International Journal of System and Software Engineering
Volume 5 Issue 2, December 2017
ISSN.: 2321-6107

A Loci Features Based Method to Convert Images of Differential Calculus Expressions to Their Text Equivalent

H. N. Sharada[1*], Basavaraj S. Anami[2], K. B. Nagasundara[3]
[1]SDM College of Engineering & Technology, Dharwad, Karnataka, India. Email: shnsdmcet@gmail.com
[2]K. L. E. Institute of Technology, Hubballi, Karnataka, India. Email: anami_basu@hotmail.com
[3]JSS Academy of Technical Education, Bangaluru, Karnataka, India. Email: nagasundarakb@gmail.com
*Corresponding Author

Abstract: Many E-documents in scientific and engineering disciplines contain mathematical expressions. Differential equations are commonly solved in pre-university and engineering studies along with many applications existing in the real world. Human beings perceive mathematical expressions as images and evaluate, which is one of the traits of human beings. The solutions range from simple to complex mathematical expressions. Automation of this trait of human beings is the dire need in the field of engineering education in the light of self directed study, a popular project of Govt. of India called SWAYAM. Proposed paper is an attempt to automate recognition of mathematical expressions from an image and consists of phases such as pre-processing, image segmentation, Loci feature extraction and text conversion. The text version is subjected to further evaluation. The proposed method gives an average accuracy of 96% conversion.

Keywords: Connected component, Loci features, Mathematical expression, Segmentation, Symbol recognition, Text conversion.

I. Introduction

The inherent ability of computers to store a large amount of information has lead to innumerable applications in the fields of banking, e-commerce, agricultural science; decision supports systems, expert systems, image processing and the like. Image is the better form of information suitable for human perception and understanding. Mathematical expression evaluation is one of the traits of human beings, with which they solve simple to complex mathematical expressions such as trigonometric functions, arithmetic functions, Fourier transformations, integration, differentiation etc. Human beings perceive mathematical expressions as images and evaluate as text using axioms, identities and rules of calculus. In general, to solve any mathematical evaluation, human brain perceives the type of the expression, checks for the sub expressions and possibility of further simplifications. According to the sub expressions and their arrangements, human beings decide on operations, identities and laws governing the domain with appropriate domain knowledge. The mathematical expressions consist of different symbols and operations such as addition, subtraction, complex operations like integration, differentiation, geometrical etc. Consider the simple expression (1).

$$Y=3 + (4 * 5) \text{--------}(1)$$

Which is perceived by human beings as an image, in which sub images refer to sub expressions, namely, Y, numbers 3, 4, 5 and operators '=', '+' and '*'. Knowledge of operator precedence, is required to perform the calculations and obtain the value of Y=23. If the mathematical knowledge is built into the computer systems, automatically one solve the expressions. In this work, we have confided to domain of calculus.

Integration and differentiation form the major topics in calculus, which are widely used in engineering and scientific fields. Differential calculus is concerned with the study of the rates at which quantities change to model the real life situations. Further, Differential calculus is classified, based on the degree of order, as First order derivatives, second order derivatives and so on, up to n^{th} order derivatives as shown in Fig. 1. In each order, differential equation can be of constant coefficient, scalar multiples, product functions, quotient function, trigonometric, logarithmic, hyperbolic, implicit etc. which is shown in Table I.

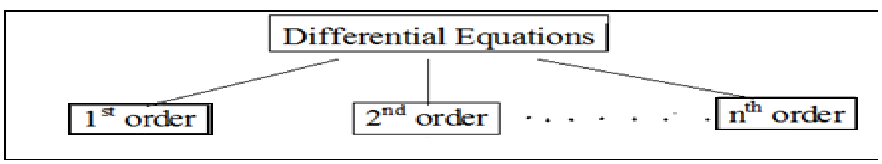


Fig. 1: Types of Differential Equations

TABLE I: TYPES OF DIFFERENTIAL FUNCTIONS

Types of Differential Functions	Example
Constant Functions.	$Y = x + x^2$, $Y = x^{1(1/2)}$.
Scalar Multiple Function.	$Y = 5x + 3x^2$
Product Functions .	Y= x sin x, Y=sin x cos x.
Quotient Functions.	$Y= 2\ x^2 / (x+\cos x)$
Trigonometric Functions.	$Y= \cos^2 x + \tan x + \cot x$
Inverse Trigonometric Functions.	$Y = \sin^{-1}x$, $Y = \cot^{-1}(x+1)$.
Hyperbolic Functions.	Y = coshx + tanh x
Inverse Hyperbolic Functions.	$Y= \cosh^{-1}x + \sinh^{-1}x$
Logrithmic Functions.	$Y = x \log x$, $Y = e^{\log x}$
Implicit Function.	$Y = e^x$ w.r.t cos x
Successive Differenation .	$Y = \sin x$, $Y^4 – Y^2 +2XY$
Composite Function (Chain Rule) .	$Y= e^{\cos x}$ U = cos 2x V = 5x

Differential equations are further classified as Ordinary Differential Equations (ODE) and Partial Differential Equations (PDE) as shown in Fig. 2. ODEs are functions of single variable, they are found in the study of dynamical systems and electrical networks, whereas PDEs involve unknown functions on two or more independent variables and used in weather forecasting, design of mechanical systems etc.

