Math Formula Script and Type Identification and Recognition

Kawther Khazri(✉) and Afef Kacem Echi

ENSIT-LaTICE, University of Tunis,
5 Avenue Taha Husseïn, BP 56 Bab mnara, 1008 Tunis, Tunisia
kawther.khazri@yahoo.fr, afef.kacem@ensit.rnu.tn

Abstract. In this work, we propose a system for math formula script and type identification based on Convolutional Neural Network, to automatically discriminate between Printed/Handwritten and Arabic/Latin formulas before their recognition by the appropriate recognizer. An identification rate of 94.6% is reached, tested on 320 formulas. For formula recognition, we focused on Arabic machine-printed formulas and we proposed a syntax-directed system, based on symbols recognition and their arrangement analysis. To recognize symbols, we combined some statistical features and a Bayes network classifier. A rate of 96.56% for symbol recognition is achieved. For formula structure analysis, the system proceeds by top-down and bottom-up parsing scheme based on operator dominance. A set of replacement rules is defined. Formula parsing consists in applying, from the dominant operator and its context, the appropriate rule to divide the formulas into sub-formulas which will be recursively analyzed by the same way. The parser used for the formula structure analysis has shown its efficiency with a recognition rate 97.63%.

Keywords: Script and type identification · Symbol recognition · Formula's structure analysis

1 Introduction

Research on script and type identification aims to create systems able to discriminate automatically between the different forms in which a document is presented, including the language and the way it is written in machine-printed or handwritten, to select the appropriate recognition system to a given document. The state of the art on the script identification shows that no work deals with math formulas. Existent works treat this problem for text. Also, few systems are interested at the same time in Arabic/Latin and Printed/Handwritten script identification. In this context, we present a new approach dealing with the problem of identification of the script: Arabic or Latin and the type: handwritten or machine-printed of math formulas. This work comes as a part of our research on off-line recognition of arabic math formulas. The rest of the paper is organized as follow. Section gives a synthesis of the existing systems for script

R. Vera-Rodriguez et al. (Eds.): CIARP 2018, LNCS 11401, pp. 857–864, 2019.
https://doi.org/10.1007/978-3-030-13469-3_99

identification and math formulas recognition. Sections 3 and 4 present the proposed identification and recognition system. Experiments are reported in Sect. 5. Finally, conclusion and future works are drawn in Sect. 6.

2 State of the Art

For Script identification, most researches focus principally in text document. As far as we know, no work handled with math documents. Script and type identification problems depend on the granularity of data sample: text-bloc, text-line, word or connected component level, the number of scripts out of which the system classifies and the way the text data is presented: handwritten or machine-printed. Based on a survey done by [1] about script and type identification, we summarized some related works (Table 1).

Table 1. Script and type identification

Script	Type	Level	System	Accuracy (%)	Ref.
Arabic English	Machine-printed	text-line word	Projection profile features, Run length and moments, etc. MLP	99.7% test on 1976 text-lines, 98.6% tested on 8320 words	[2]
Arabic Latin	Machine-printed	word	Arabic character recognition using template matching	100% tested on 478 words	[3]
Arabic Latin	Machine-printed, handwritten	text-line	Projection profile and Fractal based features, K-NN, RBF	tested on 2400 text-lines, 96.64% K-NN and 98.72% RBF	[4]
Arabic Latin	Machine-Printed, handwritten	word	Steerable pyramid transform, etc., K-NN	97.5% tested on 800 words	[5]
Arabic Latin	Machine-printed, handwritten	bloc, text-line Ccx	Morphological analysis for text-bloc and geometrical analysis for line and Ccx., K-NN	88.5% tested on 200 images and 92% tested on 113 images	[6]
Arabic Latin	Machine-printed, handwritten	word	HOG, Bayes classifier	98.4% tested on 1320 words	[7]

For math formula recognition, many researches deal with this problem, especially in Latin language [8–13]. In recent years, researches dealing with Arabic formulas have emerged. In [14], Smirnova and Watt proposed to adapt their prior

