

Received April 2, 2020, accepted April 27, 2020, date of publication May 4, 2020, date of current version May 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2992067

# A Hybrid Method for Mathematical Expression Detection in Scientific Document Images

BUI HAI PHONG<sup>1,3</sup>, THANG MANH HOANG<sup>2</sup>, AND THI-LAN LE<sup>1,2</sup>

<sup>1</sup>MICA International Research Institute, Hanoi University of Science and Technology, Hanoi 100000, Vietnam

<sup>2</sup>School of Electronics and Telecommunications, Hanoi University of Science and Technology, Hanoi 100000, Vietnam

<sup>3</sup>Faculty of Information Technology, Hanoi Architectural University, Hanoi 100000, Vietnam

Corresponding author: Bui Hai Phong (hai-phong.bui@mica.edu.vn)

This work was supported by the Domestic master's/Ph.D. Scholarship Programme of the Vingroup Innovation Foundation.

**ABSTRACT** Mathematical expressions have been widely used in scientific documents. In order to analyze the documents, automatic detection of mathematical expressions is a crucial step. The paper presents a unified system for the detection of mathematical expressions including both inline and isolated expressions in scientific document images that usually consist of heterogeneous components (e.g., figures, tables, text and expressions). In the system, a hybrid method of two stages is proposed for the effective detection of mathematical expressions. First, the layout analysis of entire document images is introduced to improve the accuracy of text line and word segmentation. Then, both isolated and inline expressions in document images are detected. Both hand-crafted and deep learning features are extensively investigated and combined to improve the detection accuracy. Furthermore, a generic performance metric is applied to evaluate the system comprehensively. The proposed method has been evaluated on two public benchmark datasets (Marmot and GTDB). The obtained accuracies of isolated and inline expressions in the Marmot dataset are 91.18% and 81.35% while those in the GTDB dataset are 89.51% and 80.20%, respectively. The performance comparison is carried out with the conventional methods to show the outstanding effectiveness of the proposed system. Moreover, extensive experiments have been performed in order to point out the effect of document image resolution and post processing techniques on mathematical expression detection.


**INDEX TERMS** Mathematical expression detection, document analysis, machine learning, neural network, fusion technique.

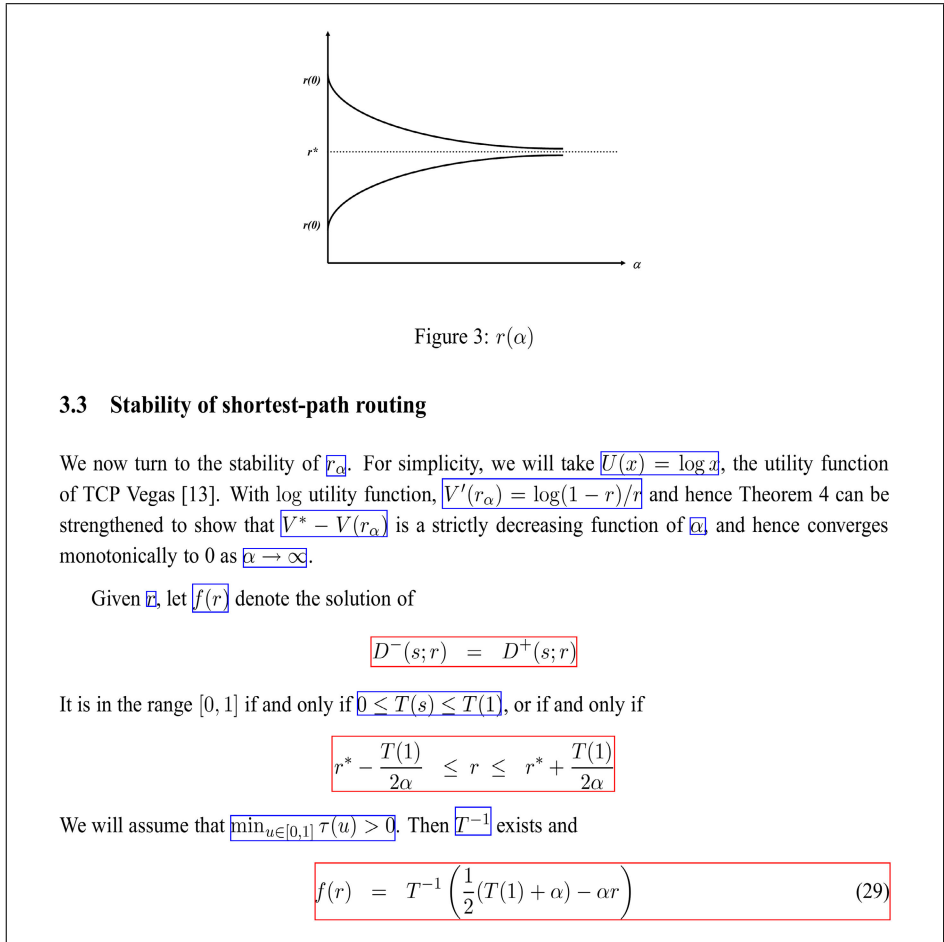
## I. INTRODUCTION

Mathematical expressions have widely used in scientific documents and an huge number of scientific documents have been produced over years. Therefore, the demand of document digitization for researching and studying purposes has continuously increased [1]. Detection of mathematical expressions in documents is considered as an essential step in the document information retrieval system. The detection typically consists of three main steps: page segmentation, classification of mathematical expressions and normal texts and post-processing. In scientific documents, mathematical expressions are classified in two categories, i.e. isolated (displayed) and inline (embedded) expressions. Isolated expressions display in separate lines, meanwhile inline expressions are mixed with other components in document

pages, e.g. texts and figures. Figure 1 illustrates some examples of isolated and inline expressions marked in red and blue, respectively. The detection of mathematical expressions has recently received many researches [1].

In the literature, the accuracy of the detection of isolated expressions has been gradually improved. However, the detection of inline expressions remains low accuracy [2]. There are many challenges in the detection of inline expression, including the variety of mathematical symbols and the complex layout of mathematical structures. In practical, inline expressions may consist of subscripts and superscripts associated with mathematical symbols or variables. As shown in Figure 1, inline expressions consist of mathematical operators (e.g.  $\sum$ ,  $\int$ ,  $\beta$ ,  $+$ ,  $-$ ,  $*$ ,  $/$ ), functions ( $\log$ ,  $\sin$ ,  $\cos$ ) and variables ( $i$ ,  $j$ ). The accuracy of detection of inline expressions can also be affected by punctuation marks and noises. Most existing detection methods have utilized heuristic rules or machine learning approaches with hand-crafted

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



**FIGURE 1.** Examples of the isolated and inline expressions in a sample document page that are marked in red and blue bounding boxes, respectively.

feature extraction. Those methods can be efficient in specific cases, however, they are not robust for various document layout. In addition, private datasets [2] have been used for testing. The *precision* and *recall* metrics have been employed for the performance evaluation. These metric are popular, but can not fully reflect the quality of the detection. In reality, mathematical expressions can be detected completely or partially correct. In some cases, the expressions cannot be detected or other components in documents are identified as expressions. The accuracy flaws have caused many difficulties in the development of mathematical detection systems.

The paper presents an extension of the work reported in [3], and compared with the previous work, there are three main improvements:

(1) Page segmentation is a prerequisite step of detection of mathematical expressions. The accurate segmentation of the text lines and words allows to obtain high accuracy of the detection of mathematical expressions. The challenges in the page segmentation for the detection of mathematical expressions are not only the complicated layout of document but also the variation in sizes, styles of the expressions. In order to overcome the obstacles, the work first over-segments the text

lines and words from input document images. Then, Then, the over-segmented text lines and words are merged by using the heuristic white space in document background. In the paper, the analysis is carried out to evaluate the impact of the results of page segmentation to the accuracy of mathematical expression detection.

(2) A hybrid method that combines both hand-crafted and deep learning features is proposed to improve the accuracy of the detection of mathematical expressions. In this work, Fast Fourier Transform (FFT) magnitude and phase are used as features for isolated expression and normal text line classification while the parameters of Gaussian distribution of peaks and valleys of both vertical and horizontal projection profiles of word images are used for inline expression and textual work classification. As Convolutional Neural Network (CNN) allows capture the rich visual features of images, in this paper, transfer learning techniques are applied on two pre-trained CNNs models that are Alexnet [5] and ResNet-18 [6] for mathematical expression and text line classification.

(3) A generic performance metric and public datasets are used to evaluate the system clearly. The proposed system is

tested with two benchmark datasets which has clear ground-truth information of mathematical expressions to obtain the in-depth evaluation of the effectiveness of the system. It is worth to mention that most detection methods have been evaluated on private datasets that are unavailable for the research [2].

The rest of the paper is organized as follows. Section II overviews significant related works. Section III presents the detail of the architecture of the proposed system. In section IV, experimental results are shown and discussed. Finally, section V gives the conclusion and the future work.

## II. RELATED WORK

This section reviews the works significantly related to the detection of isolated and inline expression in the image-based and PDF formats.

### A. DOCUMENT LAYOUT ANALYSIS

Page segmentation aims to decompose a document image into homogeneous regions by several steps. Firstly, the image pre-processing (noise removal and skew correction) is performed. Then, each component (e.g. text, figure, or table) is separated based on their structure layout. Traditional document layout analysis techniques can be divided into four types: top-down, bottom-up, multi-scale resolution and hybrid method [7]. Top-down methods split the page image into smaller components [8], [9]: a page is split into blocks, blocks are split into text lines and text lines are split into words. In general, top-down methods are useful in the segmentation of rectangular layout. However, the methods are not much effective for the complex structure document. Bottom-up methods analyze and merge local pixels in order to form larger components such as characters, words, text lines and paragraphs [10], [11]. Comparing with top-down methods, bottom-up methods show higher performance in page segmentation. However, the methods have high computational complexity. The multi-scale resolution methods analyze page structure based on the features of different resolution levels of the document image [12], [13]. Then, the features are used for text and non-text classification. Finally, text regions are split in to text lines by using a set of rules of number and intensity of pixels. The difficulty of the methods is the estimation of distance parameters between components in a document page. Hybrid methods combine the bottom-up and top-down techniques. The methods are effective for the segmentation of complex structure document [7], [14]. Connected components and delimiters (white space, tap stop) in a document page are extracted, filtered and analyzed. After that, various heuristic strategies are applied to reduce page segmentation errors. For the purpose of mathematical expression detection, text regions in the body of document are focused to analyse. A text region is segmented into text lines that are basic units for displayed expression detection. Segmented words from a text line are basic units for inline expression detection. For literature documents, there is not much variation in text lines. Thus, the text line segmentation usually achieves high

accuracy [15], [16]. In contrast, there is variation of height, distance in text lines in scientific documents that contain mathematical notations. This issue causes many errors for the text line segmentation. One of the typical error of the segmentation is that a large mathematical expression is split into many lines. Therefore, additional techniques (e.g. rule-based, learning-based methods) are integrated to improve the accuracy of text line segmentation [17], [18]. The basic idea of the techniques is that all text lines are split, then consecutive text lines are merged to form the entire expression if they are belonging to components of the mathematical expression. The text lines are merged if the vertical distance between them is smaller than a predefined threshold. Similarly, consecutive words are merged in order to form the entire expression if they belong to the expression. The words are merged if the horizontal distance between them is smaller than a predefined threshold. In recent years, deep learning approaches have been utilized for the page segmentation. The advantage of the approaches is that the page segmentation task is performed without the prior knowledge of document structure. The work in [19] has proposed a simple CNN with one layer to perform the page segmentation. Input of the CNN is a gray scale document image. The work [20] has employed a DNN based on Resnet-50 [6] to segment historical document pages.