Mathematical expressions are two-dimensional arrangements of symbols. Understanding a mathematical expression involves symbol recognition and evaluation based on rules. Recognition of mathematical expression is considered a difficult task, because there is a large character set (roman letters, Greek letters, operator symbols, etc) from which the symbols are selected and used as variables in the expressions with a variety of font sizes and font types, namely, in subscripts, superscripts, limit expressions, etc with various possible scales such as brackets, parentheses, square roots, etc.

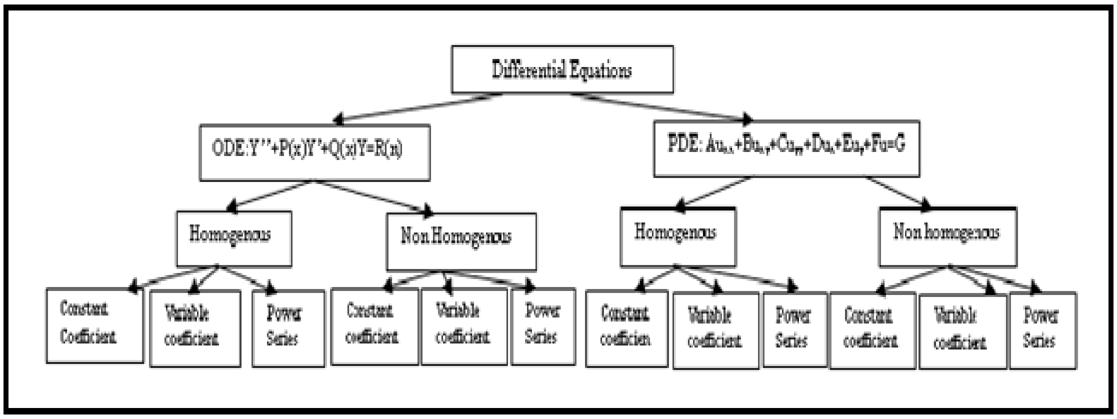


Fig. 2: Types of Differential Equations

This present work involves reading an image of printed mathematical expressions, segmentation of expression into sub expressions, recognition of sub expressions, recognition of mathematical character / symbols and converting the entire expression into equivalent text. The remaining part of the paper is organized into four sections. Section 2 gives a brief description of literature survey. Proposed methodology is given in Section 3. Results are given in Section 4. Finally, conclusion of the study and future work are presented in Section 5.

II. LITERATURE SURVEY

In order to know the state-of-the-art connected to the proposed work, we have conducted a literature survey and the gist of some papers is given as under.

Fairouz Lekhal *et al.*, (2011) have presented recognition of Persian handwritten digits based on characterization loci and mixture of experts (ME). The work has used 4 experts in which each expert is having 25 neurons in hidden layer and 5 neurons in hidden layers of gating network. Author has claimed recognition rate of 97.52%.

Richard Zanibbi *et al.*, (2002) have carried out a work on typeset and handwritten mathematical notations, in which segmentation of symbols using bounding box and symbol layout is used to construct Baseline Structure Tree (BST). It describes two dimensional arrangements of input mathematical symbols. Further, it is translated to operator tree by order and scope of operations in the input mathematical expressions for analysis. They have implemented a package called Diagram Recognition Application for Computer Understanding of Large Algebraic Expressions (DRACULAE), is as an interface and a third-party symbol recognizer for free hand formula entry system.

Xue-Dong Tian *et al.*, (2006) have presented a methodology for symbol recognition of mathematical expression, in which Projection-profile cutting and Connected components labelling algorithms are adopted for symbol segmentation. Symbol recognition is carried out using Peripheral features and 4-dimensional directional line element features. Minimum distance classifier is used in which aspect Ratio of symbols is used for classification. Developed model is tested with different data sets like, math hand book and math journal symbols and claims a success rate in the range 96.44% to 98.22%.

Manisha Bharambe (2015) has presented a method for recognition of Offline Handwritten Mathematical Expressions, in which segmentation is carried out using labelling of the isolated symbols. They have extracted features using normalized chain code (16), zone features, moment invariant features (23) and projection histogram features (127). SVM is used to recognize the given expressions. Author has claimed a recognition accuracy of 89% to 94% for letter and 82% to 95% for symbols.

Yassine Chajri and Belaid Bouikhalene (2016) have presented a method for handwritten mathematical expressions recognition. They have considered expressions from logic, analysis, algebra and probability. Segmentation is carried out using 8-way connected components and feature extraction using Mean of Radon Transform (MRT). Symbols are recognised using Support Vector Machines (SVM) with Polynomial kernel function. They have claimed recognition accuracies of 90%, 92%, 94% and 90% for logical, analytical, algebraic and probability expressions respectively.