system for Latin formula recognition [13], to online Arabic context. They used elastic matching for symbol recognition and geometrical structure analyzer for formula recognition. Their system was tested on a database of 227 symbols and achieved a recognition rate of 91.9%. Unless the good results achieved by the symbol recognizer, the use of the elastic matching can be a big limitation for the overall approach since it is strongly affected by the size of the used vocabulary. To recognize the structure of the formula, authors proposed to identify relations between symbols but they did not consider the inclusion relation which make their system unable to recognize roots. In [15], El-Sheikh proposed a system for the online recognition of one-dimensional Arabic math formulas. For symbol recognition, some statistical features are computed. Author developed a precedence grammar based on left to right scanning scheme for the syntactic recognition of math formula. The proposed system recognized 16 isolated letters, 10 digits and 11 symbols. A recognition rate of 99% was achieved. Another system for the recognition of one-dimensional Arabic math formulas was proposed by Khalifa and Bing Ru in [16] which handle with segmentation and recognition of only simple math equations. For symbol recognition, they discriminated connected components according to their proximity properties and they used a two-level neural network as classifier. They achieved a recognition rate of 89.7% for handwritten formulas and 95.2% for printed formulas. Their proposed system do not treat complex level of math formulas. In this work, we are interested by the system proposed by Belaïd et al. [8] for the online interpretation of 2D math Latin formulas. For symbol recognition, authors used morphological features and a decision tree. To interpret formulas, they proposed a syntactic parser based on a context-free grammar. It is a top-down and a bottom-up parser based on a start character which is used to select the appropriate rule and to divide the formula into sub-formulas until the whole formula recognition. A recognition rate of 93% was achieved. Authors proved the importance of contextual information to overcome the shortcoming of the symbol recognizer. Their solution for treating ambiguities, if accompanied by a robust symbol recognizer, will certainly improve the overall system. Also the efficiency of their system will be more convenient if tested on various types of formulas. Convinced by Belaïd's approach, we propose to extend and adapt this approach for Arabic in off-line context.

3 Proposed Identification System

As the content of a math formula being variable, we use a decision at connected component level. For that, we extracted then classified connected components, using a Convolutional Neural Network (CNN). An overview of the proposed CNN is given in Fig. 1. Image symbol is of size 100×100, used as input of the network. The CNN's structure is characterized by the alternation between convolution and sub sampling layers. The convolution serves to extract features from the input image and to output, using a linear filter the feature map. We used a ReLU operation after every convolution operation, to introduce non-linearity in the CNN. We then used a spatial pooling to reduce the dimensionality of each feature map but retains the most important information.

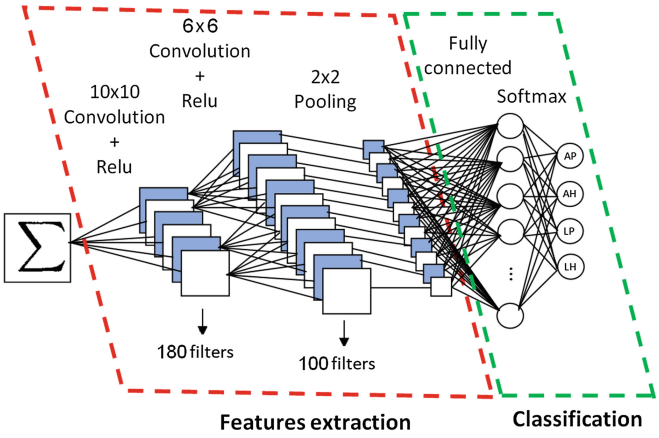

Fig. 1. CNN based system for symbol script and type identification

Once the connected components are classified, we refer to a majority vote on the decision taken for each of them to identify the script and type of the whole formula. In Fig. 2, the proposed CNN returns five Arabic Handwritten (AH) components and only one Latin Handwritten (LH) component. Thus, the formula is classified as AH. Notice that, some components are not identified, either because they are not discriminative or can be confused with other symbols. In Fig. 2, the dot above the function's name is not identified because it can be confused with the Arabic digit zero.

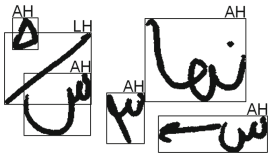

Fig. 2. Formula identification standing on CNN symbol classification.

4 Proposed Formula Recognition System

Two main stages are followed as explained in the next subsections.

4.1 Symbol Recognition

To represent symbols, we extracted 120 statistical features: Hu and Zernike moments, run-length, bi-level co-occurrence, white pixel portion and HOG, are extracted and a Bayes network classifier is used. The proposed symbol recognizer is able to identify 50 symbol classes. To select the appropriate features, we used best first algorithm, which reduced the number from 120 to 87 features to 96.86% and the run time is decreased from 0.19 s to 0.15 s in average.

Although the symbol recognizer achieved a good accuracy, its failure to distinguish certain symbols would be bothersome. In fact, some distinct symbols are in close resemblance such as the horizontal fraction bar and the minus sign. Also mirrored symbols, such as the opening and the closing parenthesis, can cause recognition problems because some used features are invariant to rotation. Observing the event of confusion, we consider some of the misrecognition cases to be too difficult to resolve without considering the context and we keep resolving some of them during the formula structure analysis.

