Dynamic Random Forest for the Recognition of Arabic Handwritten Mathematical Symbols with A Novel Set of Features

Ibtissem Ali and Mohamed Mahjoub
Laboratory of Advanced Technology and Intelligent Systems, University of Sousse, Tunisia

Abstract: *Mathematics has a number of characteristics which distinguish it from conventional text and make it a challenging area for recognition. This include principally its two dimensional structure and the diversity of used symbols, especially in Arabic context. Recognition of mathematical formulas requires solving three sub problems: segmentation, the symbol recognition and finally the symbol arrangement analysis. In this paper we will focus on the Arabic mathematical symbol recognition step. This is a challenging task due to the large symbol set with many similar looking symbols used in Arabic mathematics and also the great variability found in human writing. The strength of the selected features and the effectiveness of the classifier are the two key factors determining the performance of a handwritten symbols recognition System .In this paper we proposed a novel Shape Context (SH) descriptor and explored its combination with a modified Chain Code Histogram (CCH) and a Histogram of Oriented Gradient (HOG) at the level of descriptors extraction. For the classification we used a Dynamic Random Forest (DRF) model which has the advantage of efficiently modelling the interaction among trees to determine the right prediction. The results carried out Handwritten Arabic Mathematical Dataset (HAMF) show that the DRF proves a significant improvement in terms of accuracy compared to the standard static RF and Support Vector Machines (SVM).*

Keywords: *Arabic handwritten mathematical symbols recognition, SH, HOG, CCH, dynamic RF.*

Received February 20, 2018, accepted April 17, 2018

1. Introduction

Mathematics has a number of characteristics which distinguish it from conventional text and make it a difficult field of recognition. This include principally its two dimensional structure and the diversity of used symbols. While the recognition of handwritten Latin has been extensively investigated using various techniques, little work has been done on handwritten Arabic mathematical recognition, and none of the existing techniques are accurate enough for practical application.

Like Arabic scripts and texts, mathematical expressions are written from right to left, for example, -1 might be written as 1- and using Arabic symbols from its alphabet. These symbols are used to note the names of variables and unknown functions. As for the names of usual functions, abbreviations of the names of these functions are used (e.g., جتا، جا، ظا، ظتا، قا، قتا، نها، لو). Arabic notation uses either the same symbols as those used in current use (e.g., +, ×, - , /, ≠, =, (,), {,}, [,]) or the same symbols through an inversion sense (e.g.,< and >, → and ←), or Latin symbols reflected. These symbols are images mirrors Latin symbols, such as the sum, the square root and the integral, Figure 1 gives some examples of Arabic mathematical expressions with reflected Latin symbols. Arabic notation used in different regions, two number systems either Arabic numerals (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) or Arab-Hindu numerals (٠, ١, ٢, ٣,٤, ٥, ٦, ٧, ٨, ٩).

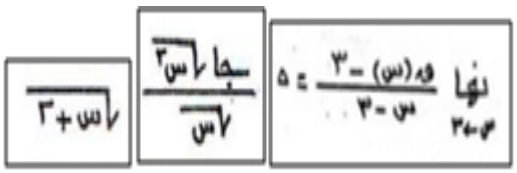


Figure 1. Examples of Arabic mathematical expressions with reflected Latin symbols.

The recognition of the mathematical expressions either Arabic or Latin amounts to solving three sub-problem:

1. Segmentation of the expression into isolated symbols.
2. Recognition of these symbols.
3. Structural analysis that determines spatial relationships between symbols and interpret the expression.

In this paper, we focused in the step of the recognition of Arabic handwritten mathematical symbols. This is a challenging machine learning problem due to:

- The large number of symbols to be classified like Arabic and Latin numerals, Arabic alphabet, arithmetic symbols, Arabic functions names,

equality operators, etc., Table 1 summarizes all the tested symbols in this paper.

- The large variety of Arabic symbols shapes.
- Similarity between some different symbols like Arabic digit one and Arabic letter Alif (١ ,ا) or like digit nine and Arabic letter Waw (9, و).
- Most Arabic characters have similar shapes [17] but differ in the position and number of dots (for example the letters: ج ،ح ،خand ب، ت، ث).
- The unlimited variation and imprecision in human handwriting what causes certain ambiguity. For example the same symbol can be written in different ways and some different symbols can have strong similarity, all depends on the writing style. Figure 2 gives some examples of ambiguity.
- Stretched large operators in Arabic notation appear in different sizes depend on their content like square root and bracket or they stretched to the same width as their lower and upper limits (see Figure 3).

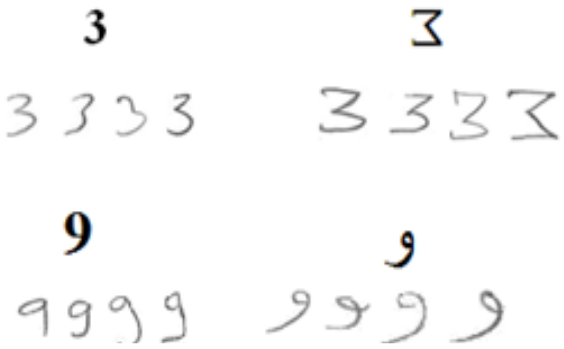


Figure 2. Examples of ambiguity samples.

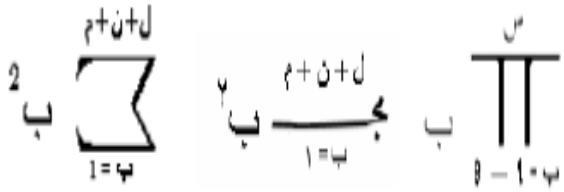


Figure 3. Stretched large operators.