Jaekyu Ha *et al.*, (1995) have developed a Math Expression Understanding System (MEUS). The primitives and associated attributes are extracted from the image. The connected component labelling algorithm is used for Primitive extraction from images of mathematical expressions. Using these attributes, initial hierarchy structure is constructed and groups are merged into key words. Developed model verifies mathematical syntax rules and includes error correction.

Ba-Quy Vuonga *et al.*, (2008) have presented a structural analysis of handwritten mathematical expressions. A focus is laid on interpreting the recognized symbols using geometrical information, such as relative sizes and positions of the symbols. It is also observed that most of the existing approaches rely on hand-crafted grammar rules to identify semantic relationships among the recognized mathematical symbols. They have used progressive symbol recognition and progressive structural analysis for representation purpose**.**

Nicolusd *et al.*, (2012) have described a method for recognition of hand written mathematical symbols. This method used Pyramid of histograms of gradients, in which orthogonal projection histogram is used as features and Support Vector Machine as a recogniser. CROHME 2012 data set is used in this method.

Sanjay S. Gharde, *et al.*, (2013**)** have presented a methodology for classification of mathematical equations, wherein support vector machine is used as recogniser. A projection histogram method is used for extracting features from handwritten mathematical equations. The classification and recognition is performed on a total of 237 symbols extracted from 28 different equations.

Anjan Dutta and Josep Llad´os (2013) have presented a sub graph matching problem in Region Adjacency Graph (RAG) applied to symbol spotting in graphical documents and in which sub graph matching is carried using the Approximate Edit Distance Algorithm (AEDA). They have given a well-defined perfectly delineated type of convexity of regions, which allows to match the spotted symbol.

From the survey, we understand that works have been carried out in recognition of symbols, and mathematical expressions, both printed and handwritten. Less number of equations, drawn from different topics, is considered. Evaluation of expressions is least attempted. The recognition of mathematical expressions and their evaluation in certain mathematical domain is still a research problem area. This is the motivation for the present work on mathematical expression recognition in differential calculus and text conversion for further evaluation.

III. Proposed Methodology

The proposed work consists of five stages, namely, pre-processing, image segmentation, Loci feature extraction, text conversion and post processing, as depicted in Fig. 3. Our data set includes printed differential expressions, which are grouped into three categories like simple, medium and complex differential expressions as shown in the Table II. We have assumed all the differential expressions are of the form *"Y = Expression"*, where 'Y' indicates function of x,f(x), on which differentiation, (d(Y)/dx), is carried out. Differential expressions are typed in Microsoft Equation 3.0 and typed expressions are converted into jpg images and form input to the method.

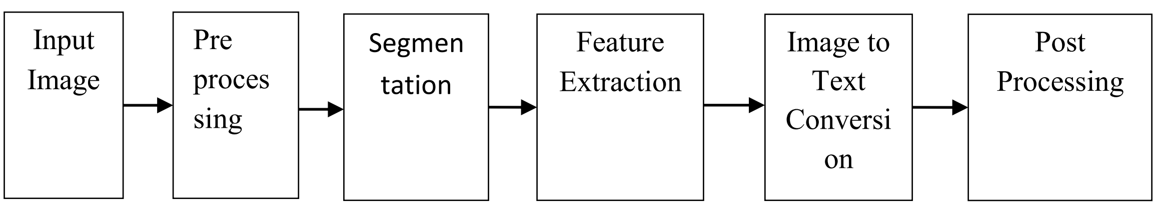


Fig. 3: Proposed Block Diagram

A. Pre-Processing

The pre-processing is to enhance the quality of the input image, which converts a gray scale image into binary image and resized to [100 X 100]. Further, it is subjected to extraction of edges and the resultant images are as shown Fig. 4 and Fig. 5.

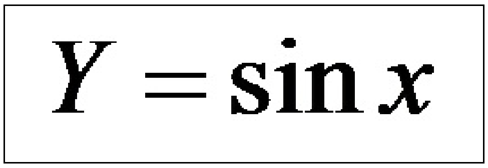


Fig. 4: Input Image of Expression

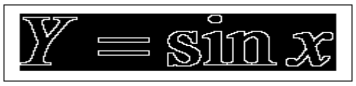


Fig. 5: Edge Image of Input Expression

B. Segmentation

Segmentation is carried out using region properties called connected component labelling, bounding box and position related properties to build spatial relation Tables (2D & 1D), which give positional information of segmented image (Xmin, (Xmax-Xmin) & Ymin, (Ymax-Ymin)) of segmented sub images and spatial coordinate values and number of connected components. For the input image, given in the Fig. 6, the connected components and spatial relationships are given in Tables III and IV respectively.

