Mathematical variable detection in scientific document images

Bui Hai Phong*

MICA International Research Institute
(HUST – CNRS/UMI2954 – Grenoble INP),
Hanoi University of Science and Technology,
Hanoi, Vietnam
Email: hai-phong.bui@mica.edu.vn
and
Faculty of Information Technology,
Hanoi Architectural University,
Hanoi, Vietnam
*Corresponding author

Thang Manh Hoang

School of Electronics and Telecommunications,
Hanoi University of Science and Technology,
Hanoi, Vietnam
Email: thang.hoangmanh@hust.edu.vn

Thi-Lan Le

MICA International Research Institute
(HUST – CNRS/UMI2954 – Grenoble INP),
Hanoi University of Science and Technology,
Hanoi, Vietnam
Email: thi-lan.le@mica.edu.vn

Abstract: Mathematical expression detection in scientific documents is a prerequisite step for developing a mathematical retrieval system that has attracted many researches recently. In the detecting process, one challenging issue is the detection of variables. The similar properties of variables and narrative text cause many errors in the detection in existing approaches. In the paper, a novel detection methodology of variables in inline mathematical expressions is proposed. The merit of the method is that it can operate directly on the variable images without the employment of character recognition. The proposed method uses the features of projection profile of images and the fine-tuning of different machine learning algorithms in the detection process. The achieved accuracy varies from 86.14% to 94% for the detection of variables in inline expressions in document images in various public benchmark datasets. The performance comparison with existing methods demonstrates the effectiveness of the proposed method.

Keywords: document analysis; mathematical expression extraction; italic detection; machine learning.

Reference to this paper should be made as follows: Phong, B.H., Hoang, T.M. and Le, T-L. (2021) 'Mathematical variable detection in scientific document images', *Int. J. Computational Vision and Robotics*, Vol. 11, No. 1, pp.66–89.

Biographical notes: Bui Hai Phong is graduated in the School of Information and Communication Technology from Hanoi University of Science and Technology, Vietnam in 2010. He obtained his MS in Information Technology from the Hanoi University of Science and Technology in 2012. He is currently a PhD student of the Hanoi University of Science and Technology. His researches interests include computer vision, pattern recognition, and machine learning.

Thang Manh Hoang received his BEng in Electronics and Telecommunications from the Hanoi University of Science and Technology, Vietnam and MSc from the Hanoi University of Science and Technology. In 2007, he was awarded his PhD in Electronics and Telecommunications from the Nagaoka University of Technology, Japan. He is currently a Lecturer at the School of Electronics and Telecommunications, Hanoi University of Science and Technology, Vietnam. His research interests include nonlinearity and its applications in electronics and communication such as cryptography, modulation, oscillation, complex network, chaos synchronisation, and recognition.

Thi-Lan Le is graduated in the Information Technology from Hanoi University of Science and Technology (HUST), Vietnam. She obtained her MS in Signal Processing and Communication from the HUST, Vietnam. In 2009, she received her PhD at the INRIA Sophia Antipolis, France in video retrieval. She is currently a Lecturer/researcher at the Computer Vision Department, HUST, Vietnam. Her research interests include computer vision, content-based indexing and, retrieval, video understanding, and human-robot interaction.

1 Introduction

Mathematical expression recognition systems have been studied for decades in order to convert images of expressions into machine-readable format. The existing systems have obtained high accuracy for independent expressions but low accuracy for expressions contained in scientific documents. Therefore, extracting mathematical expressions from scientific documents is a key step to improve the performance of mathematical expression recognition systems.

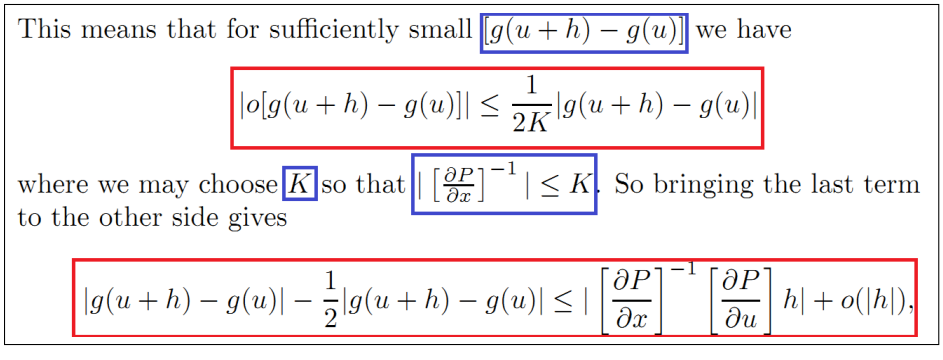
Based on the appearance of mathematical expressions relatively to text paragraphs in the documents, they are classified into two types: displayed and inline expression. A displayed expression (also, called isolated expression) displays in a separate line, while an inline expression (also, called embedded expression) is mixed with other components

(normally text) (Garain, 2009). Examples of displayed expression and inline expression are marked in red and blue in Figure 1, respectively. Several researches have obtained high accuracy in the detection of displayed expression in scientific document, however few researches focus on the detection of inline expression (Zanibbi and Blostein, 2012). Components of inline expressions can be classified into three main types: operator, variable and function. An operator is any symbol that indicates an operation to be performed (e.g., +, –, ∗, /). A variable is a quantity that may change within the context of mathematical problem or experiment. Typically, a variable is represented by a single character (Math Insight). A function can be an expression or a law that defines a relationship between variables (e.g., $y = f(x)$) (Math Insight). In the detection of the components of inline expressions, one critical issue that causes many errors is the detection of variables. In Figure 1, the alphabetical character 'K' is used as a variable of an inline expression. The properties of this kind of variable (font and size) are similar to narrative text. It causes many difficulties for the detection. Besides, some kinds of variable are represented in two-dimensional layout (e.g., A_j or x_i). Thus, the detection of various kinds of variables is a challenge.

Most of existing methods focus on the detection of special operators, symbols (e.g., '$\sum$', '$\int$' or 'β'), mathematical functions (e.g., cos, sin, log). Few methods have focused on the detection of variable. In this paper, we propose a novel method for variable detection in inline expression in scientific document images. The merit of the method is that it can perform directly on character and word images for the detection purpose without the use of character recognition as in existing methods. To achieve the purpose, a novel feature extraction of character and textual word images is firstly proposed. Then, different machine learning algorithms are employed and optimised to discriminate variables from textual words.

The rest of the paper is organised as follows. Section 2 overviews significant related works. Section 3 presents the detail of the proposed method. In Section 4, experimental results are shown and discussed. Finally, Section 5 gives the conclusion and the future work.

Figure 1 Examples of displayed expressions (in red) and inline expressions (in blue) in a sample scientific document (see online version for colours)



2 Related works

This section aims at analysing the works related to major steps of mathematical variable detection from scientific document images: document layout analysis, mathematical expression detection and italic font detection.