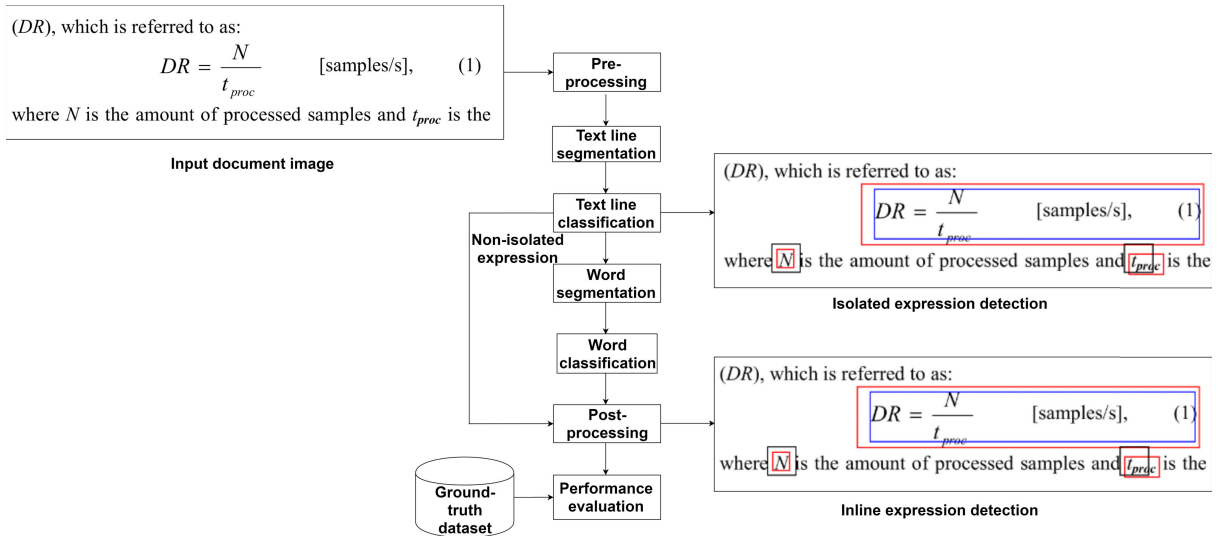
### B. MATHEMATICAL EXPRESSION DETECTION

#### 1) MATHEMATICAL EXPRESSION DETECTION IN DOCUMENT IMAGES

Mathematical expression detection has been studied for more than twenty years [21]. In the traditional detection approaches, the page segmentation is normally performed to obtain text lines and words. Then, the hand-crafted feature extraction is designed to discriminate the mathematical expressions from texts. The difference of the approaches is in the ways of feature extraction and the use of different classifiers. In the early research on the mathematical expression detection [22], all text lines and words in a document page are scanned in order to get primitive tokens. After that, each token is determined whether it belongs to an inline expression by checking predefined expression forms. The accuracy of detection is not reported in this research. Research in [23] concluded that it is difficult to detect all inline expressions without using character recognition results.

The method reported in [24] employs results of two commercial optical character recognition (OCR) systems to extract inline formula. First, existing OCR systems are applied to obtain content of document images. Then, sentences containing inline expressions are determined by computing word n-grams. For each sentence, several features of a word are extracted to determine whether the word is a part of inline expressions. The features of words mentioned in the work are:

(1) The probability of a sentence containing inline expressions.



**FIGURE 2.** Overall description of the proposed system for mathematical expression detection. The detection of isolated, inline and ground-truth expressions are marked in blue, black and red, respectively.

- (2) The confidence of OCR systems while recognizing words.
- (3) The type style (italic, bold) of words.
- (4) The space between characters of words.
- (5) The variation of position of characters in words.

If some consecutive words in a sentence are determined as inline expressions, these words can be grouped to obtain an inline expression. It is obvious that above features highly depend on results of existing OCR systems.

The method in [25] aims at detecting both isolated and inline expressions from document images. The method firstly applies the low-cost text line segmentation technique [26] for heterogeneous document images. Then, features of each text line are extracted. After the feature extraction, Support Vector Machine (SVM) classifier is used to determine if the text line is an isolated expression. Non-isolated expressions are segmented into words and features of words are extracted to check if the word belongs to an inline expression. The extracted features of words in the method are described follows:

- (1) The density of black pixels in the word image.
- (2) The proportion of the height of word to that of the whole document.
- (3) The fluctuation of “centroid” of characters in words.

The features of words are effective in the detection of special symbols but not accurate in the detection of inline expressions. The precision and recall of the detection of inline expressions reported at 80% and 48%, respectively.

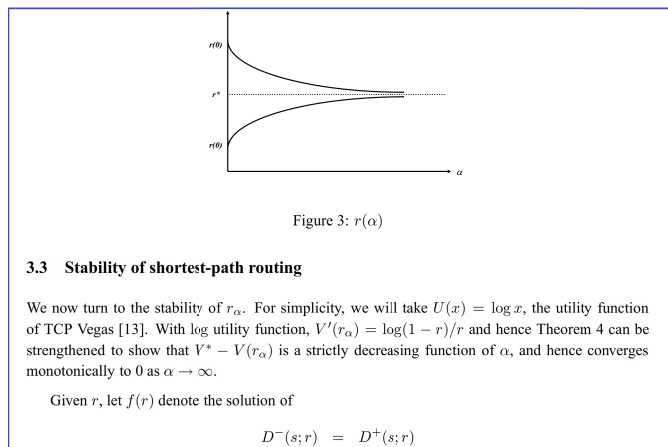
In recent years, DNNs have proved the outstanding performance in the recognition and detection of mathematical expressions tasks [27]–[29]. The work in [27] takes the advantages of CNNs in the detection of isolated and inline expressions in document images. A CNN architecture based on the U-net [30] is used for detecting mathematical expressions. The document images are divided into blocks.

The annotated information of characters in blocks are used for training the CNN. The purpose of the method is to obtain connected components of expressions. The post-processing is performed to obtain accurate expressions. For the CNN, the training on different datasets can improve the detection accuracy. Moreover, the accuracy of the detection depends on the size of image blocks in the training of CNN. The achieved precision and recall of the detection of mathematical expressions in the method are 95.2% and 91%, respectively. As stated in the method, mathematical symbols are detected with high accuracy, however the layout analysis of symbols has not been implemented to construct complete expressions. The italic and bold type styles of words can cause errors in the detection of inline expressions.

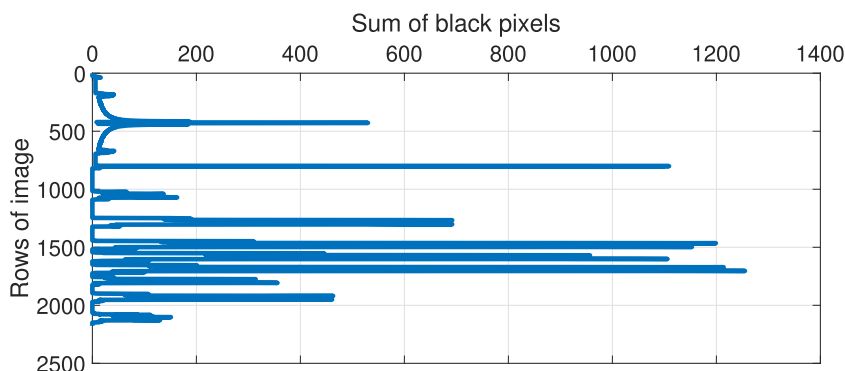
## 2) MATHEMATICAL EXPRESSION DETECTION IN NATIVE PDF DOCUMENTS

In recent years, several researches [21], [31], [32] have focused on the detection of mathematical expressions in PDF documents. For PDF documents, metadata information of textual words such as font, size, styles can be extracted precisely. Therefore, the detection of mathematical expressions in PDF documents is more accurate than that of image-based documents. The method reported in [31] extracts inline expressions in PDF documents with the use of natural language processing. After the word extraction process, word features and conditional random field (CRF) are used for inline expression detection. The achieved accuracy in detection is 88.95% on PDF files from the ACL Anthology dataset [31] but there are still many errors in the detection of variables reported in the research.

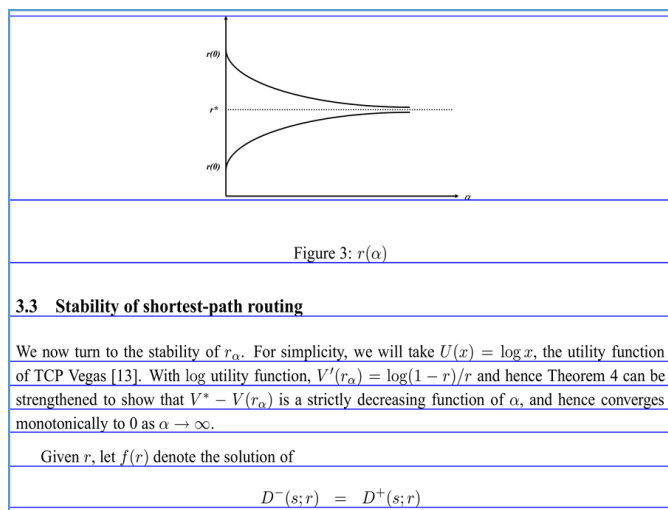
The research [33] attempts to detect mathematical expressions in PDF documents by taking the advantages of CNNs. The framework for the detection consists of two steps. In the



(a) Example of a document page.



(b) The horizontal projection profile of the sample page.

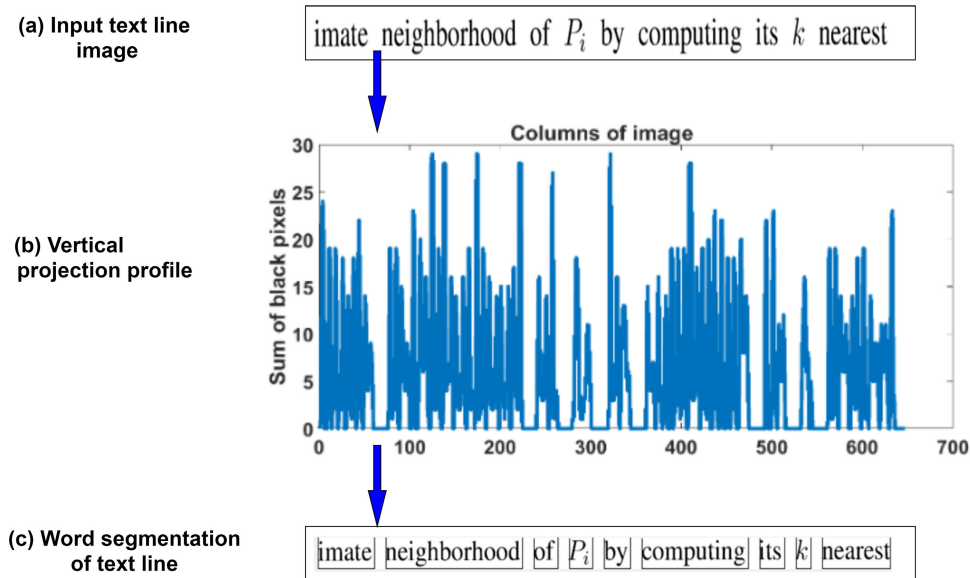


(c) The text line segmentation of the sample page.

**FIGURE 3.** Example of the text line segmentation in a sample document image. The input sample page (a), the horizontal projection profile of the page image (b) and the text line segmentation of the page (c). The x-axis represents the sum of black pixels of each row in page image and y-axis represents the rows of the image.

first step, the candidate regions for mathematical expressions are generated. For the generation of candidate regions, meta-data information including position, fonts of characters are

extracted from PDF files. In the second step, the features of candidate regions are extracted in order to obtain the entire mathematical expressions. In this step, two deep networks



**FIGURE 4.** The word segmentation of the text line image (a) based on the estimation of the vertical projection profile (b) and the results (c). The x-axis represents the columns of text line image and y-axis represents the sum of black pixels of each column.

are combined to automatically extract features of candidate regions. The first network is the CNN and the second one is the Recurrent Neural Network (RNN). The CNN is employed to extract visual features of images and the RNN is utilized to extract sequential information of characters. After that, the features are combined together to improve the accuracy of expression detection. A large dataset (12,000 document pages containing more than 22,000 mathematical expressions) is manually prepared for training the deep networks.

As above-mentioned, a number of works have been proposed for isolated and inline expression detection. However, the performance of inline expression detection is needed to be improved. In our work, a hybrid method for the mathematical expression detection has been proposed in which the accuracy of the detection of inline expressions is focused to improve. We combine the hand-crafted and deep learning features in order to obtain higher accurate detection of mathematical expressions in various document layout.

### III. SYSTEM ARCHITECTURE

The proposed system is illustrated in Figure 2. The proposed system takes a binary document image as input and outputs an image with position information of detected mathematical expressions. Like document analysis and expression detection methods, input of the proposed method is a non-skew document image. For skew and curved images, the deskew [34] and dewarping [35] algorithms must be applied. In our work, the image dewarping algorithm in [35] is adopted to correct of the distortions of input documents. By considering information of both text and non-text regions, the dewarping algorithm is designed for a wide range of document layouts. The algorithm can handle camera-captured and scanned

document images. After the pre-processing, the document is analyzed to obtain text lines for isolated expression detection. Non-isolated expressions are segmented into words for inline expression detection. After the segmentation, the combination of hand-crafted and deep learning features are applied for the isolated and inline expression detection modules. Finally, the post-processing is performed in order to obtain the accurate position information of mathematical expressions in document images.

#### A. PAGE SEGMENTATION

The projection profile [36] of a document image is applied for the page segmentation. The estimation of projection profile of images is performed recursively to analyze the structure of documents [36]. The horizontal and vertical projection profiles of an image is the horizontal and vertical distribution of black pixels. Thus, the technique is useful for the analysis of scanned documents. To obtain text regions, text and non-text regions can be classified based on the following layout features:

(1) The width and height of regions: The height of tables and figures are normally larger than that of text lines. Meanwhile, the width of non-text components are smaller than that of text lines. Thresholds can be used for confirming the text and non-text components, in this work, the threshold of the height of text line is set from 50 to 400 (pixels) and that of the width is set from 200 to 4000 (pixels).

