IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 9, 2021, accepted February 17, 2021, date of publication March 2, 2021, date of current version March 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3063413

Online Handwritten Mathematical Expression Recognition and Applications: A Survey

DMYTRO ZHELEZNIAKOV[1,2], VIKTOR ZAYTSEV[2], AND OLGA RADYVONENKO[2]

[1]Faculty of Computer Science and Cybernetics, Taras Shevchenko National University of Kyiv, 01601 Kyiv, Ukraine
[2]Samsung Research and Development Institute Ukraine, 01032 Kyiv, Ukraine

Corresponding author: Dmytro Zhelezniakov (dmitry.zhelezniakov@gmail.com)

ABSTRACT Handwritten mathematical expressions are an essential part of many domains, including education, engineering, and science. The pervasive availability of computationally powerful touch-screen devices, similar to the recent emergence of deep neural networks as high-quality sequence recognition models, result in the widespread adoption of online recognition of handwritten mathematical expressions. Also, a deeper study and improvement of such technologies is necessary to address the current challenges posed by the extensive usage of distance learning, and remote work due to the world pandemic. This paper delineates the state-of-the-art recognition methods along with the user's experience in pen-centric applications for operating with handwritten mathematical expressions. Recognition methods have been categorized into classes, with a description of their merits and limitations. Particular attention is paid to end-to-end approaches based on encoder-decoder architecture and multi-modal input. Evaluation protocols and open benchmark datasets are considered as well as the comparison of the recognition performance, based on open competition results. The use of handwritten math recognition is illustrated by examples of applications for various fields and platforms. A distinctive part of the survey is that we also considered how UI design relies on the use of different recognition approaches, which is aimed at helping potential researchers improve the performance of the introduced approaches toward the best responses in practical applications. Finally, this paper presents the prospective survey of future research directions in handwritten mathematical expression recognition and their applications.

INDEX TERMS Deep learning, handwriting mathematical expressions recognition, human-computer interaction, LSTM, neural networks, ubiquitous computing.

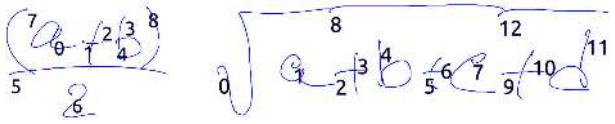
I. INTRODUCTION

Over the past decade, significant advances in sequence recognition and computer vision models based on deep neural networks (DNN), and the ubiquitous expansion of touch and pen-enabled phones and tablets have led to an increase in interest in handwritten document processing. Handwriting is a natural part of everyday human interaction. These days, in addition to widespread smartphones and tablets, new types of devices such as interactive panels, digital pens and smart writing surfaces have become widely adopted in offices and educational institutions, opening up new opportunities for technologies for recognizing specific handwritten content such as mathematics, diagrams, charts, tables, sketches, etc. At the same time, the sudden outbreak of COVID-19 pandemic reveals another scene to users and puts forward new requirements for handwriting interaction applications in education, distance learning, and remote work. To address this problem, it is necessary to make a deeper study and improve the technologies of handwriting recognition in terms of their practical applications.

Mathematical expressions (MEs) are a fundamental part of engineering, science, finance, education, and other domains. MEs differ from the textual representation by the presence of a two-dimensional (2D) structure and a large codebook (more than 1,500 symbols [1]), where the characters are often very similar to each other, especially for handwritten MEs (HME).

Handwriting input of ME is often preferred by users to keyboard and mouse, which is much slower [2]. Despite promising new developments in HME recognition, this task is still at a level where recognition errors happen quite often. Such errors can cause user dissatisfaction. A good user interface

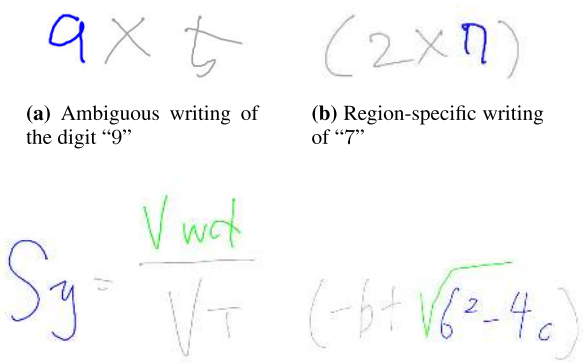
The associate editor coordinating the review of this manuscript and approving it for publication was Moussa Ayyash.



(a) An example of delayed symbols

(b) An example of a complex radical symbol, consisting of several strokes torn in time

FIGURE 1. Examples of challenges related to stroke order.



(a) Ambiguous writing of the digit "9"

(b) Region-specific writing of "7"

(c) Uncertainties in defining spatial relations: Sy or S_y, Vwd or V_{wd}

(d) Ambiguous expression with radical: $\sqrt{b^2} - 4c$ or $\sqrt{b^2 - 4c}$

(e) Segmentation uncertainty: $1 < b$ or kb

(f) An example of explicit and implicit multiplication in one expression: $\frac{4}{3}x - \frac{2}{3} \times (465 - x)$

FIGURE 2. Some examples of challenges in HME recognition.

(UI) strengthens significantly the system and user experience (UX), whereas ineffective UI can spoil UX even with almost perfect HME recognition accuracy. Hence, efficient input of HMEs should be regarded as a combination of recognition system and UI that facilitates ME input and enables the user to resolve inaccuracies quickly.

Recognition of HME could be considered from two points of view: "online" and "offline". Online recognition operates with a dynamic representation of input (the traces of pen/finger movement), and offline recognition considers a static representation (an image).

A. HME RECOGNITION CHALLENGES

More often, HME recognition is compared to handwriting text recognition. The recognition of HMEs is a much more difficult task, mainly due to the 2D layout. Such a structure poses impediments at all stages of handwritten ME processing, from preprocessing to the final expression construction. If in preprocessing of handwritten text one of the main problems is "delayed" stroke associated with diacritics, then when recognizing ME, the whole characters or subexpressions can be "delayed". An example of such a case is an expression where parentheses are written after the subexpression. Also, users can often correct a character by adding new strokes after writing the entire expression. The simplest example is to change the addition symbol "+" into the asterisk symbol "∗". It often happens that one character can be augmented several times. Examples of such characters are radicals and fractions, which expand as the user writes related subexpressions. Figure 1 shows some examples of expressions with delayed strokes, where the digit next to the gesture indicates the ordinal number of the stroke.

Another challenge is the large size of the alphabet, which makes some expressions extremely difficult to recognize even for a human. The most common examples are ambiguities associated with very similar or even the same writing of many characters. There, plenty of symbols have the same writing in lower and upper case, where an example of such characters would be "X/x", "C/c", "K/k", and many others. The use of Latin, Greek and Roman characters also adds confusion due to the similarity of many characters, such as "B/β", "p/ρ", "x/χ", and "n/η". Notation ambiguity also exist between symbols and operators such as "$x/\times$", "$1/|$", "$S/\int$", "$o/\circ$", "$t/+$", "$\angle/<$". Writing multiple symbols side by side can also cause ambiguity in the segmentation and classification of symbols. Here are some examples of such ambiguities: ")(" – "x", "13" – "B", "1 <" – "K".

Ambiguity arises not only due to a large number of very similar characters, but also due to the peculiarities of each person's handwriting or regional specifics of writing. Regional differences also include discrepancies in mathematical notation. Some countries have their own naming conventions. For example, the notation "tg" can be used for the tangent function. This feature can have a huge impact on the rules for constructing expressions.

Besides the ambiguities, associated with segmentation and character classification, the next major problem is the spatial relations classification. In mathematical notation, spatial relationships are mostly used as implicit operators. Although the number of spatial relationships used is small, relationships between elements can be very vague. Also, an implicit multiplication operator between subexpressions is often used in mathematical notation, which can also create some difficulties in determining the correct structure of an expression. Examples of various ambiguities related to segmentation, classification, and structure analysis are shown in Figure 2.

Preparing datasets for training and verification is also challenging task as it requires a significant dataset to be collected, validated and labeled. A great number of approaches require significant manual labour to annotate the collected datasets character by character.

B. PREVIOUS SURVEYS ANALYSIS

Several reviews in this field have already been published. Table 1 summarizes previous surveys. Since then, there have been significant changes in recognition methods that have

TABLE 1. Overview of existing surveys.

Publications	Year	Brief description
Blostein and Grbavec [3]	1997	The major attention in this survey is paid to the consideration of structural analysis methods for offline and online ME recognition. Very little attention is paid to the problems of segmentation and classification of symbols, classification of spatial relations, and matrices recognition.
Chan and Yeung [4]	2000	This review covers all the stages of online and offline ME recognition, from character segmentation to expression construction.
Tapia and Rojas [5]	2007	This is the first review that was completely devoted to the problems of online recognition. The survey also provides an overview of applications for the first time. But, as in the previous ones, this review does not compare different approaches. This is due to the lack of open data for comparison at the time.
Zanibbi and Blostein [6]	2012	This is the most comprehensive overview. In addition to considering online and offline recognition methods, the survey described the issues of evaluating recognition systems and reviewed existing datasets. This review also explores math retrieval systems in detail.

led to the improvement in recognition accuracy, which has been achieved mainly due to breakthroughs in DNN such as recurrent neural networks (RNN) and encoder-decoder based architectures. Also, the existing surveys mainly focus on the analysis of recognition methods, often not giving the required attention to practical applications of HME recognition and aspects of UI/UX design associated with employed approaches.

The prime aims of this work are: to trace the evolution of HME recognition methods with a focus on new approaches that have emerged over the past decade, including new end-to-end recognition approaches (Section II); to consider performance evaluation methods along with the description of open datasets for training and verification, as well as the results of open competitions (Section III); to discuss UI design approaches with regard to various recognition methods and applications to help potential researchers in improving the performance of the introduced approaches toward the best responses in practical applications (Section IV); to argue the directions for future research (Section V).

II. RECOGNITION SYSTEMS

The first works in the field of recognition of MEs date back to the second half of the 1960s [7], [8]. Although research works between the 1960s and 1980s focused mainly on the recognition of printed MEs in an image, they laid the foundation for further development. With the evolution and widespread of touch and pen devices in the early 2000s, such as Pocket PCs, the interest in handwriting input and online recognition has grown significantly. Contemporary approaches, based on sequence-to-sequence deep machine learning (DML) have led to an epoch-making improvement in the recognition accuracy in sequence recognition problems.

In general, the online HME recognition problem can be formulated as the transformation of the handwriting strokes into a tree representation such as MathML or LaTeX. Recognition of mathematical notation traditionally involves two stages [4]: (1) symbol recognition and (2) structural analysis. The symbol recognition stage contains the following tasks: (1) stroke preprocessing, which includes a variety of methods such as size normalization, stroke interpolation, slant correction, and resampling; (2) symbol segmentation, or grouping of the input strokes that belong to the same character; (3) symbol classification, or the strokes group labeling. The goal of structural analysis is to identify spatial relationships between elements, to find a mathematical interpretation of an expression, and to produce its mathematical notation. Figure 3 illustrates the classical workflow for ME recognition. Zanibbi and Blostein [6] considered recognizing MEs in the context of document recognition and emphasized on an additional task: detection of the ME in the document.

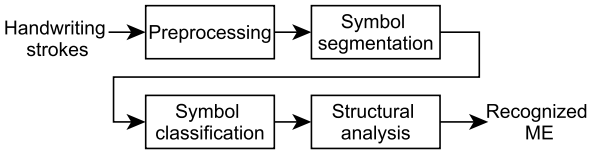


FIGURE 3. Recognition workflow in sequential solutions.

Zhang [9] indicated three epochs of mathematical recognition systems: sequential solutions, integrated solutions, and end-to-end neural network-based solutions (Figure 4). Sequential solutions are characterized by the fact that the result of the previous stage is used on the next one. This leads to the propagation and accumulation of errors. In the integrated solutions, a set of symbol hypotheses is generated, and the structure analysis module uses the best symbol candidate to construct the proper ME taking into account grammar and semantic knowledge. End-to-end solutions transform an input representation (image or set of strokes) directly

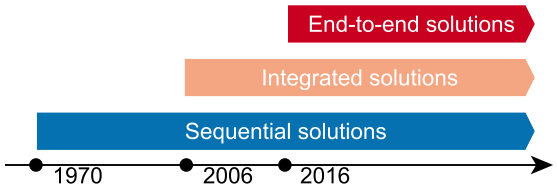
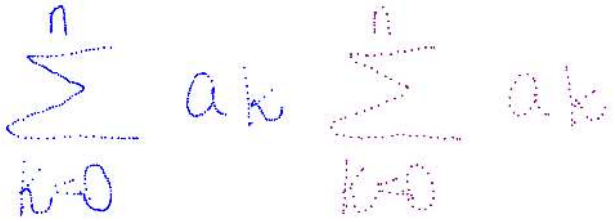


FIGURE 4. Epochs of the mathematical recognition systems evolution.



(a) Original strokes **(b)** Preprocessed strokes

FIGURE 5. Strokes preprocessing.

to mathematical notation. Whereas the set of methods and techniques for sequential and integrated solutions have been studied quite deeply in recent decades, end-to-end solutions began to be adopted for ME recognition quite recently.

A. SYMBOL SEGMENTATION & CLASSIFICATION

1) STROKES PREPROCESSING

Preprocessing is the first step in HME recognition and has a significant impact on the accuracy of the recognition model. The main goal of this step is to unify the input data in order to improve the reliability of the recognition algorithm for different input data. Typically, preprocessing of handwritten strokes in online recognition includes: removing duplicated points, and the hooks of the strokes, slant correction, smoothing input points, filling intermediate points, resampling, size normalization [10]–[12]. The Figure 5 illustrates the result of preprocessing. However, ME recognition is associated with a 2D structure where the order of strokes is not specified and depends on the user's preference. Therefore, the preprocessing stage often involves stroke reordering. Le *et al.* [13] proposed an optimized X-Y cut for reordering strokes to ensure stroke order independence. This method is based on the detection of horizontally and vertically ordered strokes. Classification of vertically ordered strokes is based on the detection of special vertical symbols such as a fraction bar, '$\int$', '$\sum$', and 'lim'. This algorithm requires the use of a character recognizer in the preprocessing stage.