The strength of the selected features and the effectiveness of the classifier are the two key factors determining the performance of a handwritten symbol recognition System. In exploring literature we find different models have been proposed like the hidden Markov models, Support Vector Machines (SVM), and Neural Networks (NNs) such as the Recurrent NNs and CNN but limited research work, which used the RFs for handwriting recognition. However, these classifiers show good results in the different domains of pattern recognition and have a lot of advantages compared to other classifiers proposed in the literature to wit:

1. The very high classification and recognition accuracy.
2. The ability to determine the variable importance.
3. The flexibility to perform several types of statistical data analysis, including regression, classification, survival analysis, and unsupervised learning.

The organization of this paper is as follows:

1. Related work in handwritten mathematical symbols, Arabic digits and characters recognition is described in section 2.
2. An Overview of our proposed approach for Arabic handwritten mathematical expressions recognition is presented in section 3.
3. The next section describes our method for the recognition of isolated handwritten Arabic mathematical symbols.
4. Section 5, present the experiments done in database Handwritten Arabic Mathematical Database (HAMF) [15] and the performance of our recognizer system. Finally, conclusions and future work are presented in section 5.

Table 1. Different symbols used in the recognition system.

Subset description	Symbols
Arabic characters	أ ت م ح ط ق د و عرسكل نصج ب
Digits Latin	0 1 2 3 4 8 3 7 8 9
Arithmetic operators	/ + - ×
Comparison operators	= ≠ ≥ >< ≤
Functions	جتا جا ظتا ظالو نها
Elastic operators	⎯ √ ∑ ∫ ∏
Others	← ∞ ( )

2. Related Work

As mathematical expressions are Two-Dimensional (2D) in nature, their online or offline recognition involves two stages: Symbol recognition and Structural analysis. Labels are assigned to the symbols by the symbol recognition process and the relationships like subscript, superscript etc., among the symbols of mathematical expressions are found in structural analysis stage. This entire process is much more complex for offline mathematical expressions than for online mathematical expressions due to lack of temporal information. A survey of existing works in this field is found in [8, 27].

Handwritten Latin mathematical recognition systems have been extensively studied and developed for many years, unlike Arabic mathematics where recognition systems of Arabic mathematical formulas are very rare. In fact, a study on the state of the art in the recognition of Arabic mathematics is very difficult.

Usually, a symbols recognition system is based on three main steps: pre-processing, features extraction, and recognition (classification).Pre-processing step is typically used to reduce noise and increase features discrimination capacity.

The most used operations are filtering and smoothing, normalization, binarization, slant correction, contour tracing and thinning. The use of the appropriate set of features is of great importance and

affects the recognition results. The performance of a classifier can rely as much on the quality of the features as on the classifier itself. Khazri *et al*. [18] proposed a system for recognizing Arabic printed mathematical formula, this system includes two stages:

1. Symbol recognition.
2. Symbol-arrangement analysis.

To accomplish symbol recognition step the authors extracted 30 statistical features (Hu moments, run-length, Zernike moments, bilevel co-occurrence, white pixel portion) and for the classification they tested K nearest Neighbors (K-NN), K*, Naive Bayes, Multilayer Perception (MLP) and Decision Tree (FT) to identify 50 symbol classes. The best recognition rate is achieved by K* with 96.18% accuracy rate. El-Sheikh [13] proposed a system for the online recognition of one-dimensional Arabic mathematical formulas.

Some statistical features (the width, aspect ratio, the relative distance between the first and the last points, the direction of the first part of character, the distance between the first point and the points of largest x or y coordinates, the number of minima and maxima in the horizontal and vertical directions, etc.,) are computed to represent symbols. To identify them, author used its own classification algorithm based on observations due to the statistical nature of the handwriting.

For the recognition of Latin symbols Davila *et al.* [11] employed Adaboost, SVM and Random Forest classifier. Offline features such as global features (angular change, line length, aspect ratio, and so on), crossing features, 2D Fuzzy histogram of points, and Fuzzy histograms of orientations were used.

Synthesized patterns were generated to train the recognizer. Álvaro *et al*. [1] proposed a set of hybrid features that combine both on-line and off-line information and using Hidden Markov Models (HMM) and Bidirectional Long Short Term Memory (BLSTM) for online handwritten mathematical symbols.

The symbol recognition rate achieved using raw images as local off-line features along the pen-tip trajectory by BLSTM significantly outperformed HMM.

In the necessity of an efficient classification a great number of classifiers have been proposed in the past for mathematical symbol recognition. Hu and Zanibbi [16] used an on-line HMM classifier proposing a novel initialization method. They performed several experiments considering 93 symbol classes. Malon *et al.* [19] described a successful approach to multi-class classification by adding binary SVM which are trained with directional histograms of the contour.

In a comparative study, Alvaro and Sanchez [2] comparing four techniques for Offline recognition of printed mathematical symbols. In addition of the use of the K-NN and SVM classification technique which are already explored in the offline mathematical symbol recognition and proved a powerful capacity in the recognition task, authors proposed to use the Weighted Nearest Neighbours (WNN) and HMM.

The proposed classification techniques are tested in two different databases, SVM and WNN achieved the best results however the worst results were obtained with HMM. Recently, a combination of CNN and BLSTM methods was presented by Nguyen *et al.* [21].

Zamani *et al*. [26] developed a handwritten digit recognition system based on the RFs and the Convolution NNs (CNNs).They performed some experiments with different pre-processing steps, feature types, and baselines. The results proved that the RFs performed better than the CNNs.A review of the different applications of the RF classifier was presented by Belgiu and Dragut [3]. This review revealed that the RF classifier could successfully handle high data dimensionality and multi-collinearity, being both fast and insensitive to over-fitting. This brief state-of-the art shows the efficiency of using the RFs in object recognition compared with other classifiers like the CNN, the SVM and others.

