IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AI BASED MATHEMATICAL EXPRESSION RECOGNITION USING HISTOGRAM PROCEDURE

[1]E.N.Ganesh

[1]Professor ECE Department Sri Venkateswara College of engineering Sriperumbudur 602117, India

Abstract: The issue of recognizing handwritten mathematical expressions is addressed in this study by utilizing an optical character recognition algorithm. Mathematical symbol segmentation was achieved through vertical and horizontal histogram projections using numerical methods, while character classification was carried out using a convolutional neural network. The evaluation was conducted on CROHME's 2011 competition database, with the neural network being trained on Kaggle's math symbols database. The algorithm's performance is measured in terms of Levenshtein distance and the number of accurate predictions. The neural network achieved an 18% success rate, which increased to 30% when symbols like sin, cos, tan, i, j, and ÷ were excluded from the database.

Index Terms **-** Mathematical Expression, Numerical Methods, Math symbols and Database.

I. Introduction

The technological progress made over the past century has been significant, leading to increased affordability and accessibility for the general population. The widespread use of smartphones and personal computers has enabled individuals to access information quickly and easily. However, this advancement has also brought about new challenges, particularly in the realm of data transfer from analog to digital formats. Recognizing handwritten expressions poses a greater challenge compared to text recognition due to the larger number of possible characters and the presence of complex mathematical symbols. This complexity necessitates the segmentation and classification of each symbol within the program, making expression recognition one of the most challenging tasks in computer vision. Considering the complexity of the issue, researchers have dedicated the past few decades to addressing it. The underlying heuristics of mathematical laws and syntax have posed a challenge in developing an algorithm that is both simple enough for computers to execute and complex enough to recognize a wide range of mathematical expressions. The most successful approaches thus far have been detailed in the paper [3,4]. Among these methods, one utilized a convolutional neural network as a classifier for mathematical symbols, while another focused on enhancing the dataset to improve the system's precision. The primary objective of the study was not to train a neural network but to address the segmentation issue using two nested recursive functions and heuristic principles for grouping segmented symbols. Two types of evaluations were conducted in the study, with one being standard and the other involving an enriched dataset. The results indicated a significantly higher precision with the latter approach, which was expected due to the non-uniform width of characters resulting from variations in handwriting styles.

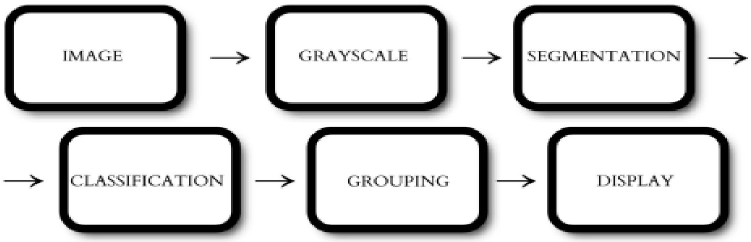

Figure 1 Flow chart of Algorithm

II Implementation Method

The intricacy of the problem necessitates a sophisticated algorithm for its resolution. The algorithm comprises two essential steps: segmentation and regrouping of segmented symbols into latex format. Prior to segmentation, certain preparations are required. Initially, the image must be converted from RGB to Grayscale, followed by noise removal. Noise reduction is a critical sub algorithm, as any remaining noise may be misinterpreted as a foreground pixel carrying information, leading to the misclassification of a math symbol not present in the dataset and rendering the results meaningless.

In addition to this, data set enrichment was utilized. The database utilized for training the neural network contained symbols that were only one pixel wide. It is well-known that handwriting styles vary significantly, as do the pencils used to write them. Therefore, a system that is invariant to these variations in styles and interpretations of symbols was necessary. The chosen solution was to employ the dilate function from the OpenCV library with a kernel K. Image processing encompasses all alterations made to an image to facilitate further processing. This process can be broken down into three main components: RGB to Grayscale conversion (for images with varying shades of gray), noise reduction, and image binarization. When working with an RGB image, the concern lies in the levels of red, green, and blue present in each pixel. To simplify the image, these three values are converted into a single value (Grayscale value) using the cvt Color function from the OpenCV library. Since hardware is a man-made creation, it is expected that it may not always function perfectly. Hardware malfunctions are referred to as noise, which can still occur despite advancements in electronics. Gaussian noise is the most common type of noise found in images, and therefore, it is the primary focus for isolation. The method employed in the experiment to remove noise is quite straightforward: the color of a specific pixel is replaced with the average value of surrounding pixels. This requires identifying similar pixels, which do not necessarily need to be adjacent. For instance, in the case of repetitive patterns, scanning a sufficiently large region of the image to find as many similar pixels as possible is crucial. Two pixels are considered similar if their values are approximately equal. This task can be accomplished using a function from the OpenCV library.

