ACIT'2017
The International Arab Conference on Information Technology
Yassmine Hammamet, Tunisia
December 22-24, 2017

Random Forests for the Recognition of Handwritten Arabic Mathematical Symbols

Ibtissem Haj Ali and Mohamed Ali Mahjoub
LATIS – Laboratory of Advanced Technology and intelligent Systems
ENISO – 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 of three sub problems: segmentation, the symbol recognition and Finally the third step is the symbol arrangement analysis. In this paper we will focus on the symbol recognition step and we will propose a novel technique for Arabic handwritten Mathematical Symbols recognition. We constructed an invariant and efficient feature set by combination of global and local directional Chain Code Histogram (CCH) and Histogram of Oriented Gradient (HOG). For classification phase, Random Forests with Dynamic version for the induction of the forest was used. The system was evaluated on HAMF handwritten Arabic mathematical dataset. The experimental results represent 94,78 % classification rate in the test set and 99.98% in the train set.

Keywords— Arabic handwritten Mathematical Symbols recognition; HOG; CCH; Random Forests.

1. Introduction

Mathematical expression recognition is a challenging recognition field that attracts a lot of interest since the 1970s. 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. In literature the recognition of mathematical Latin formulas has been widely studied in past years and several techniques and approach, but research in the recognition of Arabic mathematical expression is very poor until now. Like Arabic scripts and texts, mathematical formulas are written from right side to left side for example, -1 might be written as 1- . Arabic notation uses either the same symbols currently used in mathematics (eg +, -,≠.) or the same symbols through an inversion sense (ex. < 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 use in different regions, two number systems either Arab or Arab-Hindu.

- Arabic digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Arabic -Hindu digits: ٠, ١, ٢, ٣,٤, ٥, ٦, ٧, ٨, ٩

The recognition of the mathematical expressions either Arabic or Latin amounts to solving three sub-problem: segmentation of the expression into isolated symbols, recognition of these symbols and structural analysis that determines spatial relationships between symbols and interpret the expression.In this article, we describe a recognition system for the 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 summarize all the tested symbols. Moreover symbols represents a wide variety in size, the same symbol can appear in different sizes in different context, for example square root, fraction and limit symbols. Furthermore the handwriting nature represent a high degree of variability and imprecision 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 samples.

To achieve our objective a Random Forests classifier is used based on different sets of features. Random forest introduced by L. Breiman [11] become a popular technique for classification in many field of pattern recognition and Computer Vision like medical image analysis [13], brain tumor segmentation [14], segmentation of video objects, Traffic sign detection and recognition [15], classification of aerial images, handwritten digit recognition [16], and many others.

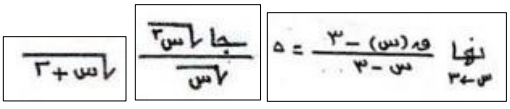

Figure 1: Examples of Arabic mathematical expressions with reflected Latin symbols.

The organization of this paper is as follows: Related work in handwritten mathematical symbols, Arabic digits and characters recognition is described in Section 2. Section 3 describes our method for the recognition of handwritten Arabic mathematical symbols. In section 4, some experiments have been done on the database HAMF [19]. Finally, conclusions and future work are presented in Section 5.

Table1 : Different symbols used in the recognition system

Subset description	Symbols
Arabic characters	ص ل ن ك س ر أ ت م ط ح ق و د ع ج ب
Digits Latin	0 1 2 3 4 8 3 7 8 9
Arithmetic operators	/ + - ×
Comparison operators	= ≠ ≥ > < ≤
Functions	لل امن انتج اج انظ اظ
Elastic operators	⟌ ∛ \ ∏
Others	← ∞ ( )