3. Overview of the Global System of the Arabic Mathematical Expressions Recognizer

The objective of the recognition of Arabic mathematical expressions recognition is to translate the formula from its manuscript form into digital editable format like LATEX, MathML etc., in this section, an overview of our approach to Arabic mathematical expression recognition is presented with its various stages. The process of ME recognition comprises three stages namely symbol segmentation, symbol recognition and finally structural analysis and generation of encoding form like LATEX or MathML code.

First, for a given input mathematical expression image, after its binarization, the first step is to segment this image into groups. The goal is that each of these groups forms exactly one symbol. In our approach, the method chosen to solve the segmentation problem is to compute the connected components of the input image. There are many mathematical symbols which are composed by more than one connected component, for example, = is composed of two horizontal line connected components. So the next step is to group connected components that can form a symbol hypothesis based on geometric rules. This hypothesis is validated by the symbol models to generate the list of the accepted hypothesis of symbols and the rejected hypothesis. The accepted symbols will be classified and labeled in the recognition symbols step which is the scope of our paper. More details about the modeling and the recognition of symbols will be given in the next section.

At this stage all symbols have been recognized and well localized, thus the task now is to identify the relationships among the recognized symbols in order to build a hierarchical structure of the symbols that represents the mathematical expression. The identification of symbol relations is based on layout analysis of the mathematical expression. A straightforward solution to this issue is the introduction of constrains that examine the relative spatial relations of the symbols. To this end, we exploit symbol's topological properties such as the centroid and the bounding box in order to infer the spatial relations among the mathematical symbols. Once the spatial relationships have been identified, the next step is to analyze these data to generate the analysis tree We opted for a bottom-up parsing respecting a 2D grammar in order to represent a validate expression. Finally we generate an encoding form like LATEX by traversing the parse tree that represents the structure of the mathematical expression. The Figure 4 describes the different steps in the proposed approach for the recognition of Arabic mathematical expression.

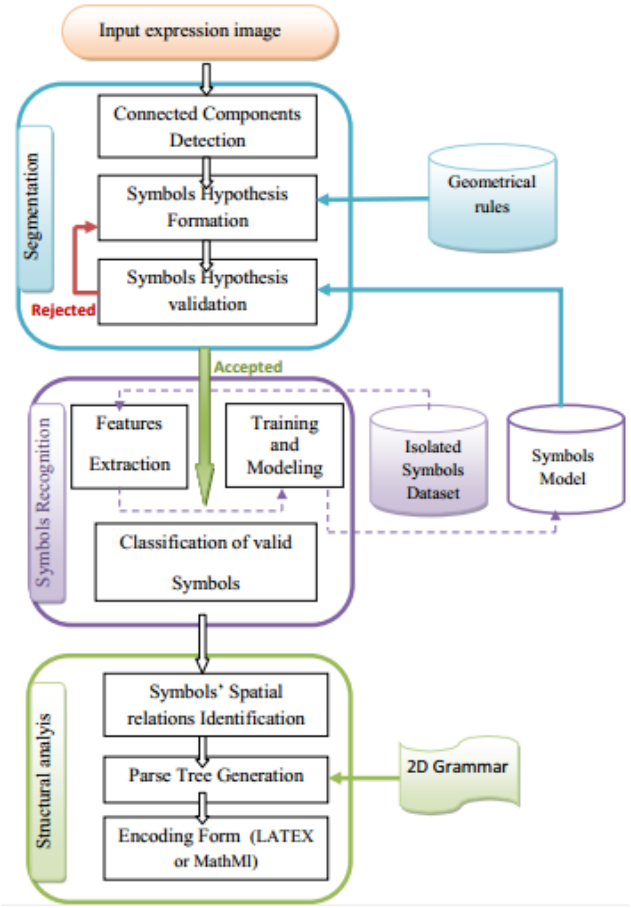


Figure 4. Global approach for the recognition of Arabic mathematical expression.

4. Dynamic RF for Arabic Mathematical Symbols Recognition with Hybrid Features

As depicted in Figure 5, the proposed method followed the typical pattern recognition system architecture that achieved in three main steps: pre-processing, features. extraction, and recognition phase.

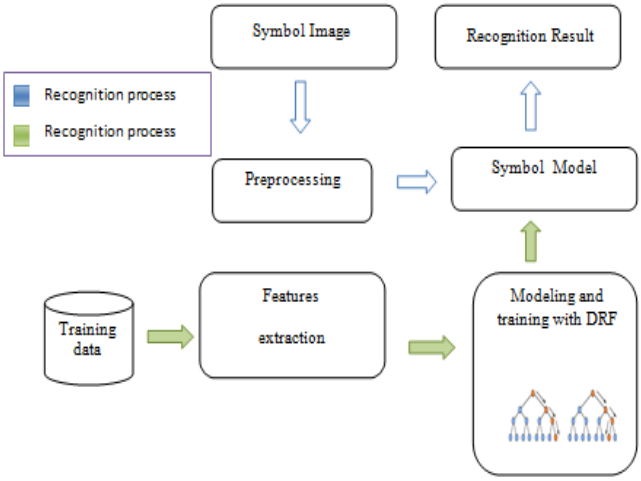


Figure 5. System architecture for the recognition of isolated mathematical symbol.

4.1. Pre-Processing

Pre-processing is one of the most important steps in the recognition of handwriting symbols that affect the next stage. First we use a median filter to remove noise introduced on the symbol image during the acquisition step, the noise in offline systems could happen because of many reasons such as the scanner quality and the papers noise. After removing noise we convert the gray scale images into binary images. In this step, we use proper thresholding and gray level normalization techniques to standardize the gray levels of background and foreground regions of the mathematical symbol images. By thresholding a gray scale image, we can obtain a binary image (with level 0 for foreground and level 1 for background). For selecting the threshold, we use the classical algorithm of Otsu [22].After that a trimming is applied to remove extra white spaces present in the image and the final step of pre-processing is normalization, this process convert the random size of image into standard size, which is used to avoid inter class variation among symbols. The trimmed symbols are normalized in this work to the size of 32×32 pixels respecting aspect ratio.

