A Comparative Study on Methods and Tools for Handwritten Mathematical Expression Recognition

Daniela S.Costa
Centro de Informática, UFPE
Recife, PE, Brazil
dsc4@cin.ufpe.br

Carlos A.B.Mello
Centro de Informática, UFPE
Recife, PE, Brazil
cabm@cin.ufpe.br

Marcelo d'Amorim
Centro de Informática, UFPE
Recife, PE, Brazil
damorim@cin.ufpe.br

ABSTRACT

Handwritten mathematical expression recognition (HMER) is a challenging task due to factors such as ambiguity, variety of writing styles, and complexity of two-dimensional writing. In this paper, we identify challenges in HMER applications through experiments that simulate real scenarios that go far beyond the usual cases found in literature: variations on luminance; different stroke width, inclination and color; different background pattern; and partially shaded images. The results of state-of-the-art methods (as TAP and Dense-WAP) and a commercial tool (MathPix) are analyzed, using the CROHME 2016 database. We proved that, although the area has had a lot of improvement in recent years, there are still issues to overcome.

CCS CONCEPTS

• **Computing methodologies**;

KEYWORDS

handwritten mathematical expression recognition, experimentation, deep learning

ACM Reference Format:
Daniela S.Costa, Carlos A.B.Mello, and Marcelo d'Amorim. 2021. A Comparative Study on Methods and Tools for Handwritten Mathematical Expression Recognition. In *ACM Symposium on Document Engineering 2021 (DocEng '21), August 24–27, 2021, Limerick, Ireland*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3469096.3474936

1 INTRODUCTION

Mathematical expression recognition is the translation of images of mathematical expressions into editable text. For handwritten expressions, we have HMER. There are several different approaches to deal with this task. According to the type of input, the methods are divided into online or off-line. In the online version, time stamp, the sequence of the writing of the strokes, the pressure of the pen on the digital surface, are possible features. In the off-line version, the digitized image is the only input.

As an example of an online method, TAP (Track, Attend, and Parse) [11] is composed by a tracker and a parser: the tracker uses a stack of bidirectional recurrent neural networks with gated recurrent units (GRU) to model the input strokes. The parser uses GRU with guided hybrid attention (GHA) to generate LaTeXnotation. It was tested in the International Competition on Recognition of Handwritten Mathematical Expressions (CROHME) 2014 and 2016 [6, 7] with the best results. However, there are cases when handwritten mathematical expressions are erroneously translated due to information loss in TAP encoder according to the authors of [4]. To solve these problems, Residual Bidirectional GRU (Res-BiGRU) is used in [4], where a bidirectional residual neural network (BiRNN) is used as encoder and an attention mechanism of TAP as decoder.

Using the offline perspective, WAP (Watch, Attend and Parse) [9] works directly with two-dimensional structures instead of trees or graphs. It is an improved encoder-decoder network with a fully-convolutional network (the watcher) that converts the input image into an intermediary representation which is transformed into a corresponding LaTeXsequence by GRUs (the parsers) with an attention mechanism that focus on the mathematical elements of the image. WAP achieved very good results in CROHME 2014 and 2016. It was improved in [10] with the use of densely connected convolutional networks to strengthen feature extraction and to facilitate gradient propagation on small training sets (called Dense-WAP).

In addition to these neural architectures, there are some tools available for handwritten mathematical expression recognition applications: ExpressMatch [1], MathBrush [5], Wolfram Alpha [13], and MathPix Snip [12]. This last one (MathPix Snip) is a commercial tool that scans images of mathematical expressions and translates them into editable code as LaTeX, Microsoft Word, HTML, etc. It is very resourceful, supporting mathematical or chemistry expressions, tables, and even some non-English languages.

This paper aims to explore some algorithms and tools for HMER under different scenarios that simulate real situations. They were defined considering the scope of a major project that aims to translate images into code, called Visual Sketch Coding [2]. For this, we have used CROHME dataset, changing: the stroke width, color or inclination, the background, and simulating a complete or partial shading. In Fig. 1, we can see a real example and the result from WAP; it is clear how the result can be completely wrong in an extreme situation.

