descriptor," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 313–320, Sydney, Australia, December 2013.
- [6] Y. J. Lee, C. L. Zitnick, and M. F. Cohen, "Shadowdraw: real-time user guidance for freehand drawing," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4, p. 1, 2011.
- [7] Z. Sun, C. Wang, L. Zhang, and L. Zhang, "Free hand-drawn sketch segmentation," in *Proceedings of the European Conference on Computer Vision*, pp. 626–639, Springer, Florence, Italy, October 2012.
- [8] J. Pu and D. Gur, "Automated freehand sketch segmentation using radial basis functions," *Computer-Aided Design*, vol. 41, no. 12, pp. 857–864, 2009.
- [9] Z. Sun, C. Wang, L. Zhang, and L. Zhang, "Query-adaptive shape topic mining for hand-drawn sketch recognition," in *Proceedings of the 20th ACM international Conference on Multimedia—MM '12*, pp. 519–528, ACM, Nara, Japan, October 2012.
- [10] L. Li, H. Fu, and C. L. Tai, "Fast sketch segmentation and labeling with deep learning," *IEEE Computer Graphics and Applications*, vol. 39, no. 12, pp. 38–51, 2018.
- [11] J. Guo, C. Wang, and H. Chao, "Building effective representations for sketch recognition," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, Austin, TX, USA, January 2015.
- [12] P. C. Stevens, R. Blagojevic, and B. Plimmer, "Supervised machine learning for grouping sketch diagram strokes," in *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, pp. 43–50, ACM, Anaheim, CA, USA, July 2013.
- [13] C. Tian, M. Masry, and H. Lipson, "Physical sketching: reconstruction and analysis of 3D objects from freehand sketches," *Computer-Aided Design*, vol. 41, no. 3, pp. 147–158, 2009.
- [14] S.-U. Cheon, B. C. Kim, D. Mun, and S. Han, "A procedural method to exchange editable 3D data from a free-hand 2D sketch modeling system into 3D mechanical CAD systems," *Computer-Aided Design*, vol. 44, no. 2, pp. 123–131, 2012.
- [15] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa, "Sketch-based image retrieval: benchmark and bag-of-features descriptors," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 11, pp. 1624–1636, 2011.
- [16] L. Rossetto, I. Giangreco, and H. Schuldt, "Cineast: a multi-feature sketch-based video retrieval engine," in *Proceedings of the 2014 IEEE International Symposium on Multimedia*, pp. 18–23, IEEE, Taichung, Taiwan, December 2014.
- [17] T. A. Hammond, D. Logsdon, B. Paulson et al., "A sketch recognition system for recognizing free-hand course of action diagrams," in *Proceedings of the Twenty-Second IAAI Conference*, Atlanta, GA, USA, July 2010.
- [18] W. Szwoch and M. Mucha, "Recognition of hand drawn flowcharts," in *Advances in Intelligent Systems and Computing*, vol. 184, pp. 65–72, Springer, Berlin, Germany, 2013.
- [19] Y. Li, Y. Z. Song, and S. Gong, "Sketch recognition by ensemble matching of structured features," in *Proceedings of the British Machine Vision Conference*, vol. 1, p. 2, Bristol, UK, September 2013.
- [20] Y. Li, T. M. Hospedales, Y.-Z. Song, and S. Gong, "Free-hand sketch recognition by multi-kernel feature learning," *Computer Vision and Image Understanding*, vol. 137, pp. 1–11, 2015.
- [21] Y. Li, Y.-Z. Song, T. M. Hospedales, and S. Gong, "Free-hand sketch synthesis with deformable stroke models," *International Journal of Computer Vision*, vol. 122, no. 1, pp. 169–190, 2017.
- [22] Z. Huang, H. Fu, and R. W. Lau, "Data-driven segmentation and labeling of freehand sketches," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 6, pp. 1–10, 2014.
- [23] R. G. Schneider and T. Tuytelaars, "Sketch classification and classification-driven analysis using Fisher vectors," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 6, pp. 1–9, 2014.
- [24] M. Bresler, T. Van Phan, D. Prusa, M. Nakagawa, and V. Hlaváč, "Recognition system for on-line sketched diagrams," in *Proceedings of the 2014 14th International Conference on Frontiers in Handwriting Recognition*, pp. 563–568, IEEE, Heraklion, Greece, December 2014.
- [25] M. Bresler, D. Průša, and V. Hlaváč, "Online recognition of sketched arrow-connected diagrams," *International Journal on Document Analysis and Recognition (IJ DAR)*, vol. 19, no. 3, pp. 253–267, 2016.
- [26] Y. Zheng, H. Yao, S. Zhao, and Y. Wang, "Discovering discriminative patches for free-hand sketch analysis," *Multimedia Systems*, vol. 23, no. 6, pp. 691–701, 2017.
- [27] M. Kleffmann, S. Röhl, M. Book, and V. Gruhn, "Evaluation of a traceability approach for informal freehand sketches," *Automated Software Engineering*, vol. 25, no. 1, pp. 1–43, 2018.
- [28] X. Wang, X. Chen, and Z. Zha, "Sketchpointnet: a compact network for robust sketch recognition," in *Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 2994–2998, IEEE, 2018.
- [29] O. Seddati, S. Dupont, and S. Mahmoudi, "Deepsketch: deep convolutional neural networks for sketch recognition and similarity search," in *Proceedings of the 2015 13th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pp. 1–6, IEEE, Athens, Greece, September 2015.
- [30] O. Seddati, S. Dupont, and S. Mahmoudi, "DeepSketch 2: deep convolutional neural networks for partial sketch recognition," in *Proceedings of the 2016 14th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pp. 1–6, IEEE, Bucharest, Romania, June 2016.
- [31] O. Seddati, S. Dupont, and S. Mahmoudi, "DeepSketch 3," *Multimedia Tools and Applications*, vol. 76, no. 21, pp. 22333–22359, 2017.
- [32] R. K. Sarvadevabhatla and J. Kundu, "Enabling my robot to play pictictionary: recurrent neural networks for sketch recognition," in *Proceedings of the 24th ACM International Conference on Multimedia*, pp. 247–251, ACM, Amsterdam, Netherlands, October 2016.