2.1 Document layout analysis

Document layout analysis is a prerequisite step for detection and recognition of mathematical expressions. In general, the document layout analysis aims to decompose the entire document image into homogeneous regions. In the step, the image pre-processing (noise removal and skew correction) is firstly performed. Then, each component (e.g., text, figure, table) is separated based on their structure layout. Document layout analysis techniques can be divided into four types: top-down, bottom-up, hybrid and multi-scale resolution method (Tran et al., 2016). Top-down methods split the entire page into smaller components (Wahl et al., 1982; Wang and Srihari, 1989). In general, top-down methods are useful in the segmentation of rectangular layout. However, the methods are not effective for the complex structure document. Bottom-up methods analyse and merge local pixels in order to form larger components such as characters, words, text lines and paragraphs (Caponetti et al., 2008; Agrawal and Doermann, 2009). Comparing with top-down methods, bottom-up methods show higher performance in page segmentation. However, the methods have high computation complexity. The multi-scale resolution methods analyse page structure based on the features of different resolution levels of the document image (Cheng and Bouman, 2001; Shi and Govindaraju, 2005). The difficulty of the methods is the estimation of distance parameters between components of document page and noises can be generated when the document size is changed. Hybrid methods combine the bottom-up and top-down techniques. The methods are effective for the segmentation of complex structure document (Tran et al., 2016; Ha et al., 2016). In the methods, connected components and delimiters (white-spaces, tap-stops) in document page are extracted, filtered and analysed. After that, various heuristic strategies are proposed to reduce page segmentation errors. For mathematical expression detection purpose, text regions in the body of document are focused to obtain. Text regions are 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 document, there is not much variation in text lines. Therefore, the text line segmentation achieves high accuracy (Breuel, 2008; Smith, 2007; Suzuki et al., 2003). In contrast, there is frequent variation of height, distance in text lines in scientific documents that contain mathematical notations. This issue causes many errors for the text line segmentation. 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 (Anoop and Anil, 2007; Lin et al., 2013). The basic idea of the techniques is that all text lines are firstly split, then consecutive text lines are merged to form the entire expression if they are components of the mathematical expression. Similarly, consecutive words are merged in order to form the entire expression if they belong to the expression.

2.2 Mathematical expression detection

Mathematical expression detection has been studied for more than 20 years (Lin et al., 2014). Methods for detecting displayed (isolated) expression have obtained robust and accurate results. However, the detection of inline expression remains a challenge (Zanibbi and Blostein, 2012). The early research on the mathematical expression detection that is reported in Lee and Wang (1998) scans all text lines from left to right to get primitive tokens. After that, each token is determined whether it belongs to an inline expression by checking predefined expression forms. However, the accuracy of detection is not reported in this research. Research in Jin et al. (2003) concluded that it is difficult to detect all inline expressions without using character recognition results.

The method reported in Garain (2009) employs results of two commercial optical character recognition (OCR) systems to extract inline formula. Firstly, existing OCR systems are applied to obtain content of document images. After that, 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 expression. The features of words mentioned in the work are:

1 The probability of a sentence containing inline expression.

2 The confidence of OCR systems while recognising the word.

3 The type style (italic, bold) of the word.

4 The space between characters of the word.

5 The variation of position of characters of the word.

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

The work in Chu and Liu (2013) firstly applies text line and word segmentation techniques for document images. Then, features of each word are extracted and support vector machine (SVM) is used to determine if the word belongs to inline expressions. The features mentioned in the work are:

1 The density of black pixel in the word image.

2 The proportion of height of the word and height of the whole document.

3 The variation of ‘centroid’ of characters in the word.

The features are effective in the detection of special symbols. However, these features are not accurate in the detection of alphabetical characters that are used as variables.

Recently, *convolutional neural network* (CNN) has emerged as a powerful technology for object recognition and particularly for mathematical expression recognition. By using this framework, multitasks in recognition are performed simultaneously, human effort for hand-crafted feature extraction is reduced. The CNNs are designed to automatically learn dominant features of images. Thus, it is not necessary to use specific feature extraction as in traditional learning-based methods. The work Wenhao et al. (2016) adopts sliding window-based CNN framework and obtains accuracy of 87% for different mathematical symbols recognition. In this

research, isolated mathematical expression images are captured by mobile phone, inline expression handling is not in the scope of the work. The work in Bui et al. (2019) uses the transfer learning of pre-trained CNNs that are Alexnet and Resnet-50 to improve the accuracy of the detection of variable in scientific document images. The CNNs are trained on a large image dataset. After that these CNNs are optimised to handle the classification task. The accuracy in the detection is much improved. However, the training process of the CNNs is time-consuming.

Scientific documents are available in both image and PDF format. Therefore, in recent years, some researches (Lin et al., 2014; Iwatsuki et al., 2017) have focused on the detection of mathematical expression in PDF documents. For PDF documents, metadata information of textual words such as font, size, type styles can be extracted precisely. Therefore, the detection of mathematical expression in PDF documents is more accurate than that of image-based documents. The method reported in Iwatsuki et al. (2017) extracts inline expression in PDF documents. The work applied natural language processing for the detection purpose. After the word extraction process, word features and conditional random field (CRF) are used for inline expression detection. The achieved precision in detection is 88.95% on PDF files from ACL Anthology dataset (Steven et al., 2008) and the detection of variables is one issue causing many errors in the research. The research (Gao et al., 2017) attempts to detect mathematical expressions from input PDF documents. The framework for the detection consists of two steps. In the first step, the candidate regions for mathematical expressions are generated. For the generation of candidate regions, metadata information including positions, 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 the step, two deep networks 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 used to extract visual feature of image and the RNN is used to extract sequential information of characters. After that, the features are combined 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 mathematical expression detection in general and inline expression detection in particular. However, they focused mainly on mathematical symbols and functions detection, the issue of detection of variable is not taken into consideration. In this paper, a fast and efficient approach is proposed for the detection.

2.3 Italic font detection

Typically, text in italic font is used to emphasise the important content in documents. Italic font detection has been attracted many researches in document analysis and OCR fields. Almost researches have focused on the detection of italic font of text paragraphs, sentences or long words (Sun and Si, 1997; Garain, 1998; Zhang et al., 2004; Fan et al., 2007). These methods heavily rely on computing the slant angle with the vertical axis of images. Then, a certain threshold is used to determine the italic properties of word images. In the method reported in Sun and Si (1997), skew angle is calculated based on the gradient orientation histogram of italic characters' images. The method in Garain (1998) directly computes the slant angles of lines in character images with vertical axis to determine the skew angles. The work in Zhang et al. (2004) firstly extracts dominant

stroke patterns of word images by applying the 2D discrete wavelet decomposition. Then, the statistical analysis of the extracted stroke patterns is performed. After that, the vertical and diagonal strokes are analysed to find out the distinctive features that discriminate the italic and non-italic properties of word images. The method sets some thresholds to determine if a word image is italic or not. The work in Harjit (2012) proposes the method to detect italic of characters in Indian language scripts. The method firstly analyses the characteristics of these scripts. Then, the italic style of characters is detected by the comparison of the number of black pixels in headline and vertical line of characters. The work is not efficient in the detection of italic style of some specific Indian characters. In general, the existing methods show high accuracy in the detection of the italic font for long words. However, these methods expose weaknesses for short words or some specific characters (e.g., 'x', 'w' characters). Moreover, the heuristic thresholds are predefined for the detection. In our work, the proposed feature extraction is aimed to discriminate variables from textual words in an efficient way. Variable displays in italic font in observed scientific documents. The proposed method can be a baseline for the detection of italic style of English characters and words.

3 Proposed method

Figure 3 shows the flowchart of the proposed framework for mathematical expression detection in document images. The proposed framework consists of two main modules: displayed expression detection and variable detection. The displayed expression module employs the fast Fourier transformation (FFT) (Buijs et al., 1974) as a feature extractor and SVM (Joachims, 2002) as a classifier. In the module, text line image is transformed from the spatial to frequency domain by using the FFT. The dominant features of mathematical symbols are obtained by using the transformation. Both FFT magnitude and phase values are used as features. Then, the extracted features are used for training the SVM model. SVM is a popular supervised machine learning model and plays an efficient role in object classification task. In our work, the model is trained for the classification of two classes: displayed expression and normal text line. It is worth to note that the displayed expression detection has been presented in the work (Bui et al., 2017). The main contribution of this paper is the method for variable detection in inline mathematical expressions. The module is marked in blue in the Figure 3. The proposed framework is briefly described as follows: to detect the variable in document images, they are firstly pre-processed and normalised. Then, the layout techniques and text line segmentation are applied in order to obtain a set of text lines in the documents. The text lines are then classified into displayed mathematical expression and normal text lines. Finally, the normal text lines are segmented into pure words and variables. Figure 2 illustrates an example input document and the text line extracted from this document.