(2) The number of the connected components in regions: text lines normally consist of more connected components than non-text elements. In our work, the heuristic threshold is chosen with the number of 5 connected components for

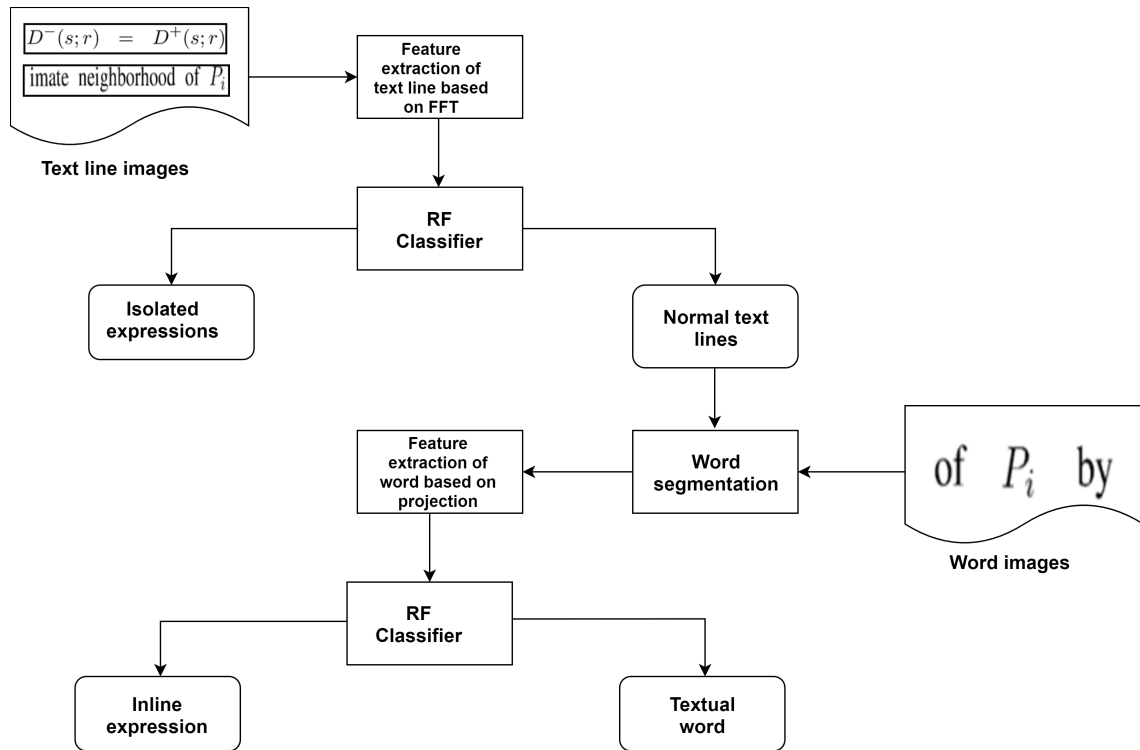


FIGURE 5. The flowchart of the isolated and inline expression detection by using hand-crafted feature extraction.

the filter. In other words, a text line typically consists of more than 5 connected components.

In fact, the heuristic filtering is used to remove non-text components (e.g. small noises, tables, figures). After the text and non-text classification, text regions are segmented into text lines. The text lines are segmented by using the threshold of vertical distance between them. The threshold is set as 20 (pixels) in this case. Examples of the text line segmentation in a document page is shown in Figure 3. The input sample page, the horizontal projection profile of the page and the results of text line segmentation are illustrated in Figure 3(a), 3(b) and 3(c), respectively. In Figure 3(b), the x-axis and the y-axis represent the sum of black pixels and the rows of the page image, respectively.

Segmented text lines are fed into the isolated expression detection module in order to identify isolated expressions. Then, text lines that are not determined as isolated expressions are segmented into words. The words are segmented by using the threshold of horizontal distance between them. The threshold is set as 10 (pixels) in this case. The segmented words are fed into the inline expression detection module in order to identify inline expressions. Examples of word segmentation in a text line is shown in Figure 4. The word segmentation of the text line, the vertical projection profile of the text line and the results of word segmentation are illustrated in Figure 4(a), 4(b) and 4(c), respectively. In Figure 4(b), the x-axis represents the columns and the y-axis represents the sum of black pixels of the page image.

## B. MATHEMATICAL EXPRESSION DETECTION

### 1) EXPRESSION DETECTION BY USING HAND-CRAFTED FEATURE EXTRACTION

The flowchart of the isolated and inline expression classification is described in Figure 5. In the hand-crafted feature extraction approach, the powerful feature extraction and classifier are applied to improve the accuracy of the classification of both isolated and inline expressions. Text line images are transformed from the spatial to the frequency domain by using the FFT [37] to classify isolated expressions and normal text lines. The dominant features of mathematical symbols are emphasized by using the transformation. Both FFT magnitude and phase values are used as features of text line images. Those also allow to clearly discriminate the isolated expression from text line images. Actually, the white space between characters of isolated expressions is larger than those of normal text line and the density of black pixels in isolated expressions is less than that of normal text line.

In order to improve the accuracy of the classification of inline expressions and textual words, feature extraction based on projection profiles of word images is applied [38]. Firstly, both vertical and horizontal projection profiles of word images are calculated. Then, the parameters of Gaussian distribution of peaks and valleys of both vertical and horizontal projection profile of word images are used as the features. Peaks and valleys are local maximum and minimum of the projection profile of word images. The features of the

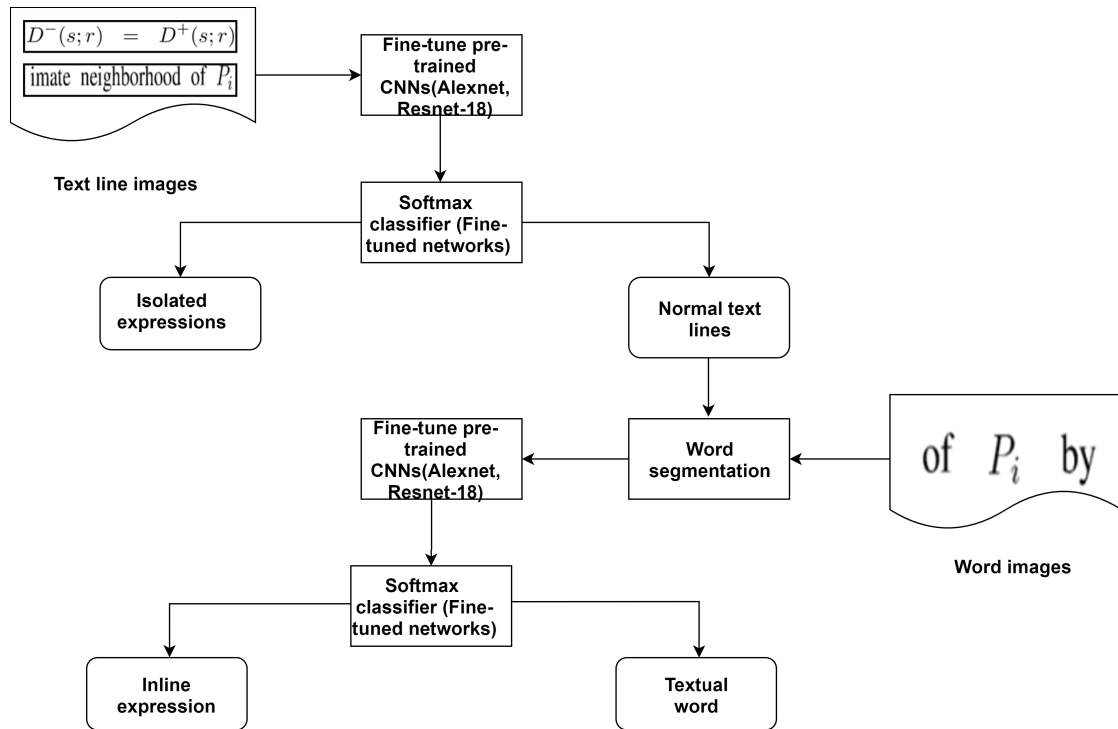


FIGURE 6. The flowchart of the isolated and inline expression classification by using the transfer learning of CNNs.

vertical and horizontal projection profiles of word images are extracted as follows:

- (1) The number of peaks in the vertical and horizontal projection profiles.
- (2) The mean (average) of values of peaks in the vertical and horizontal projection profiles.
- (3) The standard deviation of values of peaks in the vertical and horizontal projection profiles.
- (4) The number of valleys in the vertical and horizontal projection profiles.
- (5) The mean (average) of values of valleys in the vertical and horizontal projection profiles.
- (6) The standard deviation of values of valleys in the vertical and horizontal projection profiles.

By using the feature extraction, two-dimensional layout properties of inline expressions are extracted that can improve the accuracy of the classification.

After the feature extraction, Random Forest (RF) classifier is used for the classification. Compared with other machine learning classifiers such as Support Vector Machine (SVM) and k-Nearest Neighbor (k-NN), the RF shows better results in the classification [38]. The RF demonstrates the effectiveness in the classification task because it aggregates a large number of classification results of decision trees [39]. For training the RF in the classification of isolated expression, labels of two classes (isolated expression and text line) are prepared manually. The extracted features based on FFT and labels are used to train the classifier. Similarly, for training the RF in the classification of inline expression, labels of

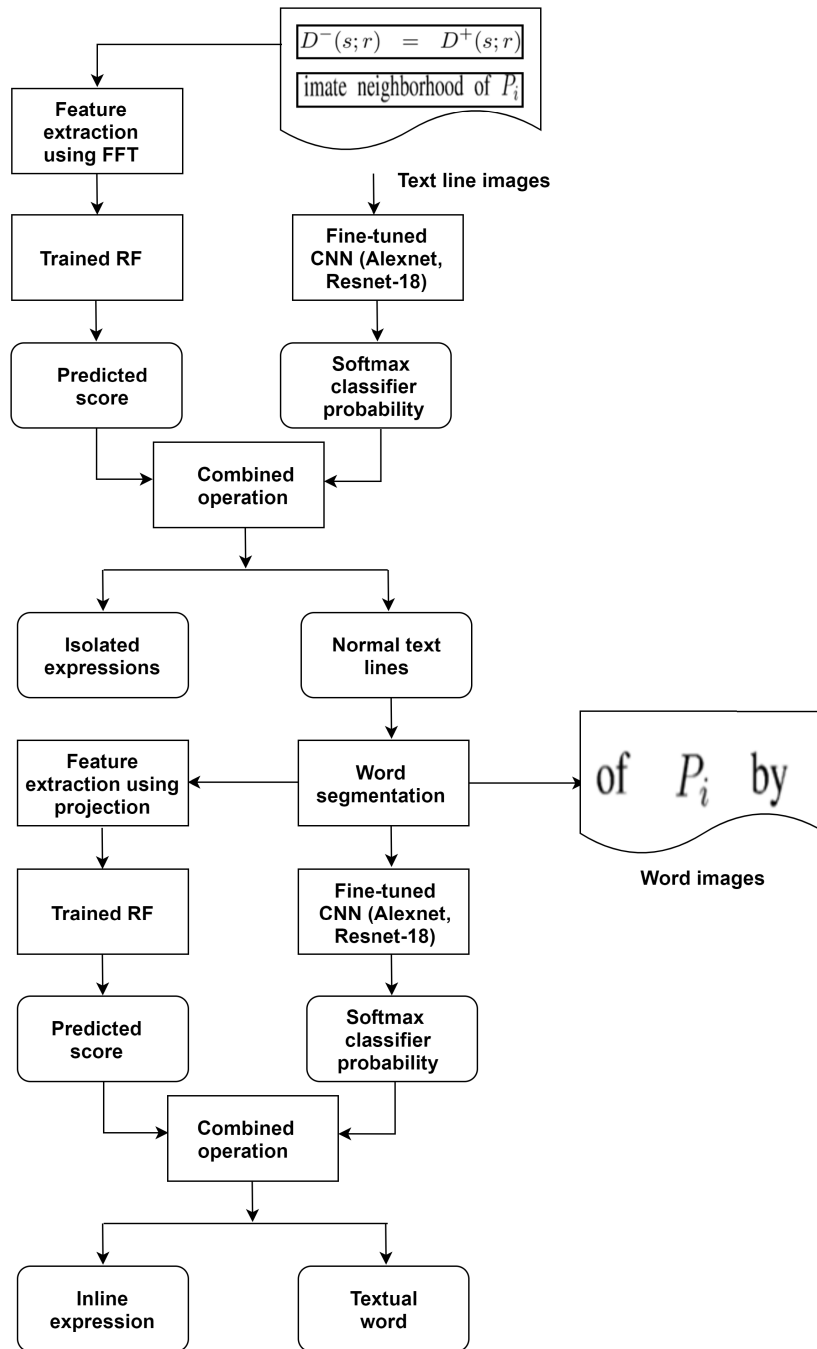
two classes (inline expression and word) are also prepared manually. Then, the extracted features based on projection profile and labels are used to train the classifier. The number of trained cycles is set as 100 and the adaptive logistic regression algorithm [40] is used for training the Random Forest. Finally, the trained model is used for the classification of testing data.