2) SYMBOL SEGMENTATION

A lot of strategies have been explored for symbol segmentation in handwritten and printed ME. Early approaches have relied on the X and Y projections (X-Y cut) technique [14]–[16], and were mainly aimed at the segmentation of printed ME. In [17]–[19] approaches for simultaneous symbol segmentation and classification were proposed. The so-called ''soft-decision'' algorithm tries to find the most probable sequence of characters and keeps all the variants of character segmentation and recognition until the final decision. Lehmberg *et al.* [18] firstly proposed Symbol Hypotheses Net (SHN). Kosmala and Rigoll [19] solved the segmentation problem using Hidden Markov Models (HMM). A Minimum Spanning Tree (MST) approach has been demonstrated by Matsakis [20]. Toyozumi *et al.* [21] presented a segmentation method based on the candidate character lattice using the position relation of strokes and mathematical structure information, which allowed to reduce the problem of over-segmentation and under-segmentation. A technique that is based on the contour feature was presented by Tian and Zhang [22]. Hu and Zanibbi [23] proposed a symbol segmentation method using AdaBoost algorithm and geometric multi-scale shape context features, which included 3 groups of features: stroke pair, local neighborhood, and global shape contexts. The SVM-based classifier for symbol segmentation was employed by Le and Nakagawa [24]. Twelve geometric features and nine additional features were applied to calculate the separation probability. The final decision on whether a pair of strokes belonged to the same segment or not was made using a threshold value. Hu and Zanibbi [25] presented a Line-of-Sight stroke graph and symbol segmentation algorithm based on these graphs and Parzen window-modified Shape Context features (PSC). PSC features are based on a log-polar coordinate system and are widely used in computer vision. This list of works can be supplemented with the results of research in the field of optical character recognition (OCR) [26]–[28].

3) SYMBOL CLASSIFICATION

There are three general groups of methods for handwriting recognition: online, offline, and combined [24]. Obviously, the set of input strokes can be converted into an image, and then offline techniques can be applied for the classification. But it is also possible to employ online methods for offline recognition. Chan [29] developed a stroke extraction algorithm for offline ME. This method includes multiple steps such as binarization, skeletonization, decomposition into segments, stroke reconstruction, order normalization, etc. As the author points out, offline to online transition approaches require retraining the online recognition model with extracted strokes. In this survey, we will focus on online and combined methods. Table 2 summarizes symbol classification methods.

Nowadays, sequence-to-sequence methods based on the Bidirectional Long Short-Term Memory (BLSTM) neural networks with Connectionist Temporal Classification (CTC) [50], [51] represent the state-of-the-art in the complex sequence labeling tasks such as speech [52] and text [53] recognition. The main feature of BLSTM models is that they can use contextual information over a long period, considering both the next part of the input sequence and the previous one. CTC refers to the scoring functions, and the use of CTC implies the introduction of an extra '*Blank*' symbol in an alphabet. The equation (1) is used as CTC loss function

TABLE 2. Categorization of methods for symbol classification.

Method	Brief description (classifier method)	Publications
Online	Hidden Markov Models	Kosmala and Rigoll [19], Winkler [30], Winkler and Lang [31], Hu and Zanibbi [32]
	Bidirectional Long Short-Term Memory	Dai Nguyen et al. [33], Zhang et al. [34]
	Bidirectional Long Short-Term Memory and Connectionist Temporal Classification	Liwicki and Bunke [35], [36]
	Nearest-neighbor classification	Smithies et al. [37]
	Multi Layer Perceptron	Awal et al. [38]
	Adaptive Resonance Theory neural architecture	Dimitriadis and Coronado [39]
	Time Delayed Neural Network	Awal et al. [40]
	Conditional Random Fields (CRF)	Lafferty et al. [41]
	Artificial Neural Networks and multi-layer perceptron as a junk classifier	Awal et al. [42]
	Template Matching	Simistira et al. [43]
	Hausdorff ARTMAP Neural Network	Thammano and Rugkunchon [44]
Combined	Support Vector Machines	Keshari and Watt [11]
	Hidden Markov Models (online + offline features)	Alvaro et al. [45]
	Long Short-Term Memory and Convolutional Neural Network	Dai Nguyen et al. [46]
	Long Short-Term Memory and Hybrid Features set	Muñoz [47]
	Artificial Neural Networks with rejection of false hypotheses	JulcaAguilar et al. [48]
	Markov Random Field and Modified Quadratic Discriminant Functions	Le and Nakagawa [24]
	Squeeze-extracted multi-feature convolution neural network	Fang and Zhang [49]

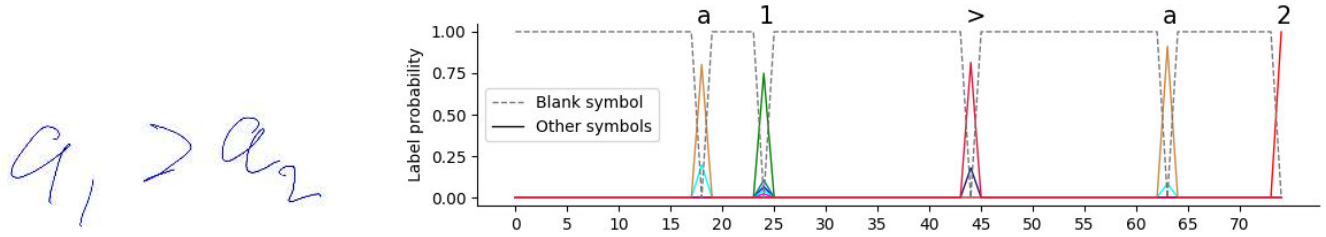
$O(S)$, which is defined as the negative log probability of the correct labeling of the training set S, where x is the input sequence, and z is the ground-truth labeling:

$$O(S) = -\ln\left(\prod_{(x,z)\in S} p(z|x)\right) = -\sum_{(x,z)\in S} \ln p(z|x) \quad (1)$$

CTC tends to find symbol prediction in the corresponding part of the input sequence [54]. However, using CTC in HME recognition can cause difficulties since HME recognition requires high segmentation accuracy. Figure 6 illustrates the output of BLSTM neural network with CTC, where the X-axis corresponds to the input points, and the Y-axis corresponds to the symbol probability for each point.

Typically, the probability spike occurs at the last point of the new character, while elsewhere the probability of '*Blank*' character tends to 1.0. Liwicki and Bunke [35] investigated a set of 25 online and pseudo-offline features and suggested a subset of 16 features for BLSTM-based recognition systems. Dai Nguyen *et al.* [33] studied BLSTM recognition method using 6 time-based features (x_i, y_i – normalized coordinates; x_i', y_i' – derivatives; d_i – the distance between the current and the next point; $EndPoint_i$ – the last stroke point flag). Zhang [9] proposed method which uses features vector from 5 features ($\sin\theta_i$, $\cos\theta_i$ – sine and cosine directors of the tangent of the stroke; $\sin\Delta\theta_i$, $\cos\Delta\theta_i$ – sine and cosine direction changes; $PenUD_i$ – state of pen down-up) and employed in-air points (non-visible strokes that connect two visible strokes). To overcome the CTC segmentation issue, she proposed the modified CTC algorithm, which was named local CTC. Zhelezniakov *et al.* [36] utilized BLSTM based on normalized features vector with 3 features (Δx_i, Δy_i, and $PenUD_i$). The segmentation problem was resolved by balancing datasets and applying postprocessing rules. Volkova *et al.* [55] suggested a lightweight neural network for segmentation correction.

Convolutional neural networks (CNN) are acknowledged for offline recognition tasks and play a vital role in computer vision applications [56]–[58]. There are several works on CNN architecture applied to online HME recognition problems. Nguyen *et al.* [59] presented a combination of CNN and LSTM for character recognition, which increased the recognition quality by 1.57%. Fang and Zhang [49] introduced a new approach to isolated symbol recognition called squeeze-extracted multi-feature convolution neural network (SE-MCNN). This approach utilizes 8-directional features [60] to convert the stroke path into a feature map and thereby compensate for the loss of dynamic information. For



(a) Input strokes **(b)** BLSTM output

FIGURE 6. An example of BLSTM output with CTC-based training.

offline mode, the corresponding 8-directional pattern images are generated using the Gabor filter. Also, as a replacement for the softmax, the Joint Loss function was proposed, so that the model can distinguish the similar handwritten symbols better and reduce the variation of features within the symbol class. Approaches that use a combination of classifiers require more computational resources, and therefore their use is limited in many cases, for example, in mobile devices.

In contrast to classical batch recognition, Phan *et al.* [61] developed an incremental recognition method for online HME. Character candidates are updated after receiving each new stroke. The proposed method has reduced the waiting time, but this approach is stroke-order-dependent and, therefore, has limitations, related to the processing of delayed strokes.

B. STRUCTURAL ANALYSIS

Structural analysis is the final step in the classic ME recognition workflow. The recognition engine classifies the spatial relationship between recognized symbols and groups of symbols. Based on spatial relations information, it generates the best interpretation of the ME in 2D notation. There are a great number of approaches to structural analysis, but all these methods can be classified into 2 groups: graph-based and grammar-driven techniques. Graph-based techniques produce a directed graph based on geometric features of strokes and/or symbols, and often such techniques implicate additional rules for error detection and correction. In turn, grammar-driven approaches rely on grammar production rules and parsing. The entire score calculation is based on symbol recognition confidence, grammar production rules confidence, relation confidence, and often language model features to produce an accurate prediction. Usually, the complexity of grammar-driven approaches is higher, but they prevent the construction of the incorrect expression like $Ca + b$). Moreover, the creation of production rules in grammars-driver approaches should be done very carefully, taking into account the specific domain (such as chemistry, geometry, physics, etc.). The total number of production rules can easily be up to several hundred. However, despite its complexity, these grammar-driver approaches are widely used and demonstrate better quality, compared to graph-based. This is

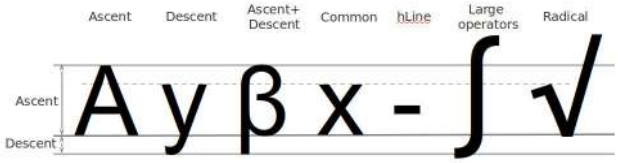


FIGURE 7. Examples of typographic classes of symbols.

substantiated by the fact that the use of grammar can remedy errors in symbol recognition or spatial relation classification.

1) SPATIAL RELATION CLASSIFICATION

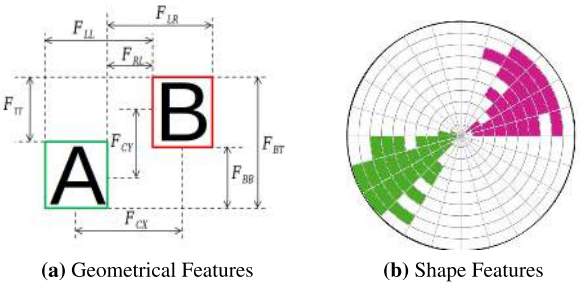
Classification of spatial relation is presented in both approaches to structural analysis. The following spatial relation classes are commonly used: '*Right*' (ab), '*Subscript*' (a_b), '*Superscript*' (a^b), '*Above*' ($\sum^a$), '*below*' ($\sum_a$), '*Inside*' ($\sqrt{a}$). The list of spatial relation classes can also include: '*Pre-Subscript* ($_ba$), '*Pre-Superscript*' (^ba). Instead of creating a separate class for the power of the root ($\sqrt[b]{a}$), a relation of type '*Pre-Superscript*' or '*Above*' is often used. Despite the small number of spatial relation classes, determining the correct type of relation is still challenging. This is largely due to the confusion between right, subscript, and superscript. Researchers have studied a variety of methods of spatial relationship classification. A brief overview is provided in Table 3. There are two sets of features for the classification of spatial relations: geometric and shape features. All the bounding box approaches used typographic symbol classes to compute geometric features (see Figure 7). Le and Nakagawa [24] introduced a concept of body boxes based on the 4 typographic symbol classes. The approaches based on shape features construct shape context as a polar histogram layout descriptor. Examples of geometric and shape features are shown in Figure 8. Approaches based on geometrical features and shape features demonstrate the order of the same accuracy.

2) GRAPH-BASED

Almost all the graph-based techniques are aimed at the construction of the oriented graph where each node represents a symbol, and each edge represents spatial relation. Such graphs are called Symbol Layout Tree (SLT) or Symbol

TABLE 3. Spatial relation classification approaches.

Feature sets	Brief description (classification algorithm)	Publication(s)
Geometric features	Fuzzy membership functions for subscript and superscript analysis	Zhang et al. [62]
	Support Vector Machines (6 Features)	Simistira et al. [63]
	Support Vector Machines (9 Features)	Álvaro et al. [64]
	Sequence of Support Vector Machines models (4 Features)	Le and Nakagawa [24]
	Bayesian classifier (2 Features)	Aly et al. [65]
	Gaussian Mixture Model	Awal et al. [40]
	Decision Tree (8 Features)	Zhelezniakov et al. [36]
	Random Forest ($\approx$ 200 features: Fuzzy Membership + Symbols typology + Geometric Features)	Lods et al. [66]
Shape features	Support Vector Machines for Polar Shape Matrix classification	Muñoz [47]
	Nearest-Neighbor	Ouyang and Zanibbi [67]
Combined features	Random forest for stroke pair and symbol pair relation classification (8 Geometric + Parzen window modified Shape Context (PSC) in polar coordinate system)	Hu and Zanibbi [68]
Others	Simultaneous character recognition and spatial relation classification with tree-based BLSTM	Zhang [9]
	Heuristic Rules	Tapia and Rojas [69]



(a) Geometrical Features **(b)** Shape Features

FIGURE 8. Examples of spatial relation classification features.

Relation Tree (SRT). The main advantage of the graph-based approach is a low complexity which usually does not exceed $O(N^2)$, where N is the number of recognized symbols.

Lee and Wang [70] suggested the system for expression construction, using recursive analysis and character grouping around special symbols such as '$\sum$', '$\int$', '$\prod$', '$\sqrt{}$', and a fraction line. Spatial relation analysis includes blocking rules for subscript and superscript for specified categories of symbols. In the final stage, the correction of recognition error is performed using the set of preassigned rules. The virtual link network was introduced by Eto and Suzuki [71], the proposed network has multiple edges with different spatial relation labels and local costs. The recognition process consists of two stages: virtual link network construction and finding the final spanning tree. The final spanning recognition graph is determined by the minimum cost, which consists of local and global costs. The local cost is defined by the ambiguity of the relation type decision between the two symbols. The global cost reflects the global structure and is calculated using the set of rules. Toyota *et al.* [72] extended the previous work and proposed an approach for incorrect ME filtering by using grammar as a verification step. Zanibbi *et al.* [73] investigated an approach for SLT construction by searching linear structures ('baselines'). Rhee and Kim [74] solved the ambiguity problem by incrementally adding a symbol hypothesis one by one and scoring each hypothesis with a heuristic function.