3.1 Projection profile of image

In order to discriminate variables and words efficiently, the projection profile (Papandreou and Gatos, 2011) of image is computed, then the feature of the projection profile is extracted. Taking the advantage of projection profile, the structure (from left to right and top to bottom) of scanned document is extracted and emphasised. Moreover, the analysis of the projection profile of image is aimed to obtain the trade-off

between the accuracy in the variable detection and time efficiency. The mathematical variable displays in both sequential and two-dimensional layout. Therefore, both vertical projection profile (VPP) and horizontal projection profile (HPP) are necessary to analysed. Vertical and horizontal projection profile of an image is the sum of black pixels along vertical and horizontal scan-lines of the image, respectively. Given an image a with the size of $M \times N$, the intensity of element at m^{th} row and n^{th} column is defined as $f(m,n)$. In the work, the text line images are binary, therefore the value of $f(m,n)$ is either 0 or 1. VPP of the image is computed as follows:

$$VPP(n) = \sum_{m} f(m,n) \tag{1}$$

Similarly, HPP of the image is computed as follows:

$$HPP(m) = \sum_{n} f(m,n) \tag{2}$$

Figure 2 Example of a document image and a text line which is extracted from the document, (a) input document image (b) extracted text line

P2 : $\texttt{request}(A,B,X) \rightarrow \texttt{accept}(B,A,X) \rightarrow \texttt{do}(B,X)$

P1' : $\texttt{request}(A,B,X) \rightarrow \texttt{accept}(B,A,X) \rightarrow \bot$

These represent two possible runs in a conversation in which A wants B to perform some physical action X. In the first case, B executes the promised action X, in the

(a)

These represent two possible runs in a conversation in which A wants B to perform

(b)

3.2 Data preparation

Document images may contain various components such as figures, tables, text regions. In our work, the hybrid document analysis technique is applied for document images to obtain text regions. Then, text line segmentation technique is applied for text regions to obtain text lines. The segmentation technique is applied based on the estimation of white-space between consecutive text lines. The displayed detection module is designed to detect displayed expressions.

Text lines that are not identified as displayed expressions are split into words. VPP is computed for each text line in order to segment the words. The VPP of an extracted text line from a document image is illustrated in Figure 4. Actually, the white-space between consecutive words in a text line is larger than that of inner characters in a word. The white-space between words is illustrated as valley in Figure 4. In the Figure, the X-axis represents the positions (in pixels) of characters in the text line horizontally and the Y-axis represents the sum of black pixels of each column of the text line image. The words of each text line are separated by using a certain threshold of white-space (in pixels) of the VPP. The words considered in this work are either textual words or variables.

Figure 3 Overview of displayed expression and variable detection modules (see online version for colours)

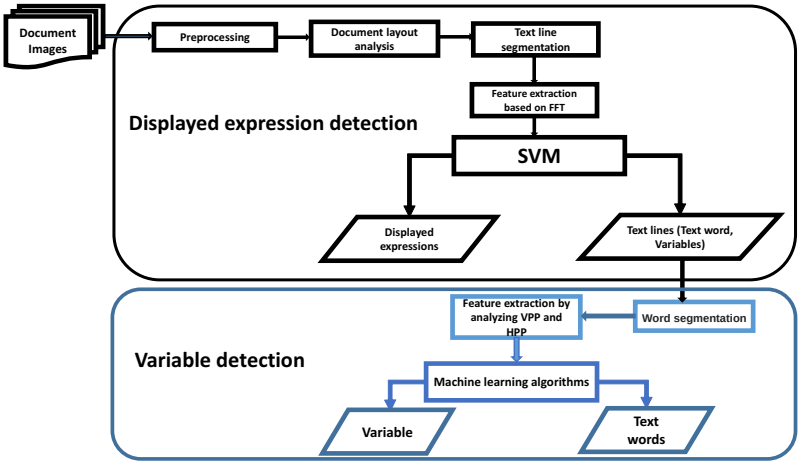
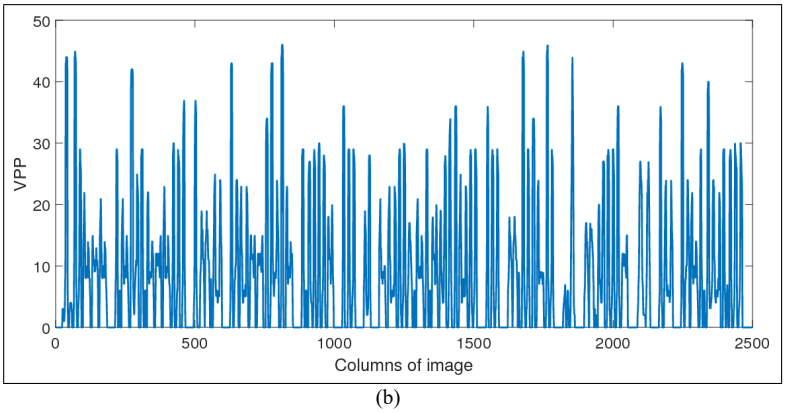


Figure 4 (a) Examples of a text line (b) VPP of the text line (see online version for colours)

These represent two possible runs in a conversation in which A wants B to perform

(a)



(b)

3.3 Feature extraction

In order to determine an extracted word from a text line is a variable or a textual word, a binary classification method is proposed. A key step in the classification is to extract the dominant features of observed words. The important feature that is used to discriminate variables from textual words is the italic font style of images. In scientific documents, variable is typically represented in italic font.

For feature extraction, firstly VPP and HPP of each variable or textual word image is computed. Then, peaks and valleys (troughs) of the VPP and HPP are determined. In mathematical definition, peaks and valleys are local maxima and minima, respectively, (Blatter, 1977) that are defined as follows.

Let $f(x)$ be a function which transforms a variable x from a subdomain $A \subset R$ to the domain R as: $f : A \to R$

Let $I = (a, b)$ be an interval that belongs to the subdomain A. A point $x_0 \in I$ is defined as the local maximum if: $f(x_0) \geq f(x), \forall x \in I$.

Similarly, a point $x_0 \in I$ is defined as the local minimum if: $f(x_0) \leq f(x), \forall x \in I$.

In our work, peaks and valleys are determined as the highest and the lowest points respectively among neighbouring points.