## 2) EXPRESSION DETECTION BY USING CONVOLUTIONAL NEURAL NETWORK

To improve the accuracy of the detection of both isolated and inline expressions, the transfer learning technique of AlexNet [5] and ResNet-18 [6] those are popular Neural Networks are employed. Comparing with AlexNet, the architecture of ResNet-18 consists of deeper layers and ResNet-18 normally shows better results in the classification task [41].

For the AlexNet, each input image is pre-processed and resized to  $[227 \times 227 \times 3]$ . The CNN consists of 25 layers with 5 convolutional layers and 3 fully connected layers. The architecture and parameters of layers of Alexnet are provided in Table 1. For ResNet-18, input image is pre-processed and resized to  $[224 \times 224 \times 3]$ . The CNN consists of 72 layers corresponding to 18 blocks. The architecture and parameters of layers of the ResNet-18 are provided in Table 2. For the isolated and inline expression detection modules, input images of the CNNs are text lines and words, respectively. For AlexNet and ResNet-18, 4096 and 512 visual features are automatically extracted from input images, respectively.





**FIGURE 7.** The flowchart of the combination of hand-crafted and deep learning features in the classification of isolated and inline expression.

Figure 6 illustrates the flowchart of the transfer learning of CNNs for isolated and inline expression detection module. The dominant features are automatically extracted by the network without any domain specific knowledge. Then, the classification is performed by *softmax* layer of the network. The learning rate and the number of *epochs* parameters of the network are set as 0.001 and 20, respectively. The stochastic gradient descent (SGDM) [42] algorithm with momentum [43] that is set as 0.9 is used for training the CNNs.

### 3) EXPRESSION DETECTION BY COMBINING THE HAND-CRAFTED AND DEEP LEARNING FEATURES

In order to leverage the advantages of both hand-crafted features and CNN models, the decision results obtained by these features will be combined using the score-based fusion technique [44], [45]. Concretely, in this work, the confidence scores obtained from hand designed features with RF and CNN features with Softmax are combined using product and average operators. Let  $p_1$  and  $p_2$  be the predicted scores of the hand-designed features with RF and fine-tuned CNN with

TABLE 1. Alexnet architecture and layer parameters.

Layer id	Layer Name	Layer type	Layer parameters
1	imageInputLayer	Input Image	227×227×3
2	conv1	Convolution	55×55×96
3	relu1	ReLU	55×55×96
4	norm1	Cross Chanel Normalization	55×55×96
5	pool1	Max Pooling	27×27×96
6	conv2	Grouped Convolution	27×27×256
7	relu2	ReLU	27×27×256
8	norm2	Cross Channel Normalization	27×27×256
9	pool2	Max Pooling	13×13×256
10	conv3	Convolution	13×13×384
11	relu3	ReLU	13×13×384
12	conv4	Grouped Convolution	13×13×384
13	relu4	ReLU	13×13×384
14	conv5	Grouped Convolution	13×13×256
15	relu5	ReLU	13×13×256
16	pool5	Max Pooling	6×6×256
17	fc6	Fully ConnectedLayer	1×1×496
18	relu6	ReLU	1×1×496
19	drop6	Dropout	1×1×496
20	fc7	Fully Connected Layer	1×1×496
21	relu7	ReLU	1×1×496
22	drop7	Dropout	1×1×496
23	fc8	Fully Connected Layer	1×1×2
24	prob	Softmax	1×1×2
25	output	Classification output	Output result

TABLE 2. ResNet-18 architecture and layer parameters.

Layer Name	Output Size	Layer parameters
conv1	112×112×64	7×7,64, stride 2
conv2_x	56×56×64	3×3maxpool, stride2 $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	28×28×128	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	14×14×256	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	7×7×512	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	1×1×512	7×7 average pool
fully connected	2	512×2 fully connections
softmax	2	Classification results

TABLE 3. Statistic of the Marmot and GTDB datasets.

Datasets	GTDB		Marmot	
	GTDB-1 (Training)	GTDB-2 (Testing)	Training	Testing
Number of pages	569	236	330	70
Number of isolated expressions	4218	2488	1322	253
Number of inline expressions	22178	9397	6951	956
Number of text fonts	30		18	
Average number of expressions per page	47.55		23.70	

Softmax, respectively. The final prediction  $p$  is determined as follows:

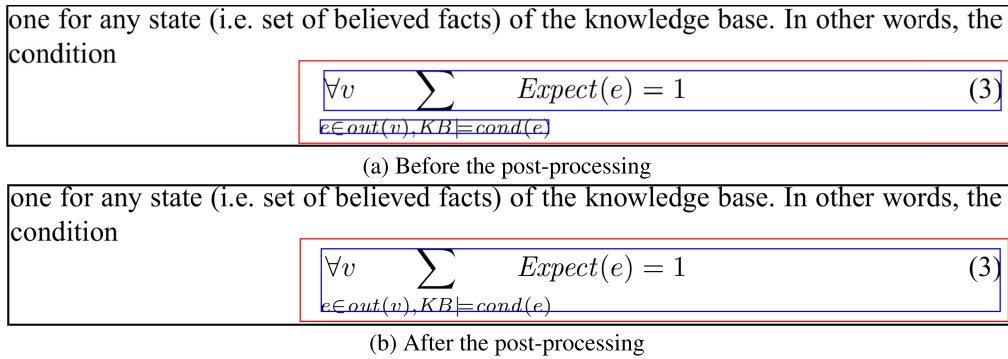
$$p = F(p_1, p_2) \quad (1)$$

where  $F$  is the combination operator.

The flowchart of the combination is described in Figure 7. The obtained score is used to classify the expression and text.

### C. POST-PROCESSING

In the detection of mathematical expressions, it is not rare that large isolated expressions are split into several text lines [21]. Some strategies are proposed in order to overcome the issue [21], [46]. The strategies have relied on the results of the character recognition to determine the conditions of merging successive text lines to become an expression. In our work, the heuristic threshold of white-space between successive



**FIGURE 8.** Example of the post-processing of a mathematical expression that is split into two text lines. The detected and ground-truth expressions are marked in blue and red, respectively. (a) before and (b) after the post-processing.

**TABLE 4.** Performance comparison between the proposed and existing methods of isolated expression detection on the Marmot dataset (highest scores are in bold).

Method	Isolated expression can be detected			Error in the detection		
	Correct	Partial	Total	Missed	False	Total
Method in [25]	26.87%	44.89%	71.76%	9.89%	18.35%	28.24%
Our method						
using FFT and RF	31.02%	42.32%	73.34%	9.04%	17.62%	26.66%
using transfer learning of AlexNet	47.22%	41.44%	88.66%	2.78%	8.56%	11.34%
using transfer learning of ResNet-18	50.89%	39.27%	90.16%	3.55%	6.29%	9.84%
combining features with average operator	<b>51.34%</b>	39.45%	90.79%	3.55%	5.66%	9.21%
<b>combining features with product operator</b>	<b>51.34%</b>	<b>39.84%</b>	<b>91.18%</b>	<b>3.14%</b>	<b>5.68%</b>	<b>8.82%</b>

**TABLE 5.** Performance comparison between the proposed and existing methods of inline expression detection on Marmot dataset (highest scores are in bold).

Method	Inline expression can be detected			Error in the detection		
	Correct	Partial	Total	Missed	False	Total
Method in [25]	1.74%	28.87%	30.61%	9.93%	59.46%	69.39%
Our method						
using projection profile and RF	11.05%	41.40%	52.45%	8.36%	39.19%	47.55%
using transfer learning of AlexNet	21.54%	56.25%	77.79%	7.60%	14.61%	22.21%
using transfer learning of ResNet-18	22.68%	57.06%	79.74%	5.59%	14.67%	20.26%
combining features with average operation	22.79%	57.96%	79.85%	5.79%	14.36%	20.15%
<b>combining features with product operation</b>	<b>22.90%</b>	<b>58.45%</b>	<b>81.35%</b>	<b>5.40%</b>	<b>13.25%</b>	<b>18.65%</b>

**TABLE 6.** Performance comparison between the proposed and existing methods of isolated expression detection on the GTDB dataset (highest scores are in bold).

Method	Isolated expression can be detected			Error in the detection		
	Correct	Partial	Total	Missed	False	Total
Method in [25]	26.22%	44.87%	71.09%	9.91%	19.00%	28.91%
Our method						
using FFT and RF	30.86%	42.12%	72.98%	9.25%	17.77%	27.02%
using transfer learning of AlexNet	47.05%	41.16%	88.21%	3.78%	8.01%	11.79%
using transfer learning of ResNet-18	50.29%	38.67%	88.96%	3.85%	7.19%	11.04%
combining features with average operator	50.34%	39.15%	89.49%	3.57%	6.94%	10.51%
<b>combining features with product operator</b>	<b>50.37%</b>	<b>39.14%</b>	<b>89.51%</b>	<b>3.16%</b>	<b>7.33%</b>	<b>10.49%</b>

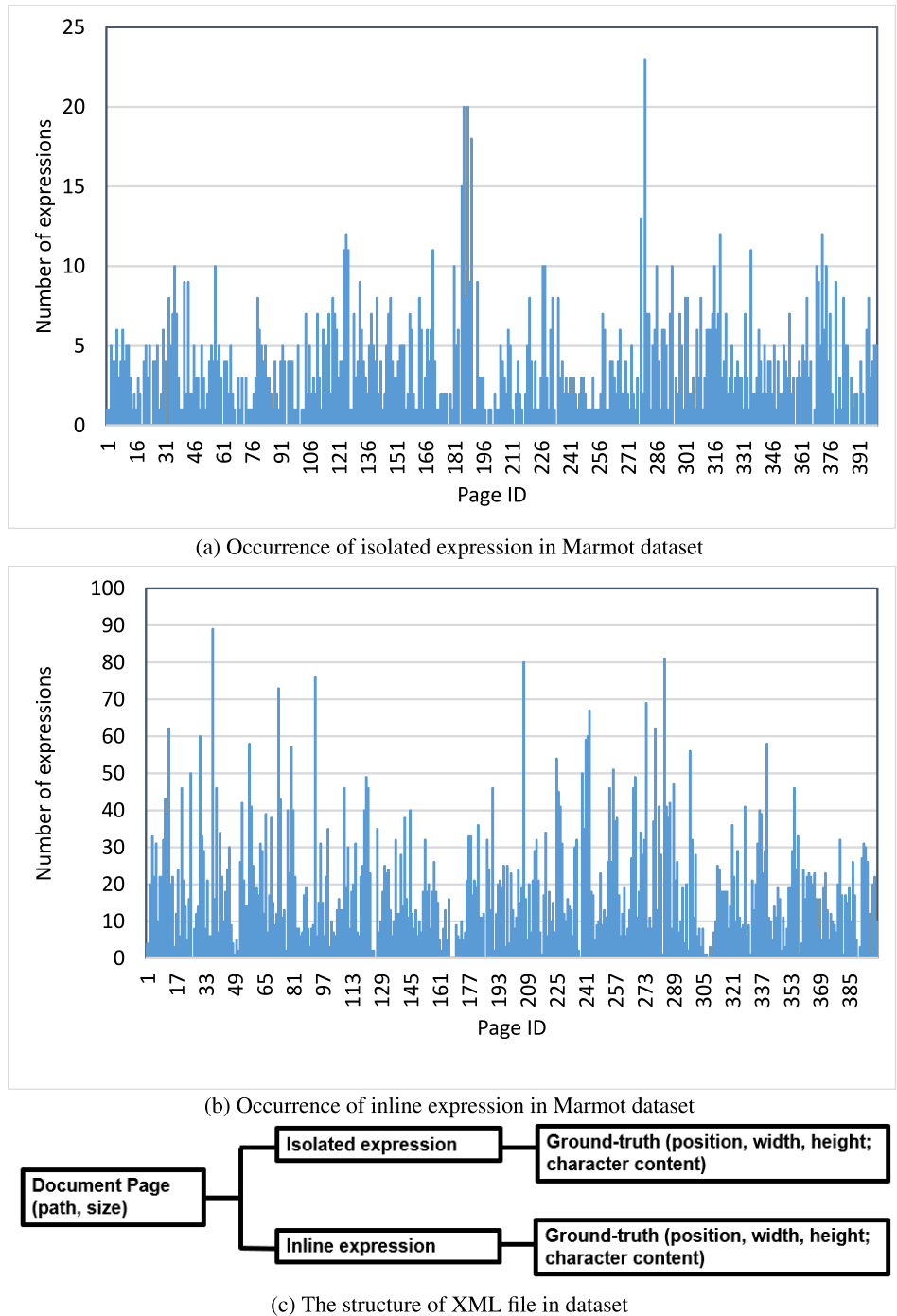
text lines are carefully considered for the post-processing. Two successive text lines are merged if the vertical distance between the text lines is smaller than a threshold (100 pixels in this work). By using the threshold, the text lines can be merged efficiently to obtain entire expressions without using any additional character recognition modules.

Let  $line_i$  and  $line_j$  be the successive text lines that are classified as isolated expressions. The text lines are considered

to merge to form an entire isolated expression if the flowing condition is satisfied:

$$|y_i - y_j| \leq H \quad (2)$$

where  $y_i$  and  $y_j$  are y-coordinates of the top-left corner of the text lines and  $H$  is the predefined threshold. The threshold is set by the observation of the average height of text lines in the whole documents.