Tapia and Rojas [69] were the first to propose the method for structure analysis of ME using Minimum Spanning Tree (MST) and symbol dominance. They used Prim's algorithm for undirected MST construction. The weight function $W(S_T, S_N)$ is based on verification of dominance during the edge weight calculation between symbols. If symbol T dominates a symbol N, then function W returns the minimal distance between nearby points of symbols S_T and S_N. Otherwise, it returns the distance between centroids of S_T and S_N. Further, this method of ME constructing was extended by Hu and Zanibbi [68]. Initially, Line-of-Sight (LOS) graph is constructed taking into account the visibility between symbols. The spatial relation classifier is employed as a weight function for edge score generation in a LOS graph. Then Edmonds' algorithm is used to find the spanning tree. The leftmost node on the main baseline is specified by a special dummy symbol, which was inserted into the LOS graph in the beginning.

Lods *et al.* [66] presented the concept of the Fuzzy Visibility Graph (FVG). Edges are defined by a spatial relation classifier, which includes a supplementary class '*Junk*'. A parser with predefined rules for removing redundant edges is applied to transform the obtained FVG to the valid ME. Zhang *et al.* [75] proposed a tree-based BLSTM system that generates SLT directly from the input strokes and does not contain the classical stages: character recognition and structural analysis. An intermediate stroke-based graph is built

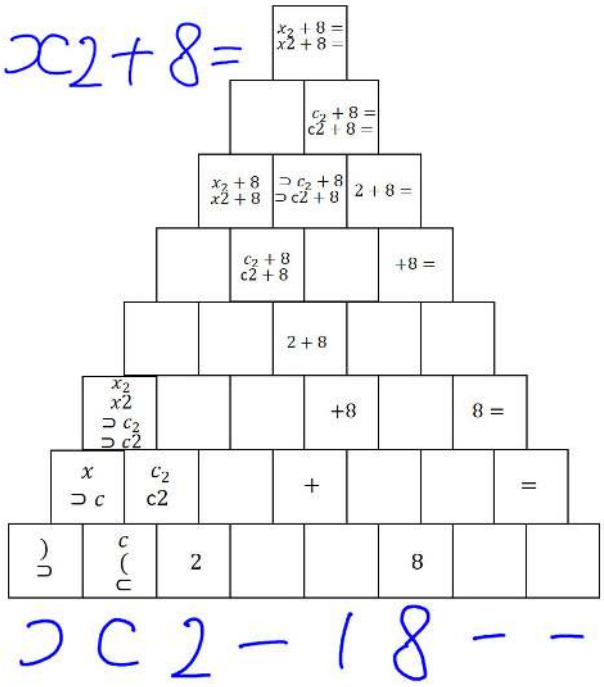


FIGURE 9. Examples of CYK table for '$x_2 + 8 =$'.

using spatial relation and temporal features. Then multiple trees are labeled by tree-based BLSTM. After that, the result is constructed from several trees based on the highest probability.

3) GRAMMAR-DRIVEN

The syntactic approaches to the recognition of two-dimensional visual languages have been studied over the past fifty years [7], [76]–[81]. All grammar-driven techniques are based on the use of context-free grammar formalism. Nowadays, the approaches based on 2D Stochastic Context-Free Grammars (SCFG) are the most studied in the ME recognition domain. Chou [82] first proposed the idea of using 2D SCFG to recognize MEs. The proposed approach utilized a two-dimensional probabilistic version of the Cocke-Younger-Kasami (CYK) algorithm. Yamamoto *et al.* [83] offered a system that simultaneous symbol recognition and structure analysis under the constraints of a 2D SCFG. In other words, stroke-based grammar was proposed. Alvaro has further studied and developed 2D SCFG-based method in numerous articles [47], [64], [84], [85]. He described a stroke-level approach based on parsing 2D Probabilistic Context-Free Grammars (PCFG) with the CYK algorithm that employs bottom-up parsing and dynamic programming. Usually, the CYK parsing algorithm is illustrated as a triangular table (see Figure 9), where each cell contains candidate hypotheses and their score. The bottom level contains non-terminal symbols that correspond to the basic elements from which the expression is built. The lowest level is built from segmented and classified symbols. However, for stroke-based grammar, the bottom level contains hypotheses for the single stroke instead of a symbol. Each next level contains a valid expression that can be generated in accordance with the grammar. The level in the CYK table determines the length of the expression. Accordingly, the last level contains a set of expressions allowed by the grammar that contain all the original elements (symbols or strokes). Production rules for the CYK algorithm are defined in Chomsky normal form (CNF) ($A \rightarrow \alpha$ and $A \rightarrow BC$, where A, B, and C are nonterminal symbols, α is a terminal symbol). Example of production rules for ME parsing algorithm:

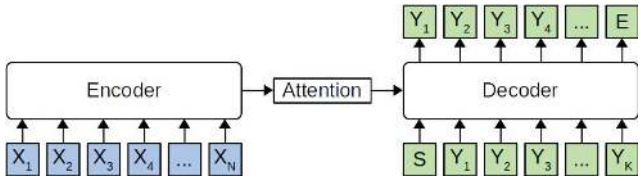


FIGURE 10. The Encoder-Attention-Decoder framework.

$$
\begin{aligned}
TERM &\Longrightarrow symbol \\
TERM &\Longrightarrow number \\
LEFT &\overset{\rightarrow}{\Longrightarrow} operator\ TERM \\
EXPR &\overset{\rightarrow}{\Longrightarrow} TERM\ LEFT \\
DENOM &\overset{\downarrow}{\Longrightarrow} hline\ TERM \\
FRAC &\overset{\downarrow}{\Longrightarrow} TERM\ DENOM \\
SUP &\overset{\nearrow}{\Longrightarrow} TERM\ TERM \\
SUB &\overset{\searrow}{\Longrightarrow} TERM\ TERM
\end{aligned}
$$

Several studies [86]–[88] have addressed the efficiency and complexity of the CYK algorithm, which is equal to $O(N^3|P|)$ for one-dimensional languages and $O(N^4|P|)$ for 2D languages, where $|P|$ is the number of production rules in the grammar and N is the number of expression primitives(strokes). In his works, Alvaro presented approaches of reducing the complexity and finally has achieved $O(N^3 \log N|P|)$ [85]. Le and Nakagawa [87] suggested pruning infeasible hypotheses in the parse tree. Reference [36] reduced recognition time by integrating the concept of symbol dominance into the parse tree.

The fuzzy relational Context-Free Grammar (r–CFG), can be differentiated from other approaches to grammar construction. This approach was introduced by MacLean and Labahn [89] and it has less computational complexity than CYK. Fuzzy r–CFG complexity is $O(N^{3+K}|P|K)$, where K is the number of right-hand side (RHS) tokens in the largest production. Another feature of this approach is more expressive grammar, which avoids creating hundreds of production rules. The proposed grammar is similar to the Backus–Naur Form (BNF), making it easier to support models for specific domains.

C. END-TO-END RECOGNITION

In recent years, end-to-end encoder-decoder based recognition systems are dominant in speech recognition, image

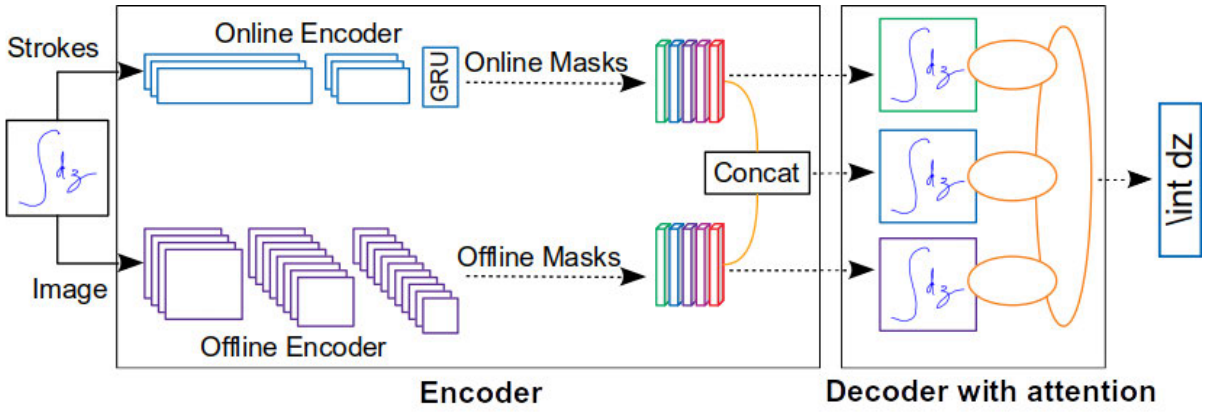


FIGURE 11. The architecture of stroke constrained attentional network (SCAN).

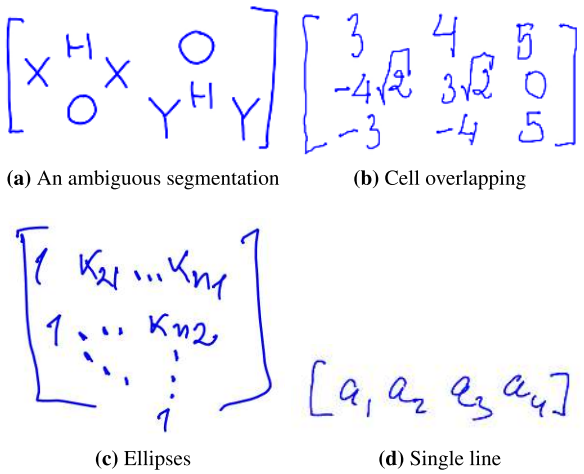
captioning, machine translation, these systems are also referred to as data-driven, which implies the absence of an explicitly defined domain knowledge. While this survey aims at online HME recognition, in this section we will take a look at end-to-end encoder-decoder based approaches, which are now mainly adopted for offline HME recognition, as they demonstrate impressive results, and some techniques can be applied for the recognition of online HME. Figure 10 illustrates the general representation of encoder-decoder framework with attention mechanism. The encoder transforms the input sequence into the hidden states, and then the decoder extracts the hidden representation through an attention mechanism and produces the result.

Zhang *et al.* conducted several experiments with several architectures. In [90], they presented Watch, Attend, and Parse (WAP) neural network model for offline ME recognition. The proposed system utilizes VGG architecture for the encoder implementation and gated recurrent unit (GRU) for the decoder. The attention model enforces the decoder to focus on a specific part of the input image to recognize a single character or spatial relationship between characters. Later, they improved the solution [91] by replacing the encoder architecture with densely connected convolutional networks (DenseNet) and applying multi-scale attention (MSA) model, which allowed the system to deal better with various scales of handwriting symbols. Their research is not limited to offline recognition. In [92], they proposed GRU-based encoder-decoder architecture for online recognition systems. It consists of a single layer GRU decoder with an implicit coverage-based attention mechanism and a multi-layer bi-directional GRU encoder. A little bit later, they improved their solution and proposed Track, Attend, and Parse (TAP) framework [93], where Guided Hybrid Attention (GHA) was integrated into the decoder. GHA consists of coverage-based spatial attention, temporal attention, and attention guider. The attention guider is used during the learning process as a regularization method for spatial attention mechanism, and temporal attention is responsible for the balancing between tracker and language model. Finally, TAP integrated with WAP and GRU-based language model (LM) in the beam search procedure, where GRU-based LM was trained on an additional text dataset.

He *et al.* [94] presented an end-to-end CNN-based framework for offline recognition of printed ME. Deng *et al.* [95] investigated different attention mechanisms for encoder-decoder architecture in offline recognition of printed ME. Le and Nakagawa [96] proposed an end-to-end solution with 3 layers: a CNN as a feature extractor, a BLSTM as an encoder, and an LSTM attention-based model as a decoder for LaTeX generation. Zhang *et al.* [34] proposed a solution based on one BSTLM architecture, which produces output as a 1D sequence describing a graph with spatial relations. All models were trained and evaluated on the open CROHME datasets (details about the CROHME datasets see in sec III-B). The final results for TAP+WAP+LM [93] approach were obtained with the ensemble of three instances of each model.

Wang *et al.* adapted the methods outlined above and focused on developing approaches based on encoder-decoder architectures that combine the advantages of both online and offline representations. He introduced multi-modal attentional network (MAN) [97] and stroke constrained attentional network (SCAN) [98]. MAN architecture uses CNN and stack of bidirectional GRUs to encode the online channel and DenseNet [99] to encode the offline channel. The decoder employs two unidirectional GRUs with a multi-modal attention mechanism. Compared to MAN approach, SCAN architecture (Figure 11) uses stroke masks to align between online and offline channels. Each online stroke mask contains information on whether the i point belongs to the j stroke or not, and an offline stroke mask specifies if (x, y) pixel belongs to j stroke. Applying such masks provides a fusion of different channels in the encoder.

Duc Le suggested the dual loss attention network [100] to improve HME recognition using printed MEs. In addition to the decoder loss, the method also comprises context matching loss to recognize semantic features from handwritten and printed MEs. In [101], the proposed methods address the



(a) An ambiguous segmentation **(b)** Cell overlapping

(c) Ellipses **(d)** Single line

FIGURE 12. **Examples of handwritten matrices.**

problem of varying symbol sizes. First, it was suggested to augment using different scaling factors, instead of normalizing to the same size. Second, the authors introduced a drop attention mechanism that restrains features with the greatest attention weight.

Table 4 shows the performance estimation (expression rate) and the number of weights in the proposed end-to-end solutions. The recognition accuracy was measured using open data sets, which will be discussed in more detail in Section III.

D. RECOGNITION OF MATRICES

HME recognition including mathematical matrices is a more complex problem, since it involves additional recognition steps associated with matrix detection and segmentation of their elements. Handwriting matrices often do not meet the requirements for the presence of space between elements, elements can overlap each other along the X-axis, Y-axis, and sometimes both axes simultaneously. Additional complexity arises from the shorthand matrix notation, which uses ellipsis ($\ldots, \vdots, \ddots$) to represent repetitive structures and sparse matrices in which zero-valued elements are omitted. Figure 12 illustrates examples of the basic problems associated with matrix recognition.