4.2. Features Extraction

The Features Extraction is a critical stage in any recognition system because it is of great importance and affects directly the recognition results. Due to the diversity in writing style, handwritten symbols are placed in a high-dimension data category and finding an optimal, effective, and robust feature set to characterize them in the recognition phase is a complex task. For this purpose we proposed a novel features set

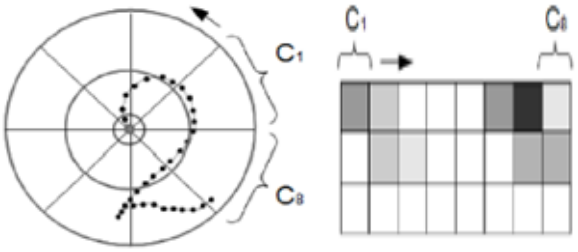
based on the Shape Context (SH) descriptor and to enhance the recognition accuracy we study the combination of this descriptor with other global and local features namely Chain Code Histogram (CCH) on its two forms global and local and the Histogram of Oriented Gradient (HOG).

4.2.1. Shape Context

SH descriptor has been applied in several Computer Vision classification problems with outstanding accuracy. Belongie *et al*. [4] first proposed the SH descriptor to characterize the points on a shape boundary for shape matching tasks, the authors evaluated SH on the recognition of handwritten digits using the MNIST dataset and 3D objects. In this approach, the object shape is represented as a set of points P={ p_i} from its contour. Each point p_i is described by a log polar histogram h_i that relates p_i to its surrounding points from the contour of the shape. The origin of a histogram is centered at the point it is describing as shown in Figure 6. It divides the space around it into partitions called bins. Each bin is identified by two parameters: distance from the centre point, and the orientation relative to the centre point. The histogram h_i of a point p_i is considered to be its SH. It counts the number of surrounding points in each bin using Equation 1.

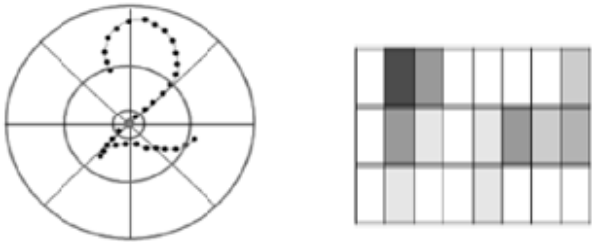
$$h_i(k) = \#\{q \neq p_i : (q - p_i) \in bin(k)\} \quad (1)$$

Roughly speaking, SH features of a symbol can be seen as a set of histograms calculated at each sampled point of the symbol, but with the particularity that histograms are defined over a log polar space, which makes SH of a point more sensitive to positions of nearby sample points (local features) than those of points farther away. Therefore, we will select a fixed number of points from different contour regions that carry more discriminative information to tackle this drawbacks In our proposed descriptor based on SH we will select 5 point from the contour of our symbol as follow: first, obtain from the symbol n-samples uniformly spaced taken from its edge elements. The first selected point is the closest point of contour to the center of gravity of the symbol's bounding box this distance is calculated using Euclidean distance. Then we split the symbol into 2×2 regions, for each region we select also the closest point to the center of gravity of this region. Once we have located our reference point we calculate for each of them the log polar histogram. The symbol area is divided into 8 angular regions and 3 radial regions, for a total of 24 bins, where a bin contains the quantity of points spatially placed in that binaccording the reference point. Finally we concatenate these different histograms to obtain the features vector based on SH. The features vector is equal to the number of bin multiplied by the number of reference points; in our case we have 120 features.



a) Show the first sampled point and log polar histogram bins used to calculate shape context.

b) The shape context histogram relative to the first point.



c) The second sampled point and log polar histogram bins.

d) The shape context histogram relative to the second point.

Figure 6. Shape Context histogram of two points of a symbol "2".

4.2.2. Histogram of Oriented Gradient

The literature investigation exposes that HOG descriptor is extensively used in numerous recognition applications because of its discriminative capability compared to other existing feature descriptors. The HOG descriptor is developed by Dalal and Triggs [10] for pedestrian detection task with the use of SVM classifier. The HOG has been productively applied in various research fields such as word spotting method [24], face recognition [12] and character recognition [20]. The HOG feature extractor represents objects by counting occurrences of gradient intensities and orientations in localized portions of an image. The HOG descriptor computes feature vectors using the following steps:

1. Split the image into small blocks of n×n cells.
2. Compute horizontal gradient Hx and vertical gradient Hy of the cells by applying the kernel [-1,0,1] as gradient detector.
3. Compute the magnitude M and the orientation θ of the gradient as:

$$M_{(x,y)} = \sqrt{H_x^2 + H_y^2} \quad (2)$$

$$\theta_{(x,y)} = arctan\frac{H_y}{H_x} \quad (3)$$

4. Form the histogram by weighing the gradient orientations of each cell into a specific orientation bin.
5. Apply L2 normalization to the bins to reduce the illumination variability and obtain the final feature vectors. In our experiments, we use 4×4 rectangular blocks and 9 orientation bins, thus yielding a 144-

dimensional feature vector. Figure 6 gives the process of HOG descriptors acquisition for the image of Arabic square root symbol.

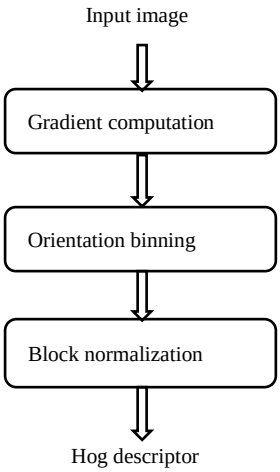


Figure 7. The process of HOG descriptor acquisition of Arabic square root symbol.

