International Journal for Multidisciplinary Research (IJFMR)

E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

A CNN-Based Approach for Handwritten Mathematical Formula Recognition and LaTeX Generation

Savan Varotariya[1], Dr. Vikas Tulshyan[2]

[1,2]Silver Oak University Ahmedabad, India

Abstract

The recognition of handwritten mathematical formulas is a challenging task due to the complexity of mathematical symbols, varied notations, and diverse handwriting styles. This paper introduces an innovative Handwritten Mathematical Formula Recognition System (HMFRS) that leverages deep learning techniques to achieve high accuracy in recognizing handwritten formulas. The system comprises three key modules: pre-processing, feature extraction, and formula recognition. In the pre-processing stage, the handwritten formula image is enhanced to improve clarity and eliminate noise. The feature extraction module utilizes Convolutional Neural Networks (CNNs) to transform the pre-processed image into a feature representation, capturing both the local and global structure of the formula. We employ a comprehensive dataset consisting of handwritten mathematical formulas across different writing styles and symbol sets to train and evaluate the HMFRS. Experimental results demonstrate that our system outperforms existing methods in terms of accuracy and robustness. The proposed system holds promise for applications in areas such as digital document processing, e-learning platforms, and scientific research, where accurate recognition of handwritten mathematical formulas is essential. Additionally, we focus on recognizing and solving handwritten quadratic equations, employing horizontal compact projection analysis and combined connected component analysis for segmentation, followed by CNN-based character classification. The results validate the effectiveness of our approach in providing accurate solutions for mathematical equations.

Keywords: Convolutional Neural Networks (CNNs), Handwritten recognition, mathematical expressions, deep learning, LaTeX conversion, symbol classification, pattern recognition, image preprocessing, CROHME dataset, optical character recognition (OCR).

INTRODUCTION

1. Systems Overview

Although digital tools and media have revolutionized the way we write, many individuals continue to prefer handwriting, particularly for mathematical equations. While Optical Character Recognition (OCR) technology has made significant strides in recognizing printed text, recognizing handwritten mathematical formulas remains a challenging area of research. This study addresses two main tasks: identifying and isolating handwritten mathematical formulas from images and converting these formulas into a machine-readable format, specifically LaTeX. These tasks can be broken down into two subproblems: formula detection and the conversion of images into markup languages.

LaTeX is commonly used in academic and scientific writing, particularly in fields like physics, mathematics, and computer science, where mathematical expressions are central to research papers. Writing complex mathematical formulas in LaTeX is often more time-consuming than handwriting them. Therefore, this project seeks to explore methods for automating the typesetting of LaTeX equations from images of handwritten formulas. Neural networks have been proven effective for similar tasks, such as handwritten digit recognition. By leveraging this technology, we propose using Convolutional Neural Networks (CNNs) to develop a method for recognizing and converting handwritten math formulas into LaTeX code. Given that neural networks require large, high-quality datasets, we also introduce a process for generating labeled datasets containing images of mathematical formulas.

2. The Recognition Problem

The primary challenge of this project is to develop a system capable of accurately recognizing and interpreting handwritten mathematical formulas from scanned documents or images. The goal is for the system to decode handwritten symbols, equations, and expressions and then convert them into a digital format that can be analyzed, processed, or used in further calculations.

Several factors make this task difficult:

Variability in Handwriting: Handwriting can vary significantly between individuals, and even a single individual may have different writing styles or speeds. The system must accommodate these variations while maintaining accuracy.

Complexity of Mathematical Notations: Mathematical symbols are often intricate, with complex structures that must be interpreted correctly. The system must handle symbols like integrals, fractions, and square roots, which can be written in different ways.

Ambiguity and Errors: Handwritten formulas may have incomplete symbols, overwritten characters, or other errors. The system must be able to handle such ambiguities and still provide accurate interpretations of the intended mathematical meaning.

Diverse Layouts: Mathematical formulas can be written in various formats, such as inline, stacked, or multi-line equations. The system must be able to accurately interpret these different layouts and recognize the spatial relationships between the symbols.

Large Symbol Set: The system must recognize a wide variety of mathematical symbols, operators, and expressions, including Greek letters and other specialized notations, and convert them into a digital format.

The goal of the Handwritten Math Formula Recognition System is to overcome these challenges and provide an accurate, efficient method for converting handwritten mathematical formulas into a format that can be used for further analysis or computational tasks.

3. Advances Systems

Several factors motivate the development of a handwritten math formula recognition system, including the following potential benefits:

Increased Efficiency and Accuracy: Automating the process of converting handwritten formulas into digital representations reduces the time and effort needed for manual transcription. The system also minimizes human errors in interpreting complex mathematical expressions, ensuring more reliable outputs.