TABLE II: SAMPLE DIFFERENTIAL EXPRESSIONS

Simple Differential Expressions	Medium Differential Expressions	Complex Differential Expressions
$Y = \sin x$	$Y = e^x \sin x$	$Y = \frac{e^x}{e^x + 1}$
$Y = \cos x$	$Y = \cot x(\sqrt{x} - 4e^x$	$Y = \frac{2^x \cot x}{\sqrt{x}}$
$Y = \tan x$	$Y = (x^2 + 3x)e^x$	$Y = \frac{\log x}{1 + \sqrt{x}}$
$Y = \sec x$	$Y = x e^x \sin x$	$Y = \frac{\log x}{1 + \log x}$
$Y = \cot x$	$Y = x \tan^{-1} x$	$Y = \frac{e^x - 1}{e^x + 1}$
$Y = cse\, x$	$Y = \sec x \tan x$	$Y = \frac{x \tan x}{\sec x + \tan x}$
$Y = \sin^2 x$	$Y = (1 - x^2) \tan x$	$Y = \frac{x + \sin x}{x + \cos x}$
$Y = \cos^2 x$	$Y = \sec x(\log x - 3x^2)$	$Y = \cos^{-1}\left(\frac{x^2 - 1}{x^2 + 1}\right)$
$Y = \tan^2 x$	$Y = (4 \sin x + 3)e^x$	
$Y = x^n$	$Y = x \sin^{-1} x$	
$Y = \frac{1}{x^n}$	$Y = x \tan^{-1} x \sin x$	
$Y = \sqrt{x^n}$	$Y = x^n \cos^{-1} x$	
$Y = \frac{1}{\sqrt{x}}$	$Y = (x^2 + 3x) \log x$	
$Y = e^x$	$Y = \log x + \frac{\sin x}{x}$	

TABLE III: 2D SPATIAL RELATIONSHIP TABLE

Number of Connected Component		Ymin	Xmin	(Ymax-Ymin)	(Xmax-Xmin)
1	Y	3	6	38	43
2	-	57	25	34	3
3	_	57	37	34	3
4	S	105	19	22	31
5	1	129	19	16	30
6	.	133	4	8	6
7	N	146	19	33	30
8	X	187	20	31	30

TABLE IV: 2D SPATIAL RELATIONSHIP TABLE

	Ymin	Ymax
Y	3	40
=	57	90
S	105	126
i	129	144
n	146	178
x	187	217

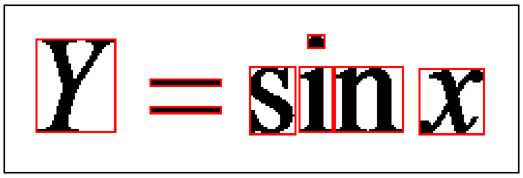


Fig. 6: Segmented Image

Considering the image of a mathematical expression as 2D arrangement of symbols, it is decomposed into sub-images of individual characters / symbols. Recursive horizontal and vertical scanning is deployed to extract the nonzero regions, so as to get a stream of atomic sub images, as shown in Fig. 7 and Fig. 8. During vertical scanning, we get seven segments and each segment is consisting of a number of connected components, which represents a sub image of an expression if the segment has just one connected component. When more than one connected components exist in the segment, the image is further horizontally scanned for the sub images in the given segment. In Fig. 8, second and fourth segments have two connected components namely '=' and 'I', which are extracted using horizontal scanning. The extracted sub images are subjected to feature extraction.

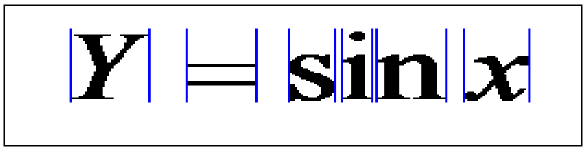


Fig. 7: Vertical Profile Cut Image of Given Expression

Fig. 8: Horizontal Profile Cut Image of Given Expression

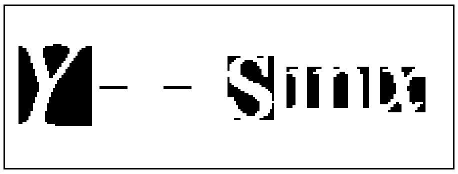


Fig. 9: Sub Images of Given Expression

C. Feature Extraction

The characteristics Loci features are used in the work [1], because of its advantages, namely, robustness to variations in styles, fonts and scaling. The Loci characteristics are generally defined in horizontal and vertical directions. They are computed with respect to the number of white to black pixels transitions in the right, down, left and up directions for each background pixel of the image. For each background pixel, a four digits number is obtained, called a Loci number. The information on background and foreground pixels is obtained to construct the feature vector. The number of transitions is limited to two and the size of the Loci vector is 81. We have constructed Loci features for 71 printed characters / symbols and segmented sub images of 100 printed mathematical expressions. The Fig. 10 shows the Loci features for the character 's' and feature vector is given in Table V.