**FIGURE 9.** The occurrence of isolated (a) and inline (b) expressions in document pages in Marmot dataset. The x-axis represents the number of expressions and the y-axis represents the Page ID in the dataset. The structure of XML file storing ground-truth information of expression (c).

Similarly, two successive words are merged to form entire inline expressions if the horizontal distance between the words is smaller than a threshold (20 pixels in the work). Examples of the post-processing are demonstrated in Figure 8. The expression is split into two text lines in Figure 8(a) and the text lines are merged to form the entire expression in Figure 8(b).

#### IV. EXPERIMENTAL RESULTS

##### A. DATASET

In this section, two public datasets that have been used for performance evaluation of mathematical expression detection are described.

The first one is Marmot public dataset [2]. It consists of 400 non-skew scientific document pages with

**TABLE 7. Performance comparison between the proposed and existing methods of inline expression detection on GTDB dataset (highest scores are in bold).**

Method	Inline expression can be detected			Error in the detection		
	Correct	Partial	Total	Missed	False	Total
Method in [25]	1.56%	28.67%	30.23%	9.97%	59.80%	69.77%
Our method						
using projection profile and RF	10.48%	41.36%	51.84%	8.26%	39.90%	48.16%
using transfer learning of AlexNet	20.46%	55.24%	75.70%	7.86%	16.44%	24.30%
using transfer learning of ResNet-18	22.16%	56.34%	78.50%	6.29%	15.21%	21.50%
combining features with average operation	22.69%	56.65%	79.34%	5.68%	14.98%	20.66%
<b>combining features with product operation</b>	<b>22.76%</b>	<b>57.44%</b>	<b>80.20%</b>	<b>5.46%</b>	<b>14.34%</b>	<b>19.80%</b>

1575 isolated and 7907 inline expressions. The resolution of each page image is around 500 dpi. In the dataset, 18 fonts of texts are used in the documents as follows: ArialMT, Courier, Helvetica, NimbusRomNo9L, Lasy9, TimesNewRomanPSMT, TimesNewRomanPS-ItalicMT, TimesNewRomanPS-BoldMT, CMMI10, MSBM10, CMR7, CMSY10, CMBX12, CMEX10, CMTI10, SymbolMT, Universal-GreekwithMathPi, GillSans-BoldCondensed. The text size varies from 4px to 22px. The training and testing datasets are described in Table 3. The number of isolated and inline expressions in each page is described in Figure 9(a) and Figure 9(b), respectively. In the figures, the y-axes represent the number of expressions and the x-axes represent the pages in the dataset. Each document page contains an average of 4 and a maximum of 20 isolated expressions. Each page contains an average 20 and a maximum of 90 inline expressions. For each page, the position information of the top left and bottom right corner of each expression is stored in the XML files that are described in Figure 9(c). The precise bounding boxes of expressions are represented by Hexadecimal numbers that consist of 16 characters. The symbols of mathematical expressions are also annotated in the XML files. The ground-truth is created by a semi-automatic tool and available for public research on the mathematical expression detection purposes.

The second one is GTDB public dataset [27]. It has recently been used for GTDB performance evaluation of researches [48]. The dataset consists of diverse font and mathematical symbol styles. In the dataset, 30 fonts of texts are used in the documents as follows: TimesNewRoman, Arial, CourierNew, AGaramondPro-Regular, HiddenHorzOCR, Helvetica, CMR6, CMBX10, CMR8, CMCS10, CMMI8, CMR10, CMMI10, CMSY7, CMEX10, CMMI7, CMR7, CMSY10, MSBM10, CMTI8, CMSY6, CMR5, CMTI10, CMSY8, CMMI6, CMMI5, MSBM7, CMSY5, CMBX8, CMTT8. The text size varies from 4px to 28px. The dataset consists of scientific articles in PDF format. Due to the copyright reason, the dataset does not directly provide the PDF articles. However, links to the articles are provided in the dataset. In order to obtain document images, the PDF articles in the dataset are converted at 600 dpi. Compared with the Marmot dataset, the GTDB dataset is larger and more challenging for expression detection due to the complexity of document layout and the diversity of scientific articles. The GTDB-1 dataset is used for training and the GTDB-2 dataset

is used for testing. The statistic of the datasets are described in Table 3. The datasets provide ground truth bounding box for both character and mathematical expression regions. The ground truth bounding box is stored in CSV files. In our work, the information of expression regions is used for the performance evaluation.

## B. EVALUATION METRIC

In order to obtain the in-depth analysis of the proposed system, the Intersection over Union (*IoU*) metric is used in our work. The metric is also known as Jaccard index that is widely used to evaluate the performance of object detection system [47]. The metric is the ratio of the overlapped and union of the detected and ground-truth regions. In our work, the detected and ground-truth expressions are represented by the coordinates of the top left corner and the size of the bounding boxes of the expressions. The *IoU* is calculated as follows:

$$IoU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \quad (3)$$

where  $B_p \cap B_{gt}$  and  $B_p \cup B_{gt}$  denotes the intersection and union of the predicted and ground-truth bounding boxes of expressions.

*IoU* value is in the closed interval [0;1] and the larger value shows better detection results. Based on the threshold of the metric, the detection results are divided into four categories as follows:

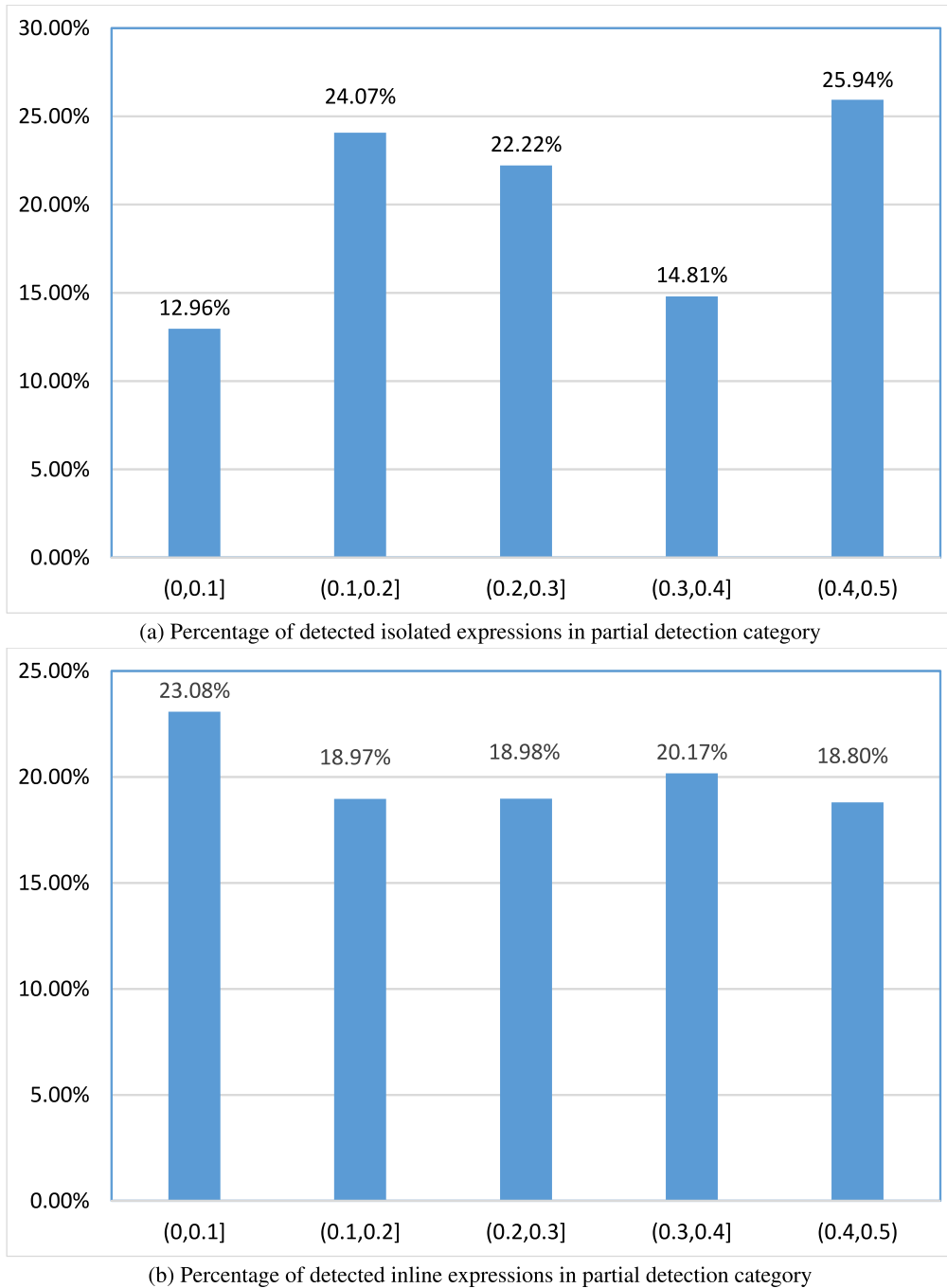
- 1) *Correct*: the *IoU* value of the detected and ground-truth regions is in the closed interval [0.5; 1].
- 2) *Partial*: the *IoU* value of the detected and ground-truth regions is in the interval (0; 0.5).
- 3) *Missed*: mathematical expression cannot be detected by the proposed system.
- 4) *False*: other components in document page are detected as mathematical expressions.

By using the evaluation metric, the quality of the detection is clearly reflected.

## C. PERFORMANCE EVALUATION

### 1) PERFORMANCE EVALUATION OF THE DETECTION OF ISOLATED AND INLINE EXPRESSIONS ON DIFFERENT PUBLIC DATASETS

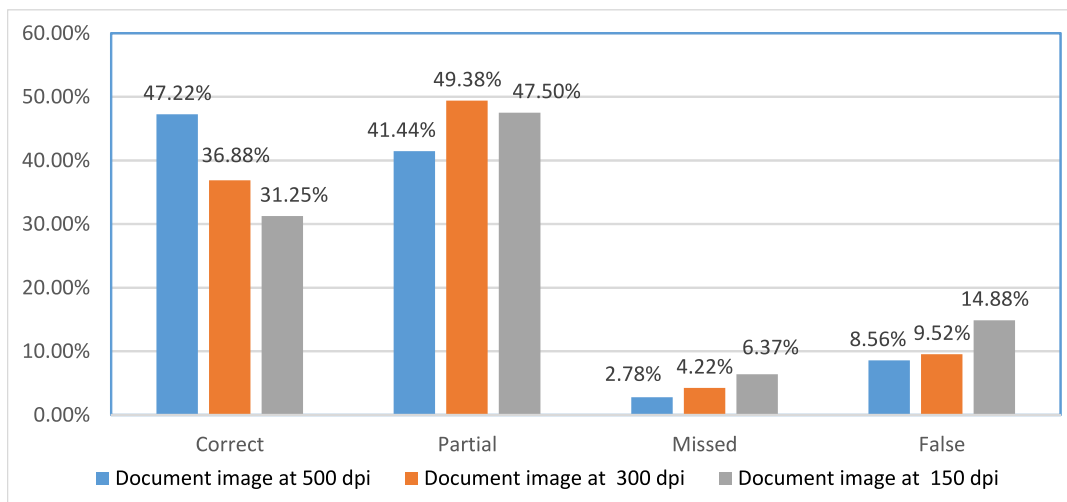
The performance comparison between the proposed and conventional methods of isolated and inline expression detection



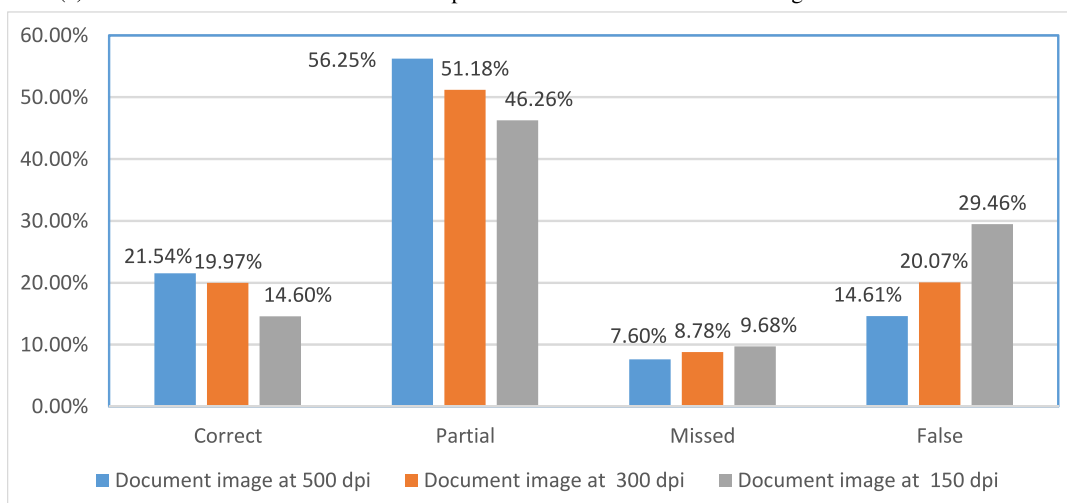
**FIGURE 10.** Percentage of detected isolated and inline expressions in partial detection category. The partial detection category is divided into five equal sub-ranges based on IoU values.