Educational and Accessibility Benefits: This system could be especially beneficial for students, educators, and researchers. By digitizing handwritten math notes and equations, it can assist in solving problems and creating educational resources. It can also improve accessibility for individuals with visual impairments by facilitating the use of text-to-speech tools or other assistive technologies.

Advancements in Scientific Research and Document Processing: The system could play a crucial role in scientific research by converting handwritten formulas in research papers, historical documents, or lab notes into digital form for analysis. It could also help in digitizing old math textbooks or notes, making them more accessible for modern use.

Integration with Mathematical Software: The recognition system could be integrated into existing mathematical software applications, such as equation solvers or tutoring platforms. This integration would allow users to input handwritten formulas directly and have them automatically converted into machine-readable formats.

Technological Growth: The development of this recognition system contributes to advancements in various technical fields, including computer vision, pattern recognition, and machine learning. Progress in this area could have broader applications in other areas of handwriting recognition and related technologies.

RELATED WORK

Handwritten mathematical formula recognition is a challenging research area in computer vision and pattern recognition due to the complex 2D spatial structure of mathematical notation and the variability of individual handwriting. The task typically involves symbol recognition, segmentation, and the structural analysis of expressions to produce accurate digital representations such as LaTeX.[9][10]

Early approaches relied on rule-based systems and handcrafted features. For instance, Okamoto and Miyazawa (1994) developed a grammar-based approach for parsing mathematical expressions, while Lavirotte and Pottier (2000) proposed a syntactic analyzer for 2D handwritten mathematics using graph grammars. These systems often struggled with variability in handwriting and symbol ambiguity.[9][10]

With the rise of machine learning, researchers began employing techniques such as Hidden Markov Models (HMMs) and Support Vector Machines (SVMs) to classify individual symbols and model their sequence. However, these approaches required extensive feature engineering and were not easily scalable.

The advent of deep learning significantly advanced the field. Convolutional Neural Networks (CNNs) have shown strong performance in symbol classification tasks due to their ability to automatically learn spatial hierarchies of features (LeCun et al., 1998). Further improvements were made using encoder-decoder architectures combined with attention mechanisms, which are capable of converting handwritten mathematical expressions into markup languages like LaTeX. Zhang et al. (2017) introduced a deep CNN with attention for handwritten math formula recognition, outperforming traditional rule-based systems.[11]

Recent work continues to explore various pipelines and architectures to improve recognition accuracy and processing speed. Table 1 below presents a comparative summary of several methodologies used in the domain, outlining their main phases and observed outcomes.[11]

Methodology Used	Results
1) Convolutional Neural Network (CNN), 2) Semi-Incremental, 3) Incremental, 4) Lines and Words Segmentation, 5) Parts-based Method, 6) Slope and Slant Correction, 7) Ensemble, 8) Zoning	CNN has excellent accuracy. Other methods like Ensemble, Zoning, and Segmentation are reliable.
Convolutional Neural Networks (CNN), 5-phase pipeline: 1) Data set enrichment, 2) Image	CNN is effective for recognizing handwritten expressions; outperforms SVM by 3-4%.

segmentation, 3) Data extraction, 4) Character-level classification, 5) Expression-level classification	
Pipeline with 4 phases: 1) Segmentation, 2) Symbol Classification, 3) Grouping of Symbols, 4) LaTeX Format using SVM and ELM classifiers	Symbol recognition accuracy: SVM ~90%, ELM ~95%.
1) Data Collection, 2) Preprocessing, 3) Segmentation, 4) Classification	CNN achieved 88% accuracy for isolated symbols.
Feedforward backpropagation neural network with 4 stages: 1) Training, 2) Preprocessing, 3) Segmentation, 4) Feature Extraction	Improved recognition rate and processing time, with better accuracy for straight line equations.
End-to-end solutions, Integrated solutions, Sequential solutions focusing on DML techniques	Growing interest in online HME recognition, with advancements in minimizing recognition errors.
Recurrent Neural Networks (RNNs) with LSTM blocks for online recognition	BLSTM outperforms MQDFs, although CNNs show better performance in certain conditions.