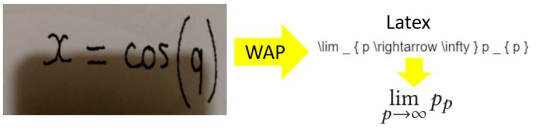

Figure 1: Example of a real case scenario in WAP.

DocEng '21, August 24–27, 2021, Limerick, Ireland

ACM ISBN 978-1-4503-8596-1/21/08…$15.00
https://doi.org/10.1145/3469096.3474936

This paper is divided as follows: next section details the methodology used in the experiments, Section 3 presents the results, while Section 4 concludes the paper.

2 METHODOLOGY

The methodology for evaluating and comparing the HMER methods consists of situations that could happen in real world applications. Since there are no HMER public dataset that contemplates such variations, these conditions were applied synthetically. Considering this, we chose the CROHME 2016 dataset [7], from which a subset of 75 expressions written by different writers was chosen for the experiments. All images contained in this subset were correctly recognized by MathPix [12], creating a baseline for the recognition experiments; none of them are part of CROHME 2014.

The choice of methods to run in the experiments considered the number of citations and the availability of the original source code. Thus, the HMER methods chosen are TAP and the Dense-WAP. According to Google Scholar, TAP is cited by 42 papers, since 2018; the original WAP, by 80 papers since 2017, with Dense-WAP cited by 55 papers since 2018 (this information was achieved in May, 2021). For commercial tool, we have chosen MathPix due to its accuracy, and robusteness..

Since TAP is an online method, we have used the stroke extractor described in [3], allowing to work with TAP in an off-line mode. The models trained and provided by the authors for each network were used. Both were trained with CROHME 2014 dataset [6] which contains 8,836 handwritten mathematical expressions. No image used for training the networks was used for testing.

2.1 Scenario with skew angle variation

For this scenario, some level of inclination was imposed on the expressions. This is a common situation, especially when we consider cases where the writing is done in a paper without guidelines. The experiments consist of submitting mathematical expressions to the following rotation angles: 5°, 10°, 15°, 20°, 25°, 30°, 35°, 40°, and 45°. It is intended to assess which levels of rotation are supported by the systems and what is the impact of this type of variation for the accuracy of the methods. This scenario has a total amount of 675 images and Figure 2.a shows some examples of rotated expressions.

2.2 Scenario with changes in stroke width

All CROHME datasets are in an online format, *i.e.*, they are in the form of a set of strokes made by points. In this case, storing the data in InkML file format has its advantages, such as the possibility of providing segmentation and labels for each symbol of the expression. However, digital images are a more usual format. In this scenario, we analyze how the stroke width variation can affect the accuracy of HMER methods. As the width of the stroke can compromise the legibility of some expressions, an interval of 1 (thinner stroke) to 4 (thicker stroke) pixels was chosen, keeping the symbols readable. This set consists of 300 images. An example of the application of this type of variation is shown in Figure 2.b.

2.3 Scenario with background patterns

Instead of an uniformly white background, this scenario tries to simulate cases where the expressions are written on lined or checked sheets of paper. The main objective is to verify the impact for the recognition tools when they find vertical or horizontal lines that are not originally part of the expressions. If not properly extracted, they can be easily misclassified as valid symbols such as the division bar. Variations in the color of the paper were also considered, where the yellowish color is used to simulate cases of colored papers. HMER tools should only segment the expressions, not being influenced by features from the sheet of paper. For this experiment, four different background patterns were used, all containing paper styles usually found in note books (some examples are presented in the Figure 2.c). For this scenario, 300 images were created.

2.4 Scenario with luminance variation

Images captured under low light conditions generally suffer from low visibility, compromising the recognition rate of objects. Considering that the user will not always have good lighting and/or a device with an efficient lighting correction algorithm, this scenario simulates images captured in low light. The objective is to verify the extent to which the HMER methods can correctly recognize the symbols of the expressions. Altogether, 5 levels of luminance were applied to the images, ranging from 0.1 (darker images) to 0.5 (lighter images), every 0.1. Considering these levels of luminance, 375 images were generated and tested (see Figure 2.d).