in the Marmot dataset is shown in Tables 4 and 5, respectively. The proposed system outperforms conventional method due to the effective strategies on document analysis and novel classification techniques. Particularly, the transfer learning of CNNs obtains the highest accuracy in the detection because the CNNs extract more visual features of images than those in other methods. The method [25] focuses to extract features of bounding boxes of characters in word images. The method is not effective for the detection of inline expressions because

there is not much variation in the visualization of inline expressions. The method using FFT and projection profiles of images obtains higher accuracy than the method [25] because it can extract two-dimensional layout features of mathematical expressions. It is clearly shown in Table 5 that the accuracy of detection of the inline expression is much improved by using the transfer learning of CNNs. The performance of the method using the transfer learning of the ResNet-18 is slightly higher than that of AlexNet. The out-performance



(a) Performance evaluation of isolated expression detection in document images at various resolution



(b) Performance evaluation of inline expression detection in document images at various resolution

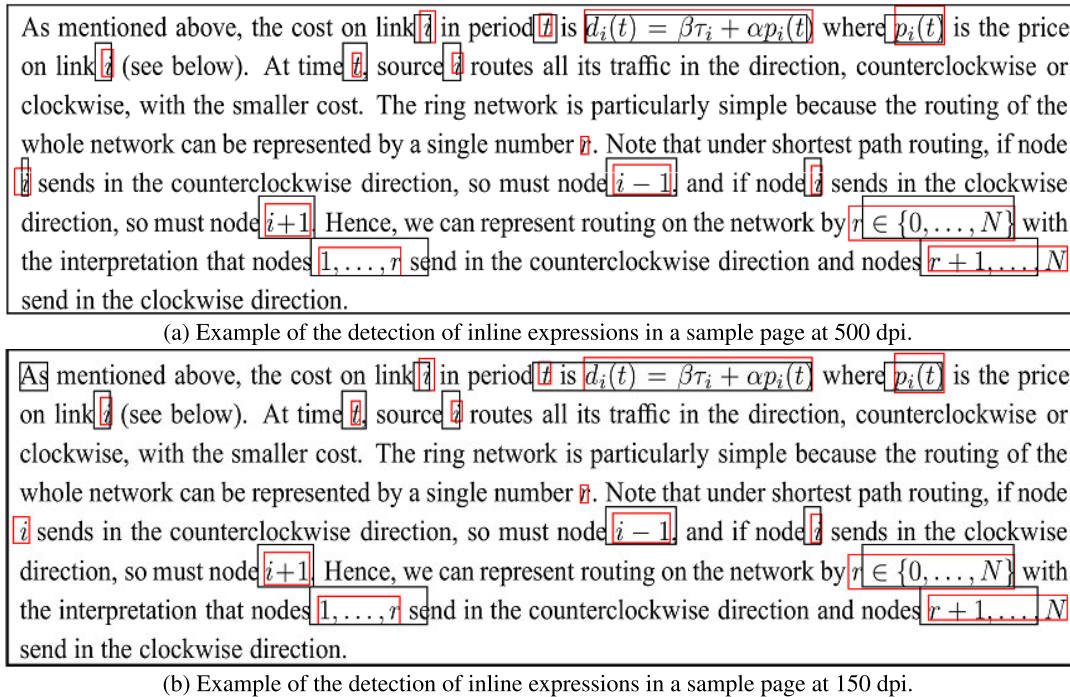
**FIGURE 11. Performance evaluation of the detection of isolated (a) and inline (b) expressions in document images at various resolution. The testing images in the Marmot dataset are rendered at 500, 300 and 150 dpi. The performance of the detection is denoted by blue, orange and gray for document images at 500, 300 and 150 dpi, respectively.**

is obtained because the deeper architecture of the ResNet-18 allows to extract visual feature better than that of AlexNet. The combination of RF and ResNet-18 allows to obtain the highest performance in the isolated expression detection because the predicted scores of two models are aggregated for the final classification and the misclassification is reduced.

For inline expressions, the percentage of partial detection is much higher than that of correct category. Thus, the percentage of inline expression detection based on the various ranges of IoU values of the partial category is evaluated in order to obtain the further analysis of the performance of the detection. The percentage of inline expression detection based on different ranges of IoU values is demonstrated in the Figure 10(b). Actually, the percentage of inline expression detection fluctuates slightly in the five ranges ((0; 0.1], (0.1; 0.2], (0.2; 0.3], (0.3; 0.4], (0.4; 0.5)) of partial detection category. The percentage of the inline expression

detection in the lowest range (0; 0.1] is not much higher than that of other ranges. The results show that the proposed method can detect inline expressions in difficult cases (e.g. the expressions consist of small mathematical symbols). The percentage of isolated expression detection based on different ranges of IoU values is demonstrated in Figure 10(a). The figure shows that the percentage of the isolated expression detection in the highest range (0.4; 0.5) is much higher than that of other ranges. The result demonstrates the effectiveness of the proposed method for isolated expression detection.

The performance comparison between the proposed and conventional methods of isolated and inline expression detection in the GTDB dataset is shown in Tables 6 and 7, respectively. Compared to the Marmot dataset, the detection of mathematical expressions in the GTDB dataset is more challenging. Actually, in the GTDB dataset, the distance between consecutive text line and word is narrower than that of the



**FIGURE 12.** Examples of the detection of inline expressions in a sample page at 500 (a) and 150 (b) dpi. The inline expressions detected by proposed system and ground-truth expressions are marked in black and red, respectively.

Marmot dataset and there is much variation in type styles (font and size of character) in the GTDB dataset. Therefore, the performance of detection of expressions in the GTDB is lower than that of the Marmot dataset.

The performance comparison between the proposed and state of the art methods on the GTDB dataset is shown in Table 8. For the GTDB dataset, the method of Samsung R&D based on graph theory [48] has shown the highest performance. However, it is worth noting that the method in [48] exploits character-level information it is provided in GTDB dataset for the detection of mathematical expression. This information is not available in others datasets such as Marmot dataset. As our method relies only on the appearance of mathematical expressions in the document images, it is general and can be applied for different datasets. In comparison with the similar method, the proposed method shows better performance than that of the Michiking system [48] because the employment of CNNs extracts features more efficiently than that of traditional rule-based and machine learning techniques.

## 2) EVALUATION OF THE IMPACT OF IMAGE RESOLUTION ON MATHEMATICAL EXPRESSION DETECTION

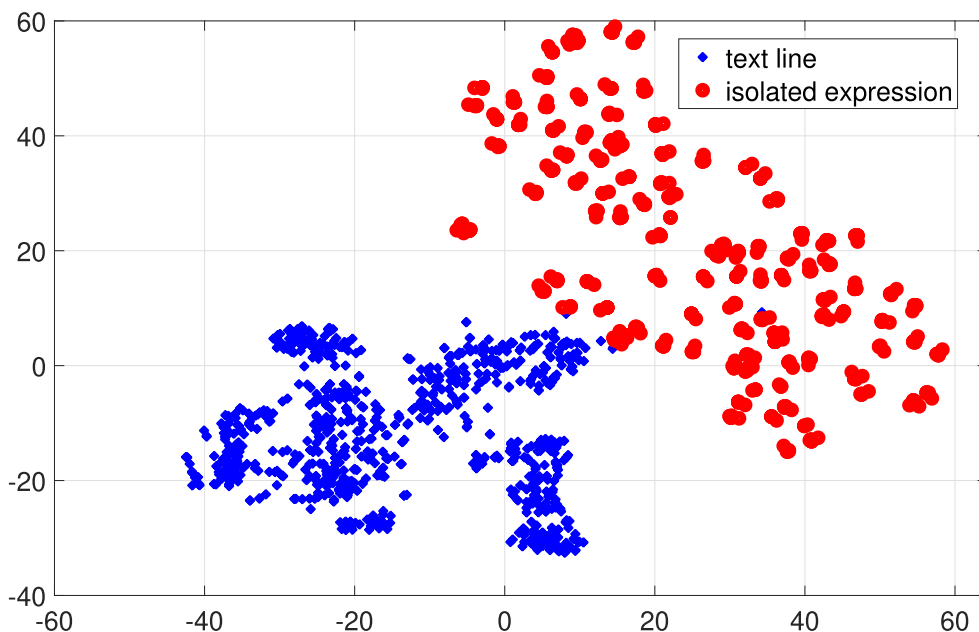
For traditional methods based on OCR technique, input document images are typically rendered at high resolution (around 600 dpi) [27] to prevent recognition errors. Our method detects mathematical expressions without using any OCR modules. Thus, the detection can be performed in low-resolution document images. In order to evaluate the

impact of the resolution to our method, the performance evaluation has carried out on document images at various resolutions. Document images are rendered at 500, 300 and 150 dpi. The performance evaluation is shown in Figure 11. As shown in the figure, the percentage of missed and failed detection slightly increases for document images at 300 dpi. Whereas, the percentage noticeably increases for document images at 150 dpi. The results have shown that our proposed method can perform the document image rendered at more than 300 dpi. For document images rendered at low resolution, the error rate has increased during the page segmentation. Therefore, the overall error rate of expression detection has significantly increased for document images at low resolution. Figure 12(a) and 12(b) illustrate the detection of inline expressions in a sample page at 500 and 150 dpi, respectively.

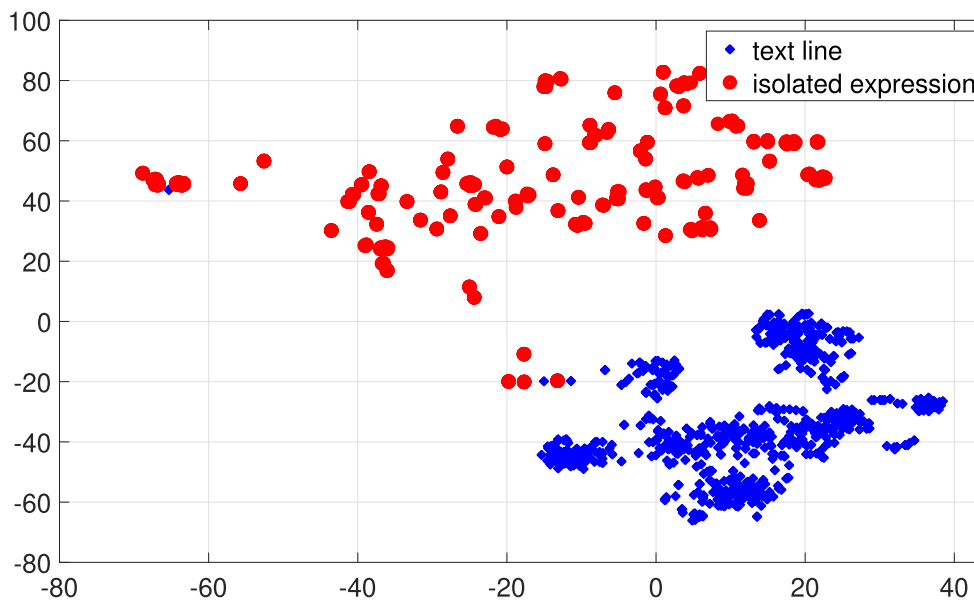
## 3) VISUALIZATION OF EXTRACTED FEATURES OF IMAGES USING THE TRANSFER LEARNING OF CNN MODEL

In order to demonstrate the effectiveness of the feature extraction of fine-tuned CNNs, the distribution of extracted features of testing images of isolated expressions and normal text lines is visualized. The extracted features of isolated expressions and normal text lines by using the ResNet-18 are illustrated in blue and red in Figure 13, respectively. For the ResNet-18, the visual features are automatically extracted at *pool5* layer at the end of the network. The dimensional reduction technique is used to visualize learned features of text line and isolated





(a) The visualization of feature extraction using dimensional reduction with the Mahalanobis distance metric.



(b) The visualization of feature extraction using dimensional reduction with the Cosine distance metric.