4.2.3. Chain Code Histogram

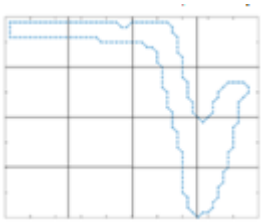
The CCH is a statistical measure for the directionality of the contour of a symbol used to determine how a pixel is connected to the next in the sequence of points. We measure the slope between two successive points, which would give the angle made by the line joining them and the x-axis. Then the set of possible slopes is (0º, 45º, 90º, 135º), which are identical to the directions (180º,225º, 270º and 315º). Thus, the directions between two successive pixels could be four or eight. Therefore, the Freeman chain code is a sequence of values, each value describes the connectivity and the direction between two consecutive pixels in the contour of the symbol.

The Freeman chain code is sensitive to the starting point computing, histogram is one solution to compensate this drawback. Then each possible value in the histogram will be normalized, which will represent the intensity of a direction.

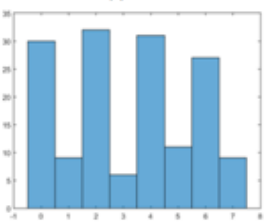
It was shown that an appropriate fusion of global and local features will compensate their short comings, and therefore improve the overall effectiveness and efficiency [23].Therefore, for the suggested system, in addition to the8-directional CCH of the whole symbol image as shown in Figure 8-a, we propose to divide the image into 16 (4×4) block, for each block the 8-directional CCH is computed Figure 8-b. As a result, we totally obtained 136 dimensional CCH vector. As a result, we totally obtained 128 dimensional CCH vector.

4.3. Modelling and Training

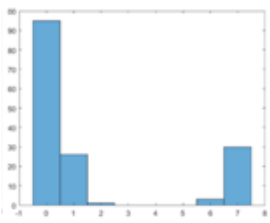
The stage "modelling and training" consist in classifying an unknown symbol and recognizing which class it belongs to. After, we extract the features from the image, as indicated in the precedent section; we introduce the principle of RF model and Dynamic Random Forest (DRF).



a) The contour of Arabicsquare root symbol.



b) Global CCH of the whole symbol image.



c) Local CCH of the top-left bloc of symbol.

Figure 8. The global and local chain code histogram descriptor: (a), (b) and (c).

4.3.1. Random Forests

Random forests introduced by Breiman [7] rate among the most recent and popular boosting methods and have proven their classification performance for difficult problems in many applications [6, 9, 14, 25]. Random forest is an ensemble training algorithm that constructs multiple decision trees, where each tree contributes with a single vote for the assignment of the most frequent class to the input data. It suppresses over fitting to the training samples by random selection of training samples for tree construction in the same way as is done in bagging, resulting in construction of a classifier that is robust against noise. Also, random selection of features to be used at splitting nodes enables fast training, even if the dimensionality of the feature vector is large.

In the training process, bagging is used to create sample sub sets by random sampling from the training sample. One sample set is used to construct one decision tree. The induction of these tree a based on the CART algorithm that modifies the feature selection procedure at each node by selecting K variables at random out of all N possible variables (independently at each node) then find the best split on the selected K variables. The recommended number of feature selections, K is the square root of the feature dimensionality. The splitting processing is repeated recursively until a certain depth is reached or until the information gain is zero. A leaf node is then created and the class probability P(c|l) is stored In the classification process, an unknown sample is input to all of the decision trees, and the class probabilities of the leaf nodes arrived at is output. The class that has the largest average of the class probabilities obtained from all of the decision trees, $P_t(c|x)$, according to Equation (4) is the classification decision.

$$(c|x) = \frac{1}{T}\sum_{t=1}^{T} P_t(c|x)\, P \quad (4)$$

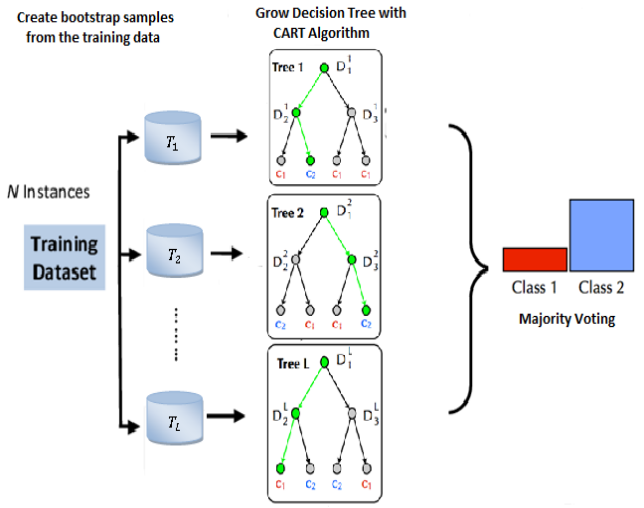


Figure 9. Architecture of the random forests.

In order to improve the performance of the RF, many extensions have been proposed in the literature, denoted by the DRFs and defined in the next section.

4.3.2. Dynamic Random Forests

Traditional Random Forests forest-RI introduced by Breiman grows trees independently from one another. Bernard *et al*. [5] have shown that when using classical RF induction algorithms, some trees degrade the performance of forest. Hence the idea of Dynamic RF is to avoid the induction of trees that could make the forest performance decrease, by forcing the algorithm to grow only trees that would suit to the ensemble already grown. The DRFis based on adaptive weighting induction of trees and the 3 main idea of this new algorithm is to adapt the induction of each tree to the current forest to construct a different set whose members cooperate well with others. The DRF uses a sequential and dynamic procedure to construct a set of random trees. The main idea for Bernard is to force the induction of the next tree to focus on the worst predicted training samples by the forest using a training data weighting. In fact, the samples that are so hard to class will be favoured in the induction of the next tree, thus affecting those having an important weight. To calculate this weight, Simon Bernard used a measure of reliability of the forest prediction which would lean on the number of votes attributed to the right class of data.