- [33] H. Jahani-Fariman, M. Kavakli, and A. Boyali, "MATRACK: block sparse Bayesian learning for a sketch recognition approach," *Multimedia Tools and Applications*, vol. 77, no. 2, pp. 1997–2012, 2018.
- [34] J. Y. He, X. Wu, Y. G. Jiang, B. Zhao, and Q. Peng, "Sketch recognition with deep visual-sequential fusion model," in *Proceedings of the 25th ACM international conference on Multimedia*, pp. 448–456, ACM, Mountain View, CA, USA, October 2017.
- [35] E. Boyacı and M. Sert, "Feature-level fusion of deep convolutional neural networks for sketch recognition on smartphones," in *Proceedings of the 2017 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 466–467, IEEE, Las Vegas, NV, USA, March 2017.
- [36] Q. Yu, Y. Yang, F. Liu, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Sketch-a-net: a deep neural network that beats humans," *International Journal of Computer Vision*, vol. 122, no. 3, pp. 411–425, 2017.
- [37] M. Sert and E. Boyacı, "Sketch recognition using transfer learning," *Multimedia Tools and Applications*, vol. 78, no. 12, pp. 17095–17112, 2019.
- [38] X. Zhang, Y. Huang, Q. Zou, Y. Pei, R. Zhang, and S. Wang, "A hybrid convolutional neural network for sketch recognition," *Pattern Recognition Letters*, 2019, In press.
- [39] E. Indermühle, M. Liwicki, and H. Bunke, "IAMonDo-database: an online handwritten document database with non-uniform contents," in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pp. 97–104, ACM, Boston, Massachusetts, USA, June 2010.
- [40] A. Graves, A. R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6645–6649, IEEE, Vancouver, Canada, October 2013.
- [41] P. Beerli, "Pairwise distance methods," *Lecture Notes on Computational Evolutionary Biology*, Florida State University, Tallahassee, FL, USA, 2005, [http://people.sc.fsu.edu/~pbeerli/BSC-5936/10-26-05/lecture\\_15.pdf](http://people.sc.fsu.edu/~pbeerli/BSC-5936/10-26-05/lecture_15.pdf).
- [42] L. Wang, Y. Zhang, and J. Feng, "On the Euclidean distance of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1334–1339, 2005.
- [43] Y. Tian, K. Wang, R. Li, and L. Zhao, "A fast incremental map segmentation algorithm based on spectral clustering and quadtree," *Advances in Mechanical Engineering*, vol. 10, no. 2, Article ID 1687814018761296, 2018.
- [44] A. Jayachandran and R. Dhanasekaran, "Brain tumor severity analysis using modified multi-texton histogram and hybrid kernel SVM," *International Journal of Imaging Systems and Technology*, vol. 24, no. 1, pp. 72–82, 2014.
- [45] M. Yazdani and F. Jolai, "Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm," *Journal of Computational Design and Engineering*, vol. 3, no. 1, pp. 24–36, 2016.
- [46] D. Mayuri and S. Madhuri, "Hand-drawn digital logic circuit component recognition using svm," *International Journal of Computer Applications*, vol. 143, no. 3, pp. 24–28, 2016.
- [47] D. Johnston and C. Alvarado, "Sketch recognition of digital logical circuits," Doctoral dissertation, MS thesis, California University, Oakland, CA, USA, 2013.
- [48] I. A. Kakadiaris, G. Passalis, G. Toderici et al., "Three-dimensional face recognition in the presence of facial expressions: an annotated deformable model approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 640–649, 2007.




**Hindawi**

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