Binarization is a technique that assigns one of two possible values (0 or 255) to each grayscale pixel in an image. This process is crucial for distinguishing between foreground and background pixels. To achieve this, a threshold is established for each pixel, allowing for accurate assignment of values. In our case, where the image's brightness may not be ideal, determining local threshold values would enable more precise separation of expressions. To accomplish this, the experiment utilized the adaptive threshold function from the OpenCV library, specifically the adaptive Thresh Gaussian C method. Moving on to segmentation, a basic neural network consists of output neurons that assess the probability of a specific object being present in an image. Each neuron represents the likelihood of the object's presence. Now, let's consider mathematical expressions as an example. The set of all possible formulas is determined by various combinations of mathematical symbols. This vast range of possibilities makes it impossible to recognize entire expressions as a whole. However, since the set of mathematical symbols is finite, segmentation becomes feasible. The previous paragraph emphasized the significance of noise reduction, and this paragraph will delve further into the topic. Firstly, let's introduce a numerical method known as histogram projection, which examines whether a particular part of an expression lies within a given column or row of an image. In this experiment, two types of histogram projections were employed: vertical and horizontal. The vertical histogram projection calculates the number of foreground pixels in each column of the image, while the horizontal histogram projection functions in a similar manner [4]. AI algorithm details dealt with [5] to [8] and histogram details in [9] to[14].