4.2 Formula Structure Analysis

This step consists of lexical, geometrical and syntactical analysis as it will be explained below.

Lexical Analysis: It attributes a lexical unit, a label which is the syntactic category to each symbol or group of symbols. For example, the label *SS* for the literal and mirrored sum symbol. For multi-part symbols ($=, \leq$, etc.), Arabic letters, having diacritic such as (أ, ب, etc.) and function names (نهَا, ظتَا, etc.), vertical regroupment is required. Horizontally adjacent digits should compose an unsigned integers. Unsigned floats consist of unsigned integers separated by a decimal point.

Geometrical Analysis: To describe spatial structure of the formula, we defined 10 spatial relations: Left, Right, Above, Below, Left and Right Superscript, Left and Right subscript, Inside and Delimited by small or great delimiters. These spatial relations, in conjunction with context, are used here to remove some confusions between symbols with similar morphologies. For example, in order for a symbol to be considered as a fraction bar, it should have no empty parts above and below.

Syntactical Analysis: The proposed parser starts by selecting the dominant operator which can be explicit, represented by a symbol like an arithmetic operator, a fraction bar, an integral, a root, a summation, a product, a new function name like ق(س), a trigonometric function such as the sinus function جَا. It can be also implicit such as subscript or a superscript or implicit multiplication. Note that Belaïd et al. [8] defined a similar concept: the starting character which is chosen based on its ability to correctly divide the formula into sub-formulas (according to the grammar) and on its priority when different characters can be used for that purpose. Thus, a priority order was defined to choose the starting character and when more than one character have the same priority, extra treatment were done to determine the best one that gives the maximum information to divide the formula and simplify its parsing. But, they only considered explicit operators. In this work, we propose to include more complex symbols such as sums, products, integrals, roots, etc. and implicit operators: subscripts,

superscripts and implicit multiplication in the choice of the start operator. We compute operator dominance in conjunction with its precedence to handle with formulas that contain many operators which are not lined up. To define dominance between two operators O_1 and O_2, we consider that O_1 dominates O_2 if O_2 lies in the range of O_1. The range of an operator is the possible emplacement of its operands. After finding the dominant operator, a top-down and a bottom-up parsing algorithm is applied to analyze the formula structure. The bottom-up parser begins by looking for the dominant operator, as explained above. Then, it chooses the corresponding rule in the grammar, considering the operator contexts. This rule provides instructions to the top-down parser to partition the formula into sub-formulas which are analyzed by the same way and so on until analyzing the whole of the formula. More details can be found in our previous works [17–19].

5 Experimental Results

To train and evaluate our systems, we used for Latin script the InftyMDB-1 [20], a database of printed math formulas and CROHME [21], a database of handwritten math formulas. View the absence of standard database of Arabic math formulas, we used our database of printed formulas scanned form math books of several Arabic countries, and of handwritten formulas written by five different writers. To evaluate the identification system, we trained our CNN using a database of 4000 samples (1000 per class, 4 classes: AH, AP, LH, LP). For the tests, we used a 1400 connected components (350 instances per class). Table 2 displays the obtained results. We also built a database of 320 formulas (80 per class, 4 classes: AH, AP, LH, LP) using the previously cited databases. Table 2 shows the obtained results. To evaluate the formula recognition system, we tested the symbol recognizer on 1016 ones extracted from 100 test formulas. 930 were correctly recognized and 86 were not recognized which means a recognition rate of 91.5% which is better than the result obtained with the same test formulas in our previous work 89.9% [17]. Some of the encountered confusions were treated during the lexical analysis guided by the characteristics of the Arabic math notation which involves diacritic and multi-parts symbols. For example, the presence of the Hamza above a letter Alef, approves its identity as letter Alef and its absence guides our system to choose the second result of the symbol recognizer. Some other encountered confusion cases have been solved during syntactical analysis guided by the conventional syntax of formulas. For example, greater than or less than signs can not just before or after an equal sign, parenthesis or bracket, an arithmetic sign. When finding these symbols, our system corrects them, referring to the alternative candidates from the symbol recognizer. When considering spatial relationships, the symbol recognition rate has grown from 95.77% to 96.56% [17].

The proposed syntax directed system was tested on a database of 161 formulas (see Table 3). Formulas of order 0 are those where operators are aligned in the same line without superscripts nor subscripts. Formulas of order 1 enclose

Table 2. Identification rate of the CNN based system.

Class	Symbol classification (%)	Formula classification (%)
AH	93	100
AP	95	97.5
LH	92.6	93.75
LP	92.9	87.5
Average	93.4	94.6

subscripts, superscripts and roots. Formulas of order 2 allow operator below and above the horizontal fraction bar and formulas of order 3 include integrals, summations, etc.