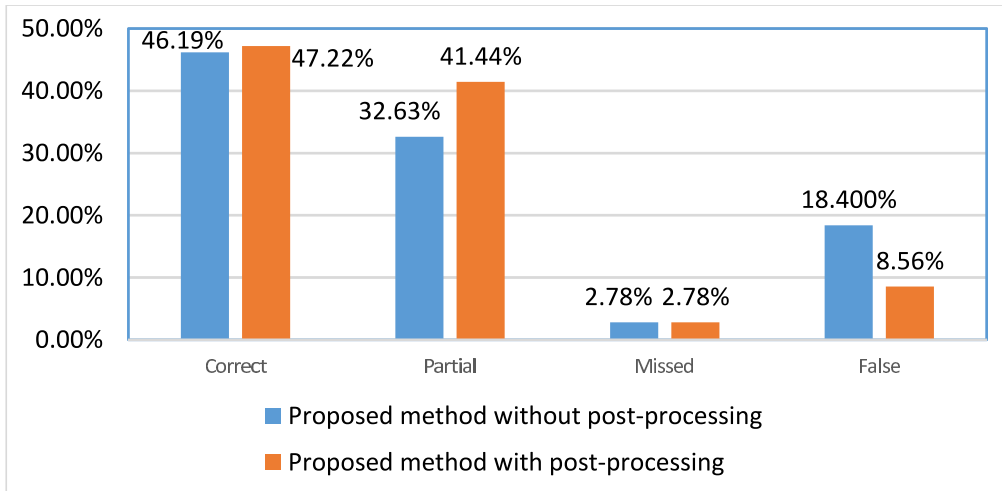
**FIGURE 13. The feature distribution of isolated and text line images. The extracted features of isolated expressions and normal text lines in Marmot dataset by using the ResNet-18 are illustrated in red and blue, respectively. The visualization of extracted feature of text lines and isolated expression by using the t-SNE dimensional reduction with the Mahalanobis (a) and Cosine (b) distance metrics.**

**TABLE 8. Performance comparison of the proposed and the state of the art methods on the GTDB dataset.**

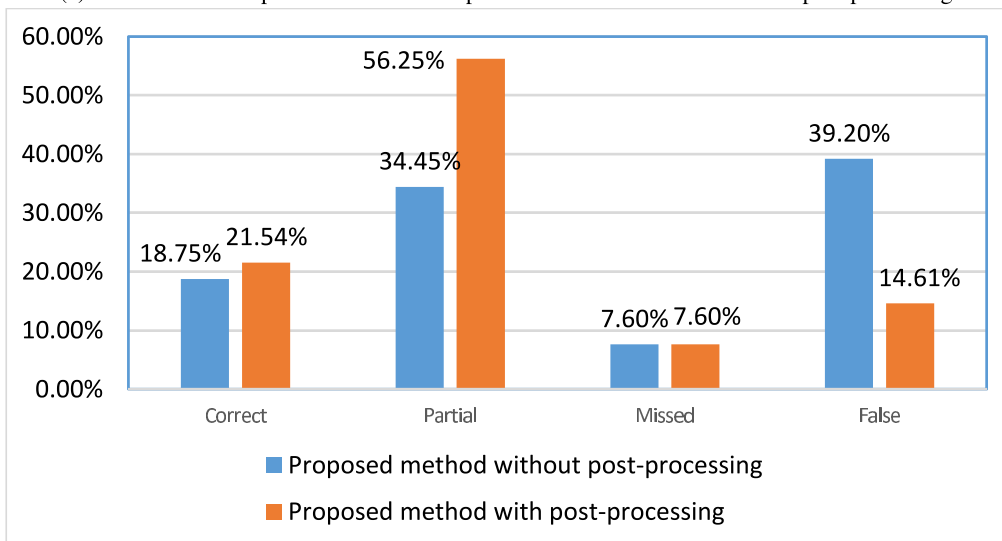
Method	Expression detection results with $IoU \geq 0.5$	Expression detection results with $IoU \geq 0.75$
Method of Samsung based on graph theory [48]	94.36%	94.17%
Michiking system [48]	36.87	19.10%
Proposed method	50.17%	43.19%

expression images. In this case, 512 extracted features of each testing image of isolated expressions and text lines are reduced to 2 features for the visualization purpose by using the T-distributed Stochastic Neighbor

Embedding (t-SNE) technique [49]. The t-SNE has demonstrated out-performance on various datasets compared to other dimensional reduction techniques (e.g. Classical scaling [50], Principal component analysis [51]). Eight hundred



(a) Performance comparison of isolated expression detection before and after post-processing



(b) Performance comparison of inline expression detection before and after post-processing

**FIGURE 14.** Performance comparison of the proposed method before (in blue) and after (in orange) the post-processing in the detection of isolated (a) and inline (b) expressions.

**TABLE 9.** The average time of the detection of expressions in a document page in the Marmot dataset by different methods (Bold value indicates the smallest detection time).

Methods	Average detection time per page (second)	
	Isolated detection	Inline detection
Method in [25]	9.56	24.8
Our method		
using FFT, projection profile and RF	<b>2.6</b>	<b>13.04</b>
using AlexNet	14.34	49.6
using ResNet-18	22.17	56.5
combining features	22.30	56.6

images of each class of isolated expression and normal text lines are used in the visualization. The images are normalized to the size of  $[224 \times 224 \times 3]$  as the ResNet-18 requirement. The technique aims to respect the similarities between points in the visual space with the reduction from high-dimension to low-dimension. It is clearly shown in Figure 13 that most of testing images are separated into two classes. Various

distance metrics are used for the t-SNE technique. In our work, two popular distance metrics including the Mahalanobis and Cosine are employed. The visualization of extracted feature of text lines and isolated expressions by using the t-SNE dimensional reduction with the Mahalanobis and Cosine distance metric is shown in Figure 13 (a) and 13(b), respectively. The t-SNE technique

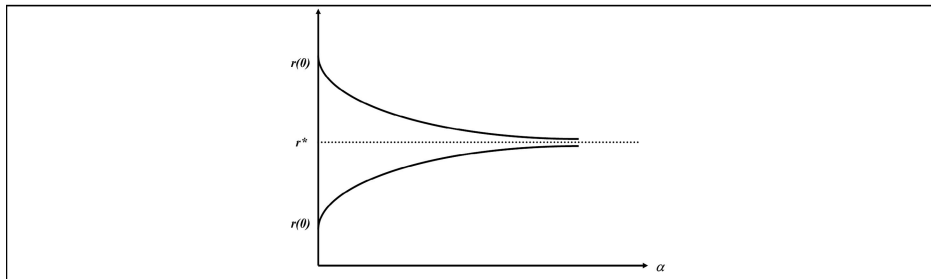


Figure 3:  $r(\alpha)$

### 3.3 Stability of shortest-path routing

We now turn to the stability of  $r_\alpha$ . For simplicity, we will take  $U(x) = \log x$ , the utility function of TCP Vegas [13]. With log utility function,  $V'(r_\alpha) = \log(1-r)/r$  and hence Theorem 4 can be strengthened to show that  $V^* - V(r_\alpha)$  is a strictly decreasing function of  $\alpha$  and hence converges monotonically to 0 as  $\alpha \rightarrow \infty$ .

Given  $\alpha$ , let  $f(r)$  denote the solution of

$$D^-(s; r) = D^+(s; r)$$

It is in the range  $[0, 1]$  if and only if  $0 \leq T(s) \leq T(1)$ , or if and only if

$$r^* - \frac{T(1)}{2\alpha} \leq r \leq r^* + \frac{T(1)}{2\alpha}$$

We will assume that  $\min_{u \in [0,1]} \tau(u) > 0$ . Then  $T^{-1}$  exists and

$$f(r) = T^{-1} \left( \frac{1}{2}(T(1) + \alpha) - \alpha r \right) \tag{29}$$

(a) Examples of the isolated and inline expression detection in one-column page.

- *reproducible*: running several times the same benchmark program on a given device under the same conditions, performance results should not change considerably (except for uncertainty contributions);
- *portable*: the benchmark has to be independent of a particular technology or architecture.

Besides these basic properties, a kernel benchmark test should also be:

$$DR = \frac{N}{t_{proc}} \quad [\text{samples/s}], \tag{1}$$

where  $N$  is the amount of processed samples and  $t_{proc}$  is the processing time (e.g. the time to compute an FFT algorithm on  $N=1024$  complex samples). Notice that this parameter is reliable because it is inversely proportional to the processing

(b) Examples of the isolated and inline expression detection in two-column page.

**FIGURE 15. Examples of the isolated and inline expression detection in one-column (a) and two-column (b) pages in the Marmot dataset. The detection of isolated, inline and ground-truth expressions are marked in blue, black and red, respectively.**

with the Cosine distance metric allows to visualize the separation between two classes better than that of the Mahalanobis distance metric.

#### 4) EVALUATION OF THE IMPACT OF THE POST-PROCESSING TO THE DETECTION OF MATHEMATICAL EXPRESSION

The outcome of the post-processing in the detection of mathematical expressions is clearly shown in Figure 14. The post-processing allows to obtain better accuracy of the detection of isolated and inline expressions. The percentage of partial detection of isolated expressions is increased by 8.81% and the false detection is decreased by 9.84%. Particularly, the post-processing of the inline expression detection allows to obtain the considerable improvement of accuracy.

The percentage of partial detection of inline expressions is increased by 21.80% and the false detection is decreased by 24.59%. The percentage of expressions in missed detection category is not affected by the post-processing because the post-processing aims at merging detected components of expressions. Actually, a large number of expressions consists of multiple words in scientific documents. Therefore, the post-processing is necessary to improve the accuracy of the detection of inline expressions.

#### 5) TIME EFFICIENCY

In order to compare the performance of the methods, the execution time of testing phase of methods is evaluated. The methods are implemented in Matlab R2019a environment on

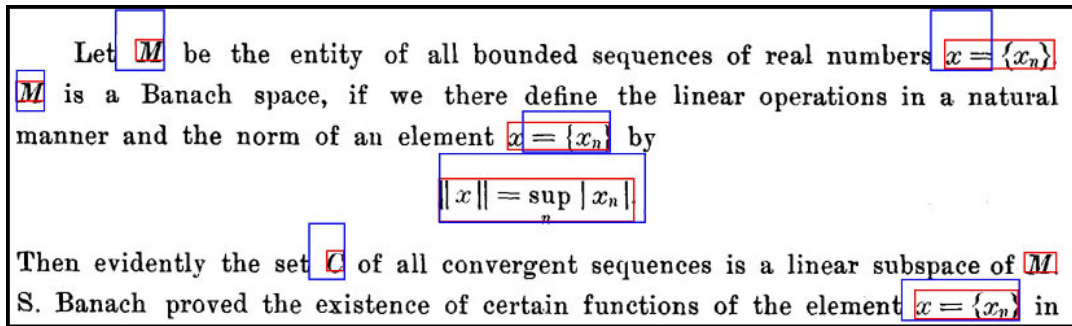
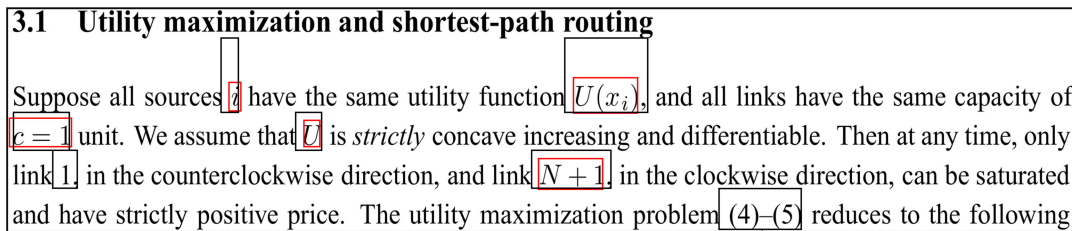
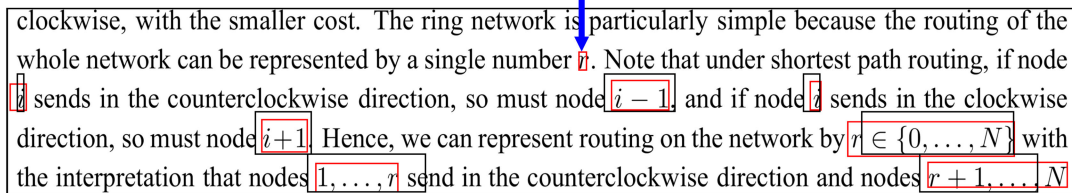


FIGURE 16. Examples of the expression detection in a sample page in the GTDB dataset. The detection and ground-truth expressions are marked in blue and red, respectively.



(a) Example of the false detection of inline expressions.



(b) Example of the missed detection of inline expressions.

FIGURE 17. Examples of the false (a) and missed (b) detection of inline expression. The inline expressions detected by proposed system and ground-truth expressions are marked in black and red, respectively.

a computer with 6GB RAM and Core i3-2.67 GHz processor. The average execution time of the detection of inline expressions in a document page in the Marmot dataset by using the methods is shown in Table 9. As shown in the table, the methods using the transfer learning of CNNs perform slower than those of hand-crafted feature extraction methods. Particularly, for inline expression detection, the classification of expressions and textual words using CNNs requires more time than those of hand-crafted feature extraction methods. The main reason of the time-consuming execution of using the CNNs is that the CNNs aim at extracting more features than those of hand-crafted feature extraction methods. The method using the transfer learning of Alex has shown the slightly higher performance in time execution than that of the ResNet-18. In fact, the ResNet-18 consists of more layers than the Alexnet, the feature extraction takes more time to extract features than that of the Alexnet. For machine learning methods, the proposed method based on the FFT and

projection profile as feature extraction and RF as classifier has shown the highest results. The proposed method has achieved the most effective results in the training and testing time because it focuses on extracting the feature distribution of peak and valley values of projection profiles of an image instead of performing a whole image. Meanwhile, the method reported in [25] is slow in comparison with the proposed method because it extracts all bounding boxes of characters of inline expressions. The combination of hand-crafted and deep learning features allows to obtain the highest accuracy in the detection and it takes the running time similar to the transfer learning of CNNs.