Training with the DRF is different from training with the Forest_RI on the process of partition-rule selection. Each rule divides the group of training data into two groups based on certain evaluation criteria. These criteria in using the Forest_RI are based on the effectives of each class of problems in each group.

In using the DRF, those effectives are replaced by the sum of weights of data sum contained in the sub groups. The process of DRF described in the Algorithm1.

Algorithm 1: Dynamic random forests

Input:
A: training set
M: number of randomly selected variables
L: number of trees
$W_\alpha(c(x,y))$ *: weighting function*
Output: Random Forest.
Begin
1: for $x_i \in A$ *do*
2: $D_1(x_i) = 1/N$ */// weighting vector*
3: for (l<= L) do
4: tree $\longleftarrow$ *empty tree*
5: $Z \longleftarrow 0$
6: $A_l \longleftarrow$ *weighted bootstrap*
7: tree.root $\longleftarrow$ *RndTree (tree.root,* A_l *)*
8: Random Forest $\longleftarrow$ *Random Forest* $\cup$ *tree*
9: for $x_i \in A$ *do*
10: $D_{l+1}(x_i) \longleftarrow W_\alpha(c(x_i,y_i))$
11: $Z \longleftarrow Z + D_{l+1}(x_i)$
12: for $x_i \in A$ *do*
13: $D_{l+1}(x_i) \leftarrow \frac{D_{l+1}(x_i)}{Z}$ */// normalizing weights*
14: return Random Forest
End

5. Experimental Results

In this section we compare the different proposed set of features namely the novel SH, the HOG and the proposed global and local CCH. We evaluate different combination between these descriptors to determine who gives the best results. We explore the DFR to adjust his parameters and compare its performance to the Forest_RI and SVM to proven our approach. All the experimental tests were performed on the public HAMF dataset [15] using a 2.7 GHz Intel i5 processor.

5.1. Dataset

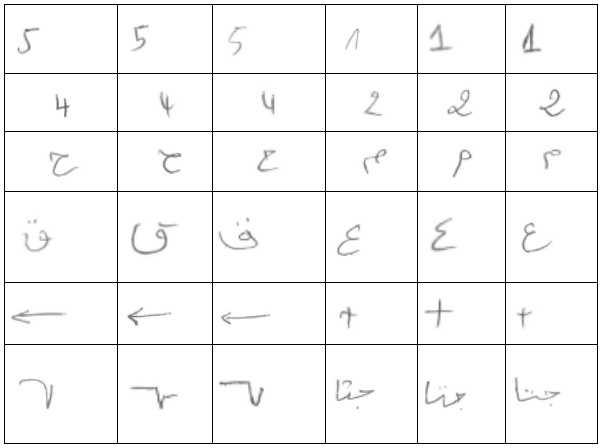
We evaluate our method on a large handwritten dataset of Arabic mathematical expression and isolated symbols named HAMF [15]. This dataset is freely available and it is the first dataset related to Arabic handwritten mathematical. The HAMF database consists of two subset, the first contains 4 238 images of handwritten Arabic mathematical formula written by 66 different writers and the second formed by 20 300 isolated symbols images.

Table 2 gives some example of handwritten isolated symbols. In this experiment we use the second set of isolated mathematical symbol. This dataset contains 49 different classes of isolated mathematical symbols divided into seven different set presented in Table 1.

To evaluate our approach the set of isolated mathematical symbol was divided into two subsets of data, one for training and the other for testing. The separation of the data was carried out by random sampling, with two thirds of the data for training

(13532 samples) and the remaining third for the test (6768 samples).

Table 2. Example of handwritten isolated symbols.



5.2. Results and Discussion

The DRF works according to two main parameters: the number L of trees in the forest, and the number K of features pre-selected for the splitting process in the tree. According to the literature, the number K is set to the square root of the feature dimensionality. First we study the behaviour of the DRF according to the number of trees, using each time one of three proposed descriptor:

1. The novel SH which consists of a vector of 120 features.
2. The set of global and local CCH which consist of 136 dimensional vectors.
3. The third set composed by 144 HOG descriptors.

Concerning the number *L* of trees, we have picked eleven increasing values, from 10 to 500 trees. The results are shown in Figure 10. We notice that using the novel SH done the best recognition accuracy compared to using HOG and CCH. We can see a global tendency of the recognition rate to rise for an increasing number of trees. It appears that this increase is not linear but logarithmic. One can conclude from this, that with respect to an increasing number of trees, the DRF converges. It seems on this figure that the rise of the recognition rate begins to considerably slow down from 200 trees in the forest.

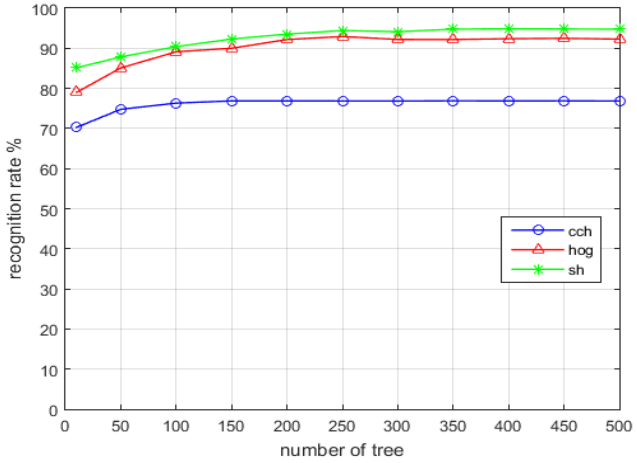


Figure 10. The recognition rate of Arabic mathematical symbols using DRF according different number of trees and different features set.