Despite the number of publications, related to the recognition of MEs, only a few of them pay attention to the recognition of matrices. Most of the approaches described in the publications are related to the recognition of the table representation in printed documents [70], [104]–[106]. Currently, all the existing approaches (online and offline) use special symbols (like [,], (,), |) to define the boundaries of the matrix to begin the analysis of the structure itself. The main differences are only in the method of analysis of the matrix structure (segmentation). The concept of attractor points and area projection function for rows clustering was proposed in [69]. Column clustering is based on the gaps between symbols. Tausky *et al.* [107] compared two methods for the matrices structure analysis. The first method is based on the comparison of the distance between adjacent symbols with some threshold value, which was calculated as the average width and height of symbols. The second method utilizes the expectation-maximization (EM) algorithm. Clustering was performed separately for rows and columns and used projections of the geometric centers of the symbols on the X and Y-axis. Li *et al.* [108] described an approach for matrix structure analysis based on sequential row identification and then column identification. To overcome the overlapping of cell elements that can be caused by superscripts or subscripts, they introduced '*rowBox*' which is calculated from the bounding boxes of the elements from the main baseline, rather than include all elements from the cell's expression tree. This approach requires making an assumption about the relationship between the elements inside the matrix cell.

A comparison of the accuracy of matrices recognition methods is given in Table 5. Matrix recognition accuracy evaluation uses additional metrics, which will be discussed in Section III. There are currently no publications on end-to-end solutions for matrix recognition. Also, all of the approaches described above can be classified as sequential solutions. It is not possible to make a comprehensive comparison of them because there were no published common test datasets at that time and no published evaluation.

E. MULTI-MODAL INPUT

Multi-modal interfaces are characterized by the use of several modalities to support input or output. Typing, handwriting, speech, and vision are the main modalities for human-computer interaction. Multi-modal systems are now in trend since they can reduce the ambiguity that may be characteristic of certain modalities. The voice modality can be used to simple text input, but the input of more complex elements (such as tables, charts, mathematics) using voice as a single modality is required speaking conventions to overcome the ambiguity and inconsistency [112]. During voice input user usually skips a lot of information related to the ME layout (for example, fences). The main difficulties in HME voice input are associated with the identification of spatial relations (especially for right, subscript, and superscript), and the classification of characters with a similar way of writing. Medjkoune *et al.* [113] suggested and explored various methods that reduce the uncertainty during MEs input that occurs in single-modality mode (Fig. 13). Their method is based on the simultaneous input of ME using two modalities, and keyword extraction from the recognized text by the voice modality. Further, these keywords are used to minimize handwriting ambiguity. Through the use of several modalities, they managed to achieve an improvement in the quality of recognition.

III. EVALUATION OF RECOGNITION SYSTEMS

Evaluation of handwritten MEs recognition systems is a non-trivial task. There is a set of issues that is common to all the types of recognition systems, such as creating or selecting representative datasets and metrics. Recognition of MEs has

TABLE 4. Comparison of end-to-end solutions. *ER* – Expression rate.

Solution	Number of weights in neural networks	ER on CROHME 2014	ER on CROHME 2016	ER on CROHME 2019
BLSTM [34]	≈ 300 000	30.57%	–	–
WAP [90]	≈ 1 400 000	46.55%	44.55%	–
GRU [92]	–	52.43%	–	–
MSA [91]	≈ 6 200 000	52.80%	50.10%	–
TAP [93]	≈ 5 700 000	55.37%	50.22%	–
TAP+WAP [93]	≈ 11 100 000	60.34%	55.27%	–
TAP+WAP+LM [93]	–	61.16%	57.02%	–
PAL [102]	–	47.06%	–	–
Watch Step by Step [103]	–	49.46%	–	–
MAN [97]	–	52.43%	49.87%	–
Enhanced MAN [97]	–	54.05%	50.86%	–
Online SCAN [98]	–	51.22%	46.12%	46.49%
Online SCAN + TAP [98]	–	52.64%	47.17%	47.62%
Offline SCAN [98]	–	47.67%	46.64%	47.62%
Offline SCAN + WAP [98]	–	49.39%	49.60%	49.62%
Online and Offline SCAN (decoder fusion) [98]	–	55.38%	52.22%	53.88%
Online and Offline SCAN (encoder fusion) [98]	–	57.20%	53.97%	56.21%
Dual loss attention network [100]	–	51.88%	51.53%	–
Scale augmentation and drop attention [101]	–	60.45%	58.06%	–

TABLE 5. Comparison of solutions with recognition of matrices. *ER* – Expression rate; *SymRR* – Symbol recall rate; *MRR* – Matrix recall rate; *RowRR* – Row recall rate; *ColRR* – Column recall rate; *CellRR* – Cell recall rate.

Solution	ER	SymRR	MRR	RowRR	ColRR	CellRR
	CROHME 2014					
Universitat Politécnica de Valéncia [109]	31.15	87.43	73.14	70.59	50.84	55.35
MyScript [109]	53.28	89.81	92.57	92.00	69.16	71.07
Yakovchuk et al. [110]	–	–	97.71	91.53	89.16	90.51
	CROHME 2016					
MyScript [111]	68.40	94.86	97.52	95.61	90.71	87.49
Wiris [111]	59.40	87.03	85.67	87.16	82.22	84.68
Yakovchuk et al. [110]	–	–	98.07	94.82	95.30	94.37

its difficulties and features in the evaluation process, which are associated with the 2D structure of ME, a wide range of character classes, the ambiguity of mathematical notations, and others. An example of ambiguity in LaTeX notation is the fact the same expression can be represented in multiple ways. Such a fraction can be represented using the \frac and \over commands ('a \over b' or '\frac{a}{b}'). Moreover, the proper selection of performance metrics can identify weaknesses of the verified system. Several works are devoted to an in-depth analysis of ME recognition system evaluation issues [114], [115].

A. METRICS

The choice of metrics highly depends on the goals and the method of recognition used. Expression rate (2) is generally used for the evaluation of HME recognition systems. This metric is defined as the percentage of recognized MEs matching ground-truth up to the symbols, relations, and structure:

$$ER = \frac{Number\ of\ correctly\ recognized\ expressions}{Total\ number\ of\ expressions} \quad (2)$$

ER metric is employed to evaluate the overall solution, but it does not provide information to identify weaknesses of the system. Another integrated metric is a structure recognition rate (3). It measures the percentage of expressions, where ME tree was recognized correctly, ignoring the label of characters in the tree nodes.

$$SRR = \frac{Number\ of\ correctly\ recognized\ structures}{Total\ number\ of\ expressions} \quad (3)$$

The most common metrics were proposed at the stage of sequential solutions dominance, and in addition to *ER* and

TABLE 6. Performance evaluation methods.

Matching structure	Primitive	Description
Graph	Object	EMERS [116] is a method for performance evaluation based on the edit distance cost between two trees. Objects are vertices and edges of the tree. Alvaro et al. [117] proposed the method for the tree comparison using an additional tree, which is equivalent to the ground-truth tree. The main goal of this method is to take into account the peculiarities of ME when the expressions have different structures but are semantically equivalent.
	Stroke	Zanibbi et al. [118] presented an approach for the graph-based evaluation, where vertices of the graph are input primitives (strokes) instead of symbols. This approach allows measuring all the metrics related to character segmentation, classification, and structural analysis using a single graph. In these works, 3 metrics (segmentation, layout, and classification) were proposed, which are presented as Hamming distance, and one integrated Expression Level Distance Metric, which combines 3 metrics. In the proposed method, minor segmentation errors, that did not affect the final recognition result, influence the integrated ER. Also, such an approach requires stroke level annotation of test samples, and the accuracy of such annotation affects the final estimation of ER.
Text	Symbol	Levenshtein edit distance is often used for accuracy measurement during working with one-dimensional elements such as text. Kumar et al. [119] described the method for the encoding 2D structure of ME into text. Simplicity and time complexity $O(N^2)$ are the main benefits of text-based approaches.
Image	Pixel	Image-based Mathematical Expression Global Error (IMEGE) evaluation method based on image comparison was proposed by Álvaro Muñoz et al. [120]. This method utilizes the Image Distortion Model to perform matching between images. This approach allows the detection of incorrectly recognized zones but does not allow the identification of bottlenecks during testing. The peculiarity of this method is that the final score implicitly takes into account the similarity of many characters.

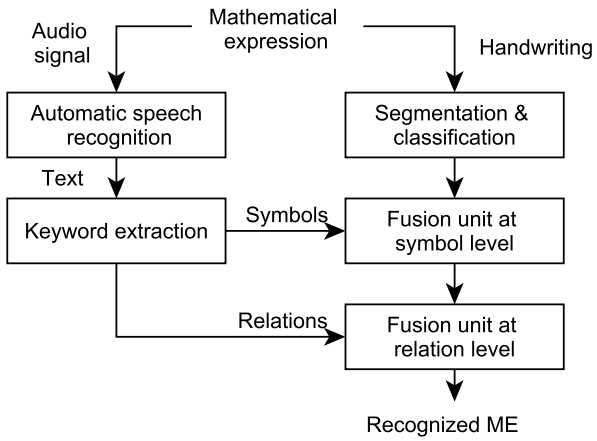


FIGURE 13. Multi-modal ME recognition workflow.

SRR, include segmentation accuracy, symbol classification accuracy, and the accuracy of structure analysis [121]. Each of the mentioned metrics is directly related to a certain recognition step. The development of recognition methods, the complexity of comparison, the ambiguity of notation led to the creation of a set of metrics and evaluation methods that can be classified as follows: *Graph-based* – matching between ground-truth and recognized result is made by matching graphs; *Text-based* – plain text representation is used for matching; *Image-based* – evaluation is based on a comparison of two generated pictures of ground-truth and test expression. Classification and brief description of evaluation metrics are presented in Table 6.

There is no way to use stroke-based methods to evaluate the accuracy of end-to-end solutions because these solutions directly transform input gestures to mathematical notation. The stroke-based evaluation has been used for evaluation from the start of CROHME. With the evolution and widespread of end-to-end systems, since 2019 the stroke-based evaluation has been replaced by object-based [122].

Layout recall rates are used to determine the quality of the table structure analysis related to matrices recognition, where there are four specific object types of interest. Such metrics include *Matrix recall*, *Row Recall*, *Column Recall*, and *Cell Recall* [109]. Typically, layout metrics utilize the result of stroke level segmentation according to the required elements of the matrix structure [110]. Metrics related to the quality of segmentation and character classification are also indicated. The quality of character recognition in matrices is usually lower than in regular MEs, since the matrix structure makes it difficult to detect character [111].

B. DATASETS

Collecting a large ground-truth dataset is expensive and requires a lot of effort. To create such a dataset, many aspects

need to be taken into account: such as the number of characters supported and their frequencies, the spatial relations used, the length of the expression, the domains (geometry, physics, etc.), the number of writers, the available equipment, workforce for proof-reading, and labeling. The most time-consuming steps are handwritten samples gathering itself, proof-reading, and labeling of ME at different levels (symbol segmentation and classification). These steps are often the cause of errors, as they are most often performed manually. End-to-end approaches do not require a detailed ME annotation, but they do require a huge number of training samples. This significantly reduces the time for sample annotation and the potential for human error but increases the cost of gathering and verification collected samples.

One of the most common approaches is to generate ground-truthed synthetic examples [123], [124]. However, such an approach is more suitable for OCR recognition of printed documents [125], [126] because handwriting input is very diverse and depends on the region, age, etc. The quality of the recognition systems is highly dependent on the variety of real handwriting in the train set. This insight has led to the creation of tools that automate, or at least simplify, data annotation tasks. ExpressMatch [127] is a system designed to simplify creation and management datasets with ground-truth annotation. The system provides semi-automatic annotation while writing based on the use of time between strokes to perform segmentation. The UI tool that supports automatic annotation, manual verification, and correction has been proposed in [128]. This system is also based on the time interval between strokes for symbol segmentation, which forces users to take breaks between writing characters. Hirata and Honda [129] proposed an approach for automatic ME labeling based on normalized graph matching. The search for the best fit is performed using the matching cost function, which includes cost ratios of vertices and edges. The cost function for edges is based on geometric features and requires a reference sample to calculate. A reference example for matching can be generated, using manual annotation or synthesized. Moreover, this technique also requires the correct character segmentation in the examples, which is achieved by requiring users to take a break between writing characters.

1) CROHME

In 2011, during the preparation of the first CROHME, Mouchere *et al.* presented a new dataset, which was a union of several open datasets such as MfrDB [130], Mathbrush [131], HAMEX [132], Expressmatch [127] and CIEL [38]. Since then, the datasets released as part of CROHME, have become the de facto standard benchmark for various studies and comparisons. The proposed evaluation methods are used as a standard, and the results, shown during the competition, become state-of-the-art. Within each next competition, a new test dataset was prepared, and the training dataset was expanded with the test dataset from the previous competition. Datasets are distributed as a set of Ink Markup Language (InkML) [133] files, where each file contains the following information: ground-truth in LaTeX and MathML formats; input tracepoints; the segmentation and assigned labels of each symbol in the expression.

Despite a gradual increase, this dataset is still quite small compared to training data for other domains (ImageNet [134] contains over 14 million images, AudioSet [135] contains more than 2 million sound clips). Especially it concerns end-to-end HME solutions that require significantly larger datasets than sequential solutions. Le *et al.* [136] proposed a set of techniques for extending existing datasets by synthesizing new samples. To extend the dataset, they offer local and global distortions. Where local distortions are applied for symbols and include shear, shrink, perspective, rotation, and their combinations. Global distortions apply to the entire ME and contain scaling and rotation. Based on the CROHME 2014 and 2016 datasets and the proposed techniques, they synthesized and published new datasets named Artificial Online Handwritten Mathematical Expressions.

Quiniou *et al.* [132] presented the public datasets named HAMEX for tasks, related to multimodal input of ME. It contains 4 350 MEs in online handwritten and audio spoken forms (in French) from 58 respondents.

C. COMPETITIONS

Nowadays, CROHME is the main driving force for the development of ME recognition systems both online and offline. Table 7 contains a brief competition summary, including the number of participants, the main metrics of datasets, the best results, and others. Each time the size of datasets increases, and the number of participants remains almost the same.