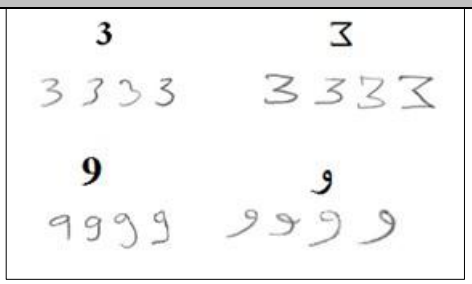

Figure 2: Examples of ambiguity samples

2. Related work

A several number of approaches have been proposed for handwriting Online Latin mathematical symbol recognition. F. Álvaro, et al [1] used a set of hybrid features that combine both on-line and off-line information, for each point 7 online features are calculated that include normalized coordinates, normalized first derivatives, normalized second derivatives and curvature, then combined it with off-line information extracted from a context-window centered in this point. lately authors use Recurrent Neural Networks (RNN) based on the long Short-Term Memory architecture for the classification and compare its performance with the HMM classifier. As a result RNN done the best accuracy rate 89.4%. An efficient elastic matching algorithm was developed by MacLean and Labahn [4], in this research authors carried out an experimentation taking into account only the single-stroke symbols of the MathBrush database[2], resulting in 70 different classes. Using the same database Hu and Zanibbi [8] released several experiments based on Hidden Markov Model (HMM) and variant of segmental K-means for the initialization of the Gaussian Mixture Models' parameters . Based on KNN classifier Smithies et al. [9] proposed a fast user-trained algorithm using features vector of 50 dimensions.

The techniques of handwriting recognition Online have been widely explored latest years. When the recognition is realized from a offline form, most of the research has focused in printed mathematical expressions and mathematical symbols. U.Garain et al [3] proposed a multiple_classifier approach that use a group of classifiers arranged hierarchically. the first classifier recognize some of the frequently occurring symbols by the looking for the presence of certain strokes . Then the symbols that are not recognized by the first Classifier are passed to a group of three classifiers. For each classifier a different set of features are calculated, the second classifier is based on a run-number feature vectors to classify symbols. Feature vector used by the third classifier is based on percentage of black pixels calculated from rectangular grids extracted from the symbol bounding box, two more descriptor, density of black pixels in the entire symbol and the height-to width ratio are added to give 27 dimensional feature vectors. The final classifier used wavelet transform that characterizes different physical structures of symbol images at different resolution levels. For Alvaro et al., each bounding box was normalized and the Euclidean distance between two images is calculated based on the difference of each pixel [5]. This method obtained a 94.24% symbol recognition rate in InftyCDB-1 database. Malon et al described a successful approach to multi-class classification by adding binary support vector machines (SVM) which are trained with directional histograms of the contour. In [7]authors Comparing four Techniques for Offline recognition of printed mathematical symbols. In addition of the use of the knn[9] and svm[6] 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 Neighbors (WNN) and Hidden Markov models(HMM) which have been widely used for symbol classification in online mathematical expression

recognition [2] but their use in the offline mathematical expression recognition remains unexplored. the proposed classification techniques are tested in two different database, SVM and WNN achieved the best results However the worst results were obtained with HMM.

Besides the classification techniques, the use of the appropriate set of features is of great importance and effects the recognition results. The performance of a classifier can rely as much on the quality of the features as on the classifier itself. A good set of features should represent characteristics that are particular for one class and be as invariant as possible to changes within this class [10]. Commonly used features in offline Arabic character recognition are: projections, invariant moments , Fourier descriptors, zoning feature , and contour direction histogram . A feature set made to feed a classifier can be a mixture of such features. For exemple A. Boukharouba and A. Bennia [12] compined a features set based on transition information in the vertical and horizontal directions combined with Freeman chain code for the recognition of arbic/farsi digits by an svm classifier

3. Proposed Method

As depicted in Figure 3, the proposed method followed the typical pattern recognition system architecture that achieved in three main steps: preprocessing, features extraction, and recognition phase.

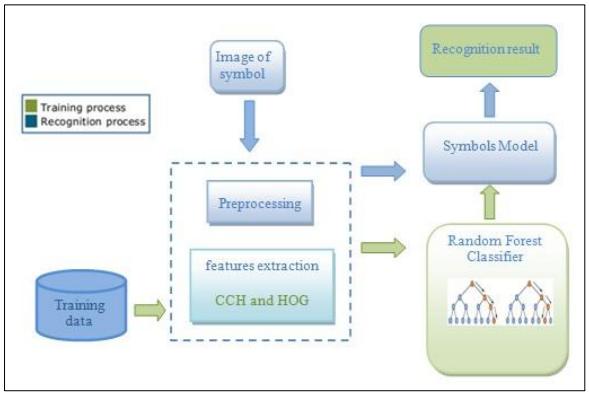

Figure 3: The proposed method

Preprocessing

Preprocessing 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. then we convert the grayscale 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 grayscale 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. After binarization, the next step of preprocessing is trimming to adjust the bounding box of the symbol.