Figure 2 H Equation

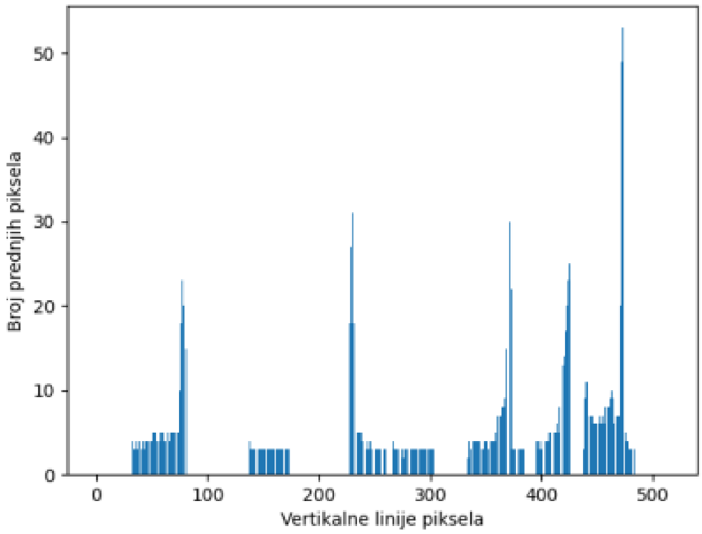

Figure 3 Vertical Histogram

If we consider the example shown in figure 2 and examine its horizontal histogram projection, it can be concluded that there is only one expression present. This is evident from the presence of a single peak on the diagram. A peak refers to a consecutive set of rows in the image where each row contains at least one foreground pixel. Similarly, by analyzing the vertical histogram projection, it is possible to identify seven peaks which directly correspond to the seven symbols in the mathematical expression. In this particular instance, a single horizontal and vertical projection were sufficient to detect all the symbols in the image. However, it should be noted that this may not always be the case with mathematical expressions. For instance, when dealing with a fraction within an expression, it becomes apparent that after performing the vertical and horizontal projections, most symbols can be successfully segmented. However, the numerator, denominator, and the fractional line will be recognized as a single block. To address this issue, a new approach is introduced by incorporating an additional horizontal projection. This enhancement improves the algorithm, but there is a possibility that either the numerator or the denominator may contain a new mathematical expression. Theoretically, this sequence can continue indefinitely. Consequently, a finite number of horizontal and vertical projections would only provide an approximation of the problem. Therefore, recursive programming in the Python programming language was utilized in this experiment. As with any recursive process, it is essential to define the base case, which is the condition that terminates the recursion. One possible approach is to halt the recursion when no new smaller block images can be obtained, i.e., when only one peak is present in both histogram graphs. However, it is important to note that this approach assumes perfect conditions and assumes that each symbol is written flawlessly. In reality, it is highly likely that some parts of one symbol may intersect with the vertical projection of another symbol. In such cases, the recursion would be terminated even though the segmentation process is incomplete. A convolutional neural network (CNN) was utilized for the purpose of classification. CNN belongs to the category of deep neural networks and is among the various deep learning algorithms that enable the utilization of self-optimizing weights and biases to recognize specific objects, thereby allowing the computer to possess a similar perception as humans. The key distinction between CNN and conventional neural networks lies in the inclusion of convolution. Convolution is a mathematical operation involving two functions that results in a third function indicating how one function's shape is altered by the other. In the context of CNN, convolution is employed to generate a feature map from the input data using specific filters and kernels. In numerical instances, these features may represent distinct curves or sharp angles. The computer's neural system closely resembles that of humans, comprising a network containing a vast number of neurons. Each neuron evaluates the received impulse and its significance to determine whether to activate and transmit the impulse through the network or to inhibit the flow by keeping the axon closed. Essentially, each neuron can be perceived as a combination of inputs, weights, and an activation function. Neurons vary based on their location within the network layer, with the input, hidden, and output layers constituting the fundamental layers in every neural network. While there is only one input and output layer, the number of hidden layers may vary depending on the network's structure and the nature of the problem. In the case of CNN, the network comprises convolutional, activation, pooling, and fully connected layers. The experimental architecture employed is [Conv − Relu − Pool] × 2 − [FC − Relu] × 2 − [FC], incorporating 2 × 2 max pooling with a stride length of 2, and the implementation was carried out using the PyTorch library.

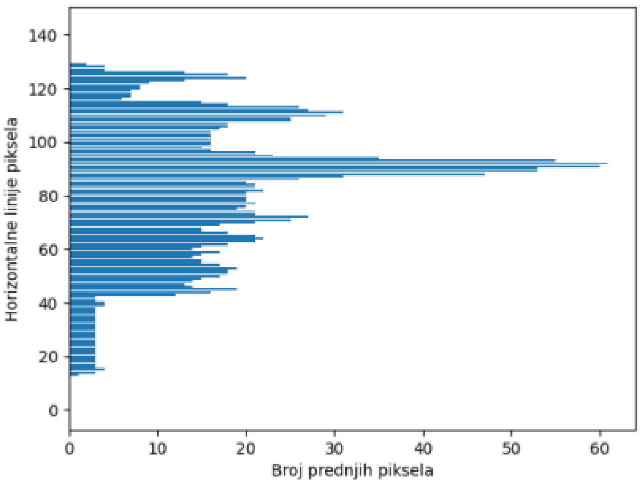

Figure 4 Horizontal Histogram

Despite being a relatively simple task for humans to recognize and group symbols from an image into an expression, this process proves to be more challenging for computers due to two main reasons. Firstly, not all symbols are written without lifting the hand from the paper, making it harder for a computer to accurately identify them. Secondly, a significant number of mathematical expressions have multiple dimensions, such as exponents, indices, and fractions. To address these challenges, we have developed a neural network that takes into account the multidimensional nesting characteristic of mathematical expressions. We have stored the distance of each symbol from the axis passing through the center of mass of the previous symbol. By analyzing this information, we can determine whether the current symbol is a continuation of the mathematical expression, an exponent, or an index of the previous symbol. To calculate the width and height of the symbols, we utilize the method of connected components. This involves identifying the northernmost, southernmost, westernmost, and easternmost pixels of each symbol and calculating the differences between them. When the axis of the previous symbol intersects with the current symbol, we classify it as a continuation of the mathematical expression. Conversely, if the axis of the previous symbol is either below or above the current symbol, we classify it as an exponent or an index of the previous symbol. However, the problem becomes more challenging when dealing with disconnected symbols. For the purpose of explanation, let's assume that the expression has no multidimensional nesting and that the baseline crosses every symbol in the expression. In this case, there are potential statements that require correction.