In 2019, a system based on the encoder-decoder end-to-end approach won for the first time with $ER = 80.73\%$ and $SRR = 91.49\%$. The winner system was trained on the CROHME training dataset only. Also, the winner combined online and offline recognition models, where each model is an attention-based encoder-decoder (RNN-based encoder for the online model and CNN-based encoder for offline model), and a single layer RNN-based language model. The gap between the winner and the closest competitors is less than 1.6%. Also, in 2019, the organizers abandoned the use of character segmentation and classification accuracy metrics, since the presented end-to-end solutions provide an answer in the form of a LaTeX string without binding information about input strokes to output elements.

In different years, the conditions vary slightly. So in 2014 and 2016, tasks with matrix recognition were included. This competition provides a comparison based on accuracy metrics. However, this is not enough to get a complete picture of the approaches. There is no information about other important indicators, such as recognition time, memory consumption, and model size, necessary to understand hardware requirements. This is especially important when you need to evaluate the possibility of using the solution on devices with limited resources, such as mobile devices or interactive panels.

TABLE 7. CROHME: datasets parameters and competition results: ***SR*** **– Segmentation Rate;** ***SCR*** **– Symbol classification rate;** ***SSCR*** **– Symbol segmentation and classification rate;** ***SpRR*** **– Spatial relation rate.**

		2011	2012	2013	2014	2016	2019
Number of symbol classes		56	75	101	101	101	101
Number of isolated symbols in test set		–	–	–	10 061	10 019	15 483
Number of isolated symbols in train set		–	–	–	85 781	85 802	180 440
Number of isolated symbols in validation set		–	–	–	–	10 061	18 435
Number of test MEs		348	488	671	986	1 147	1 199
Number of train MEs		921	1 336	8 836	8 836	8 836	9 993
Number of validation MEs		–	–	–	–	986	986
Number of matrices test MEs		–	–	–	175	250	–
Number of matrices train MEs		–	–	–	362	362	–
Number of matrices validation MEs		–	–	–	–	175	–
Number of writers (test set)		–	–	–	–	50	80
Isolated formulae: Best results	Number of participants	5	7	8	8	5	8
	ER(%)	22.41	62.50	60.36	62.68	67.65	80.73
	$\leq$ 1 error	–	78.89	80.33	72.31	75.59	88.99
	$\leq$ 2 errors	–	81.76	84.95	75.15	79.86	90.74
	$\leq$ 3 errors	–	81.97	86.14	76.88	–	–
	ER(%) (trained on CROHME)	19.83	–	23.40	37.22	49.61	80.73
	SRR(%)	–	80.33	–	–	88.14	91.49
	SR(%)	87.82	98.84	97.86	98.27	98.89	–
	SCR(%)	92.56	96.85	–	–	–	–
	SSCR(%)	–	–	93.03	93.47	95.47	–
	SpRR (%)	–	–	88.65	94.82	96.81	–
Matrix recognition: Best results	Number of participants	–	–	–	2	2	–
	ER(%)	–	–	–	53.28	68.40	–
	ER(%) (trained on CROHME)	–	–	–	31.15	56.40	–
Paper		[137]	[138]	[139]	[109]	[111]	[122]

IV. HUMAN-COMPUTER INTERACTION

Most researchers focus on recognition methods and their quality, as well as on the functional features of a particular system. Fewer works focus on aspects related to the integration of recognition systems in the application and user interactions. Little attention is given to the features of using systems on mobile devices with limited screen size and computational resources. The goal of this section is to pose UI/UX design issues with respect to the recognition approaches.

A. RECOGNITION MODULES AND USER EXPERIENCE

Typically, researchers refer to two UX design methods for integrating recognition systems into applications: iterative (real-time) and batch recognition. Iterative input consistently provides feedback and gives the user a chance to see what went wrong and what needs to be corrected [140]. When using the batch technique, recognition is performed after the entire ME has been written, and often the user has to manually start the recognition process. Bott *et al.* [141] studied the user's preferences for recognition mode depending on the accuracy of the recognition system and the number of MEs. This work demonstrated that the quality of recognition weakly affects the preferences of users for recognition mode, but during the input of multiple ME, users prefer the iterative mode. Most systems today provide interactivity. Even if recognition is performed on the whole ME, the intermediate results of recognition are displayed to the user as they are written. For modern systems, it is possible to distinguish three main features: edit mode, recognition mode, and representation mode. Table 8 provides an overview of the modes used in the considered solutions and applications.

1) EDIT MODES

The ability to correct errors more comfortably and quickly is a key requirement for UI. The easiest way to support the editing mode is to use the so-called *eraser* to remove one or more strokes and then re-draw again. The second method relies on different kinds of *menus* to select the correct character or the correct ME from the list of candidates. The 'undo' operation is also applied to this mode, as it is performed through the application menu (shortcut). The next option is to use handwritten editing *gestures* (Figure 14). This method is most natural for handwriting but requires a gesture recognition system and conflict resolution since a lot

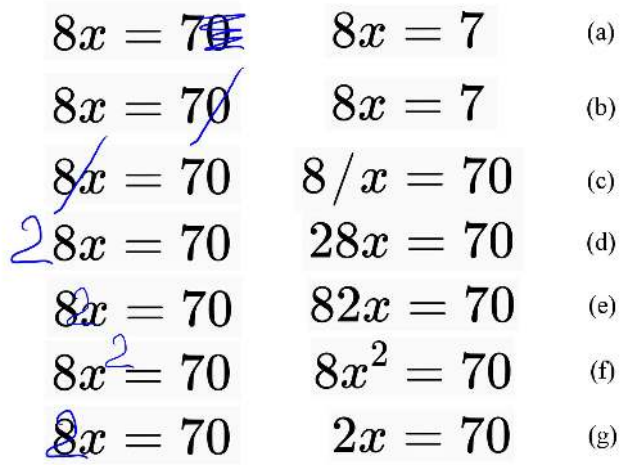


FIGURE 14. Examples of edit gestures and their interpretations: a) delete symbol(s) using *scribble* gesture; b) delete symbol(s) with a *slash* gesture; c) *slash* to insert '/' between characters; d) append a symbol; e) insert a symbol between two others; f) insert a symbol as a subexpression; g) replace a symbol.

of editing gestures are similar to mathematical symbols. The last method is *keyboarding* (hardware or software for mobile devices). All these approaches can complement each other and be used simultaneously in the same application. However, some recognition methods are incompatible with all of these editing methods. For example, existing end-to-end solutions do not support edit '*gestures*' mode. Also, the *menu* mode is more difficult to implement due to the complex structure of ME, so it is most often used to select a recognition candidate for a single character or an entire expression.

2) RECOGNITION MODES

Iterative mode assumes the recognition of characters sequentially as they are drawn. That is, only new strokes are transmitted for recognition, and ME is updated, taking into consideration newly recognized characters. Using this approach reduces latency. But it can lead to recognition quality degradation due to limited context while using RNN architecture for character recognition. In *batch* mode all strokes are used for the new ME recognition. The response time in this mode can greatly increase, but it is easier to implement and allows you to make some changes without resorting to editing mode (for example minus sign '−' can be easy modified to equality sign '=' or plus '+' sign). However, in this mode, the recognition result with the addition of one gesture can be significantly different from the previous recognition. BLSTM-based approaches are most susceptible to this effect. As a result, this behavior can distract the user. Batch mode is used more often than iterative and can be applied for any type of recognition solution (sequential, integrated, and end-to-end). The iterative mode cannot be implemented using current end-to-end solutions. For grammar-driven approaches, it is necessary to take into account the fact that the ME during input may have the wrong structure (input is not yet complete, and some elements can be missed). This leads to additional ambiguities in the analysis of the ME. For instance, the recognition algorithm can interpret an expression with an overline as an uncompleted fraction with an empty numerator.

TABLE 8. Iterative recognition features: *Rec.* – recognition mode; *Rep.* – representation mode; *G* – gesture edit mode; *G** – scribble gesture only for the stroke/symbol erasing; *M* – menu edit mode; *M** – undo/redo menu operations only; *K* – keyboard edit mode; *B* – batch recognition mode; *I* – iterative recognition mode; *H* – handwritten representation mode; *P* – printed representation mode.

Reference	Edit	Rec.	Rep.
Smithies et al. [37]	G M	B	H
Toyozumi et al. [142]	M	B	H
Chan and Yeung [143]	–	B	H
Phan et al. [61]	–	I	H
Bott and Laviola Jr [144]	G*	B	H
LaViola Jr and Zeleznik [145]	G	–	H
Zeleznik et al. [146]	G	I	H
MathBrush [147]	G M	B	H
Microsoft Office [148]	M	B	H
Microsoft Math Solver [149]	M*	B	H
MyScript Calculator [150]	G	I	P
Samsung S Note [151]	M	B	P
SMath [152]	M	I	P

3) REPRESENTATION MODES

Editing HME with a stylus or finger is called *handwritten* mode. In this mode, the recognition result can be displayed in a separate area of the screen in printed form 15. But the simultaneous presence of two representations of the same ME can distract the user (which of the two instances should be corrected?). The presence of the second instance also creates inconvenience on mobile devices due to the limited screen size. In *printed* mode, input strokes are transformed into a printed representation and, further editing operations are performed on top of it. Such mode removes the ambiguities related to the choice of instance for editing, reduces the number of elements on the screen. At present, there are no end-to-end solutions that support *printed* representation mode.

Gesture editing is not limited to the generally accepted *scribble* gesture. LaViola Jr and Zeleznik [145] presented the interface, which includes a set of gestures to create a graph, simplify an expression, solve an equation, and others. In [153] a system based on pen and touch modalities was proposed. Separate processing of signals from the pen and finger made it possible to simplify the interface and get rid of many ambiguities, in which the pen is used to input strokes, and finger gestures to control visual elements, movements, etc. Zanibbi *et al.* [154] studied the feedback method by transformation handwritten symbols in ME to corresponding their position in a printed version. Such a transformation is done by translation and scaling of each symbol and preserves the style of user drawn symbols. The purpose of this method is to provide immediate feedback about the recognition result but

TABLE 9. Applications: ***Avail.*** **– Availability; *P* – Prototype; *F* – Free; *C* – Commercial; *D* – Desktop; *B* – Board; *M* – Mobile; *MC* – Mobile(cloud); *MD* – Mobile(on-device).**

Application/Publication	Year	Avail.	Platform	Purpose and brief functionality description
PenCalc [143]	2001	P	D	Simple solving
MathPad^2 [145]	2004	P	D	Animated visualization, solving, simplifying, factoring
E-Chalk [156]	2005	–	B	Lecturer assistant with computational functionality and plotting
Anthony et al. [157]	2007	P	D	Tutoring systems in algebra learning
AlgoSketch [158]	2008	P	D	Pen-based interface for 2D algorithmic description language, which supports iterations, conditions, trace tool.
Newton's Pen [159]	2008	P	D	Diagrams and equilibrium equations
PenProof [160]	2010	P	D	Geometry proving system which correlate figures and ME
VectorPad [161]	2010	P	–	Operations on vectors and 2D/3D visualization
SetPad [162]	2012	P	D	Explore discrete math problems by sketching
LogicPad [163]	2012	P	D	Visualization and verification of Boolean Algebra
PhysicsBook [164]	2014	P	D	Pen-based tutoring system for physics domain that supports sketch recognition, understanding, and animation.
MathBrush [147]	2012	F	M	Solving, simplifying, factoring, plotting
Microsoft Ink to Math [148]	–	C	D	ME input, solving, 2D/3D plotting, simplifying, factoring
Samsung S Note [151]	2013	F	MD	ME input in note application
Nebo [165]	2016	C	MD	ME input and solving in note application
MyScript Calculator [150]	2018	F	MD	Simple ME solving
Microsoft Math Solver [149]	2019	F	MC	Solving, simplifying, factoring, plotting
SMath [152]	2020	P	MD	Simple solving

to avoid sudden changes in the appearance that occur during the transformation into a printed representation. Taranta and LaViola Jr [155] suggested a hint-based approach for simplified iterative input using visible bounding boxes around a specific part of ME.

B. APPLICATIONS

The first application prototype for ME input was introduced in 1993 [166]. This prototype supported iterative input of ME as well as edit, delete, and move gestures. MathPad^2 [145] prototype is the mode-less interface that focused on the combination of HME and free-form mathematical sketching. It allows animated visualization of ME and graph creation with pen gestures only. Besides, the system supports a wide range of functions, including solving, simplifying, factoring. Anthony *et al.* [157] proposed pencil-and-paper interfaces for application in tutoring systems. Li *et al.* [158] presented a system for sketching algorithms in pseudocode-like description style, which is based on the 2D traditional mathematical notation. Document processors can also be equipped with handwriting input recognition methods, where the input is carried out through a special mode with creating a temporary visual control (window). Applications for inputting and managing user notes also include ME recognition [151], [165]. To support diverse content, such applications combine multiple recognition modules for text, diagrams, tables, charts. However, the type of input element is mostly controlled by the user. In one case, a dedicated input area is created in the document; gestures inside this area are always

(a) Microsoft Math Solver **(b)** MyScript Calculator 2

FIGURE 15. **Examples of user interface: a) batch recognition and handwriting representation; b) iterative recognition and printed representation.**

recognized as formulae. In another case, the user should select the required strokes to perform transformation into the mathematical notation. The reason for this UI design is the difficulty of automatic classification of input strokes for text and formulae. Therefore, document structure analysis is often based on a binary classification of gestures into *Text* and *Non-Text* [167], [168]. The list of applications and brief description are presented in Table 9. Figure 15 demonstrates UI examples for mobile devices.

As one can see, in recent years, the number of mobile applications that focus on handwriting with a pen or finger has increased. These applications are optimized for use on small screens and under limited computational resources. Using cloud computing is one of the most common methods

to reduce processor load. But cloud-based approaches require constant access to the Internet and are associated with the risks of leakage of private information. In addition, users tend to keep their data private and often do not want to pass them to third parties [169]. The latest emerging trend in AI mobile applications is the rollout of on-device recognizers, which is privacy-preserving and always available, even without the Internet connection [170].

V. SUMMARY AND DISCUSSION

A. CONCLUSION

This survey demonstrates that interest in online HME recognition has been growing over the past 40 years. The development of integrated solutions made it possible to combine various methods (grammatical, statistical) minimizing the accumulation of recognition errors. The research community has now focused on DML techniques, resulting in third-generation end-to-end solutions. Although end-to-end solutions have taken the lead in many areas, in HME recognition, these methods are still at an early stage of development and have some limitations. One of the inherent features of these methods is their high computational complexity, which often prevents them from being used for on-device mobile computing. The second unresolved problem is the limited use-cases in UX design. Despite the fact that they have established the new state-of-the-art in the recognition performance, lagging behind integrated solutions is not yet significant.