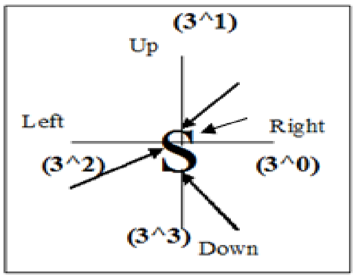


Fig. 10: Loci Features for the Character 's'

Table V: Loci Features of Character 's'

Loci Feature (Fi)	Value	Sl.No.	Value	Sl.No.	Value	Sl.No.	Value
F_1	0	F_{22}	0	F_{42}	0	F_{62}	0
F_2	0	F_{23}	0	F_{43}	0	F_{63}	0
F_3	0	F_{24}	0	F_{44}	161	F_{64}	0
F_4	0	F_{25}	0	F_{45}	0	F_{65}	0
F_5	0	F_{26}	0	F_{46}	0	F_{66}	0
F_6	0	F_{27}	0	F_{47}	0	F_{67}	0
F_7	0	F_{28}	0	F_{48}	0	F_{68}	162
F_8	0	F_{29}	0	F_{49}	0	F_{69}	0
F_9	0	F_{30}	0	F_{50}	0	F_{70}	0
F_{10}	0	F_{31}	0	F_{51}	0	F_{71}	0
F_{11}	0	F_{32}	0	F_{52}	0	F_{72}	0
F_{12}	0	F_{33}	0	F_{53}	0	F_{73}	0
F_{13}	0	F_{34}	0	F_{54}	0	F_{74}	0

Loci Feature (Fi)	Value	Sl.No.	Value	Sl.No.	Value	Sl.No.	Value
F_{14}	18	F_{35}	0	F_{55}	0	F_{75}	0
F_{15}	0	F_{36}	0	F_{56}	0	F_{76}	0
F_{16}	0	F_{37}	0	F_{57}	0	F_{77}	0
F_{17}	0	F_{38}	24	F_{58}	0	F_{78}	0
F_{18}	0	F_{39}	0	F_{59}	0	F_{79}	0
F_{19}	0	F_{40}	0	F_{60}	0	F_{80}	0
F_{20}	0	F_{41}	585	F_{61}	0	F_{81}	0
F_{21}		0				Loci Number=41	

Table VI: Non Zero Loci Features of Expression

Sub Image	Non Zero Loci Features
Y	F_{11},F_{12},$F_{14}$$F_{15}$,$F_{17}$,$F_{38}$,$F_{39}$,$F_{41}$,$F_{42}$,$F_{48}$,$F_{50}$,$F_{66}$,$F_{75}$.
-	F_{31}.
-	F_{31}.
S	F_{14},F_{41}, F_{44}, F_{42}, F_{68}.
I	F_{31}, F_{33}, F_{34}, F_{41}, F_{42}.
N	F_{32}, F_{33}, F_{35}, F_{40}, F_{41}, F_{42}, F_{43}, F_{45}, F_{49}, F_{53}, F_{59},F_{60},F_{68},F_{69},F_{72},F_{77}.
X	F_{14}, F_{41}, F_{44}, F_{68}.

D. Image to Text Conversion

This stage has the knowledge base already created from Loci features obtained for all the character / symbols along with their equivalent text information. The Loci features obtained from a sub image are compared with the knowledge base and equivalent text information is obtained. Thus, a stream of text equivalent of character / symbols for given input image, given in Fig. 4, is obtained, which is shown in Fig. 11.

Y- - S in x

Fig. 11: Converted Text Output of Given Expression

E. Post-Processing

The output obtained from the image-to-text conversion stage is single dimensional arrangement of character / symbols, representing square roots, subscripts, and superscripts. We have developed rules, based on mathematical formulae and the spatial relationships existing amongst the sub images, to obtain the equivalent text expression for an image of the given input expression. The spatial information amongst sub images obtained during segmentation stage, due to vertical / horizontal scanning, is not adequate to predict the presence of square-roots, subscripts and superscripts in the input expressions. The main task in this stage is to identify the super script, fraction and expression under square root. In order to perform appropriate post processing, we have made the following assumptions.

- All the mathematical expressions are of the form "Y = Mathematical expression", where "Y" is a differential function on which first order differentiation is to be carried out i.e. (d(Y)/dx). The assumed text equivalent of some of the mathematical symbols like superscript / powers and square roots (√) are given in the Table VII.

Table VII: Text Equivalent for Mathematical Symbols

Sl.No.	Character / Symbol	Equivalent Output	Example	Equivalent Text
1	Power	^	$\sin^2 x$	sin^2(x)
2	Square root	Sqrt	$\sqrt{\sin x}$	sqrt (sin x)
3	Inverse Function	^(-1)	$\sin^{-1}x$	sin^(-1)x
4	S followed by ln	in	sln	sin

- We describe an image in spatial coordinate system, in which it is described in terms of x and y coordinates. Consider the sub image of mathematical expression 'Y' is represented in spatial coordinate system in terms of its [Ymin, Xmin] and [Ymax, Xmax] as shown in Fig. 12. Similarly, the remaining sub images of input expression are given Table VII.