Features Extraction

The Features Extraction is critical stages 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. We combine the Chain Code Histogram (CCH) on its two forms global and local and the Histogram of Oriented Gradient (HOG) to generate the features vector for our system. In the next sections we describe CCH and HOG clearly.

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 values which are shown in figure 4. 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.

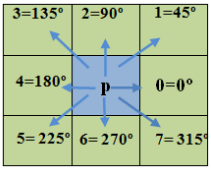

Figure 4: The 8-chain code directions

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 [17]. Therefore ,for the suggested system, in addition to the 8-directional CCH of the whole symbol image as shown in figure 5(a) , we propose to divide the image into 4 (2 * 2) block, for each block the 8-directional CCH is computed figure 5(b). as a result, we totally obtained 40 dimensional CCH vector.

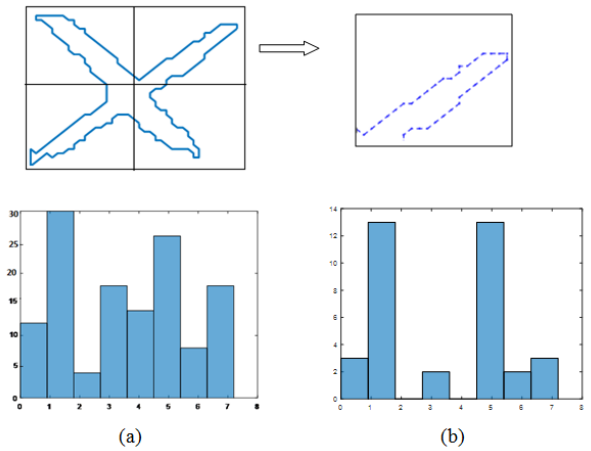

Figure 5: the chain code histogram descriptor multiplication symbol (a) global CCH of the whole symbol image, (b) local CCH of the top-right bloc of symbol

Histogram of Oriented Gradient

HOG feature descriptor proposed by Dalal and Triggs for detection of humans [18] is one of successful features for object detection and recognition. the key idea of HOG is that the shape of objects in an image can be characterized by the distribution of local intensity gradients representing the dominant edge. To compose HOG, the image of mathematical symbol is splitted into *N*N* cells, for each cell the horizontal and vertical gradient of the image is computed by convolving the image with the respective gradient masks and . To compute the gradient orientation at each pixel, a transformation into polar coordinates is used through Eq. 1,2.

| ( )| √( ) ( ) ( ) (1)

dir ( ) ( ) (2)

( )

The possible orientations, due to the selected gradient filter, are nine. Later a histogram of 9 bins gradient directions is computed for the pixels inside the cell. To normalize the histogram of each cell, a L2 Norm is used. Finally the histograms of all cells are concatenated to form the descriptor. Figure 6 gives the process of HOG descriptor acquisition.

The dimensionality of the descriptor is a function of the number of cells *N*N* and the number of bins *k* in the histogram. In this approach, we set the number of cells to 4*4 and the number of bins to 9 dimensional HOG for each cell. The 16 (4*4) histograms with nine bins were then concatenated to make a 144 dimensional features vector.

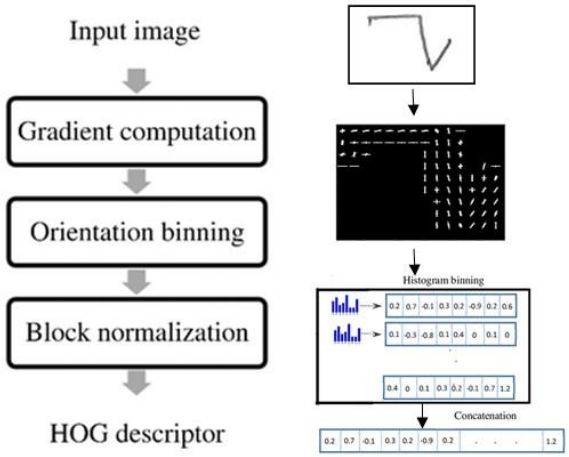

Figure 6: the process of HOG descriptor acquisition.