With the technological transition from desktop solutions to mobile platforms, the number of pen-centric mobile applications is constantly increasing. These applications allow the user to perform a variety of tasks from simple calculations to the input of free-form diverse content. Although the deployment of next-generation networks (such as 4G, 5G) is nearly ubiquitous, developers often prefer to provide solutions with on-device recognition and calculation. In such a way, they can avoid security and privacy concerns associated with cloud computing.

The research in the field of user-centric and task-centric handwriting interfaces continues to bring it closer to natural input pen and paper interface. Using multi-modality (for example, voice) allows reducing the recognition errors during or simplify their correction.

B. FUTURE RESEARCH

In summary, this survey demonstrates that there continue to be significant advances in HME recognition approaches that use DNN. This progress together with the increase of user requirements opens up new possibilities for further research. In this section, we have summarized and briefly described some of them.

1) END-TO-END RECOGNITION

Approaches for mobile platforms are in great demand now. The current complexity of such solutions is burdensome for many devices and requires cloud computing. Furthermore, end-to-end solutions focus on the whole expression recognition, and the adaptation of these approaches to iterative input expands the scope of these solutions.

2) CLOUD SOLUTIONS

The latest competition showed that a combination of online and offline HME recognition methods provides a significant increase in recognition accuracy. Also, cloud-based solutions are expected to take full advantage of federated learning to create more robust models.

3) INTEGRATED SOLUTIONS

Such methods still show significant progress from year to year and have not yet passed their saturation point.

4) PERSONALIZATION ORIENTED APPROACHES

This branch of HME recognition is not yet developed. Such techniques as reinforcement learning and few-short learning can make the system more flexible and provide the ability to adapt to a specific user. This is especially important for sequence recognition systems with a large codebook. The development of approaches related to personalization will help reduce the ambiguity associated with the peculiarities of the handwriting of each user significantly.

5) CONTEXT-AWARE APPROACHES

The context-aware methods for HME recognition still have not received much attention from the research community. The focus shift from recognizing single HME to support multiple HME input reckoning with the document context will help to avoid many ambiguities associated with the similarity of many mathematical characters and a 2D structure complexity.

6) HME COMPLEXITY

Currently, commercial systems support the recognition of about 200 different types of characters. Verification of approaches is carried out on open datasets that contain only 101 types of characters, while the mathematical notation implies the use of more than 1500 kinds of symbols. Existing solutions provide a general recognition model that is independent of a specific field, although many scientific and engineering disciplines have adapted mathematical notation to suit their needs. The rules for constructing expressions, the used subset of symbols in many areas are significantly different. Thereby it is expected that the number of characters, supported expression types, and engineering domains will increase.

7) HME DETECTION

One of the main tasks for researchers is the detection of HME to ensure seamless input and recognition of different document elements (text, math, tables, sketches, and others) during HW input. Current research is mainly focused on localizing mathematical expressions in printed text that does

not involve iterative input. This situation leads to the creation of complex user interfaces and does not allow the full use of the concept of ''pen and paper''.

8) SKETCH-BASED APPLICATIONS

Educational applications that used both sketch-based input and HME recognition are the most widely represented since mathematics in high need there. For instance, deeper integration of HME recognition and sketch-based applications can help provide interactive visualizations of physical phenomena based on mathematical concepts. It should be noted that the current state of the technology allows it to be applied in an increasing number of industries, such as chemistry, finance, and others.

9) MULTI-MODALITY

Multi-modality is developing in many spheres such as Visual Question Answering. Moreover, users have different preferences on how to input information depending on the current activity [171]. So applications will allow seamless switching between keyboard/mouse input and handwriting by providing multiple modalities support. Voice control can also be actively involved in the UX design of HME recognition systems.

10) MIXED AND AUGMENTED REALITY

We see the potential in facilitating interaction with HME in augmented and mixed reality. In particular, this can lead to revealing a lot of opportunities for improving the productivity of education. Combining on-screen pen-or-touch input and HME recognition with gaze-and-touch-based editing, augmented by visualization of ME dependencies, or prediction/autocompletion results, can improve user experience and open up new challenges in HME recognition.

Based on this survey, it can be argued that recognition of HME is already a fairly mature technology that has been moved from prototyping to commercial mobile solutions. But despite significant progress, interactive recognition and editing of HME remain a challenging task that requires the joint efforts of researchers from different areas. We believe that the lessons we are learning in HME recognition are likely to be relevant to a wide range of other sequence recognition, computer vision, and natural language processing tasks.

REFERENCES

[1] B. Beeton, A. Freytag, and M. Sargent, ''Unicode support for mathematics,'' The Unicode Consortium, Mountain View, CA, USA, Tech. Rep. 25, 2017.

[2] L. Anthony, J. Yang, and K. R. Koedinger, ''Evaluation of multimodal input for entering mathematical equations on the computer,'' in *Proc. ACM CHI Extended Abstr. Hum. Factors Comput. Syst.*, 2005, pp. 1184–1187.

[3] D. Blostein and A. Grbavec, ''Recognition of mathematical notation,'' in *Handbook of Character Recognition and Document Image Analysis*. Singapore: World Scientific, 1997, pp. 557–582.

[4] K.-F. Chan and D.-Y. Yeung, ''Mathematical expression recognition: A survey,'' *Int. J. Document Anal. Recognit.*, vol. 3, no. 1, pp. 3–15, Aug. 2000.

[5] E. Tapia and R. Rojas, ''A survey on recognition of on-line handwritten mathematical notation,'' Free Univ. Berlin, Berlin, Germany, Tech. Rep. B-07-01, 2007.

[6] R. Zanibbi and D. Blostein, ''Recognition and retrieval of mathematical expressions,'' *Int. J. Document Anal. Recognit.*, vol. 15, no. 4, pp. 331–357, Dec. 2012.

[7] R. H. Anderson, ''Syntax-directed recognition of hand-printed two-dimensional mathematics,'' in *Proc. ACM Symp. Interact. Syst. Exp. Appl. Math.*, 1967, pp. 436–459.

[8] S.-K. Chang, ''A method for the structural analysis of two-dimensional mathematical expressions,'' *Inf. Sci.*, vol. 2, no. 3, pp. 253–272, Jul. 1970.

[9] T. Zhang, ''New architectures for handwritten mathematical expressions recognition,'' Ph.D. dissertation, Lab. Sci. Numérique Nantes (LS2N), Univ. de Nantes, Nantes, France, 2017.

[10] M. Koschinski, H.-J. Winkler, and M. Lang, ''Segmentation and recognition of symbols within handwritten mathematical expressions,'' in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 1995, pp. 2439–2442.

[11] B. Keshari and S. Watt, ''Hybrid mathematical symbol recognition using support vector machines,'' in *Proc. 9th Int. Conf. Document Anal. Recognit. (ICDAR)*, vol. 2, Sep. 2007, pp. 859–863.

[12] B. Q. Huang, Y. B. Zhang, and M. T. Kechadi, ''Preprocessing techniques for online handwriting recognition,'' in *Proc. 7th Int. Conf. Intell. Syst. Design Appl. (ISDA)*, Oct. 2007, pp. 793–800.

[13] A. D. Le, H. Dai Nguyen, and M. Nakagawa, ''Modified XY cut for re-ordering strokes of online handwritten mathematical expressions,'' in *Proc. IEEE Int. Workshop Document Anal. Syst.*, Apr. 2016, pp. 233–238.

[14] M. Okamoto, ''Recognition of mathematical expressions by using the layout structure of symbols,'' in *Proc. IEEE Int. Conf. Document Anal. Recognit.*, 1991, pp. 242–250.

[15] M. Okamoto and A. Miyazawa, ''An experimental implementation of a document recognition system for papers containing mathematical expressions,'' in *Proc. Comf. Struct. Document Image Anal.*, 1992, pp. 36–53.

[16] J. Ha, R. M. Haralick, and I. T. Phillips, ''Understanding mathematical expressions from document images,'' in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, 1995, pp. 956–959.

[17] H.-J. Winkler, H. Fahrner, and M. Lang, ''A soft-decision approach for structural analysis of handwritten mathematical expressions,'' in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, vol. 4, 1995, pp. 2459–2462.

[18] S. Lehmberg, H.-J. Winkler, and M. Lang, ''A soft-decision approach for symbol segmentation within handwritten mathematical expressions,'' in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. Conf. Proc.*, vol. 6, May 1996, pp. 3434–3437.

[19] A. Kosmala and G. Rigoll, ''On-line handwritten formula recognition using statistical methods,'' in *Proc. 14th Int. Conf. Pattern Recognit.*, vol. 2, 1998, pp. 1306–1308.

[20] N. E. Matsakis, ''Recognition of handwritten mathematical expressions,'' M.S. thesis, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 1999.

[21] K. Toyozumi, N. Yamada, T. Kitasaka, K. Mori, Y. Suenaga, K. Mase, and T. Takahashi, ''A study of symbol segmentation method for handwritten mathematical formula recognition using mathematical structure information,'' in *Proc. 17th Int. Conf. Pattern Recognit.*, 2004, pp. 630–633.

[22] X. Tian and Y. Zhang, ''Segmentation of touching characters in mathematical expressions using contour feature technique,'' in *Proc. 8th ACIS Int. Conf. Softw. Eng., Artif. Intell., Netw., Parallel/Distrib. Comput. (SNPD)*, Jul. 2007, pp. 206–209.

[23] L. Hu and R. Zanibbi, ''Segmenting handwritten math symbols using AdaBoost and multi-scale shape context features,'' in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 1180–1184.

[24] A. D. Le and M. Nakagawa, ''A system for recognizing online handwritten mathematical expressions by using improved structural analysis,'' *Int. J. Document Anal. Recognit.*, vol. 19, no. 4, pp. 305–319, Dec. 2016.

[25] L. Hu and R. Zanibbi, ''Line-of-sight stroke graphs and parzen shape context features for handwritten math formula representation and symbol segmentation,'' in *Proc. 15th Int. Conf. Frontiers Handwriting Recognit. (ICFHR)*, Oct. 2016, pp. 180–186.

[26] A. Nomura, K. Michishita, S. Uchida, and M. Suzuki, ''Detection and segmentation of touching characters in mathematical expressions,'' in *Proc. IEEE Int. Conf. Document Anal. Recognit.*, 2003, pp. 126–130.

[27] U. Garain and B. Chaudhuri, ''Segmentation of touching symbols for ocr of printed mathematical expressions: An approach based on multifactorial analysis,'' in *Proc. IEEE Int. Conf. Document Anal. Recognit.*, Dec. 2005, pp. 177–181.