2.5 Scenario with partial shading

This scenario considers that, when a user takes a photo of a paper, he/she can project a shadow in the scene. Unlike the previous scenario (whose illumination is uniform all over the image), the images generated by this experiment explore the ability of HMER's methods to fully recognize the expression and not just the part that is under good lightining conditions. Also, we want to check if the shape of the shadow can provoke errors due to some kind of misclassification, especially in the boundary between the shadow and the paper. For this scenario, three types of shading were tested: shadows projected horizontally, vertically, and diagonally. Each shade generated synthetically has an intensity of illumination that is randomly chosen within the range of 0.1 (darkest) to 0.5 (lightest), every 0.1. Experiments for this scenario consisted of 75 images (see Figure 2.e).

2.6 Scenario with variation on ink color

Usually, the problem of recognizing handwritten mathematical expressions is divided into three tasks: line segmentation, single symbol recognition and structural analysis. The experiments proposed by this scenario are designed so that the ink of the pen is as close as possible to the color of the lines of the paper. In this way, with little contrast between them, we want to check how well the algorithms can differentiate what are strokes from mathematical expressions and what are background lines. The colors black, blue and red are considered for the ink (Figure 2.f) because they are commonly found in ballpoint pens. The variations proposed by this scenario were added to the initial set of 75 images, maintaining the total number of images.

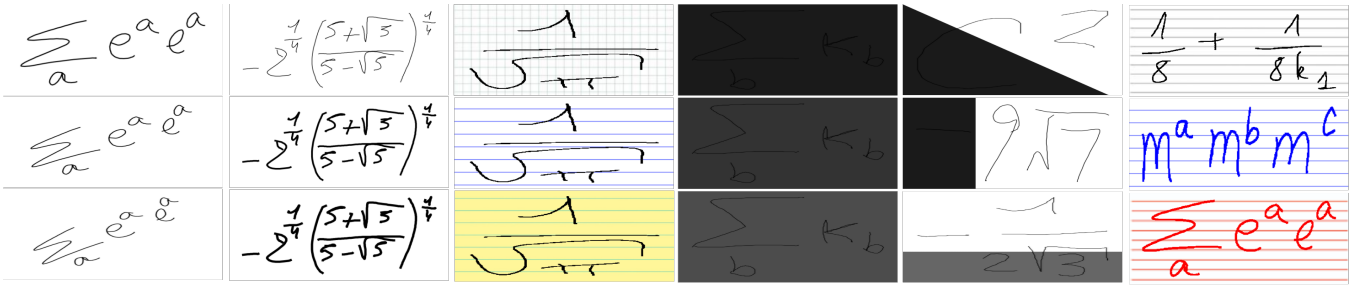

Figure 2: Examples for each scenario proposed by this paper.

3 EXPERIMENTS AND RESULTS

In this section, we present and compare the results obtained by Dense-WAP, TAP, and MathPix methods in the datasets detailed before. As stated, our objective is to verify how the performance of these systems is affected under extreme conditions.

To measure the performance of handwritten expression recognition tools, we report the test results in terms of the expression recognition rate (ExpRate), *i.e.*, the percentage of mathematical expressions correctly recognized, which provides a global performance metric. For ExpRate, the higher the better.

Before applying the scenarios, we checked the performance of Dense-WAP, TAP, and MathPix for the chosen subset of images in its original form (black ink and white paper, no noise). MathPix recognized all the symbols in the database. On the other hand, Dense-WAP and TAP had difficulties in the recognition reaching rates of 12% and 44% of correctness, respectively.

For the skew variation scenario, for each handwritten expression, rotations of 5°, 10°, 15°, 20°, 25°, 30°, 35°, 40°, and 45° were applied. Table 1 shows the results for each rotation angle except for the 40° and 45° because all methods failed in these cases. All the approaches have decreased their results with some inclination of the text. However, MathPix has better results up to 30° of inclination. Even though its performance decreased from 100% to 81.33% with a 5° skew angle. Dense-WAP has lower results for every case. TAP, although not the most efficient, demonstrates resilience in the face of high rotation angle, with no recognition with a rotation of 40°.

Skew angle	Dense-WAP (%)	TAP (%)	MathPix (%)
5°	10.66	37.33	81.33
10°	9.33	32.0	68.0
15°	5.33	29.33	57.33
20°	0.0	16.0	38.66
25°	0.0	6.66	17.33
30°	0.0	2.6	6.66
35°	0.0	1.33	0.0

Table 1: ExpRate for scenarios with skew angle variation.