Random Forests Classification

Random forests introduced by Breiman in 2001[11] rate among the most recent and popular boosting methods and have proven their classification performance for difficult problems in many applications [14][15][16].

A random forest is a classifier consisting of a collection of tree-structured classifiers *{ h(x ,), k = 1, . . .}* where the { } are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x. the idea behind the Random Forest is growing up an ensemble of classification trees, where each tree contributes with a single vote for the assignment of the most frequent class to the input data. For the training of random forests each tree in the forests is built on random subsets of the original training set (the Bagging principle) with a CART-like induction algorithm figure 7 gives the architecture of Random forests . This tree induction method, is a CART-based algorithm that modifies the feature selection procedure at each node by selecting ***mtry*** variables at random out of all ***M*** possible variables (independently at each node) then find the best split on the selected ***mtry*** variables . The induction of tree is

performed without pruning step so each tree is a maximal one. Consequently this algorithm works according to two main parameters : the number ***L*** of trees in the forest, and the number ***mtry*** of features pre-selected for the splitting process.

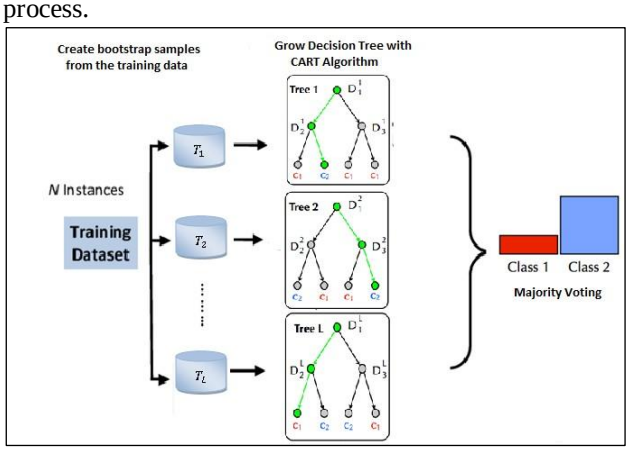

Figure 7: Architecture of the Random forests

When the training set for the current tree is drawn by sampling with replacement, about one-third of the cases are left out of the sample. This oob (out-of-bag) data is used to get a running unbiased estimate of the classification error as trees are added to the forest. It is also used to get estimates of variable importance. The Random Forests is trained using *bootstrap aggregation*, where each new tree is fit from a bootstrap sample of the training observations ( ). The *out-of-bag* (OOB) error is the average error for each calculated using predictions from the trees that do not contain in their respective bootstrap sample. This allows the Random Forests to be fit and validated whilst being trained. These popular classifiers have been shown to be very robust against noisy data as well as overfitting and have a good performance in high dimensional data

3.3.1. Dynamic Random Forests

Traditional Random Forests algorithm introduced by Breiman grows trees independently from one another. Hence, each new tree is arbitrarily added to the forest. In [21], authors have shown that when using classical RF induction algorithms, some trees degrade the performance of forest, and that a well selected subset of trees can outperform the initial forest. Hence the idea of Dynamic Random Forests which 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 [20]. Therefore, a weighting of each randomly selected training instance according to the predictions given by all the trees already added to the forest. Thus, one possible measure can be a ratio of trees in the existing forest that have predicted the true class. This ratio is defined by:

( ) ____ Σ (( )) (3)

| |

Where *I(.)* is the indicator function; *x* is an input data point and *y* its true class; ( ) is the i-th classifier output; and stands for the set of out-of-bag trees of x, i.e. the trees for which x is an out-of-bag instance. The lower value of ( ) the more the next tree will have to focus on the instance x, since it means that it was incorrectly classified by a large number of trees in the current forest. Consequently, the weighting function that will attribute a weight to x has to decrease with respect to ( ). For our system , we used the following function:

(( )) ( ) (4)

4. Experimental Results

This section presents the results obtained by the proposed approach. Evaluation of the Dynamic Random Forests using different parameters as well as the combination between the HCC et HOG descriptors to justify the choice of the proposed system. All the tests were performed on the public HAMF dataset[19]. using a 2.7 GHz Intel i5 processor

Dataset

We evaluate our method on a large handwritten dataset of Arabic mathematical expression and isolated symbols named HAMF [19]. 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 will use the second set because our approach is interested in the recognition 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 (13 532 samples) and the remaining third for the test (6768 samples).