[28] M. Suzuki, F. Tamari, R. Fukuda, S. Uchida, and T. Kanahori, "INFTY: An integrated OCR system for mathematical documents," in *Proc. ACM Symp. Document Eng.*, 2003, pp. 95–104.
[29] C. Chan, "Stroke extraction for offline handwritten mathematical expression recognition," *IEEE Access*, vol. 8, pp. 61565–61575, 2020.
[30] H.-J. Winkler, "HMM-based handwritten symbol recognition using on-line and off-line features," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. Conf. Proc.*, Dec. 1996, pp. 3438–3441.
[31] H.-J. Winkler and M. Lang, "Online symbol segmentation and recognition in handwritten mathematical expressions," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 1997, pp. 3377–3380.
[32] 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," in *Proc. Int. Conf. Document Anal. Recognit.*, Sep. 2011, pp. 457–462.
[33] H. Dai Nguyen, A. Duc Le, and M. Nakagawa, "Recognition of online handwritten math symbols using deep neural networks," *IEICE Trans. Inf. Syst.*, vol. E99.D, no. 12, pp. 3110–3118, 2016.
[34] T. Zhang, H. Mouchère, and C. Viard-Gaudin, "Using BLSTM for interpretation of 2-D languages," *Document Numérique*, vol. 19, no. 2, pp. 135–157, 2016.
[35] M. Liwicki and H. Bunke, "Feature selection for hmm and BLSTM based handwriting recognition of whiteboard notes," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 5, pp. 907–923, Aug. 2009.
[36] D. Zhelezniakov, V. Zaytsev, and O. Radyvonenko, "Acceleration of online recognition of 2D sequences using deep bidirectional LSTM and dynamic programming," in *Proc. Int. Work-Conf. Artif. Neural Netw.*, 2019, pp. 438–449.
[37] S. Smithies, K. Novins, and J. Arvo, "A handwriting-based equation editor," *Graph. Interface*, vol. 99, pp. 84–91, Jun. 1999.
[38] A.-M. Awal, H. Mouchère, and C. Viard-Gaudin, "Towards handwritten mathematical expression recognition," in *Proc. 10th Int. Conf. Document Anal. Recognit.*, 2009, pp. 1046–1050.
[39] Y. A. Dimitriadis and J. López Coronado, "Towards an art based mathematical editor, that uses on-line handwritten symbol recognition," *Pattern Recognit.*, vol. 28, no. 6, pp. 807–822, Jun. 1995.
[40] A.-M. Awal, H. Mouchère, and C. Viard-Gaudin, "A global learning approach for an online handwritten mathematical expression recognition system," *Pattern Recognit. Lett.*, vol. 35, pp. 68–77, Jan. 2014.
[41] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. Int. Conf. Mach. Learn.*, 2001, pp. 282–289.
[42] A.-M. Awal, H. Mouchère, and C. Viard-Gaudin, "A hybrid classifier for handwritten mathematical expression recognition," *Proc. SPIE*, vol. 7534, Jan. 2010, Art. no. 753410.
[43] F. Simistira, V. Katsouros, and G. Carayannis, "A template matching distance for recognition of on-line mathematical symbols," in *Proc. IEEE Int. Conf. Frontiers Handwriting Recognit.*, Dec. 2008, pp. 415–420.
[44] A. Thammano and S. Rugkunchon, "A neural network model for online handwritten mathematical symbol recognition," in *Proc. Int. Conf. Intell. Comput.*, 2006, pp. 292–298.
[45] F. Alvaro, J.-A. Sanchez, and J.-M. Benedi, "Classification of on-line mathematical symbols with hybrid features and recurrent neural networks," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 1012–1016.
[46] H. Dai Nguyen, A. D. Le, and M. Nakagawa, "Deep neural networks for recognizing online handwritten mathematical symbols," in *Proc. 3rd IAPR Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2015, pp. 121–125.
[47] F. A. Muñoz, "Mathematical expression recognition based on probabilistic grammars," M.S. thesis, Dept. Sistemas Informáticos y Computación, Univ. Politécnica de Valéncia, Valencia, Spain, 2015.
[48] F. JulcaAguilar, N. S. T. Hirata, C. ViardGaudin, H. Mouchere, and S. Medjkoune, "Mathematical symbol hypothesis recognition with rejection option," in *Proc. 14th Int. Conf. Frontiers Handwriting Recognit.*, Sep. 2014, pp. 500–505.
[49] D. Fang and C. Zhang, "Multi-feature learning by joint training for handwritten formula symbol recognition," *IEEE Access*, vol. 8, pp. 48101–48109, 2020.
[50] M. Liwicki, A. Graves, S. Fernàndez, H. Bunke, and J. Schmidhuber, "A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks," in *Proc. Int. Conf. Document Anal. Recognit.*, 2007, pp. 367–371.
[51] A. Graves, "Connectionist temporal classification," in *Proc. Supervised Sequence Labelling Recurrent Neural Netw.*, 2012, pp. 61–93.
[52] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4774–4778.
[53] H. T. Nguyen, C. T. Nguyen, and M. Nakagawa, "ICFHR 2018–competition on Vietnamese online handwritten text recognition using HANDS-VNOnDB (VOHTR2018)," in *Proc. 16th Int. Conf. Frontiers Handwriting Recognit. (ICFHR)*, Aug. 2018, pp. 494–499.
[54] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 369–376.
[55] V. Volkova, I. Deriuga, V. Osadchyi, and O. Radyvonenko, "Improvement of character segmentation using recurrent neural networks and dynamic programming," in *Proc. IEEE 2nd Int. Conf. Data Stream Mining Process. (DSMP)*, Aug. 2018, pp. 218–222.
[56] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[57] S. Balaban, "Deep learning and face recognition: The state of the art," *Proc. SPIE Biometric Surveill. Technol. Hum. Activity Identificat.*, vol. 9457, May 2015, Art. no. 94570B.
[58] K. Xu, J. Ba, R. Kiros, K. Cho, and A. Courville, "Show, attend and tell: Neural image caption generation with visual attention," *Comput. Sci.*, vol. 2015, pp. 2048–2057, Feb. 2015.
[59] H. Nguyen, A. Le, and M. Nakagawa, "Combination of LSTM and CNN on recognizing mathematical symbols," in *Proc. Inf.-Based Induction Sci. Mach. Learn.*, 2014, pp. 287–292.
[60] Z.-L. Bai and Q. Huo, "A study on the use of 8-directional features for online handwritten chinese character recognition," in *Proc. 8th Int. Conf. Document Anal. Recognit. (ICDAR)*, 2005, pp. 262–266.
[61] K. M. Phan, A. D. Le, B. Indurkhya, and M. Nakagawa, "Augmented incremental recognition of online handwritten mathematical expressions," *Int. J. Document Anal. Recognit.*, vol. 21, no. 4, pp. 253–268, Dec. 2018.
[62] L. Zhang, D. Blostein, and R. Zanibbi, "Using fuzzy logic to analyze superscript and subscript relations in handwritten mathematical expressions," in *Proc. 8th Int. Conf. Document Anal. Recognit. (ICDAR)*, 2005, pp. 972–976.
[63] F. Simistira, V. Papavassiliou, V. Katsouros, and G. Carayannis, "Recognition of spatial relations in mathematical formulas," in *Proc. 14th Int. Conf. Frontiers Handwriting Recognit.*, Sep. 2014, pp. 164–168.
[64] F. Álvaro, J.-A. Sánchez, and J.-M. Benedí, "Recognition of on-line handwritten mathematical expressions using 2D stochastic context-free grammars and hidden Markov models," *Pattern Recognit. Lett.*, vol. 35, pp. 58–67, Jan. 2014.
[65] W. Aly, S. Uchida, A. Fujiyoshi, and M. Suzuki, "Statistical classification of spatial relationships among mathematical symbols," in *Proc. 10th Int. Conf. Document Anal. Recognit.*, 2009, pp. 1350–1354.
[66] A. Lods, E. Anquetil, and S. Mace, "Fuzzy visibility graph for structural analysis of online handwritten mathematical expressions," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 641–646.
[67] L. Ouyang and R. Zanibbi, "Identifying layout classes for mathematical symbols using layout context," in *Proc. IEEE Western New York Image Process. Workshop*, Jun. 2009, pp. 1–4.
[68] L. Hu and R. Zanibbi, "MST-based visual parsing of online handwritten mathematical expressions," in *Proc. 15th Int. Conf. Frontiers Handwriting Recognit. (ICFHR)*, Oct. 2016, pp. 337–342.
[69] E. Tapia and R. Rojas, "Recognition of on-line handwritten mathematical expressions using a minimum spanning tree construction and symbol dominance," in *Proc. Int. Workshop Graph. Recognit.*, 2003, pp. 329–340.
[70] H.-J. Lee and J.-S. Wang, "Design of a mathematical expression understanding system," *Pattern Recognit. Lett.*, vol. 18, no. 3, pp. 289–298, Mar. 1997.
[71] Y. Eto and M. Suzuki, "Mathematical formula recognition using virtual link network," in *Proc. 6th Int. Conf. Document Anal. Recognit.*, 2001, pp. 762–767.
[72] S. Toyota, S. Uchida, and M. Suzuki, "Structural analysis of mathematical formulae with verification based on formula description grammar," in *Proc. Int. Workshop Document Anal. Syst.*, 2006, pp. 153–163.

[73] R. Zanibbi, D. Blostein, and J. R. Cordy, "Recognizing mathematical expressions using tree transformation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 11, pp. 1455–1467, Nov. 2002.

[74] T. H. Rhee and J. H. Kim, "Efficient search strategy in structural analysis for handwritten mathematical expression recognition," *Pattern Recognit.*, vol. 42, no. 12, pp. 3192–3201, Dec. 2009.

[75] T. Zhang, H. Mouchere, and C. Viard-Gaudin, "Tree-based BLSTM for mathematical expression recognition," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 914–919.

[76] A. Belaid and J.-P. Haton, "A syntactic approach for handwritten mathematical formula recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 1, pp. 105–111, Jan. 1984.

[77] M. Tomita, "Parsing 2-dimensional language," in *Proc. Current Issues Parsing Technol.*, 1991, pp. 277–289.

[78] K. Marriott, B. Meyer, and K. B. Wittenburg, "A survey of visual language specification and recognition," in *Proc. Conf. Vis. Lang. Theory*, 1998, pp. 5–85.

[79] S. Lavirotte and L. Pottier, "Mathematical formula recognition using graph grammar," *Proc. SPIE Document Recognit.*, vol. 3305, pp. 44–52, Apr. 1998.

[80] J. F. Hull, "Recognition of mathematics using a two-dimensional trainable context-free grammar," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 1996.

[81] S. C. Reghizzi and M. Pradella, "A CKY parser for picture grammars," *Inf. Process. Lett.*, vol. 105, no. 6, pp. 213–217, Mar. 2008.

[82] P. A. Chou, "Recognition of equations using a two-dimensional stochastic context-free grammar," *Proc. SPIE Vis. Commun. Image Process.*, vol. 1199, pp. 852–865, Nov. 1989.

[83] R. Yamamoto, S. Sako, T. Nishimoto, and S. Sagayama, "On-line recognition of handwritten mathematical expressions based on stroke-based stochastic context-free grammar," in *Proc. IEEE Int. Workshop Frontiers Handwriting Recognit.*, Oct. 2006, pp. 852–865.

[84] F. Alvaro, J.-A. Sánchez, and J.-M. Benedi, "Recognition of printed mathematical expressions using two-dimensional stochastic context-free grammars," in *Proc. Int. Conf. Document Anal. Recognit.*, Sep. 2011, pp. 1225–1229.

[85] F. Álvaro, J.-A. Sánchez, and J.-M. Benedí, "An integrated grammar-based approach for mathematical expression recognition," *Pattern Recognit.*, vol. 51, pp. 135–147, Mar. 2016.

[86] P. Liang, M. Narasimhan, M. Shilman, and P. Viola, "Efficient geometric algorithms for parsing in two dimensions," in *Proc. 8th Int. Conf. Document Anal. Recognit. (ICDAR)*, 2005, pp. 1172–1177.

[87] A. D. Le and M. Nakagawa, "Speedup of parsing for recognition of online handwritten mathematical expressions," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 896–901.

[88] M. Celik and B. Yanikoglu, "Probabilistic mathematical formula recognition using a 2D context-free graph grammar," in *Proc. Int. Conf. Document Anal. Recognit.*, Sep. 2011, pp. 161–166.

[89] S. MacLean and G. Labahn, "A new approach for recognizing handwritten mathematics using relational grammars and fuzzy sets," *Int. J. Document Anal. Recognit.*, vol. 16, no. 2, pp. 139–163, 2013.

[90] J. Zhang, J. Du, S. Zhang, D. Liu, Y. Hu, J. Hu, S. Wei, and L. Dai, "Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition," *Pattern Recognit.*, vol. 71, pp. 196–206, Nov. 2017.

[91] J. Zhang, J. Du, and L. Dai, "Multi-scale attention with dense encoder for handwritten mathematical expression recognition," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 2245–2250.

[92] J. Zhang, J. Du, and L. Dai, "A GRU-based encoder-decoder approach with attention for online handwritten mathematical expression recognition," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 902–907.

[93] J. Zhang, J. Du, and L. Dai, "Track, attend, and parse (TAP): An end-to-end framework for online handwritten mathematical expression recognition," *IEEE Trans. Multimedia*, vol. 21, no. 1, pp. 221–233, Jan. 2019.

[94] W. He, Y. Luo, F. Yin, H. Hu, J. Han, E. Ding, and C.-L. Liu, "Context-aware mathematical expression recognition: An end-to-end framework and a benchmark," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 3246–3251.

[95] Y. Deng, A. Kanervisto, J. Ling, and A. M. Rush, "Image-to-markup generation with coarse-to-fine attention," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 980–989.

[96] A. D. Le and M. Nakagawa, "Training an end-to-end system for handwritten mathematical expression recognition by generated patterns," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 1056–1061.

[97] J. Wang, J. Du, J. Zhang, and Z.-R. Wang, "Multi-modal attention network for handwritten mathematical expression recognition," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 1181–1186.

[98] J. Wang, J. Du, and J. Zhang, "Stroke constrained attention network for online handwritten mathematical expression recognition," 2020, *arXiv:2002.08670*. [Online]. Available: http://arxiv.org/abs/2002.08670

[99] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Dec. 2017, pp. 4700–4708.

[100] A. D. Le, "Recognizing handwritten mathematical expressions via paired dual loss attention network and printed mathematical expressions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 566–567.

[101] Z. Li, L. Jin, S. Lai, and Y. Zhu, "Improving attention-based handwritten mathematical expression recognition with scale augmentation and drop attention," 2020, *arXiv:2007.10092*. [Online]. Available: http://arxiv.org/abs/2007.10092

[102] J.-W. Wu, F. Yin, Y.-M. Zhang, X.-Y. Zhang, and C.-L. Liu, "Image-to-markup generation via paired adversarial learning," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2018, pp. 18–34.

[103] H. Wang and G. Shan, "Recognizing handwritten mathematical expressions as LaTex sequences using a multiscale robust neural network," 2020, *arXiv:2003.00817*. [Online]. Available: http://arxiv.org/abs/2003.00817

[104] K. Toshihiro and S. Masakazu, "A recognition method of matrices by using variable block pattern elements generating rectangular area," in *Proc. Int. Workshop Graph. Recognit.*, 2001, pp. 320–329.

[105] T. Kanahori and M. Suzuki, "Detection of matrices and segmentation of matrix elements in scanned images of scientific documents," in *Proc. IEEE Int. Conf. Document Anal. Recognit.*, 2003, pp. 433–437.

[106] C. Malon, S. Uchida, and M. Suzuki, "Mathematical symbol recognition with support vector machines," *Pattern Recognit. Lett.*, vol. 29, no. 9, pp. 1326–1332, Jul. 2008.

[107] D. Tausky, G. Labahn, E. Lank, and M. Marzouk, "Managing ambiguity in mathematical matrices," in *Proc. ACM Eurographics Workshop Sketch-Based Interface Modeling*, 2007, pp. 115–122.

[108] C. Li, R. Zeleznik, T. Miller, and J. J. LaViola, "Online recognition of handwritten mathematical expressions with support for matrices," in *Proc. 19th Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–4.

[109] H. Mouchere, C. Viard-Gaudin, R. Zanibbi, and U. Garain, "ICFHR 2014 competition on recognition of on-line handwritten mathematical expressions (CROHME 2014)," in *Proc. 14th Int. Conf. Frontiers Handwriting Recognit.*, Sep. 2014, pp. 791–796.

[110] O. Yakovchuk, A. Cherneha, D. Zhelezniakov, and V. Zaytsev, "Methods for lines and matrices segmentation in RNN-based online handwriting mathematical expression recognition systems," in *Proc. IEEE 3rd Int. Conf. Data Stream Mining Process. (DSMP)*, Aug. 2020, pp. 255–261.

[111] H. Mouchere, C. Viard-Gaudin, R. Zanibbi, and U. Garain, "ICFHR2016 CROHME: Competition on recognition of online handwritten mathematical expressions," in *Proc. 15th Int. Conf. Frontiers Handwriting Recognit. (ICFHR)*, Oct. 2016, pp. 607–612.

[112] R. Fateman, "How can we speak math," *J. Symbolic Comput.*, vol. 25, no. 2, pp. 1–5, 1998.

[113] S. Medjkoune, H. Mouchère, S. Petitrenaud, and C. Viard-Gaudin, "Multimodal mathematical expressions recognition: Case of speech and handwriting," in *Proc. Int. Conf. Hum.-Comput. Interact.*, 2013, pp. 77–86.

[114] A.-M. Awal, H. Mouchere, and C. Viard-Gaudin, "The problem of handwritten mathematical expression recognition evaluation," in *Proc. 12th Int. Conf. Frontiers Handwriting Recognit.*, Nov. 2010, pp. 646–651.