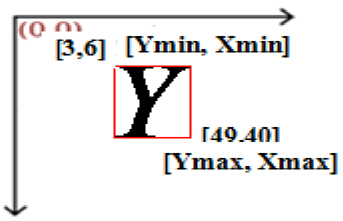
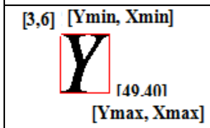
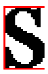
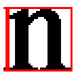
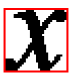


Fig. 12: Spatial Coordinate Representation of Image

Consider the image of mathematical expression given in Fig. 4, and its converted text output show in the Table VIII.

TABLE VIII: SUB IMAGES AND ITS SPATIAL COORDINATE VALUES

Sub Image	Text Output	[Ymin, Xmin, (Ymax-Ymin), (Xmax-Xmin)]	[Ymin, Ymax]	Description
	Y	[3, 6, 38,43]	[3,40]	The first text output is 'Y' and from the origin it is towards y-axis is 3 to 40 and towards x-axis is 6 to 49.
	–	[57,25,34,3]	[57,90]	Text output is '-' and from the origin it is towards y-axis is 57 to 90 and towards x-axis is 25 to 28.
	–	[57,37,34,3]	[57,90]	Text output is '-' and from the origin it is towards y-axis is 57 to 90 and towards x-axis is 37 to 38, i.e. previous symbol and present symbol has same (Ymin ,Ymax), different (Xmax,Xmin). It indicates that symbol is "=".
	S	[105,19,32,30]	[105,126]	This is the first sub symbol of expression and its maximum spatial co-ordinate is ((Xmax-Xmin)=30).
	L	[129,19,16,31]	[129,144]	This is the second sub symbol of expression and its maximum spatial co-ordinate is ((Xmax-Xmin)=31). It indicates previous and present symbol is in same base line.
	.	[133,4,8,6]	[133,141]	This is the third sub symbol of expression and its maximum spatial co-ordinate is ((Xmax-Xmin)=6). It indicates previous and present symbol is not in same base line, it is above previous symbol.
	N	[146,19,33,30]	[146,178]	This is the fifth sub symbol of expression and its maximum spatial co-ordinate is ((Xmax-Xmin)=30). It indicates previous and present symbol is in same base line.
	X	[187,20,31,30]	[187,217]	This is the fifth sub symbol of expression and its maximum spatial co-ordinate is ((Xmax-Xmin)=30). It indicates previous and present symbol is in same base line.

F. Rules Developed for Decision Making

i. If text output is of the first sub image, keep it as it is, because always first character / symbol in considered expression is Y.

ii. If the text output is "- -" and both have the same (Ymin, Ymax) and different (Xmin, Xmax), then it represents the symbol "=", from the Table VII.

Once the '=' symbol is obtained, the next text output is the given mathematical expressions and its spatial relationships are decided from its coordinates' values. Find the baseline reference with respect to "=" symbol and all the text output's spatial relationship is decided with respect to the base line. The text output obtained is Right hand side expression. According to considered mathematical expressions, first character / symbol cannot be a superscript; it can be a character / symbol, square root, brackets or fraction line. The first text output is kept as it is and second text output character / symbol's coordinate values are compared with previously identified character / symbol, in which its Xmax values are same or almost same, then it is in the same line.

iii. If the position (Xmax) of the text output is different from the preceding text ouput, then find the currently identified text output as super / subscript, fraction etc.

For the square root, find the overlapped components under square root. From the spatial coordinate values, extract the region property of connected components under the square root using morphological operations and represent the text output within brackets ( ). When the line segment is found, it can be a fraction line or a minus symbol.

iv. If it has overlapped sub images, then it is fraction lines. Text outputs are represented in the brackets using '/' symbols. If it is a minus symbol (-) keep it as it is.

The text output so obtained is called intermediate expression, which is shown in expression (2).

Y= sin x --------(2)

IV. Result Analysis

The developed method is tested with 100 images of printed mathematical expressions and 100 segmented images of printed mathematical expressions. Recognition accuracy ranges from 94% to 98%, which includes simple mathematical expressions, medium mathematical expressions and complex mathematical expressions. Recognized text output is verified with mathematics experts and their opinions for sample expressions are given in Table IX. The graph of the recognition accuracy is shown in Fig. 13.