Table 1. Summary of existing handwritten mathematical expression recognition methods and their performance

This comparison highlights the dominance of deep learning-based methods such as CNNs, which consistently outperform traditional classifiers like SVM and MQDF. Additionally, ensemble techniques and multi-phase pipelines that incorporate preprocessing, segmentation, and symbol grouping have proven effective in boosting recognition accuracy. Researchers are increasingly exploring hybrid models that combine CNNs with recurrent networks or attention mechanisms for better context awareness and structure parsing.[12]

Competitions like CROHME (Competition on Recognition of Online Handwritten Mathematical Expressions) have played a key role in benchmarking these systems, pushing the field forward with publicly available datasets and evaluation metrics (Mouchère et al., 2016). Commercial solutions such as MyScript and academic tools like Detexify reflect the practical demand for accurate handwriting-to-LaTeX conversion, especially in education and digital documentation.[12]

This project builds upon these advancements by implementing a deep learning-based pipeline designed to recognize handwritten mathematical expressions from scanned images or photos, leveraging CNNs for symbol classification and a structured approach to parsing expressions into LaTeX format.[11]

PROPOSED ARCHITECTURAL FRAMEWORK

A. Introduction

Handwritten mathematical expression recognition (HMER) methods typically consist of three main components: symbol segmentation, symbol recognition, and structure analysis (Chan & Yeung, 2000). Convolutional Neural Networks (CNNs) are highly effective in learning and extracting features directly from raw image data (Krizhevsky et al., 2017). These networks convert low-level features into high-level representations, enabling CNNs to perform complex tasks in computer vision. To address the challenge of insufficient coverage, Zhang et al. (2017) introduced the WAP encoder-decoder model, which incorporates a CNN encoder based on the Visual Geometry Group (VGG) architecture and a decoder utilizing a Recurrent Neural Network (RNN) with a Gated Recurrent Unit (GRU). This model encodes the input image of a mathematical expression to extract high-dimensional features and then decodes them into the corresponding LaTeX sequence.

Building on this, Zhang et al. (2018) proposed the DenseWAP model to improve upon CNN performance. In this approach, the VGG network in the WAP model is replaced with the Dense Convolutional Neural Network (DenseNet). Furthermore, Wu et al. (2019) introduced the PAL and PAL-V2 models, which combine deep learning with adversarial learning to address variations in handwriting styles. The PAL-V2 model utilizes a CNN-based decoder to tackle issues such as vanishing and exploding gradients often encountered in RNNs (Wu et al., 2020).[4]

B. Framework Overview

The architecture of the Handwritten mathematical expression recognition consists of the following key components:

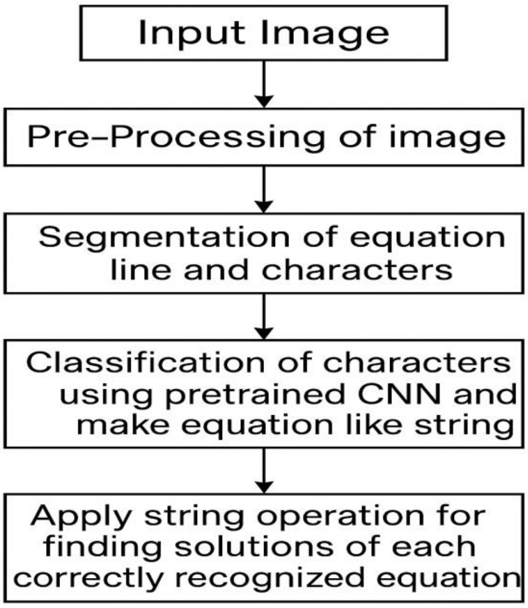

Figure 1. Flowchart of handwritten mathematical expression recognition process

1. Data augmentation:

Data augmentation is a technique used in data analysis to expand the size of a dataset by creating new, slightly altered versions of the existing data or generating synthetic data. This process involves applying various transformations such as geometric modifications, flipping, cropping, rotation, noise injection, and random erasure to enhance images. The goal is to improve the robustness and generalization of machine learning models by exposing them to a wider variety of data. After applying data augmentation, the dataset is typically split into training and testing sets for further analysis and model evaluation.[2][6]

2. Data preprocessing:

Data preprocessing is essential for preparing raw handwriting data for machine learning. It includes steps like resizing images, normalizing pixel values, and conducting exploratory data analysis (EDA) to understand the dataset. These processes ensure that the data is consistent and ready for model training.[6]

2.1 Noise Removal: Before moving to the next stage, the images must undergo preprocessing to enhance clarity and ensure proper spacing between edges, characters, and other elements. This step may involve various techniques, some of which are categorized as morphological operations, to improve the image quality for further analysis.[2]