In order to improve the performance of our system we study different combinations between the proposed descriptors. The first combination is between the HOG and CCH (hog-cch) the second is between the SH and the CCH (sh-cch) and the third incorporate the SH and HOG (sh-hog). As depicted in Figure 11 the fusion of the HOG descriptor and SH (sh-hog)outperform the two others combination and gives high recognition that reaches 97.95% using 250 trees in the DRF. Although the symbol recognizer achieved a good accuracy, its failure to distinguish certain common symbols. In fact, certain distinct symbols are in close resemblance such Arabic letter Waw and digit 9 point (9, ٩), minus sign and horizontal fraction bar, Arabic digit one and Arabic letter Alif (ا,١), etc., Observing the event of confusion, we remark that confused symbols have roughly similar morphologies. We consider some of the misrecognitions to be too difficult for any classifier to resolve without considering symbol context.

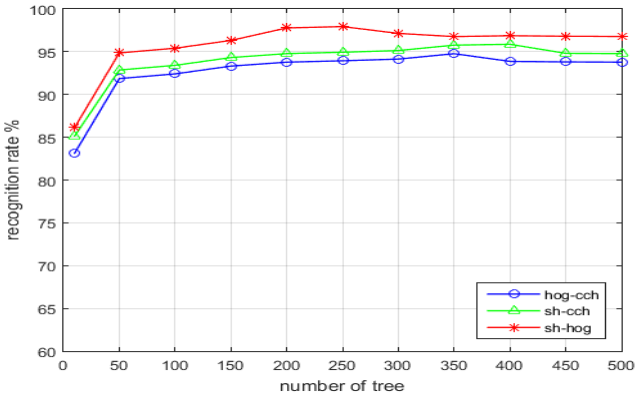


Figure 11. The recognition rate of Arabic mathematical symbols using DRF according different number of trees and hog-cch, sh-cch and sh-hog features.

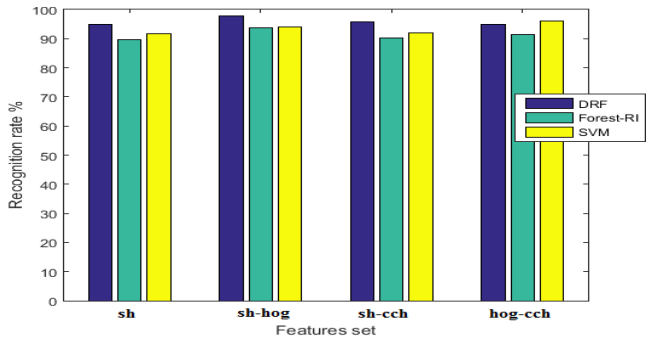


Figure 12. Comparing the performance of DRF, Forest-RI and SVM for the recognition of Arabic mathematical symbols using different set of features.

A comparison between the DRF, Forest-RI and SVM on order of the recognition of handwritten Arabic mathematical symbols is established using sh descriptor, sh_hog, sh-cch and hog-cch. The Figure 12 gives the result, as shown in this figure the use of SH descriptor combined with the HOG done the best results with the three different classifiers. The DRF gives the best recognition rate on almost of the cases

Except in the case of using the hog-cch descriptor. We also notice that the DRF and SVM outperform the static model of RF the forest-RI in all the cases.

Table 3 shows recognition rate provided by DRF, static RF (Forest-RI) and SVM based on the combination of SH descriptor and HOG on the HAMF dataset, which is composed of seven subsets as illustrated in Table 1. The two models of RF trained with 250 trees and 16 feature scores ponds to the square root of the features dimension, the classification rates is computed independently for each subset and for the whole road symbols of the HAMF dataset. The recognition of the Arabic characters set have the lowest rate using the three different classifiers because of the great number of classes to recognize compared to other set like arithmetic operator which compute only 4classes. Moreover the Arabic characters have a complicated shape with some similarity between them.

Table 3. Recognition rate corresponding to different subset.

	Recognition rate %		
	DRF	**Fore-RI**	**SVM**
Arabic characters	96.45	93.32	95.78
Digits Latin	98.82	95.85	96.46
Arithmetic operators	99.32	98.54	94.72
Comparison operators	98.27	96.37	96.98
Functions	97.53	95.83	96.37
Elastic operators	99.99	96.24	95.30
Others	99.81	96.20	97.21
All symbols	97.95	93.87	94.15

6. Conclusions

In this paper, we presented an efficient system for Arabic handwritten mathematical symbols recognition. For this purpose we proposed a novel hybrid set of features based on the fusion of our modified SH descriptors, CCH and HOG. We studied also the DRF for the recognition of isolated mathematical symbols using different fusion of the proposed descriptors.

Following experiments on the HAMF dataset, we draw the following conclusions:

1. Compared with Forest-RI and SVM, DRF can improve the offline recognition of Mathematical symbols.
2. Combining both SH descriptor and HOG improve classification performance by talking their advantage.
3. Analyzing the cases of system failure we conclude that the system needs to incorporate contextual information to remove ambiguity.

Finally, our future work will be focused on the integration of this system in mathematical expression recognition system, where the recognition of the whole system will help to solve the problem of similar shaped classes.

References

[1] Álvaro F., Sanchez J., and Benedi J., "Classification of On-Line Mathematical Symbols with Hybrid Features and Recurrent Neural Networks," *in Proceedings of the International Conference on Document Analysis and Recognitio*n, Washington, pp. 1012-1016, 2013.

[2] Álvaro F. and Sanchez J., "Comparing Several Techniques for Offline Recognition of Printed Mathematical Symbols," *in Proceedings of the 20th International Conference on Pattern Recognition*, Istanbul, pp. 1953-1956, 2010.

[3] Belgiu M. and Dragut L., "Random Forest in Remote Sensing: A Review of Applications and Future Directions," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 114, pp. 24-31, 2016.

[4] Belongie S., Malik J., and Puzicha J., "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509-522, 2002.