The feature vector is formed by features of projection profile of variable and textual word images. For each image, these features are described as follows:

1 The number of peaks in the VPP and HPP.

2 The mean (average) of values of peaks in the VPP and HPP.

Let P be a vector that is made up by N peak values (P_i), the mean is defined as follows:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} P_i \tag{3}$$

3 The standard deviation of values of peaks in the VPP and HPP.

Let P be a vector that is made up by N peak values (P_i) and μ is the mean of the elements of vector P. The standard deviation is defined as follows:

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (P_i - \mu)^2} \tag{4}$$

4 The number of valleys in the VPP and HPP.

5 The mean (average) of values of valleys in the VPP and HPP.

6 The standard deviation of values of valleys in the VPP and HPP.

For instance, the peaks and valleys of VPP of word images are illustrated in red and green, respectively, in Figure 5. It is clear that the number of peaks and valleys of word 'and' is more than that of variable 'a'. The difference between the features of VPP of character 'a' in italic and non-italic styles in Arial font is shown in Figure 6(b) and Figure 6(d), respectively. For the character 'a' in Arial font with italic and non-italic style, the feature of VPP of the image is different. In Figure 5 and Figure 6, the X-axis represents the positions (in pixels) of each column and the Y-axis represents the sum of black pixels of each column of the image. The detail values of the proposed feature of VPP of variable and textual word images are described in Table 1. The detail values of the extracted feature of VPP of character 'a' in italic and non-italic styles in Arial font are described in Table 2. For variable that displays in two-dimensional layout, in our work, the feature in both VPP and HPP is extracted in order to improve the classification accuracy. For instance, bounding boxes of characters of variable a_i, peaks and valleys in VPP and HPP of the variable are shown in Figures 7(a), 7(b) and 7(c), respectively. In the figures, the X-axis represents the positions (in pixels) of each column/row and the Y-axis represents the sum of black pixels of each column/row of the image.

Table 1 Features of VPP of variable and word images in Figure 5

Features	*Variable 'a in Figure 5(a)*	*Word 'and' in Figure5(c)*
Number of peaks	3	7
Mean of peaks' values	18	18
Standard deviation of peaks' values	11.36	7.44
Number of valleys	2	6
Mean of valleys' values	4	2.2 0
Standard deviation of valleys' values	0	1.83

Table 2 Comparison of VPP features between italic and non-italic styles of character 'a' of Arial font

Features	*Character 'a' in italic style*	*Character 'a' in non-italic style*
Number of peaks	3	3
Mean of peaks' values	18.33	19.33
Standard deviation of peaks' values	9.29	7.02
Number of valleys	2	2
Mean of valleys' values	10.50	10
Standard deviation of valleys' values	0.70	0

The feature extraction of VPP and HPP is proposed by the observation that the peaks and valleys of projection profile of characters approximately obey the Gaussian (normal) distribution (Ghahramani, 2000; Roger, 2008). For a peak P_i in vector P, the probability density function of the corresponding Gaussian distribution is described as follows:

$$f(P_i|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(P_i-\mu)^2}{2\sigma^2}} \tag{5}$$

The first parameter μ is the average value of elements in the vector P and the second parameter σ is the standard deviation of the values. The features (2) and (3) are proposed by using the parameters of the distribution of peaks. The features (5) and (6) are proposed by using the parameters of the distribution of valleys.

3.4 Classification

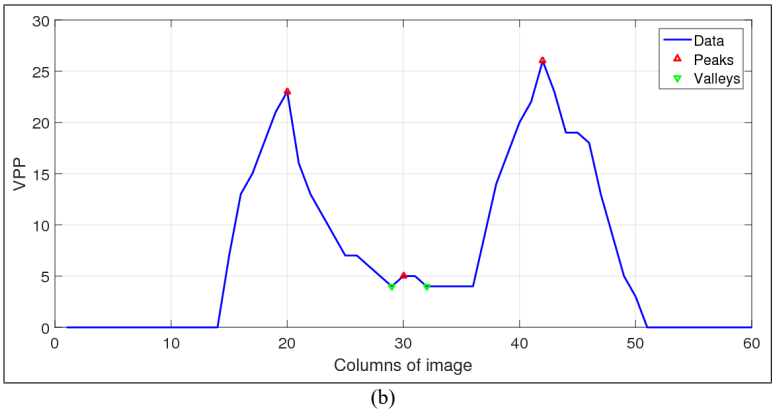
After feature extraction process, in order to discriminate variables from textual words, different machine-learning-based algorithms are used in the work. In the section, different classification models are applied: SVM (Joachims, 2002), k-nearest neighbour (kNN) (Friedman and Finke, 1977), decision tree (Breiman and Stone, 1984) and random forest (RF) (Breiman, 2001). For the classifiers, tuned parameters play an important role to achieve high performance. Therefore, different parameters of the each classifier are considered in order to determine the optimal values for the classification.

SVM is a supervised classifier model that is popular in classification, regression problems. Several researches on the classification of mathematical expressions have employed the model (Lin et al., 2012). The model separates data into groups by

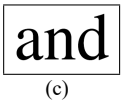
maximising the margins between these groups. In the work, linear and radial basis function (RBF) kernels of the model are tested for the classification.

Figure 5 Examples of peaks and valleys of VPP of variable and word images, (a) image of variable '*a*' (b) peaks and valleys of VPP of the variable image (c) image of word 'and' (d) peaks and valleys of VPP of the word image (see online version for colours)

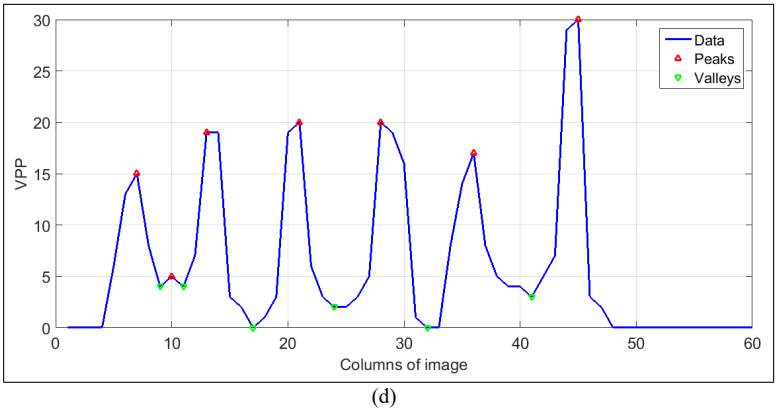
(a)



(b)

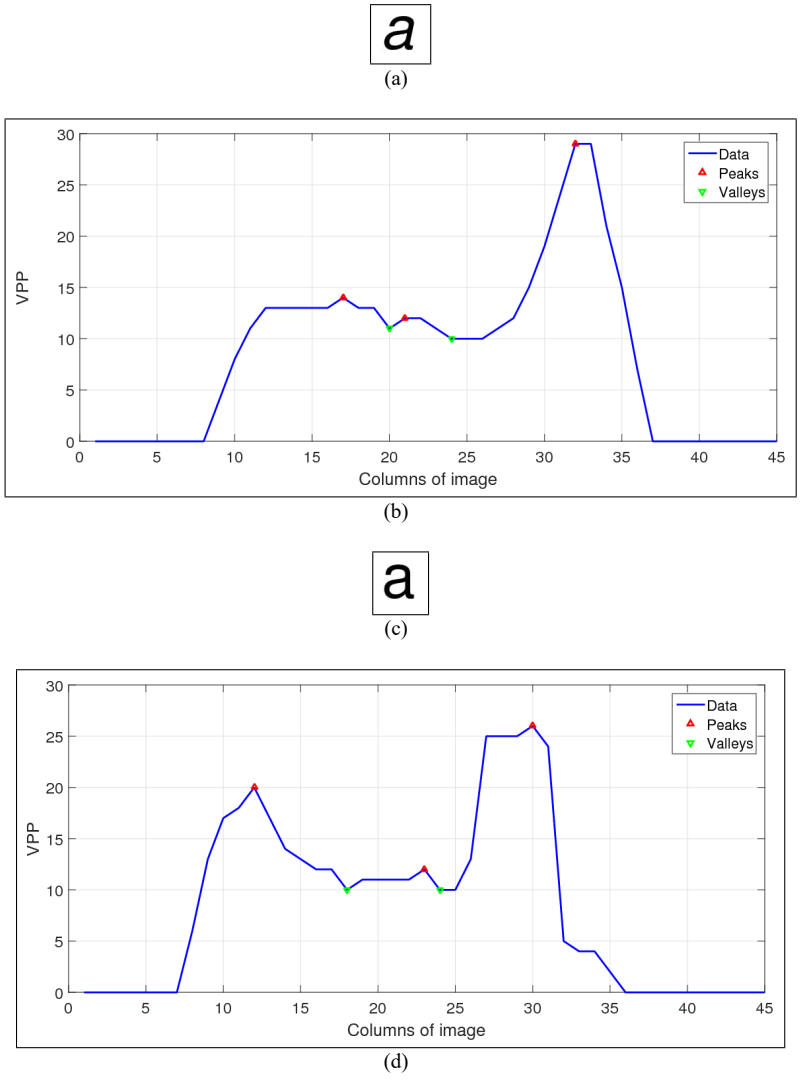


(c)