Table 3. Parsing results.

Order0	Order1	Order2	Order3	Average
98% (99 form.)	95.76% (42 form.)	99.62% (8 form.)	100% (12 form.)	97.63%

6 Conclusion and Future Work

In this work, the focus was on the problem of math formula script and type identification and recognition. We firstly proposed an identification system able to automatically discriminate between printed and handwritten, Arabic and Latin math symbols based on CNN, then exploited the obtained result to identify the script and type of the whole formula before employing a particular recognizer. We then addressed the problem of formula recognition. The proposed recognition system was tested on complex math formulas containing implicit multiplication, subscripts and superscripts and gives satisfactory results. We also explained how our system offers the possibility to detect and correct some symbol recognition errors during the different steps of formula's structure analysis. Adding more features, testing other feature selection algorithms and choosing faster classifier should enhance the performance of the proposed system. Based on our experimentations, we showed that the CNN-based identification system results were promising with 94.6% identification rate. Also we argue the robustness of the recognition system, carrying tests on a reasonable number of practical math formulas. In fact, our system proves its efficiency with a recognition rate of 97.63%. In future work, we plan to work on improving the performance of the proposed CNN-based system working on the CNN's filters and architecture.

References

1. Ubul, K., Tursun, G., Aysa, A., Impedovo, D., Pirlo, G., Yibulayin, T.: Script identification of multi-script documents: a survey. IEEE ACCESS **5**, 6546–6559 (2017)
2. Elgammal, A.M., Ismail, M.A.: Techniques for language identification for hybrid Arabic-English document images. In: ICDAR, pp. 1100–1104 (2001)
3. Moalla, I., Elbaati, A., Alimi, A.A., Benhamadou, A.: Extraction of Arabic text from multilingual documents. In: ICSMC (2002)
4. Moussa, S.B., Zahour, A., Benabdelhafid, A., Alimi, A.M.: Fractal-based system for Arabic/Latin, printed/handwritten script identification. In: ICPR (2011)
5. Benjelil, M., Mullot, R., Alimi, A.: Language and script identification based on Steerable Pyramid Features. In: ICFHR (2012)
6. Kanoun, S., Ennaji, A., Lecourtier, Y., Alimi, A.M.: Script and nature differentiation for Arabic and Latin text images. In: IWFHR, pp. 309–313 (2002)
7. Saïdani, A., Kacem, A.: Arabic/Latin and machine-printed/handwritten word discrimination using HOG-based shape descriptor. In: ELCVIA (2015)
8. Belaïd, A., Haton, J.P.: A syntactic approach of handwritten mathematical formula recognition. PAMI **6**(1), 105–111 (1984)
9. Stria, J., Prusa, D., Hlavac, V.: Combining structural and statistical approach to online recognition of handwritten mathematical formulas. In: CVWW (2014)
10. Celik, M., Yanikoglu, B.: Probabilistic mathematical formula recognition using a 2D context-free graph grammar. In: ICDAR (2011)
11. Tian, X., Wang, F., Liu, X.: An improved method of formula structural analysis. In: ICDAR, pp. 161–166 (2011)
12. Awal, A., Mouchere, H., Gaudin, C.: Towards handwritten mathematical expression recognition. In: ICDAR, pp. 1046–1050 (2009)
13. Wan, B., Watt, S.: An interactive mathematical handwriting recognizer for the pocket PC. In: MathML International Conference (2002)
14. Smirnova, E., Watt, S.: Aspect of mathematical expression analysis in Arabic handwriting. In: ICDAR, vol. 2, pp. 1183–1187 (2007)
15. El-Sheikh, T.S.: Recognition of handwritten Arabic mathematical formulas. In: UK IT Conference, pp. 344–351 (1990)
16. Khalifa, M., Bing Ru, Y.: A hybrid segmentation system of offline Arabic mathematical expression recognition. CJIPCV **2**(4), 30–35 (2011)
17. Ayeb, K.K., Echi, A.K., Belaïd, A.: A syntax directed system for the recognition of printed Arabic mathematical formulas, In: ICDAR, pp. 186–190 (2015)
18. Kacem, A., Khazri, K., Belaïd, A.: Reconnaissance de formules mathématiques arabes par une approche dirigée syntaxe. In: CIFED (2010)
19. Khazri, K., Kacem, A., Belaïd, A.: Recognition of machine-printed Arabic mathematical formulas. In: ICTIA (2014)
20. Infty Project Homepage. http://www.inftyproject.org/en/database.html. Accessed 15 Oct 2018
21. CROHME Homepage. http://www.isical.ac.in/~crohme/. Accessed 15 Oct 2018