Table IX: Sample Tested Mathematical Expressions

Types of Expressions	Input Image	Equivalent Text Output	Expert 1	Expert2	Expert3
Simple Expressions	$Y = x^7$	Y=x^7	Correct Output	Correct Output	Correct Output
	$Y = 1/x^n$	Y=1/(x^n)	Correct Output	Correct Output	Correct Output
	$Y = \sin x$	Y=sinx	Correct Output	Correct Output	Correct Output
	$Y = \cos x$	Y=cosx	Correct Output	Correct Output	Correct Output
	$Y = \tan x$	Y=tanx	Correct Output	Correct Output	Correct Output
	$Y = e^x$	Y=ex	Wrong Output	Wrong Output	Wrong Output
Medium Expressions	$Y = (1 - x^2)\tan x$	Y=(1-x^2)tanx	Correct Output	Correct Output	Correct Output
	$Y = \sec x \tan x$	Y=secxtanx	Correct Output	Correct Output	Correct Output
	$Y = \sec x(\log x - 3x^2)$	Y=secx(log(x)-3x^2)	Correct Output	Correct Output	Correct Output
	$Y = \cot x(\sqrt{x} - 4e^x)$	Y=cotx(sqrt(x)-4e^x)	Correct Output	Correct Output	Correct Output
	$Y = e^x \sin x$	Y=ex sinx	Wrong Output	Wrong Output	Wrong Output
	$Y = xe^x \sin x$	Y=xexsinx	Wrong Output	Wrong Output	Wrong Output
	$Y = x\tan^{-1} x$	Y=xtan^(-1)x	Correct Output	Correct Output	Correct Output
Complex expressions	$Y = \frac{\log x}{1 + \log x}$	Y=(log(x))/(1+log(x))	Correct Output	Correct Output	Correct Output
	$Y = \frac{x + \sin x}{x + \cos x}$	Y=(x+sinx)/(x+cosx)	Correct Output	Correct Output	Correct Output
	$Y = \frac{\log x}{1 + \sqrt{x}}$	Y=(log(x))/(1+sqrt(x))	Correct Output	Correct Output	Correct Output
	$Y = \frac{x \tan x}{\sec x + \tan x}$	Y=cotx(sqrt(x)-4e^x)	Correct Output	Correct Output	Correct Output

Types of Expression	Input image	Equivalent Text Ouput	Expert 1	Expert 2	Expert 3
	$Y = \frac{e^x}{e^x + 1}$	Y=ex/(ex+1)	Wrong Output	Wrong Output	Wrong Output
	$Y = \frac{2^x \cot x}{\sqrt{x}}$	Y=(2^x cotx)/(sqrt(x))	Correct Output	Correct Output	Correct Output
	$Y = \cos^{-1}\left(\frac{x^2 - 1}{x^2 + 1}\right)$	Y=cos^(-1)((x^2-1)/(x^2+1))	Correct Output	Correct Output	Correct Output
	$Y = \tan\left(\frac{a + b\cos x}{b - a\cos x}\right)$	Y=tan^(-1)((a+hcosx)/ ((h-acosx))	Wrong Output	Wrong Output	Wrong Output

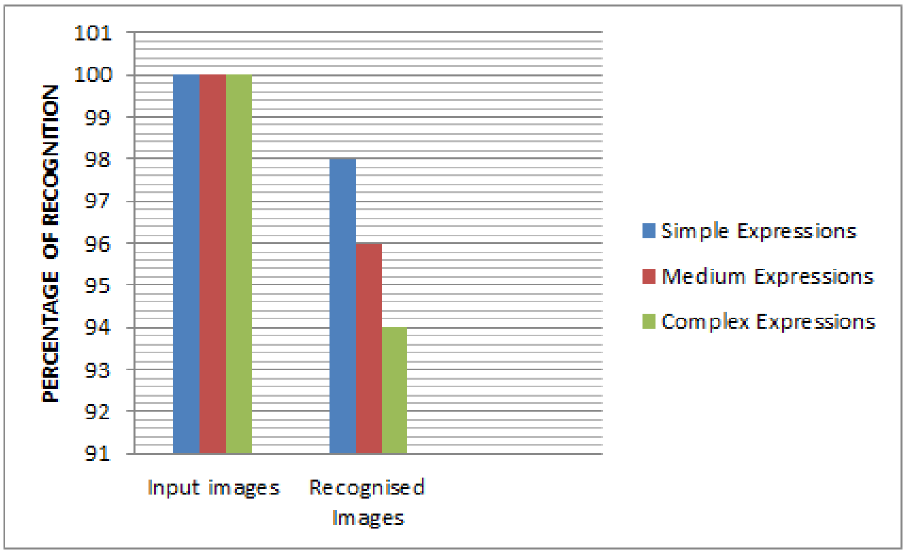


Fig. 13: Graph of Tested Mathematical Expressions

Rule based classification is carried out to recognize appropriate types of differential expressions and converting them to text representations. The average accuracy of the recognition is found to be 96%. We have compared proposed work with the related works in the literature. Table X gives the comparison with the related work. The text output is suitably represented for further evaluation, which is beyond the scope of this paper.