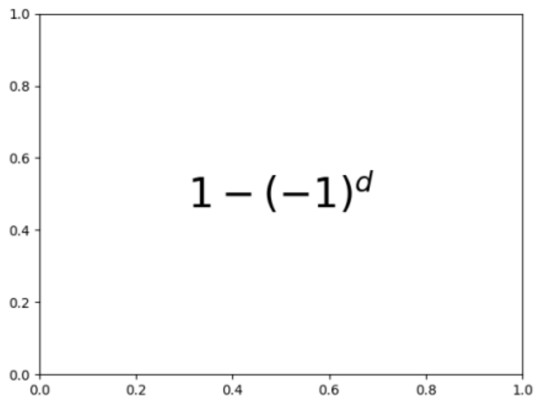

Figure 5 Final Expression

Table 1 Database

!	(	)	+	,	-	=	[	]	{
}	0	1	2	3	4	5	6	7	8
9	A	α	|	b	β	C	cos*	d	δ
$\div^*$	e	$\exists$	f	$\forall$	/	G	γ	$\geq$	$>$
H	i^*	$\in$	∞	$\int$	j^*	k	l	λ	$\ldots^*$
$\leq$	lim*	log	$<$	M	μ	N	$\neq$	o	p
ϕ	π	$\pm$	$\prime$	q	R	$\rightarrow$	S	σ	sin*
$\sqrt{x}$	$\sum$	T	tan*	θ	$\times$	u	v	w	x
y	z								

Table 2 Results

Results		
enrichment/symbols	all symbols	selected symbols
with enrichment	17.29%	29.90%
without enrichment	8.45%	13.17%

1.In the case where we encounter two minus signs with centers of mass within the range of each other along the x-axis, but differing along the y-axis, we can connect them and interpret them as a sign of equality. This is because we know that a series of two minuses is an incorrect mathematical notation and should not appear in any expression.
2. When it comes to trigonometric functions, if we come across a series of characters such as c, o, and s, we can combine them into "cos". The same principle applies to other trigonometric functions like "tan" and "log". For the sine function, we have a specific case where the letter "i" cannot be written without lifting the pen off the paper.
3. In this scenario, the classification string would consist of the characters "s", "|", and "n". We would consider this combination as "sin". In the case of the characters "<", "-", or ">", "-", if they appear in a string, we would classify them as "≤" and "≥". The reverse order also applies.
4. If there is a string of characters "+", "i", "-", or vice versa, they would be merged into the symbol "±".

The first part of the research focuses on comparing the accuracy of the system in classifying mathematical expressions written using all eighty-two possible symbols from the database, as well as expressions that do not contain certain symbols. It is important to note that the samples in the database are provided in XTML files, so it was necessary to convert them to jpg format before applying the algorithm described earlier.

III Results

The neural networks trained with widened and regular symbols exhibit similar accuracy levels of approximately 98% and 97% respectively. However, a significant difference in accuracy is noted when recognizing complete mathematical expressions. The original database contains symbols with a thickness of only one pixel, requiring expressions to be viewed from a distance to appear thinner. This is not the case in the database of written mathematical expressions, where symbols are proportionally larger in the image. Consequently, the expressions are magnified and appear bold. By doubling the symbol database and widening each symbol by one pixel, the neural network becomes invariant to symbol thickness and distances from which images were captured. The slight difference in neural network accuracy is attributed to the number of training examples used. Both networks were trained for three epochs without reaching saturation, indicating that the network with more training examples performed better. Despite the small difference in accuracy percentages, the impact on the overall system accuracy is negligible. The second part of the study focuses on the influence of specific symbols on system precision. Eight symbols commonly used in mathematics, posing complexity issues, were removed from the database. Regardless of symbol widening, a difference in system accuracy is observed due to the various ways trigonometric functions can be written, such as different representations of the sine function.