As described in Section 2.2, the tests involving variation of stroke width were divided considering 4 values of width. Table 2 shows the results obtained by each HMER method in terms of global accuracy (ExpRate). Dense-WAP was unable to recognize expressions when they had widths equal to 1 and 4 pixels, reaching approximately 15% of accuracy for 3 pixels width. TAP tends to be better for expressions written with thinner strokes, reaching around 50% of accuracy for 1-pixel width. However, even in its best case, TAP is still far from the accuracy presented by MathPix, which demonstrated an average accuracy of 76.83%. Finally, MathPix appeared to suffer little impact by the variation proposed in this scenario.

Stroke width	Dense-WAP (%)	TAP (%)	MathPix (%)
1	0.0	48.0	76.0
2	12.0	40.0	77.33
3	14.67	37.33	74.66
4	0.0	33.33	73.33

Table 2: ExpRate for scenario with stroke width variation.

For the experiments with background patterns, Dense-WAP was unable to recognize expression which contains customized background pattern. Meanwhile, TAP managed to hit 20% of the submitted images, showing that it can deal with a certain level of background not used in its training. In turn, MathPix tool reached the best recognition with 65.33% ExpRate.

The images generated synthetically with luminance variation try to simulate situations where the image is captured under low light conditions. The results obtained by the experiments are presented in Table 3. Dense-WAP is not included in this table, because it was unable to recognize any case of luminance variation. On the other hand, the performance of TAP and MathPix is affected by the decrease of the illumination: for low luminance factor ($L = 0.5$) both of them had a decrease in ExpRate. However, in the tested range, they have small variation in the metric. Even in the worst scenario tried ($L = 0.1$), they were not affected by very sharp deteriorations. it is interesting to see that they have higher scores for cases where lighting factor is lower; in the scenario with lower L, Mathpix still reached 80% of accuracy. Perhaps the fact that the lighting variation is applied uniformly throughout the image has contributed to the methods still being able to identify the symbols even in such poorly illuminated cases.

Unlike previous experiments that change the lighting uniformly, experiments concerning the scenario of partial shading consist of evaluating the HMER methods with images that contain partially shaded regions. The idea is that, when considering non-uniform

Luminance factor (L)	TAP (%)	MathPix (%)
0.1	46.66	80.0
0.2	48.0	77.33
0.3	46.66	77.33
0.4	46.66	78.66
0.5	41.33	77.33

Table 3: ExpRate for scenario with luminance variation.

illumination, parts of the handwritten expressions can be hidden in the darkest regions and consequently they are ignored by HMER methods. As in the case of luminance variation, Dense-WAP was unable to recognize any of the partially shaded images. Both TAP and MathPix had difficulty recognizing handwritten expressions, presenting 18.66% and 24% of accuracy, respectively.

The tests with differernt ink colors are based on cases where the ink of the pen is similar to the lines of a note book. This situation, depending on the strategy adopted by the HMER method, can make the task of segmenting strokes even more difficult. Since, in the scenario with background pattern, Dense-WAP was not able to recognize the expressions, the experiments were conducted just with TAP and MathPix. The difficulties are best observed if we compare these results with those in the scenario with background. Including only the background, TAP obtained 20% accuracy, but when we also consider the ink of the pen, this value drops to 12%. MathPix, which previously had 65.33% of accuracy, now has 40%.

4 CONCLUSIONS

In this paper, we propose a comparative study for Handwritten Mathematical Expression Recognition. For that, we tested two state-of-the-art methods (Dense-WAP and TAP) and the commercial tool MathPix in six scenarios that simulate real-world situations. The experiments were conducted on the CROHME 2016 dataset and the results were evaluated in terms of the expression recognition rate (ExpRate).