[5] Bernard S., Adam S., and Heutte L., " Dynamic Random Forests," *Pattern Recognition Letters*, vol. 33, no. 12, pp. 1580-1586, 2012.

[6] Bernard S., Heutte L., and Adam S., "Using Random Forests for Handwritten Digit Recognition," *in Proceedings of the 9th International Conference on Document Analysis and Recognition*, Parana, pp. 1043-1047, 2007.

[7] Breiman L., "Random Forests," *Machine Learning Journal*, vol. 45, no. 1, pp. 5-32, 2001.

[8] Chan K. and Yeung D., "Mathematical Expression Recognition: A Survey," *International Journal of Document Analysis and Recognition*, vol. 3, no. 1, pp. 3-15, 2000.

[9] Criminisi A. and Shotton J., *Decision Forests for Computer Vision and Medical Image Analysis*, Springer, 2013.

[10] Dalal N. and Triggs B., "Histograms of Oriented Gradients for Human Detection," *in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, pp. 886-893, 2005.

[11] Davila K., Ludi S., and Zanibbi R., "Using Off-line Features and Synthetic Data for On-Line Handwritten Math Symbol Recognition," *in Proceedings of 14th International Conference on Frontiers in Handwriting Recognition*, Heraklion, pp. 323-328, 2014.

[12] Deniz O., Bueno G., Salido J., and De la Torre F., "Face Recognition Using Histograms of Oriented Gradients," *Pattern Recognition Letters Journal*, vol. 32, no. 12, pp. 1598-1603, 2011.

[13] El-Sheikh T., "Recognition of Handwritten Arabic Mathematical Formulas," *in Proceeding of the UK IT 1990 Conference*, Southampton, pp. 344-351, 1990.

[14] Greenhalgh J. and Mirmehdi M., "Traffic Sign Recognition Using MSER and Random Forests," *in Proceedings of the 20th European Signal Processing Conference*, Bucharest, pp. 1935-1939, 2012.

[15] Hadj I. and Mahjoub M., "Database of Handwritten Arabic Mathematical Formula Images," *in Proceedings of the 13thInternational Conference Computer Graphics, Imaging and Visualization*, Beni Mellal, pp. 145-149, 2016.

[16] Hu L. and Zanibbi R., "HMM-Based Recognition of Online Handwritten Mathematical Symbols Using Segmental K-Means Initialization and a Modified Pen-Up/Down Feature," *in Proceedings of the International Conference on Document Analysis and Recognition*, Beijing, pp. 457-462, 2011.

[17] Jayech K., Mahjoub M., and Ben Amara N., "Arabic Handwritten Word Recognition Based on Dynamic Bayesian Network," *The International Arab Journal of Information Technology*, vol. 13, no. 6B, pp. 1024-1031, 2016.

[18] Khazri K., Kacem A., and Belaïd A., "A Syntax Directed System for the Recognition of Printed Arabic Mathematical Formulas," *in Proceedings of the International Conference on Document Analysis and Recognition*, Tunis, pp. 186-190, 2015.

[19] Malon C., Uchida S., and Suzuki M., "Mathematical Symbol Recognition with Support Vector Machines," *Journal of Pattern Recognition Letters*, vol. 29, no. 9, pp. 1326-1332, 2008.

[20] Minetto R., Thome N., Cord M., Leite N., and StolfiJ., "T-HOG: An Effective Gradient-Based Descriptor for Single Line Text Regions," *Pattern recognition Journal*, vol. 46, no.3, pp. 1078-1090, 2013.

[21] Nguyen H., Le A., and Nakagawa M., "Recognition of Online Handwritten Math Symbols Using Deep Neural Networks," *Journal of IEICE Transactions on Information and Systems*, vol. E99.D, no. 12, pp. 3110-3118, 2016.

[22] Otsu N., "A Threshold Selection Method From Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62-66, 1979.

[23] Ping D., "A Review On Image Feature Extraction And Representation Techniques," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 8, no. 4, pp. 385-396, 2013.

[24] Terasawa K. and Tanaka Y., "Slit Style HOG Feature for Document Image Word Spotting," *in Proceedings of the 10th International Conference on Document Analysis and Recognition*, Barcelona, pp.116-120, 2009.

[25] Tustison N., Shrinidhi K., Wintermark M., Durst C., Kandel B., Gee J., Grossman M., and Avants B., "Optimal Symmetric Multimodal Templates and Concatenated Random Forests for Supervised Brain Tumor Segmentation with ANTsR," *Neuroinformatics Journal*, vol. 13, no. 2, pp. 209-225, 2015.

[26] Zamani Y., Souri Y., Rashidi H., and Kasaei S., "Persian Handwritten Digit Recognition by Random Forest and Convolutional Neural Networks," *in Proceedings of the 9th Iranian Conference on Machine Vision and Image Processing*, Tehran, pp. 37-40, 2015.

[27] Zanibbi R. and Blostein D., "Recognition and Retrieval of Mathematical Expressions," *International Journal of Document Analysis and Recognition*, vol. 15, no. 4, pp. 331-357, 2012.

Ibtissem Ali Received the Diploma of computer science Engineering and Diploma of master respectively in 2010 and 2013 from the National Engineering School of Sousse - Tunisia. She is currently a PH D student and member of research laboratory LATIS (Laboratory of Advanced Technology and Intelligent Systems) team of analysis and processing of document. Her research interests include handwritten mathematical recognition, Arabic optical character recognition, document analysis, computer vision and pattern recognition.

Mohamed Mahjoub is an associate professor in Signal and Image processing at the National Engineering School of Sousse (ENISo) and member of the Laboratory of Advanced Technology and Intelligent Systems (LATIS). His research interests include dynamic Bayesian network, computer vision, pattern recognition, HMM and data retrieval. He is a member of IEEE and his main results have been published in international journals and conferences.