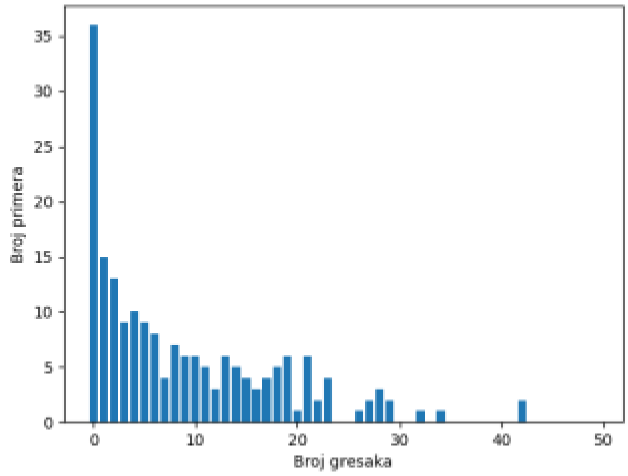

Figure 5 L Distance with all symbols

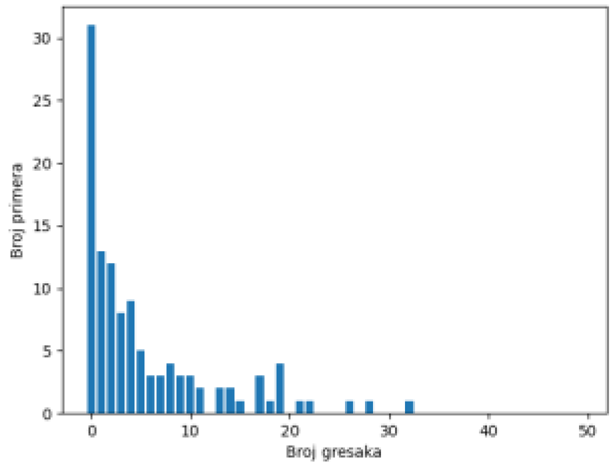

Figure 6 L distance with selected symbols

If individuals choose to write in small print, it is conceivable to associate sin with the symbol '('. Consequently, the neural network will struggle to accurately identify the expression sin(since it is not present in the symbol database. It is important to highlight that the overlapping of symbols does not necessarily pertain solely to trigonometric functions. In fact, the relatively low results primarily stem from the issue of overlapping symbols, as the computer cannot discern whether a series of intertwined lines belongs to a single symbol or a group of symbols. The final segment of the study focuses on the percentage accuracy of recognition. In the preceding sections, only the final output was compared to determine if the model aligns completely with the label. However, in this section, the Levenshtein distance was employed to measure the extent to which the system's outputs deviate from the exact given value. The Levenshtein distance calculates the minimum number of changes required in the output latex code to match the corresponding label [2]. Upon examining the provided graphs, it becomes evident that although the overall accuracy is relatively low, the number of examples decreases exponentially as the number of errors per example increases. Additionally, the average percentage accuracy of recognition for a single mathematical expression is 75%. This demonstrates that the utilization of this system would significantly expedite the aforementioned digitalization process

IV Conclusion

Testing neural networks on small samples reveals slight variations in their accuracy. These differences can be attributed to the distinct characteristics of each mathematical symbol, which enable the neural network to make precise classifications. The experiments conducted in this study utilized a convolutional neural network with two convolution-pooling layers and three fully connected layers. The ReLU activation function was employed throughout. Three different measurements were carried out in this research. The first measurement focused on the disparity in classification accuracy between neural networks trained with widened symbols and those trained with symbols of constant width. The findings demonstrated that a neural network trained with widened symbols exhibited twice the proficiency in classifying mathematical expressions compared to a network trained with symbols of constant width. This outcome is justified by the understanding that individuals possess unique writing styles and use different pens, resulting in variations in symbol width. The second measurement examined the precision of the system in relation to the specific mathematical symbols used in the expressions. It was observed that symbols with complex spatial structures and intricate LaTeX notations compromised the precision of both systems. These results align with previous research in the field. Additionally, the study employed the Levenshtein distance to assess the system's accuracy at the individual mathematical expression level, which yielded a 75% accuracy rate. This highlights the potential of the system to significantly expedite the process of digitization. Furthermore, the study identified the existence of overlapping symbols as a major challenge. To address this issue and enhance the classification accuracy of the system, future improvements could involve the development of an algorithm capable of effectively separating overlapping symbols.