In the first scenario (skew angle variation), the experiments consisted of submitting the expressions to different skew angles ranging from 5° to 45°. From the results, it can be seen that for rotations up to 15°, all evaluated methods are able to show some level of recognition (even with small hit rates). However, when the rotation of the expressions increases, the performance of such methods is strongly affected. For the second scenario (stroke width variation), the results showed that the MathPix tool was the only one that did not seem to suffer in its performance. For the third scenario (different background patterns), only TAP and MathPix can somehow segment and correctly recognize some of the images. The experiments for the fourth scenario (luminance variation) indicate that Dense-WAP was unable to recognize any of the expressions which contain luminance variance. Unlike this, TAP and MathPix can achieve some recognition, even for poorly illuminated images. Taking into account cases where there are partial shadows in the images, the results for the fifth scenario (partial shading) point to difficulties in recognition for all methods, with low values of accuracy. Finally, the experiments in the sixth scenario (different ink color) confirmed that, if an expression is written with an ink with the color close to the color of the guidelines of the background pattern, HMER methods have a lot of difficulty.

Considering all the scenarios covered in this paper, the ones that were most challenging for HMER methods were those that involved the presence of background pattern, partially shaded images and to deal with colors from ink and background that are similar. It would be interesting to have HMER methods, in the future, capable of dealing with such situations since they can easily occur in real world applications.

As future work, we have two possible approaches. The first is to create a pipeline to improve the quality of the images for further recognition. For the scenarios explored in this paper, the pipeline runs based on the following sequence: (i) ink segmentation (to separate the ink from a background pattern, creating a colored image); (ii) binarization through a local method that could deal with illuminance variation; (iii) skew detection and correction is performed in the black and white image; (iv) finally, the stroke width must be adjuted to a standard best value through morphological operations. The second possible solution is the creation of a full dataset with several samples of each case for new training of Dense-WAP and TAP (this will probably require changes in the networks).

The database is already available for research purposes in: https://drive.google.com/file/d/1hocUxfoPBEC2fdVVVWh6GsNEAYx84gm9/view?usp=sharing. The access to the files requires a password which will be provided by the authors after prior contact.

5 ACKNOWLEDGMENTS

The authors would like to thank: the Foundation for Science and Technology Support in Pernambuco (Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco - FACEPE) for the financial support of the work; NVIDIA Corporation for the donation of a Titan XP GPU used for this research; and MathPix for providing a license for the experiments.

REFERENCES

[1] Aguilar, F.D.J.J., and Hirata, N.S.T. ExpressMatch: A System for Creating Ground-Truthed Datasets of Online Mathematical Expressions. In: DAS 2012, pp.155–159 (2012)
[2] D'Amorim, M., Abreu, R., and Mello, C.A.B. Visual sketching. In: ICSE '20: 42nd International Conference on Software Engineering, pp.101–104 (2020)
[3] Chan, C. Stroke extraction for offline handwritten mathematical expression recognition. In: IEEE Access, v. 8, pp.61565–61575 (2020)
[4] Hong, Z., You, N., Tan, J., and Bi, N. Residual BiRNN based Seq2Seq Model with Transition Probability Matrix for Online Handwritten Mathematical Expression Recognition. In: ICDAR, pp.635–640 (2019)
[5] Labahn, G., Lank, E., MacLean, S., Marzouk, M., and Tausky, D. MathBrush: a system for doing math on pen-based devices. In: DAS 2008 (2008)
[6] Mouchère, H. et al. ICFHR 2014 competition on recognition of on-line handwritten mathematical expressions. In: 2014 ICFHR. IEEE, pp.791–796. (2014)
[7] Mouchère, H., Viard-Gaudin, C., Zanibbi, R., and Garain, U. ICFHR2016 CROHME: Competition on Recognition of Online Handwritten Mathematical Expressions. In: 15th ICFHR, pp.607–612, Shenzhen (2016)
[8] Zhang, T., Mouchere, H., and Viard-Gaudin, C. Tree-based BLSTM for mathematical expression recognition. In: ICDAR, pp.914–919 (2017)
[9] Zhang, J., et al. Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition. In: Pattern Recognition **71**, pp.196–206 (2017)
[10] Zhang, J., Du, J., and Dai, L. Multi-Scale Attention with Dense Encoder for Handwritten Mathematical Expression Recognition. In: ICPR, pp.2245-2250 (2018)
[11] Zhang, J., Du, J., and Dai, L. Track, Attend, and Parse (TAP): An End-to-End Framework for Online Handwritten Mathematical Expression Recognition. In: IEEE Transactions on Multimedia **21**(1), pp.221–233 (2019)
[12] MathPix Snip: https://mathpix.com/
[13] Wolfram Alpha: https://rhttps://www.wolframalpha.com/