2.2 Image Segmentation: To accurately classify symbols, it's crucial to identify which traces form a complete symbol. A simple heuristic is used, where intersecting traces are assumed to belong to the same symbol. However, this approach may fail in two cases: when symbols overlap or when symbols like "=", "log", "sin", "cos", etc., don't overlap but are still part of the same symbol. To address this, an additional step is added to handle non-overlapping strokes. If the classifier identifies the strokes as a specific symbol with a probability above 0.7, it is segmented as that symbol. Segmentation is a vital step in Handwritten Mathematical Expression (HME) recognition, as it involves dividing the image into smaller, homogeneous components like numbers, letters, and symbols. This process is crucial for the recognition and categorization of symbols in a mathematical expression.[2][7]

2.3 Feature extraction: Feature extraction involves representing each character as a feature vector, which acts as its identifier during the recognition process. The main goal is to select a set of features that maximizes recognition accuracy while minimizing the number of elements used. However, this task is challenging due to the inherent variability and imprecision of handwriting. In our approach, InkML data, which consists of individual mathematical symbols, is converted into a format suitable for model input. This involves transforming the data into normalized images represented as pixel arrays. To improve recognition, we apply pixel blurring, which alters the pixel values from binary (0s and 1s) to a distribution centered around 1, with added noise that gradually decreases as the distance from the pixel increases, based on Euclidean distance. For CNN models, the number of strokes is included as a feature in the final affine layer.[2][7]

3. CNN Model:

Convolutional Neural Networks (CNNs) are employed in this project to effectively recognize handwritten mathematical expressions by learning spatial hierarchies of features from input images. Prior to classification, online patterns are preprocessed and resized to 48×48 pixel grayscale images with a standardized line thickness of 3 pixels to ensure uniformity across training samples..[8]

The CNN architecture consists of multiple layers that are organized to progressively extract, condense, and interpret image features:

- Input Layer: Acts as a buffer, accepting preprocessed image data and preparing it for convolution operations.
- Convolutional Layers: Apply trainable filters (kernels) across the input image to generate feature maps. Each kernel extracts specific visual patterns such as edges, corners, or textures relevant to mathematical symbols.
- ReLU Activation: Introduces non-linearity by replacing negative values in the feature maps with zero. This helps the model learn complex patterns more effectively.
- · · Pooling Layers: Perform dimensionality reduction through operations like MaxPooling, making the extracted features more robust to local distortions and shifts, and reducing computational overhead.
- Fully Connected Layers (Dense Layers): After feature extraction, these layers interpret the condensed representations and perform symbol classification. The final layer uses a Softmax activation function to output the probability distribution over predefined symbol classes.

The hierarchical structure of CNNs allows the system to autonomously learn relevant visual features without requiring manual engineering, making them well-suited for recognizing the diverse and spatially complex symbols found in handwritten mathematical expressions.

4. Classification:

The classification stage involves using a Convolutional Neural Network (CNN) as the classifier, specifically a model named SpNet-CNN, which can recognize 83 distinct classes. The architecture of SpNet-CNN consists of several layers designed to efficiently extract and process features from input images. It begins with an input layer, followed by multiple convolutional layers that apply a 3x3 filter to detect patterns. These are interspersed with max pooling layers, which use a 2x2 subsampling size to reduce the spatial dimensions, effectively halving the image size at each step. After feature extraction, a fully connected layer processes the information, and the final softmax layer assigns probabilities to different symbol classes for accurate recognition. This structured design ensures high performance in symbol classification.

5. Conversion to LaTex expression:

The system creates the recognised math formula as output, which can then be printed, shown on a screen, or processed further for use in other applications like solving the math equation, exporting it in LaTeX or MathML format, or being stored in a database.

Experimental Results

To evaluate the performance of the proposed handwritten mathematical formula recognition system, two sets of experiments were conducted: binary classification and multiclass classification. These experiments were performed using a custom CNN architecture and a dataset derived from the CROHME dataset, which includes various handwritten mathematical expressions.

A. Binary Classification

In the binary classification experiment, the system was trained to differentiate between two selected mathematical symbols. The CNN model was designed with an input shape of (256, 256, 3). It included:

- Convolutional layers with ReLU activation
- A MaxPooling2D layer with pool size (3,3)
- Dense layers with 64 neurons
- A final layer with a sigmoid activation function

The model was compiled using the Adam optimizer and binary cross-entropy loss. Training was conducted over 10 epochs. The accuracy improved steadily over each epoch, while both training and validation loss consistently decreased, indicating good model convergence and learning capacity.

B. Multiclass Classification

For the multiclass experiment, the model was extended to recognize five distinct formula types, labeled using placeholder tags such as “fom,” “kom,” “som,” etc. The final dense layer was modified to have five output neurons with a softmax activation function, appropriate for multiclass classification tasks.