[115] A. Lapointe and D. Blostein, "Issues in performance evaluation: A case study of math recognition," in *Proc. 10th Int. Conf. Document Anal. Recognit.*, 2009, pp. 1355–1359.

[116] K. Sain, A. Dasgupta, and U. Garain, "EMERS: A tree matching–based performance evaluation of mathematical expression recognition systems," *Int. J. Document Anal. Recognit.*, vol. 14, no. 1, pp. 75–85, 2011.

[117] F. Alvaro, J.-A. Sanchez, and J.-M. Benedi, "Unbiased evaluation of handwritten mathematical expression recognition," in *Proc. Int. Conf. Frontiers Handwriting Recognit.*, Sep. 2012, pp. 181–186.

[118] R. Zanibbi, A. Pillay, H. Mouchere, C. Viard-Gaudin, and D. Blostein, "Stroke-based performance metrics for handwritten mathematical expressions," in *Proc. Int. Conf. Document Anal. Recognit.*, Sep. 2011, pp. 334–338.
[119] P. P. Kumar, A. Agarwal, and C. Bhagvati, "A string matching based algorithm for performance evaluation of mathematical expression recognition," *Sadhana*, vol. 39, no. 1, pp. 63–79, Feb. 2014.
[120] F. Á. Mu noz, J. A. S. Peiró, and J. M. I. B. Ruiz, "IMEGE: Image-based mathematical expression global error," Univ. Politécnica de Valéncia, Valencia, Spain, Tech. Rep. DSIC-PRHLT, 2011.
[121] K.-F. Chan and D.-Y. Yeung, "Error detection, error correction and performance evaluation in on-line mathematical expression recognition," *Pattern Recognit.*, vol. 34, no. 8, pp. 1671–1684, Aug. 2001.
[122] M. Mahdavi, R. Zanibbi, H. Mouchere, C. Viard-Gaudin, and U. Garain, "ICDAR 2019 CROHME+TFD: Competition on recognition of handwritten mathematical expressions and typeset formula detection," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 65–74.
[123] S. MacLean, G. Labahn, E. Lank, M. Marzouk, and D. Tausky, "Grammar-based techniques for creating ground-truthed sketch corpora," *Int. J. Document Anal. Recognit.*, vol. 14, no. 1, pp. 65–74, 2011.
[124] A. Kumar, A. Balasubramanian, A. Namboodiri, and C. Jawahar, "Model-based annotation of online handwritten datasets," in *Proc. IEEE Int. Workshop Frontiers Handwriting Recognit.*, 2006.
[125] P. Heroux, E. Barbu, S. Adam, and E. Trupin, "Automatic ground-truth generation for document image analysis and understanding," in *Proc. 9th Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2007, pp. 476–480.
[126] O. Okun and M. Pietikainen, "Automatic ground-truth generation for skew-tolerance evaluation of document layout analysis methods," in *Proc. 15th Int. Conf. Pattern Recognit. (ICPR)*, 2000, pp. 376–379.
[127] F. D. Aguilar and N. S. Hirata, "Expressmatch: A system for creating ground-truthed datasets of online mathematical expressions," in *Proc. Int. Workshop Document Anal. Syst.*, 2012, pp. 155–159.
[128] A. Le Duc and M. Nakagawa, "A ground-truthing tool for making a database of online handwritten mathematical expressions," *PRMU2012-205*, vol. 112, pp. 147–150, Dec. 2013.
[129] N. S. Hirata and W. Y. Honda, "Automatic labeling of handwritten mathematical symbols via expression matching," in *Proc. Int. Workshop Graph-Based Represent. Pattern Recognit.*, 2011, pp. 295–304.
[130] J. Stria, M. Bresler, D. Prusa, and V. Hlavac, "MfrDB: Database of annotated on-line mathematical formulae," in *Proc. Int. Conf. Frontiers Handwriting Recognit.*, Sep. 2012, pp. 542–547.
[131] G. Labahn, E. Lank, S. MacLean, M. Marzouk, and D. Tausky, "MathBrush: A system for doing math on pen-based devices," in *Proc. 8th IAPR Int. Workshop Document Anal. Syst.*, Sep. 2008, pp. 599–606.
[132] S. Quiniou, H. Mouchere, S. P. Saldarriaga, C. Viard-Gaudin, E. Morin, S. Petitrenaud, and S. Medjkoune, "HAMEX—A handwritten and audio dataset of mathematical expressions," in *Proc. Int. Conf. Document Anal. Recognit.*, Sep. 2011, pp. 452–456.
[133] S. M. Watt, *Ink markup language (InkML)*, document W3C 10, 2011.
[134] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[135] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 776–780.
[136] A. D. Le, B. Indurkhya, and M. Nakagawa, "Pattern generation strategies for improving recognition of handwritten mathematical expressions," *Pattern Recognit. Lett.*, vol. 128, pp. 255–262, Dec. 2019.
[137] H. Mouchere, C. Viard-Gaudin, D. H. Kim, J. H. Kim, and U. Garain, "CROHME 2011: Competition on recognition of online handwritten mathematical expressions," in *Proc. IEEE Int. Conf. Document Anal. Recognit.*, 2011, pp. 1497–1500.
[138] H. Mouchere, C. Viard-Gaudin, D. Kim, J. Kim, and U. Garain, "ICFHR 2012 competition on recognition of on-line mathematical expressions (CROHME 2012)," in *Proc. IEEE Int. Workshop Frontiers Handwriting Recognit.*, 2012, pp. 811–816.
[139] H. Mouchere, C. Viard-Gaudin, R. Zanibbi, U. Garain, D. H. Kim, and J. H. Kim, "ICDAR 2013 CROHME: Third international competition on recognition of online handwritten mathematical expressions," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 1428–1432.
[140] D. Zhelezniakov, V. Zaytsev, O. Radyvonenko, and Y. Yakishyn, "InteractivePaper: Minimalism in document editing UI through the handwriting prism," in *Proc. 32nd Annu. ACM Symp. Interface Softw. Technol.*, Oct. 2019, pp. 13–15.
[141] J. N. Bott, D. Gabriele, and J. J. LaViola, "Now or later: An initial exploration into user perception of mathematical expression recognition feedback," in *Proc. 8th Eurographics Symp. Sketch-Based Interface Modeling (SBIM)*, 2011, pp. 125–132.
[142] K. Toyozumi, T. Suzuki, K. Mori, and Y. Suenaga, "A system for real-time recognition of handwritten mathematical formulas," in *Proc. 6th Int. Conf. Document Anal. Recognit.*, 2001, pp. 1059–1063.
[143] K.-F. Chan and D.-Y. Yeung, "PenCalc: A novel application of on-line mathematical expression recognition technology," in *Proc. 6th Int. Conf. Document Anal. Recognit.*, 2001, pp. 774–778.
[144] J. N. Bott and J. J. Laviola, Jr., "The woz math recognizer: A mathematics handwriting recognition wizard of OZ tool," Univ. Central Florida, Orlando, FL, USA, Tech. Rep. CS-TR-11-03, 2011.
[145] J. J. LaViola and R. C. Zeleznik, "MathPad 2: A system for the creation and exploration of mathematical sketches," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 432–440, Aug. 2004.
[146] R. Zeleznik, T. Miller, C. Li, and J. J. LaViola, "MathPaper: Mathematical sketching with fluid support for interactive computation," in *Proc. Int. Symp. Smart Graph.*, 2008, pp. 20–32.
[147] MathBrush Labs. (2020). *MathBrush*. [Online]. Available: https://apps.apple.com/ca/app/mathbrush/id578957934
[148] Microsoft Corporation. (2019). *Microsoft Office*. [Online]. Available: https://products.office.com/
[149] Microsoft Corporation. (2020). *Microsoft Math Solver*. [Online]. Available: https://math.microsoft.com/
[150] MyScript. (2020). *MyScript Calculator 2*. [Online]. Available: https://www.myscript.com/calculator
[151] Samsung Electronics. (2020). *S Note*. [Online]. Available: https://galaxy.store/snot
[152] D. Zhelezniakov, A. Cherneha, V. Zaytsev, T. Ignatova, O. Radyvonenko, and O. Yakovchuk, "Evaluating new requirements to pen-centric intelligent user interface based on end-to-end mathematical expressions recognition," in *Proc. 25th Int. Conf. Intell. Interface*, Mar. 2020, pp. 212–220.
[153] R. Zeleznik, A. Bragdon, F. Adeputra, and H.-S. Ko, "Hands-on math: A page-based multi-touch and pen desktop for technical work and problem solving," in *Proc. 23nd Annu. ACM Symp. Interface Softw. Technol.*, 2010, pp. 17–26.
[154] R. Zanibbi, K. Novins, J. Arvo, and K. Zanibbi, "Aiding manipulation of handwritten mathematical expressions through style-preserving morphs," in *Proc. Int. Conf. Graph. Interface*, 2001, pp. 127–134.
[155] E. M. Taranta and J. J. LaViola, "Math boxes: A pen-based user interface for writing difficult mathematical expressions," in *Proc. 20th Int. Conf. Intell. Interace*, Mar. 2015, pp. 87–96.
[156] E. Tapia and R. Rojas, "Recognition of on-line handwritten mathematical expressions in the E-Chalk system–an extension," in *Proc. 8th Int. Conf. Document Anal. Recognit. (ICDAR)*, 2005, pp. 1206–1210.
[157] L. Anthony, J. Yang, and K. R. Koedinger, "Adapting handwriting recognition for applications in algebra learning," in *Proc. ACM Int. EMME Workshop*, 2007, pp. 47–56.
[158] C. Li, T. S. Miller, R. C. Zeleznik, and J. J. LaViola, Jr., "AlgoSketch: Algorithm sketching and interactive computation," in *Proc. SBM*, 2008, pp. 175–182.
[159] W. Lee, R. de Silva, E. J. Peterson, R. C. Calfee, and T. F. Stahovich, "Newton's pen: A pen-based tutoring system for statics," *Comput. Graph.*, vol. 32, no. 5, pp. 511–524, Oct. 2008.
[160] Y. Jiang, F. Tian, H. Wang, X. Zhang, X. Wang, and G. Dai, "Intelligent understanding of handwritten geometry theorem proving," in *Proc. 15th Int. Conf. Intell. Interface*, 2010, pp. 119–128.
[161] J. N. Bott and J. J. LaViola, Jr., "A pen-based tool for visualizing vector mathematics," in *Proc. Int. Sketch-Based Interface Modeling Symp.*, 2010, pp. 103–110.
[162] T. J. Cossairt and J. J. LaViola Jr, "SetPad: A sketch-based tool for exploring discrete math set problems," in *Proc. ACM Int. Symp. Sketch-Based Interface Model.*, 2012, pp. 47–56.
[163] B. Kang and J. LaViola, "LogicPad: A pen-based application for visualization and verification of Boolean algebra," in *Proc. ACM Int. Conf. Intell. Interface*, 2012, pp. 265–268.
[164] S. Cheema, "Pen-based methods for recognition and animation of handwritten physics solutions," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Univ. Central Florida, Orlando, FL, USA, 2014.

[165] MyScript. (2020). *Nebo*. [Online]. Available: https://www.nebo.app
[166] Y. Nakayama, "A prototype pen-input mathematical formula editor," in *Proc. ED-MEDIA*, 1993, pp. 400–407.
[167] A. Delaye and C.-L. Liu, "Text/non-text classification in online handwritten documents with conditional random fields," in *Proc. Chin. Conf. Pattern Recognit.*, 2012, pp. 514–521.
[168] V. Khomenko, A. Volkoviy, I. Degtyarenko, and O. Radyvonenko, "Handwriting text/non-text classification on mobile device," in *Proc. Int. Conf. on Art. Intel. Pattern Recognit.*, 2017, p. 42.
[169] N. Malkin, J. Deatrick, A. Tong, P. Wijesekera, S. Egelman, and D. Wagner, "Privacy attitudes of smart speaker users," *Privacy Enhancing Technol.*, vol. 2019, no. 4, pp. 250–271, Oct. 2019.
[170] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, Q. Liang, D. Bhatia, Y. Shangguan, B. Li, G. Pundak, K. C. Sim, T. Bagby, S.-Y. Chang, K. Rao, and A. Gruenstein, "Streaming end-to-end speech recognition for mobile devices," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6381–6385.
[171] K. Yatani and K. N. Truong, "An evaluation of stylus-based text entry methods on handheld devices studied in different user mobility states," *Pervas. Mobile Comput.*, vol. 5, no. 5, pp. 496–508, Oct. 2009.

DMYTRO ZHELEZNIAKOV received the M.Sc. degree in information control systems and technologies from the Kyiv National University of Building and Architecture. He is currently pursuing the Ph.D. degree in computer science with the Taras Shevchenko National University of Kyiv. From 1999 to 2010, he was a Software Engineer and then the Head of the System Development Department, EnranTelecom. In 2010, he joined the Samsung Research and Development Institute Ukraine, Ukraine. His current research interests include handwriting recognition, user experience design, and algorithms optimization for on-device mobile computing.

VIKTOR ZAYTSEV received the M.Sc. degree in computer science from Donetsk National Technical University, Ukraine, in 2000. From 2000 to 2012, he was a System Analyst and a Software Engineer with Nokia Siemens Networks, MTS. He is currently a Software Engineer with the Samsung Research and Development Institute Ukraine, Ukraine. His research interests include recurrent neural networks and time-series analysis.

OLGA RADYVONENKO received the M.Sc. degree in computer science from National Aerospace University, Kharkiv, Ukraine, in 2001, and the Ph.D. degree in artificial intelligence from the Kharkiv National University of Radio Electronics, Ukraine, in 2008.

From 2001 to 2014, she was an Assistant Professor and then an Associate Professor with the Computer Science Department and the Vice-Dean of the Aircraft Control Systems Faculty, National Airspace University. From 2010 to 2013, she has served as a Chairman of a Section Fuzzy Applications at the East–West Fuzzy Colloquium in IPM Hochschule Zittau/Goerlitz—University of Applied Science, Germany. In 2014, she joined the Samsung Research and Development Institute Ukraine, Ukraine. She is involved in research on deep neural networks, document recognition technologies, and intelligent user interfaces. She was a recipient of the Best Young Scientist Award Kharkiv, Ukraine, in 2007, and the Young Scientist Award of the Cabinet of Ministers of Ukraine in 2012.