(d)

Figure 6 (a) Image of character 'a' in italic style in Arial font (b) Peaks and valleys of VPP of character 'a' in italic style in Arial font (c) Image of character 'a' in non-italic style in Arial font (d) Peaks and valleys of VPP of character 'a' in non-italic style in Arial font (see online version for colours)

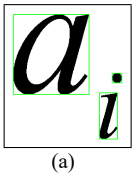


Notes: Image of character 'a' in italic (a) and non-italic (c) styles in Arial font. Peaks and valleys of VPP of the character in italic (b) and non-italic (d) styles.

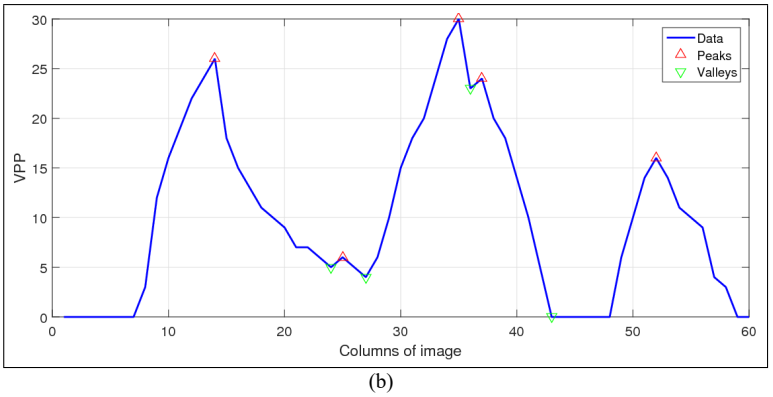
kNN is a popular and powerful prototype method in pattern recognition. The method is memory-based and does not require fitting models as other learning algorithms. In the method, each class of data is represented by several samples. A new sample is classified in the class based on the distances to k nearest neighbour prototypes. Although, the principle of the method is simple, it shows high performance in the

classification problem. Especially, the method is very powerful for the cases that the decision boundary is irregular. There are several functions for the metric of the distance such as Euclidean, Minkowski. In the work, the Euclidean distance is used to identify the nearest samples in the training dataset. Besides, the value of k affects so much on the accuracy. In the work, different values of k (these values are 1, 3, 6, 9, 12, 15, 18, 21) are tested to find the optimal parameter.

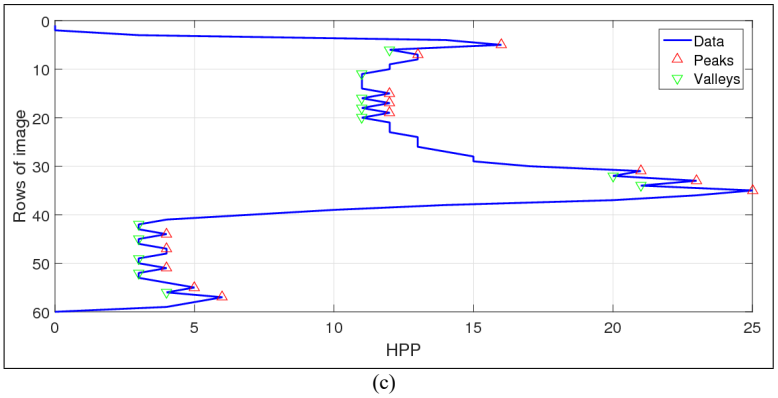
Figure 7 (a) Example of a variable with index (b) Peaks and valleys of VPP of the variable (c) Peaks and valleys of HPP of the variable (see online version for colours)



(a)



(b)



(c)

Notes: Bounding boxes of characters of variable a_i (a), peaks and valleys of VPP (b) and HPP (c) of the variable.

Decision tree is original from the tree data structure that contains root and child nodes. The algorithm classifies data items by posing a series of questions about the features

of the items. Each question is contained in a tree node. Every internal node points to one child node for each possible answer to its question. The tree is formed by the predefined data. The tree starts from the root nodes and traverses to all child nodes from left to right. The testing samples are split into categories upon features. The tree is effective in the representation of human-readable rules of classification, however, the model becomes costly in time and memory resources when classifying a large data.

RF has been early mentioned since 1990. In the classification method, an ensemble of decision trees are combined. Therefore, RF has shown the high performance in classification and overcome over-fitting problems. In the method, every decision tree is formed by randomly selecting data from the input data. The number of trees in RF is a significant parameter that affects the accuracy of the classification. In the work, several values of the parameter (10, 50, 200, 500 and 1,000) are tested to find out the optimal parameter.

For each model, training data of variable and textual word images and labels are prepared manually. Then, the training data and labels are fed into classifier models. Finally, the trained model is used for the classification of testing data. The flowchart of the classification is shown in Figure 8.

The overall classification process of variables and textual words is executed by the pseudocode in Algorithm 1.

Algorithm 1 Algorithm of detecting variable from document images

procedure variable detection
  Input: Word Images
  Output: Classification Results
  Training Feature Vector: $F = \emptyset$;
  Label Vector: $L = \emptyset$;
  Testing Feature Vector: $T = \emptyset$;
    ▷ Features of training images are extracted by analysing VPP and HPP
  for each *image(i)* in *training dataset of word images* **do**
    VPP ← *ComputeVPP(image(i))*;
    HPP ← *ComputeHPP(image(i))*;
    $F(i)$ ← *FeatureExtraction(VPP,HPP)*;
  end for
    ▷ Variable and word images are labelled for the classification
  for each *image(i)* in *training dataset of word images* **do**
    $L(i)$ ← *LabelImage(image(i))*;
  end for
    ▷ Features of testing images are extracted by analysing VPP and HPP
  for each *image(i)* in *testing dataset of word images* **do**
    VPP ← *ComputeVPP(image(i))*;
    HPP ← *ComputeHPP(image(i))*;
    $T(i)$ ← *FeatureExtraction(VPP,HPP)*;
  end for
    ▷ Feature and label vectors are used for training a classification model
  TrainedModel ← *TrainModel(F,L)*;
    ▷ Trained model is used to discriminate variables from textual words
  ClassificationResults ← *Classify(TrainedModel,T)*
  Return *Classification Results*
end procedure

Figure 8 Flowchart of the classification of variable and word images (see online version for colours)