#### D. ERROR ANALYSIS AND DISCUSSION

For examples of isolated and inline expression detection, the results on document page images in the Marmot and GTDB datasets are shown in Figure 15 and Figure 16, respectively. The detection result in the Marmot dataset is more

accurate than that of the GTDB one. It is clearly shown in the figures that isolated expressions are detected with high accuracy. However, there are some errors encountered in the detection of inline expressions. The errors can be classified into two classes:

1) The ambiguity in the detection of some numbers and variables is encountered in the real context. The number symbols and single characters can be used in both mathematical expressions and narrative texts. The factor can cause errors in the detection. An example of the false detection of inline expressions is shown in Figure 17(a). In this case, normal texts containing mathematical symbols are detected as inline expressions.

2) Small mathematical symbols cannot be detected in some cases because of the noises generated in the page segmentation. Concretely, during the word segmentation and merging, small symbols are possibly omitted. An example of the missed detection of inline expression is shown in Figure 17(b). In this case, the small variable  $r$  cannot be detected.

## V. CONCLUSION AND FUTURE WORKS

We have presented a unified system that detects both isolated and inline mathematical expressions in document images. The improvements in the page segmentation and the classification of mathematical expressions and texts are combined to improve the performance of the overall detection system. The combination of hand-crafted and deep learning features is proposed to improve the performance of the detection. In the hand-crafted feature extraction method, the feature extraction based on FFT and RF classifier are applied for the isolated expression detection. The feature extraction based on projection profile and RF classifier are applied for the inline expression detection. The transfer learning of CNNs including the Alexnet and ResNet-18 has efficiently employed in the detection of both isolated and inline expressions. The performance of overall system is evaluated on two public datasets those are the Marmot and GTDB. The generic performance metrics based on IoU are applied to evaluate the system clearly. The obtained results have shown that the performance of the detection of the proposed system is significantly improved comparing with the conventional methods.

In the future, the performance of the system can be further improved by applying various strategies. Different deep neural networks can be combined to improve the accuracy of both the page segmentation and the expression detection. The context information can be integrated to improve the accuracy of the inline expression detection.

## REFERENCES

- [1] 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.
- [2] X. Lin, L. Gao, Z. Tang, X. Lin, and X. Hu, "Performance evaluation of mathematical formula identification," in *Proc. 10th IAPR Int. Workshop Document Anal. Syst.*, Gold Coast, QLD, Australia, Mar. 2012, pp. 287–291.
- [3] B. H. Phong, T. M. Hoang, and T.-L. Le, "A unified system for mathematical expression detection in scientific document images," in *Proc. Korea-Vietnam Int. Joint Workshop Commun. Inf. Sci.*, Hanoi, Vietnam, 2019, pp. 14–16.
- [4] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Proc. Int. Conf. Artif. Neural Netw.* Rhodes, Greece, Oct. 2018, pp. 270–279.
- [5] A. Krizhevsky and I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, Dec. 2012, pp. 1097–1105.
- [6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
- [7] T. A. Tran, I. S. Na, and S. H. Kim, "Page segmentation using minimum homogeneity algorithm and adaptive mathematical morphology," *Int. J. Document Anal. Recognit. (IJ DAR)*, vol. 19, no. 3, pp. 191–209, Sep. 2016.
- [8] F. M. Wahl, K. Y. Wong, and R. G. Casey, "Block segmentation and text extraction in mixed text/image documents," *Comput. Graph. Image Process.*, vol. 20, no. 4, pp. 375–390, Dec. 1982.
- [9] D. Wang and S. N. Srihari, "Classification of newspaper image blocks using texture analysis," *Comput. Vis., Graph., Image Process.*, vol. 47, no. 3, pp. 327–352, Sep. 1989.
- [10] L. Caponetti, C. Castiello, and P. Górecki, "Document page segmentation using neuro-fuzzy approach," *Appl. Soft Comput.*, vol. 8, no. 1, pp. 118–126, Jan. 2008.
- [11] M. Agrawal and D. Doermann, "Voronoi++: A dynamic page segmentation approach based on Voronoi and docstrum features," in *Proc. 10th Int. Conf. Document Anal. Recognit.*, Barcelona, Spain, Jul. 2009, pp. 1011–1015.
- [12] H. Cheng and C. A. Bouman, "Multiscale Bayesian segmentation using a trainable context model," *IEEE Trans. Image Process.*, vol. 10, no. 4, pp. 511–525, Apr. 2001.
- [13] Z. Shi and V. Govindaraju, "Multi-scale techniques for document page segmentation," in *Proc. 8th Int. Conf. Document Anal. Recognit. (ICDAR)*, Seoul, South Korea, Aug./Sep. 2005, pp. 1020–1024.
- [14] H. Dai-Ton, N. Duc-Dung, and L. Duc-Hieu, "An adaptive over-split and merge algorithm for page segmentation," *Pattern Recognit. Lett.*, vol. 80, pp. 137–143, Sep. 2016.
- [15] T. M. Breuel, "The OCRopus open source OCR system," *Proc. SPIE*, vol. 6815, Jan. 2008, Art. no. 68150F.
- [16] R. Smith, "An overview of the tesseract OCR engine," in *Proc. 9th Int. Conf. Document Anal. Recognit. (ICDAR)*, vol. 2, Sep. 2007, pp. 629–633.
- [17] A. M. Nambodiri and A. K. Jain, "Document structure and layout analysis," in *Digital Document Processing*. London, U.K.: Springer, 2007, pp. 29–48.
- [18] X. Lin, L. Gao, Z. Tang, J. Baker, M. Alkalai, and V. Sorge, "A text line detection method for mathematical formula recognition," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Washington, DC, USA, Aug. 2013, pp. 339–343.
- [19] K. Chen, M. Seuret, J. Hennebert, and R. Ingold, "Convolutional neural networks for page segmentation of historical document images," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Kyoto, Japan, Nov. 2017, pp. 965–970.
- [20] S. A. Oliveira, B. Seguin, and F. Kaplan, "DhSegment: A generic deep-learning approach for document segmentation," in *Proc. 16th Int. Conf. Frontiers Handwriting Recognit. (ICFHR)*, Niagara Falls, NY, USA, Aug. 2018, pp. 7–12.
- [21] X. Lin, L. Gao, Z. Tang, J. Baker, and V. Sorge, "Mathematical formula identification and performance evaluation in PDF documents," *Int. J. Document Anal. Recognit.*, vol. 17, no. 3, pp. 239–255, Sep. 2014.
- [22] 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.
- [23] J. Jin, X. Han, and Q. Wang, "Mathematical formulas detection," in *Proc. Int. Conf. Document Anal. Recognit.*, Edinburgh, U.K., 2003, pp. 1138–1141.
- [24] U. Garain, "Identification of mathematical expressions in document images," in *Proc. 10th Int. Conf. Document Anal. Recognit.*, Barcelona, Spain, Jul. 2009, pp. 1340–1344.
- [25] W.-T. Chu and F. Liu, "Mathematical formula detection in heterogeneous document images," in *Proc. Conf. Technol. Appl. Artif. Intell.*, Taipei, Taiwan, Dec. 2013, pp. 140–145.

- [26] K. L. Bouman, G. Abdollahian, M. Boutin, and E. J. Delp, "A low complexity sign detection and text localization method for mobile applications," *IEEE Trans. Multimedia*, vol. 13, no. 5, pp. 922–934, Oct. 2011.
- [27] W. Ohyama, M. Suzuki, and S. Uchida, "Detecting mathematical expressions in scientific document images using a U-net trained on a diverse dataset," *IEEE Access*, vol. 7, pp. 144030–144042, 2019.
- [28] B. H. Phong, T. M. Hoang, and T.-L. Le, "Mathematical variable detection based on convolutional neural network and support vector machine," in *Proc. Int. Conf. Multimedia Anal. Pattern Recognit. (MAPR)*, Ho Chi Minh City, Vietnam, May 2019, pp. 1–5.
- [29] 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)*, Cancun, Mexico, Dec. 2016, pp. 3246–3251.
- [30] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, 2015, pp. 234–241.
- [31] K. Iwatsuki, T. Sagara, T. Hara, and A. Aizawa, "Detecting in-line mathematical expressions in scientific documents," in *Proc. ACM Symp. Document Eng. (DocEng)*, Valletta, Malta, 2017, pp. 141–144.
- [32] B. H. Phong, A. Aizawa, T. M. Hoang, and T.-L. Le, "Mathematical variable detection in PDF scientific documents," in *Proc. Int. Conf. Intell. Inf. Database Syst.*, Yogyakarta, Indonesia, Apr. 2019, pp. 694–706.
- [33] L. Gao, X. Yi, Y. Liao, Z. Jiang, Z. Yan, and Z. Tang, "A deep learning-based formula detection method for PDF documents," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Kyoto, Japan, Nov. 2017, pp. 553–558.
- [34] A. Papandreou and B. Gatos, "A novel skew detection technique based on vertical projections," in *Proc. Int. Conf. Document Anal. Recognit.*, Sep. 2011, pp. 384–388.
- [35] T. Kil, W. Seo, H. I. Koo, and N. I. Cho, "Robust document image dewarping method using text-lines and line segments," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 865–870.
- [36] T.-Y. Chang, Y. Takiguchi, and M. Okada, "Physical structure segmentation with projection profile for mathematic formulae and graphics in academic paper images," in *Proc. 9th Int. Conf. Document Anal. Recognit. (ICDAR)*, vol. 2, Parana, Brazil, Sep. 2007, pp. 1193–1197.
- [37] B. H. Phong, T. M. Hoang, and T.-L. Le, "A new method for displayed mathematical expression detection based on FFT and SVM," in *Proc. 4th NAFOSTED Conf. Inf. Comput. Sci.*, Hanoi, Vietnam, Nov. 2017, pp. 90–96.
- [38] B. H. Phong, T. M. Hoang, and T.-L. Le, "Mathematical variable detection in scientific document images," *Int. J. Comput. Vis. Robot.*, to be published.
- [39] P. T. Noi and M. Kappas, "Comparison of random forest, K-nearest neighbor, and support vector machine classifiers for land cover classification using Sentinel-2 imagery," *Sensors*, vol. 18, no. 2, p. 18, 2017.
- [40] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *Ann. Statist.*, vol. 28, no. 2, pp. 337–407, 2000.
- [41] P. Napolitano, F. Piccoli, and R. Schettini, "Anomaly detection in nanofibrous materials by CNN-based self-similarity," *Sensors*, vol. 18, no. 2, p. 209, 2018.
- [42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [43] K. Murphy, *Machine Learning: A Probabilistic Perspective*, 1st ed. Cambridge, MA, USA: MIT Press, 2012.
- [44] S. L. Lee, M. R. Zare, and H. Muller, "Late fusion of deep learning and handcrafted visual features for biomedical image modality classification," *IET Image Process.*, vol. 13, no. 2, pp. 382–391, Feb. 2019.
- [45] A. G. S. de Herrera and H. M. Müller, "Fusion techniques in biomedical information retrieval," in *Fusion in Computer Vision*. Cham, Switzerland: Springer, Mar. 2014, pp. 209–228.
- [46] Z. Liu and R. Smith, "A simple equation region detector for printed document images in tesseract," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Washington, DC, USA, Aug. 2013, pp. 245–249.
- [47] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
- [48] 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)*, Sydney, NSW, Australia, Sep. 2019, pp. 1–6.
- [49] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.
- [50] W. S. Torgerson, "Multidimensional scaling: I. Theory and method," *Psychometrika*, vol. 17, no. 4, pp. 401–419, Dec. 1952.
- [51] C. K. I. Williams, "On a connection between Kernel PCA and metric multidimensional scaling," *Mach. Learn.*, vol. 46, nos. 1–3, pp. 11–19, 2002.



**BUI HAI PHONG** graduated from the School of Information and Communication Technology, Hanoi University of Science and Technology, Vietnam, in 2010. He received the M.S. degree in information technology from the Hanoi University of Science and Technology, in 2012, where he is currently pursuing the Ph.D. degree. His research interests include computer vision, pattern recognition, and machine learning.



**THANG MANH HOANG** received the B.Eng. degree in electronics and telecommunications and the M.Sc. degree from the Hanoi University of Science and Technology, Vietnam, and the Ph.D. degree in electronics and telecommunications from the Nagaoka University of Technology, Japan, in 2007. He is currently a Lecturer with the School of Electronics and Telecommunications, Hanoi University of Science and Technology. His research interests include non-linearity

and its applications in electronics and communications, such as cryptography, modulation, oscillation, complex networks, chaos synchronization, and recognition.



**THI-LAN LE** graduated in information technology from the Hanoi University of Science and Technology (HUST), Vietnam. She received the M.S. degree in signal processing and communication from HUST, and the Ph.D. degree in video retrieval from INRIA Sophia Antipolis, France, in 2009. She is currently a Lecturer/Researcher with the Computer Vision Department, HUST. Her research interests include computer vision, content-based indexing, and retrieval, video understanding, and human-robot interaction.

...