Table X: Related Work Comparison

Paper	Method	Remarks
Recognizing Mathematical Expressions using Tree Transformation [2]	Segmentation of symbols using bounding box and connected components. Symbol layout is used to construct Baseline Structure Tree (BST). Lexed BST Operator tree, it indicates order and scope of operations in the input mathematical expressions. They have implemented Diagram Recognition Application for Computer Understanding of Large Algebraic Expressions (DRACULAE).	Lexed BST and Operator Tree is ready to parse.
Research on Symbol Recognition for Mathematical Expressions [3]	Symbol segmentation: Projection-profile cutting and Connected components labelling algorithm. Symbol recognition using Peripheral features and directional line element features. Aspect Ratio of symbol is used to classify.	Symbol Recognition and classification.
Segmentation of Offline Printed and Handwritten Mathematical Expressions [4]	Segmentation is carried out using labelling the isolated symbols. Normalized chain code, Zoning features, Moment invariant features, Projection histogram. SVM is used to recognize the expression.	Expression Segmentation Rate and symbol Classification.

Paper	Method	Remarks
Handwritten Mathematical Expressions Recognition [5]	Mathematical expressions belongs to mathematical logic, mathematical analysis, mathematical algebra, mathematical probability are considered. Segmentation is carried out using 8-way connected components. Feature extraction using Mean of Radon Transform (MRT). Symbols recognition is carried out using Support Vector Machines (SVM).	Symbol Recognition and Expression Recognition.
Present Work	Segmentation is using connected component labelling and bounding box. Loci features to recognize character / symbol. Intermediate text output. Differential calculus expressions are used.	Expression segmentation, conversion of images to equivalent text output ready for parsing and evaluation. 96% accuracy of recognition.

V. Conclusion

The work has considered mathematical expressions from differential calculus and has given 96% accuracy using characteristics Loci features. The method is robust to variations in styles, fonts and scaling and has given good results in recognizing numerals, symbols and isolated characters. The images of mathematical expressions are converted into equivalent text representations. The developed rules are used to convert the images into equivalent text, which is further used for evaluation.

References

[1] F. Lekhal, Md. E. Hitmy, and O. E. Melhaoui, “Arabic numerals recognition based on an improved version of the loci characteristic,” *International Journal of Computer Applications*, vol. 24, no. 1, pp. 36-41, June 2011.

[2] R. Zanibbi, D. Blostein, and J. R. Cordy, “Recognizing mathematical expressions using tree transformation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 11, pp. 1455-1467, 2002.

[3] X. D. Tian, H. Y. Li, X. F. Li, and L. P. Zhang, “Research on symbol recognition for mathematical expressions,” *Proceedings of the First International Conference on Innovative Computing, Information and Control, 2006 (ICICIC’06),* 30 August - 1 September 2006.

[4] M. Bharambe, “Segmentation of offline printed and handwritten mathematical expressions,” *International Journal of Computer Applications National Conference on Digital Image and Signal Processing*, 2016.

[5] Y. Chajri, and B. Bouikhalene, “Handwritten mathematical expressions recognition,” *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 9, no. 5, pp. 69-76, 2016.

[6] J. Ha, R. M. Haralick, and I. T. Phillips, “Understanding mathematical expressions from document images,” *Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR ‘95)*, pp. 956-959, 1995.

[7] B. Q. Vuong, S. C. Hui, and Y. He, “Progressive structural analysis for dynamic recognition of on-line handwritten mathematical expressions,” *Pattern Recognition Letters*, vol. 29, pp. 647-655, 2008.

[8] R. Ebrahimpour, Md. R. Moradian, A. Esmkhani, and F. M. Jafarlou, “Recognition of Persian handwritten digits using characterization loci and mixture of experts,” *International Journal of Digital Content Technology and its Applications*, vol. 3, no. 3, pp. 42-46, September 2009.

[9] N. D. Jimenez, and L. Nguyen, “Recognition of hand written mathematical symbol with PHOG feature,” *Proceedings of the 14th International Conference on Pattern Recognition (ICPR)*, vol. 2, 2012.

[10] X. Lin, L. Gao, Z. Tang, X. Lin, and X. Hu, “Performance evaluation of mathematical formula identification,” *10th IAPR International Workshop on Document Analysis Systems (DAS)*, 27-29 March 2012.

[11] P. F. Felzenszwalb, and R. Zabih, “Dynamic programming and graph algorithms in computer vision,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 721-740, 2011.

[12] S. S. Gharde, V. B. Pallavi, and K. P. Adhiya, “Evaluation of classification and feature extraction techniques for simple mathematical equations,” *International Journal of Applied Information Systems*, vol. 1, no. 5, pp. 34-38, February 2012.

[13] A. Dutta, J. Llad´os, H. Bunke, and U. Pal, “Near convex region adjacency graph and approximate neighborhood string matching for symbol spotting in graphical documents,” *2013 12th International Conference on Document Analysis and Recognition (ICDAR),* 25-28 August 2013.