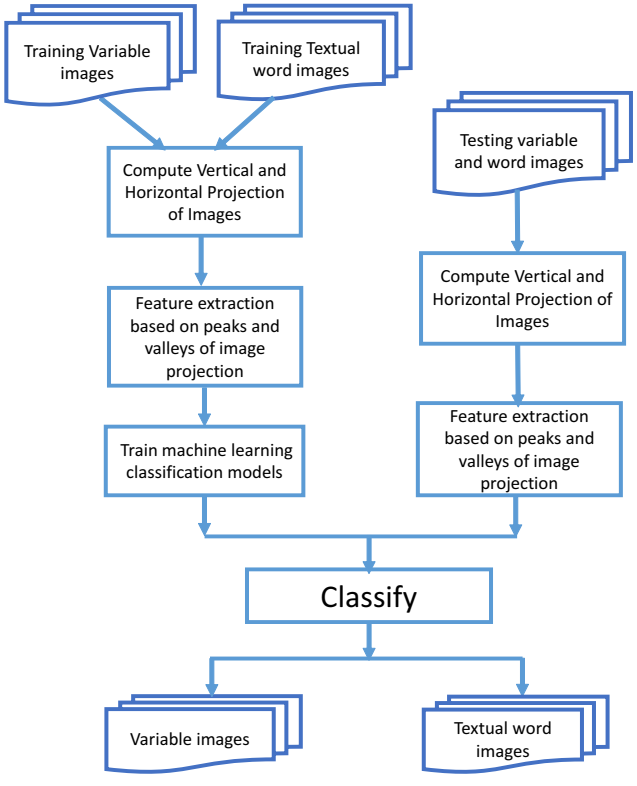
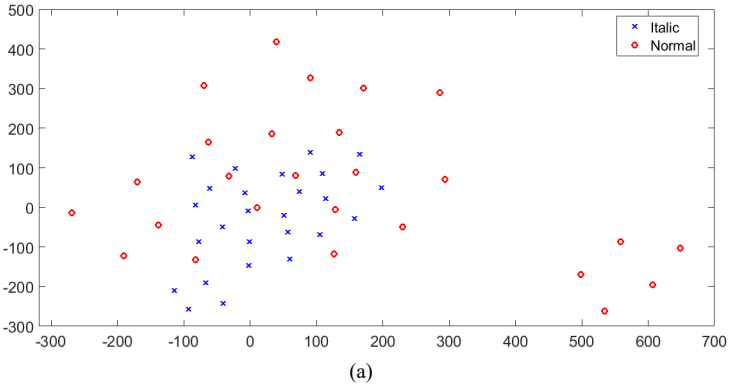
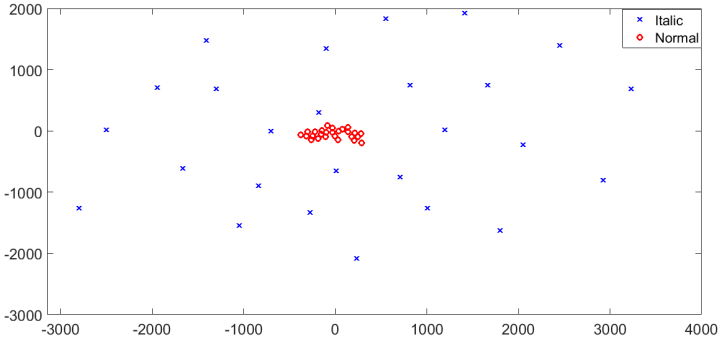


Figure 9 Feature distribution of each English character in italic and non-italic styles in, (a) feature distribution of each English character in italic and non-italic styles in lower-case (b) feature distribution of each English character in italic and non-italic styles in upper-case in Arial font (see online version for colours)



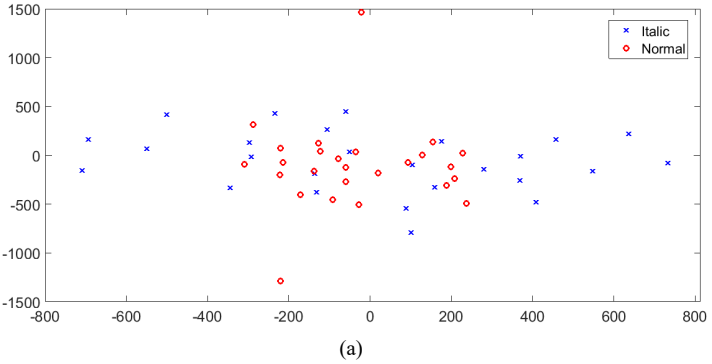
(a)

Figure 9 Feature distribution of each English character in italic and non-italic styles in, (a) feature distribution of each English character in italic and non-italic styles in lower-case (b) feature distribution of each English character in italic and non-italic styles in upper-case in Arial font (continued) (see online version for colours)

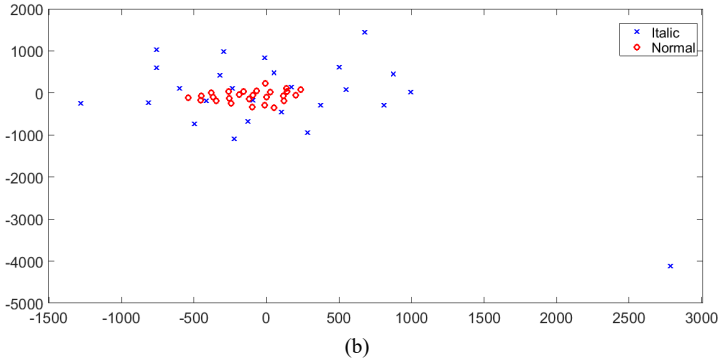


(b)

Figure 10 Feature distribution of each English character in italic and non-italic styles in, (a) feature distribution of each English character in italic and non-italic styles in lower-case (b) feature distribution of each English character in italic and non-italic styles in upper-case in Times New Roman font (see online version for colours)



(a)



(b)

4 Evaluation result

4.1 Benchmark datasets

In this paper, input data for detecting of variable is text lines containing textual words and variables. The document images are extracted from Marmot (Lin et al., 2012) dataset. The dataset contains 400 scientific documents with 1,575 displayed and 7,907 inline expressions that are crawled from Citeseer website. The documents are provided in PDF and PNG formats. The document image is generated with high resolution (500 dpi).

4.2 Experimental results

In this section, experiments have been carried out on the Marmot (Lin et al., 2012) datasets. For the detection of variables from textual words, the italic style and the length of words are two important features. Therefore, the proposed features of English characters that are displayed in normal and italic styles are firstly illustrated in the testing. Then, the testing of the classification with different parameters of machine learning algorithms is performed to determine optimal parameters. Finally, the performance evaluation of the discrimination of variables from textual words is presented. The classification results of the proposed method are compared with the existing methods (Sun and Si, 1997; Zhang et al., 2004). The executing time is also evaluated to determine the performance of each method.

4.2.1 Testing of feature extraction on clear character images

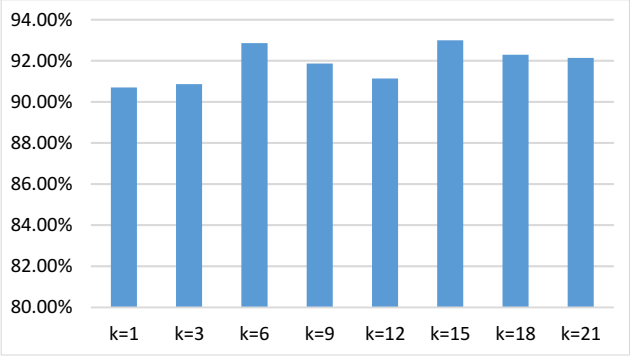
In order to test the effectiveness of feature extraction, clear images of English characters in Times New Roman and Arial fonts are generated by computer. The dataset consists of 26 characters in lower-case (from ‘a’ to ‘z’) and upper-case (from ‘A’ to ‘Z’) in italic and normal styles for each font. For simplicity, these characters are generated by text editor. The font and style properties are selected by using the text editor. The characters are converted to eight-bit grey-scale image. The size of image is normalised to 45×45 pixels. Figures 9 and 10 show the feature distribution of each English character in Arial and Times New Roman fonts, respectively. The feature distribution is represented by using the dimensional reduction technique called principal component analysis (PCA) (Jolliffe, 2002). The technique allows to visualise many features in the two-dimensional view. The feature distribution indicates that the italic and non-italic styles can be clearly detected by using the proposed features. Actually, the distance in the feature distribution reflects the possibility of the italic style detection. The feature distribution indicates that the detection of italic style of characters in Arial font is more accurate than that of Times New Roman font.