Like the binary model, this CNN used Conv2D and MaxPooling layers followed by flattening and dense layers. Training was performed with the same configuration (10 epochs), but performance exhibited more variability. Although accuracy increased over time, the loss curve fluctuated, suggesting that the model was more sensitive to the increased complexity of classifying multiple classes.

C. Observations

The binary classifier achieved stable accuracy, with a clear and consistent reduction in both training and validation loss.

The multiclass model, while effective, showed non-uniform loss reduction and greater variability in accuracy across epochs.

Both models demonstrate that CNNs are capable of learning symbol-level features from handwritten expressions.

Preprocessing techniques such as resizing, normalization, and noise reduction contributed significantly to improved results.

Model Type	Accuracy	Loss Trend	Output Activation
Binary	~80-90%	Decreasing consistently	Sigmoid
Multiclass	~80-85%	Fluctuates, improves overall	Softmax

Table 2. Comparison of binary and multiclass classification models

CONCLUION

Handwritten math formula recognition is a crucial technology that facilitates the automatin, digitization, and interpretation of handwritten mathematical expressions. This technology is particularly valuable in applications that require an understanding of mathematical notation, such as automated grading, educational tools, and accessibility solutions for visually impaired individuals. Various approaches, including rule-based methods, machine learning, and deep learning, have been employed to improve recognition accuracy and efficiency. While rule-based systems rely on predefined rules to classify symbols, machine learning techniques enhance recognition by training models on large datasets. Deep learning, particularly neural networks, has revolutionized this field by enabling automatic learning of complex structures and relationships between mathematical symbols.

Despite these advancements, handwritten math formula recognition remains a challenging task due to variations in individual handwriting styles, the complexity of mathematical expressions, and the need for precise spatial interpretation of symbols. However, continuous improvements in deep learning and artificial intelligence have significantly enhanced the accuracy and robustness of these systems. Modern recognition techniques now achieve high levels of precision, making them more reliable and effective in real-world applications.

The impact of handwritten math recognition extends beyond academic settings. It plays a vital role in automating grading systems, assisting educators and students in learning environments, improving accessibility for individuals with disabilities, and streamlining data entry in research and academic institutions. As research in this field progresses, further innovations are expected to refine recognition algorithms, enhance processing speeds, and expand the applicability of this technology. Ultimately, handwritten math formula recognition will continue to contribute to advancements in education, accessibility, and computational automation, making mathematical interpretation more efficient and widely accessible.

REFERENCES

1. S. S. Rosyda and T. W. Purboyo, “A review of various handwriting recognition methods,” International Journal of Applied Engineering Research, vol. 13, no. 2, pp. 1155–1164, 2018.
2. IRJET, “Recognition of handwritten mathematical expression and using machine learning approach,” International Research Journal of Engineering and Technology (IRJET), 2021.
3. D. H. Nguyen, M. Nakagawa, and A. L. Duc, “Recognition of online handwritten math symbols using deep neural networks,” IEICE Transactions on Information and Systems, Oct. 2016.
4. Z. Ye, “An intelligent-detection network for handwritten mathematical expression recognition,” Scho-

ol of Information Science and Technology, Fudan University, China, unpublished.
5. W. Zhao and L. Gao, “Handwritten mathematical expression recognition with bidirectionally trained transformer,” unpublished.
6. S. Nasi and S. H. N., “Offline handwritten mathematical expression recognition using CNN and Xception,” International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 142, unpublished.
7. C. Lu and K. Mohan, “Recognition of online handwritten mathematical expressions using convolutional neural networks,” Stanford University, unpublished.
8. D. H. Nguyen, A. L. Duc, and M. Nakagawa, “Recognition of online handwritten math symbols using deep neural networks,” IEICE Transactions on Information and Systems, 2016.
9. M. Okamoto and K. Miyazawa, “Recognition of printed mathematical expressions using structure analysis,” IEICE Transactions on Information and Systems, vol. E77-D, no. 7, pp. 749–755, 1994.
10. S. Lavirotte and L. Pottier, “Mathematical formula recognition using graph grammar,” in Proc. Int. Conf. Document Analysis and Recognition (ICDAR), 2000.
11. Z. Zhang, J. Du, and L. R. Dai, “Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition,” Pattern Recognition, vol. 71, pp. 196–206, 2017.
12. H. Mouchère, R. Zanibbi, U. Garain, and C. Viard-Gaudin, “Advancing the state of the art for handwritten math recognition: The CROHME competitions, 2011–2014,” International Journal on Document Analysis and Recognition, vol. 19, no. 2, pp. 173–189, 2016.