REFERENCES

[1] Ink markup language, http://www.w3.org/TR/InkML/
[2] Levenshtein distance, https://en.wikipedia.org/wiki/Levenshtein distance [
3] Catherine Lu and Karanveer Mohan, Recognition of Online Handwritten Mathematical Expressions Using Convolutional Neural Network, cs231n project report Stanford 2015
[4] Md Bipul Hossain, Feroza Naznin, Y.A. Joarder, Md Zahidul Islam , Md Jashin Uddin, Recognition and Solution for Handwritten Equation Using Convolutional Neural Network, 2018
[5] Rajagopal, M., Ghate, S.N., P, R, Ganesh E.N. *et al.* Multi-class segmentation skin diseases using improved tuna swarm-based U-EfficientNet. *J. Eng. Appl. Sci.* **71**, 71 (2024). https://doi.org/10.1186/s44147-024-00399-6
[6] G. Mohan, Muhammadu Sathik Raja, S. Swathi, E.N. Ganesh,
A novel breast cancer diagnostic using convolutional squared deviation neural network classifier with Al-Biruni Earth Radius optimization in medical IoT system, e-Prime - Advances in Electrical Engineering, Electronics and Energy, Volume 7, 2024, 100440, ISSN 2772-6711, https://doi.org/10.1016/j.prime.2024.100440
[7] R.Sudharshan and E.N.Ganesh “A Swish RNN based customer churn prediction for the telecomindustry with a novel feature selection strategy” CONNECTION SCIENCE2022, VOL. 34, NO. 1, 1855–1876https://doi.org/10.1080/09540091.2022.2083584
[8] S. Nivash, E. N. Ganesh etal “Implementation and Analysis of AI-Based Gesticulation Control for Impaired People” Hindawi Wireless Communications and Mobile Computing Volume 2022, Article ID 4656939, 15 pages https://doi.org/10.1155/2022/4656939.
[9] Shanmugasundaram, Pradeep, E.N.Ganesh “Modelling and analysis of space vector pulse width modulated inverter drives system using MatLab/Simulink” International Journal of Advanced Intelligence ParadigmsArticle 2022
DOI: 10.1504/IJAIP.2022.123023
[10] B. Jaishanthi, E. N. Ganesh “Priority-based reserved spectrum allocation by multi-agent throughreinforcement learning in cognitive radio network” in AUTOMATIKA2019, VOL. 60, NO. 5, 564–569https://doi.org/10.1080/00051144.2019.1674512.
[11] Senthilkumar, E.N.Ganesh and Uma maheswari “Early and Accurate Model of Malignant Lung Nodule Detection System with Less False Positives” in Braz. arch. biol. technol.
61 • 2018 • https://doi.org/10.1590/1678-4324-2018160536
[12] A.R.Venmathi, E.N.Ganesh” Hybrid wavelet features for the classification of microca*lcification clusters in digital m*ammograms*” in* Journal of Computational and Theoretical NanoscienceArticle2017
DOI: 10.1166/jctn.2017.6495
[13] Prem Kmar. Rajkumar, Ganesh E.N. “Measurement of power radiation in Base Transceiver Station using Quad Phone and
Quadcopter” in Journal of Green Engineering Article 2016 DOI: 10.13052/jge1904-4720.522
[14] B.Sathyasri and Ganesh E.N. “Fauna inspired probabilistic and randomly computed channel assignment and multipath
Routing for multi-channel multi-radio mesh networks” Asian Journal of Information TechnologyArticle2016
DOI: 10.3923/ajit.2016.3883.3898