4.2.2 Testing of optimal parameter of machine learning algorithms

For the kNN classifier, k value plays an important role in the performance. In our testing, different values of k have been tested to find out the optimal value. Figure 11 shows the relationship between the classification precision (y-axis) and k value (x-axis).

The precision increases sharply until $k = 15$ and after that it decreases slightly. Therefore, the optimal value is selected at $k = 15$ with the highest precision is 93%.

Figure 11 The relationship between the classification precision (y-axis) and k value (x-axis) of the kNN classifier (see online version for colours)



For SVM, the linear and RBF kernel are examined. The obtained precision is 81.43% and 86.14% respectively. It indicates that the RBF outperforms linear kernel.

For RF classifier, the number of trees significantly affects the performance. In our work, the number of trees is tested as 10, 50, 200, 500, 1,000 in order to determine the optimal value. Figure 12 shows the relationship between the out-of-bag (OOB) error and the number of trees in the RF. The OOB is used for estimating the prediction error of RF. The OOB decreases sharply when the number of trees increases from 1 to 100. When the number of trees increases to 200, the OOB slightly fluctuates. The obtained prediction is highest (94%) when the number of trees is 200. Therefore, the optimal value for the number of trees of RF is selected at 200.

Figure 12 The relationship between the classification error (y-axis) and the number of trees of RF (x-axis) (see online version for colours)

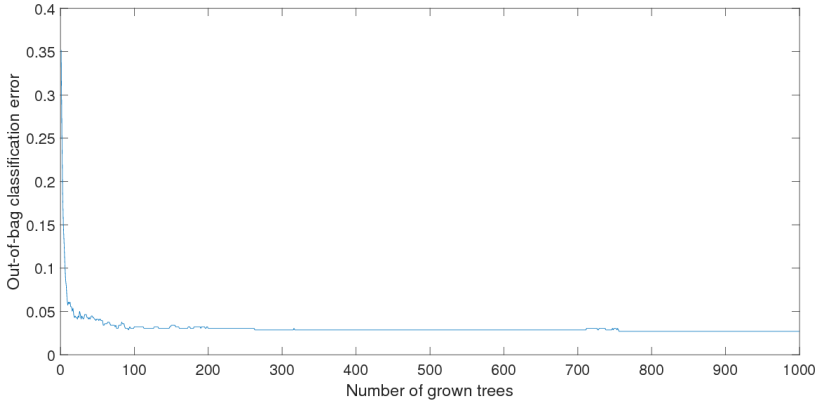


Table 3 The classification results by using different machine learning algorithms

Machine learning algorithms	*Precision*	*Recall*	*F1 score*
kNN	93.00%	86.04%	89.38%
SVM	86.14%	76.34%	80.94%
Decision tree	88.71%	77.75%	82.87%
Random forest	94.00%	87.97%	90.89%

4.2.3 Performance evaluation of discrimination of variable from textual word

To test the performance of the proposed method, the implementation of feature extraction and the binary classification of variable and textual word is carried out. 340 images of each type of variable and textual word are used for training machine learning algorithms. 800 variable and textual images are used for testing the algorithms.

In our work, the precision (P), recall (R) and F1 score are used for the performance evaluation. Precision is the proportion of the true positives against all the positive results; recall is the proportion of the true positives against all the true results and F1 score is the harmonic mean of precision and recall.

The obtained accuracy rates in the classification by using kNN, SVM, decision tree and RF are shown in Table 3. The RF classifier demonstrates the highest accuracy in the classification. In contrast, SVM shows the lowest accuracy.

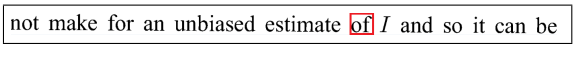
The overall performance comparisons between the proposed method and the existing methods in Sun and Si (1997) and Zhang et al. (2004) are shown in Table 4. The method in Sun and Si (1997) computes the orientation of the gradient of italic characters image to identify the slant angle with vertical axis. In fact, the method is not efficient for the small slant angles. In our method, the statistical features are proposed to discriminate variable from textual word. Therefore, the obtained performance of our proposed method is higher than that of method in Sun and Si (1997). The method in Zhang et al. (2004) applies the discrete wavelet transformation (DWT) level 2 for italic and non-italic word images. The method focuses on the stroke analysis of characters. Therefore, the existing methods are not efficient for short words in different fonts and for variables that display in two-dimensional layout.

Table 4 Performance comparison of proposed and existing methods

Methods	*Precision*	*Recall*	*F1 score*
Method using orientation of gradient (Sun and Si, 1997)	86.38%	76.81%	81.31%
Method using DWT (Zhang et al., 2004)	92.63%	83.45%	87.80%
Proposed method	*94.00%*	*87.97%*	*90.89%*

Notes: Ital value indicates the highest scores of the proposed method.

Figure 13 Example of an error that a textual word (in red) is misrecognised as a variable (see online version for colours)



4.2.4 Time efficiency

In order to compare the executing time of the methods, the execution time of testing phase of methods are evaluated. The methods are implemented in MATLAB R2018a environment on a computer with 6 GB RAM and Core i3-2.67 GHz processor. The execution time of the methods are shown in Table 5.

Table 5 Execution time of methods in the classification of variable and text in testing dataset

Methods	*Execution time (seconds)*
Method using orientation of gradient in Sun and Si (1997)	37.52
Method using DWT in Zhang et al. (2004)	6.95
Proposed method	*4.48*

Notes: Ital value indicates the highest performance of the proposed method.

The proposed method achieves the highest performance in time efficiency for the discrimination of variables from word images. The computation, feature extraction and classification of profile projection of image are faster than those of methods using orientation of gradient in Sun and Si (1997) and DWT in Zhang et al. (2004). The better performance of the proposed method is obtained because only discriminant values of VPP and HPP of images are used in the classification.

4.3 Error analysis and discussion

In this section, some errors in the variable detection are shown and analysed. In reality, the proposed feature extraction depends on the clearness of testing datasets. For clear images, the number and the values of peaks and valleys of VPP and HPP are persistent. Therefore, the proposed features overcome the ambiguities between variable and textual words. For noisy data, the number and the values of peaks and valleys of VPP and HPP of images can be affected. It can cause the misclassification (e.g., some variables such as i, x are possibly misrecognised as textual words). Moreover, short words such as '*is*' or '*of*' may cause more errors (misclassification as variables) than long words. One error of the misclassification is shown in Figure 13. In figure, the word '*of*' (in red) is misrecognised as a variable.

The proposed method shows better results in the classification compared with existing methods in Sun and Si (1997) and Zhang et al. (2004). Moreover, the method outperforms in the executing time. Our proposed method for variable detection uses the statistical analysis, therefore, existing methods that use the geometric features (Chu and Liu, 2013) or OCR techniques (Garain, 2009) are not mentioned for the performance comparison. In general, these methods require much time and resources.