5	5	5	1	1	1
4	4	4	2	2	2

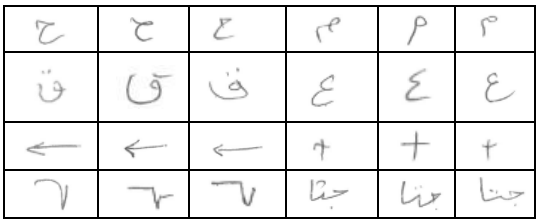

TABLE 2: EXAMPLE OF HANDWRITTEN ISOLATED SYMBOLS

Recognition results

The random forests works according to two main parameters : the number L of trees in the forest , and the number mtry of features pre-selected for the splitting process in the tree. As there is no verified formula to find optimal parameter values for the forest, numerous trials were done with different adjustment of random forest parameters . First we study the behavior of the method according to the number of trees, based in three different set of features: the set of global and local CCH which consist of 40 dimensional vector, the second set composed by 144 HOG descriptor and the third set combine the two sets of features previously cited to discuss the influence of the features quality on the performance of the classifier. Concerning the number *L* of trees, we have picked eleven increasing values, from 10 to 500 trees and we fixed arbitrarily the *mtry* parameter on 10 features.

Figure 8 presents the average classification accuracy of the random forest classifier respect to the number of trees and with different features. We can see a global tendency of the recognition rate to raise for an increasing number of trees. It appears that this increasement is not linear but logarithmic. One can conclude from this, that with respect to an increasing number of trees, the Random Forests accuracy converges. It seems on this figure that the rise of the recognition rate begins to considerably slow down from 200 trees in the forest. We conclude also that the combination of the CCH set features and the set of HOG descriptors done the best recognition accuracy.

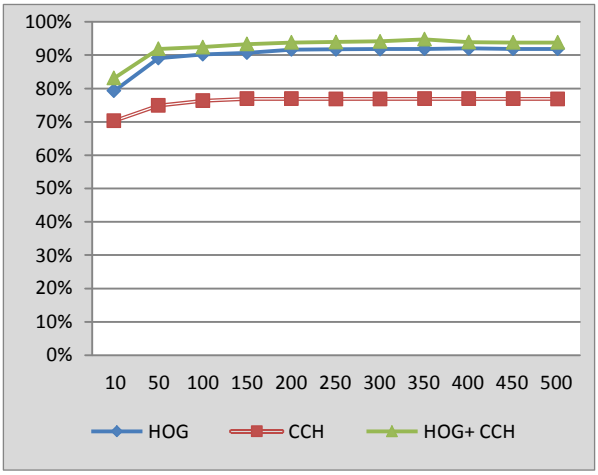

Figure 8: The average classification accuracy of the random forest classifier according different number of trees with different features set.

Table 3 shows recognition rate provided by applying the proposed recognition method based on CCH and HOG features to the HAMF dataset, which is composed of seven subset as illustrated in Table1. The random forests classifier trained with 300 trees and 10 features randomly pre-selected for the splitting procedure, the classification rates is computed independently for each subset and for the whole road symbols of the HAMF dataset.After careful examination of the samples that were incorrectly recognized, we concluded that most of these samples are hard to recognize even by a human expert reader.