5 Conclusions and future work

We have presented the method for detecting variables in inline expressions. In the work, novel features of projection profile of variable and text images are proposed. The feature extraction is combined with the optimisation of various machine learning algorithms including kNN, SVM, Decision Tree and RF that allow to detect accurately variables

in scientific document images. The experimental results with the classification precision that varies from 86.14% to 94% on public benchmark datasets show the effectiveness of the method. Comparing to the traditional methods, the proposed method is more powerful to discriminate variable from textual word.

In the future, the proposed feature extraction can be developed for other applications in the document analysis domain such as: character recognition, type styles extraction of characters. Moreover, the trend of storing scientific papers changes to other formats (e.g., PDF, XML). Therefore, the method can be considered to be suitable for such formats to respond to the new trend.

Acknowledgements

This work was supported by the Domestic Master/PhD Scholarship Programme of Vingroup Innovation Foundation.

References

Agrawal, M. and Doermann, D. (2009) 'Voronoi++: a dynamic page segmentation approach based on Voronoi and Docstrum features', *International Conference on Document Analysis and Recognition*.

Anoop, M.N. and Anil, K.J. (2007) 'Document structure and layout analysis', in B.B. Chaudhuri (Ed.): *Digital Document Processing*, Advances in Pattern Recognition, Springer, London, pp.29–48.

Blatter, C. (1977) 'Analysis 1', *Heidelberger Taschenb Cher (Book 151)*, Springer.

Breiman, L. and Stone, C. (1984) *Classification and Regression Trees*, CRC Press, Boca Raton, FL.

Breiman, L. (2001) 'Random forests', *Machine Learning*, Vol. 45, No. 1, pp.5–32.

Breuel, T.M. (2008) *The OCRopus Open Source OCR System*, Document Recognition and Retrieval XV, 68150F.

Bui, H.P. et al. (2017) 'A new method for displayed mathematical expression detection based on FFT and SVM', *NAFOSTED Conference on Information and Computer Science*.

Bui, H.P. et al. (2019) 'Mathematical variable detection based on convolutional neural network and support vector machine', *International Conference on Multimedia Analysis and Pattern Recognition*.

Buijs, H. et al. (1974) 'Implementation of a fast Fourier transform (FFT) for image processing applications', *IEEE Transactions on Acoustics, Speech, and Signal Processing*.

Caponetti, L. et al. (2008) 'Document page segmentation using neuro-fuzzy approach', *Applied Soft Computing*, Vol. 8, No. 1, pp.118–126.

Cheng, H. and Bouman, C.A. (2001) 'Multiscale Bayesian segmentation using a trainable context model', *IEEE Transactions on Image Processing*, Vol. 10, No. 4, pp.511–525.

Chu, W. and Liu, F. (2013) 'Mathematical formula detection in heterogeneous document images', *Conference on Technologies and Applications of Artificial Intelligence (TAAI)*.

Fan, K.C. et al. (2007) 'Italic detection and rectification', *Journal of Information Science and Engineering*, Vol. 23, No. 2, pp.403–419.

Friedman, J. and Finke, R.A. (1977) 'An algorithm for finding best matches in logarithmic expected time', *ACM Transactions on Mathematical Software*, pp.209–226.

Gao, L. et al. (2017) 'A deep learning-based formula detection method for PDF documents', *International Conference on Document Analysis and Recognition*.

Garain, U. (1998) 'Automatic detection of italic, bold and all-capital words in document images', *International Conference on Pattern Recognition*.

Garain, U. (2009) 'Identification of mathematical expressions in document images', *International Conference on Document Document Analysis and Recognition*.

Ghahramani, S. (2000) *Fundamentals of Probability*, Prentice Hall, New Jersey.

Ha, D.T. et al. (2016) 'An adaptive over-split and merge algorithm for page segmentation', *Pattern Recognition Letters*, 1 September 2016, Vol. 80, pp.137–143

Harjit, S. (2012) 'Detection of bold and italic character in Gurmukhi script', *Journal of Computer Engineering*, Vol. 1, No. 6, pp.28–31.

Iwatsuki, K., Sagara, T., Hara, T. and Aizawa, A. (2017) 'Detecting in-line mathematical expression in scientific documents', *Proceedings of the 2017 ACM Symposium on Document Engineering*, 4–7 September 2017, pp.141–144, Valletta, Malta.

Jin, J., Han, X. and Wang, Q. (2003) 'Mathematical formulas detection', *International Conference on Document Analysis and Recognition*.

Joachims, T. (2002) 'Optimizing search engines using clickthrough data', *ACM Conference on Knowledge Discovery and Data Mining*.

Jolliffe, I.T. (2002) 'Principal component analysis', *Springer Series in Statistics*, 2nd ed., Springer, New York.

Lee, H. and Wang, J. (1998) 'Design of a mathematical expression understanding system', *Pattern Recognition Letters*, Vol. 18, No. 3, pp.289–298..

Lin, X. et al. (2012) 'Performance evaluation of mathematical formula identification', *International Workshop on Document Analysis System*.

Lin, X. et al. (2013) 'A text line detection method for mathematical formula recognition', *International Conference on Document Analysis and Recognition*.

Lin, X. et al. (2014) 'Mathematical formula identification and performance evaluation in PDF documents', *International Journal on Document Analysis and Recognition*, Vol. 17, No. 3, pp.239–255.

Papandreou, A. and Gatos, B. (2011) 'A novel skew detection technique on vertical projections', *International Conference on Document Analysis and Recognition*.

Roger, L. (2008) *Linguistics 251 Lecture Notes*, Fall, University of California San Diego.

Shi, Z. and Govindaraju, V. (2005) 'Multi-scale techniques for document page segmentation', *International Conference on Document Analysis and Recognition*.

Smith, R. (2007) 'An overview of the Tesseract OCR engine', *International Conference on Document Analysis and Recognition*.

Steven, B. et al. (2008) 'The ACL anthology reference corpus: a reference dataset for bibliographic research in computational linguistics', *International Conference on Language Resources and Evaluation*.

Sun, C. and Si, D. (1997) 'Skew and slant correction for document images using gradient direction', *International Conference on Document Analysis and Recognition*.

Suzuki, M. et al. (2003) 'INFTY: an integrated OCR system for mathematical documents', *Proceedings of the 2003 ACM Symposium on Document Engineering*, Grenoble, France, 20–22 November 2003, pp.95–104.

Tran, T.A. et al. (2016) 'Page segmentation using minimum homogeneity algorithm and adaptive mathematical morphology', *International Journal on Document Analysis and Recognition* , Vol. 19, No. 3, pp.191–209.

Wahl, F. et al. (1982) 'Block segmentation and text extraction in mixed text/image documents', *Computer Graphics and Image Processing*, Vol. 20, No. 4, pp.375–390.

Wang, D. and Srihari, S. (1989) 'Classification of newspaper image blocks using texture analysis', *Computer Vision, Graphics, and Image Processing*, Vol. 47, No. 3, pp.327–352.

Wenhao, H. et al. (2016) 'Context-aware mathematical expression recognition: an end-to-end framework and a benchmark', *International Conference on Pattern Recognition (ICPR)*.

Zanibbi, R. and Blostein, D. (2012) 'Recognition and retrieval of mathematical expressions', *International Journal on Document Analysis and Recognition*, Vol. 15, No. 4, pp.331–357.

Zhang, L. et al. (2004) 'Italic font recognition using stroke pattern analysis on wavelet decomposed word images', *International Conference on Pattern Recognition*.