Table 3: Recognition rate corresponding to different subset

	recognition rate	
	Test	**Train**
Arabic characters	93.45	100
Digits Latin	97.82	100
Arithmetic operators	100	100
Comparison operators	98.27	100
Functions	97.53	100
Elastic operators	99.99	100
Others	99.81	100
All symbols	**94,78**	**99.98**

5. Conclusion

In this paper, we proposed an efficient system for Arabic handwritten mathematical symbols recognition. We combined global and local block-based CCH vector with HOG features. For the classification, we used a random forests classifier. Extensive experiments with various number of trees and features pre-selected were performed on the HAMF dataset[19]. The highest accuracy was reached using 300 trees and 10 random splitting features with 94,78 % in the test set and 99.98% in the train set.

References

[1] F. Álvaro, et al, Classification of On-line Mathematical Symbols with Hybrid Features and Recurrent Neural Networks, in Proceedings12th International Conference on Document Analysis and Recognition, 2013

[2] S. MacLean, G. Labahn, E. Lank, M. Marzouk, and D. Tausky, Grammar-based techniques for creating ground-truthed sketch corpora, International Journal on Document Analysis and Recognition, vol. 14, pp. 65–74, 2011

[3] U. Garain, B.B. Chaudhuri, and R.P. Ghosh. A Multiple-Classifier System for Recognition of Printed Mathematical Symbols, in Proceedings ICPR, vol. 1, pp 380-383, 2004.

[4] S. MacLean and G. Labahn, Elastic matching in linear time and constant space, Document Analysis Systems, Ninth IAPR Workshop on (short paper), 2010

[5] F. Alvaro, J.-A. Sanchez, and J. Benedi, Recognition of printed mathematical expressions using twodimensional stochastic contextfree grammars, in International Conference on Document Analysis and Recognition, pp. 1225 –1229, 2011.

[6] C. Malon, S. Uchida, and M. Suzuki, Mathematical symbol recognition with support vector machines, Pattern Recognition Letters,vol. 29, pp. 1326–1332, July 2008.

[7] F. Alvaro and J. Sanchez, Comparing several techniques for offline recognition of printed mathematical symbols, in 20th International Conference on Pattern Recognition, pp. 1953 –1956, Aug. 2010.

[8] L. Hu and R. Zanibbi, HMM-Based Recognition of Online Handwritten Mathematical Symbols Using Segmental K-Means Initialization and a Modified Pen-Up/Down Feature, International Conference on Document Analysis and Recognition, vol. 0, pp. 457–462, 2011.

[9] S. Smithies, K. Novins, and J. Arvo, A handwriting-based equation editor, Proc. Graphics Interface, pp. 84–91, June 1999

[10] F. Lauer, C.Y. Suen, G. Bloch, A trainable feature extractor for handwritten digit recognition, Pattern Recognition, pp. 1816–1824, 2007

[11] L. Breiman, Random Forests, Mach. Learn. J., vol. 45, pp. 5–32, 2001

[12] Boukharouba A, Bennia A. Novel feature extraction technique for the recognition of handwritten digits, Appl Comput Inform , 2015.

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

.

[14] N. J. Tustison, K. Shrinidhi, M. Wintermark, C. R. Durst, B. M. Kandel, J. C. Gee, M. C. Grossman, and B. B. Avants, Optimal symmetric multimodal templates and concatenated random forests for supervised brain tumor segmentation with ANTsR, Neuroinformatics, vol.13 no. 2, pp.209–225, 2015.

[15] A. Ellahyani, M. El-Ansari and I. El-Jaafari, Traffic sign detection and recognition based on random forests, J. Appl. Soft Comput, vol. 46, 805–815, 2016.

[16] S. Bernard, L. Heutte, S. Adam, Using random forests for handwritten digit recognition, ICDAR 2007: Ninth International Conference on Document Analysis and Recognition, vol. 1–2, pp. 1043–1047, 2007

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

[18] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR, vol. 1, pp. 886–893, 2005.

[19] I. Hadj Ali, M.A. Mahjoub, Database of handwritten Arabic mathematical formula images, in 13th International Conference Computer Graphics, Imaging and Visualization, pp.145-149, 2016.

[20] S. Bernard, S. Adam, L. Heutte. Dynamic random forests. Pattern Recognition Letters, vol. 33, no. 12, pp. 1580-1586, 2012.

[21] S. Bernard, L. Heutte, S. Adam, On the selection of decision trees in random forests, International Joint Conference on Neural Network, pp. 302–307, 2009.