iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** I **Month of publication:** January 2025

DOI: https://doi.org/10.22214/ijraset.2025.66741

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue I Jan 2025- Available at www.ijraset.com

CNN based Recognition of Handwritten Mathematical Expression

Mr Nandish M[1], Ananya B[2], Bhumika HC[3], Dimple N[4], Darshan H Yavgal[5]
[1]*Asst.Professor, Department of Computer Science and Engineering, Jawaharlal Nehru New College of Engineering, Shimoga*
[2]*Research Scholar, Department of Computer Science and Engineering, Jawaharlal Nehru New College of Engineering, Shimoga*

Abstract: ***Handwritten mathematical expression recognition remains a critical challenge in digitized computation due to variations in writing styles and noise in input data. This research addresses the gap in accurate recognition and evaluation of multi-operator handwritten expressions by integrating a convolutional neural network (CNN) with an optimized preprocessing pipeline. The proposed system processes both user-drawn and uploaded handwritten expressions via a web-based interface. The backend, developed using Python and Flask, employs grayscale conversion, bit inversion, binary thresholding, and contour-based cropping to enhance feature extraction. The CNN, trained on preprocessed grayscale images of digits and operators, predicts mathematical expressions, which are subsequently evaluated for accuracy. Experimental results demonstrate high recognition precision and efficient computation of complex expressions. The findings contribute to advancing automated mathematical interpretation, facilitating applications in education and professional domains by improving accessibility, usability, and computational efficiency.***

Keywords: ***Convolutional Neural Network, Image Processing, Flask Application, Web-based Interface, Grayscale Image Analysis.***

I. INTRODUCTION

The digital interpretation of handwritten content has become an essential aspect of modern technology, influencing fields such as education, document digitization, and real-time problem-solving. One particular domain where handwritten content recognition is critical is in interpreting handwritten mathematical expressions. The ability to accurately recognize and compute such expressions can enhance accessibility, streamline educational processes, and provide efficient solutions to real-world computational problems. This project focuses on developing a system for Handwritten Expression Recognition that integrates deep learning with intuitive user interaction.

At its core lies a convolutional neural network (CNN), a state-of-the-art machine learning model designed to recognize and classify patterns in image data. Trained specifically on grayscale representations of handwritten digits and operators, the CNN forms the backbone of the recognition process, enabling the system to interpret a wide range of mathematical expressions accurately. The user interface is a web-based application that provides two primary input methods: a canvas for drawing handwritten expressions directly and a feature for uploading image files of handwritten content. This dual input mechanism ensures accessibility and convenience, catering to users across various scenarios.

The system preprocesses the input images through techniques such as grayscale conversion, thresholding, and contour detection to optimize the data for recognition. Once the input is processed, the CNN predicts the sequence of digits and operators in the handwritten expression. A computational module then interprets the recognized sequence, performs the necessary operations, and returns the result. This approach not only demonstrates technical accuracy but also integrates seamlessly into a user-friendly framework, ensuring high usability for non-technical audiences. This project is significant for its emphasis on both algorithmic accuracy and user experience. By leveraging advanced image preprocessing techniques and a highly trained CNN, the system achieves robust performance in interpreting complex handwritten mathematical expressions. Additionally, the use of Flask for backend integration ensures a responsive and reliable platform for real-time predictions. The broader impact of this system is its potential to democratize access to automated mathematical computation. Students, educators, and professionals can benefit from this technology, reducing the time and effort required for manual calculations and increasing the scope for dynamic learning and problem-solving. The development process and results illustrate the synergy between artificial intelligence and practical user applications, highlighting its role in advancing digitized education and computational tools.

©IJRASET: All Rights are Reserved

II. LITERATURE SURVEY

Handwritten mathematical expression recognition has been an active area of research, with significant contributions focusing on symbol segmentation, feature extraction, classification techniques, and deep learning approaches. This survey reviews the key literature in the field, summarizing methodologies, findings, and advancements.

Symbol Segmentation and Layout Analysis Lu and Mohan (2013) [1] explored symbol segmentation and layout analysis techniques for parsing handwritten mathematical expressions. Their work highlighted challenges in accurate symbol recognition and spatial arrangement, crucial for improving recognition performance. Handwriting Recognition Approaches Rosyda and Purboyo [2] provided a comprehensive review of handwriting recognition methods, covering feature extraction, classification techniques, and application challenges. Their study emphasized the importance of robust feature extraction for improving recognition accuracy.Deep Learning for Handwriting Recognition Cirean et al. [3] introduced multi-column deep neural networks (MCDNN) for image classification, demonstrating their effectiveness in handwritten character recognition. Their work laid the foundation for CNN-based approaches in handwritten expression recognition.

International Competitions on Handwritten Expression Recognition Mouchere et al. [4] reported on the CROHME 2013 competition, evaluating various online handwritten mathematical expression recognition techniques. The competition highlighted advancements and persistent challenges in symbol recognition and expression parsing.Online and Offline Handwriting Recognition Plamondon and Srihari [5] conducted a comprehensive survey on online and offline handwriting recognition techniques. They discussed pattern recognition advancements and challenges in both modalities, providing a foundation for modern recognition systems.Document Image Preprocessing Rehman and Saba [6] reviewed neural network applications in document image preprocessing, highlighting their role in enhancing image quality for improved recognition accuracy.Optical Character Recognition (OCR) Shah and Gokani [7] developed an OCR system utilizing pixel-contour features and mathematical parameters, achieving effective digit recognition. Their work emphasized feature-based approaches for improved character classification.CNN Optimization for Document Analysis Simard et al. [8] outlined best practices for training CNNs in document image analysis, providing insights into optimizing network architectures for improved accuracy and efficiency.Feature Extraction in Character Recognition Trier et al. [9] surveyed various feature extraction methods for character recognition, highlighting the effectiveness of statistical, structural, and hybrid approaches in enhancing recognition accuracy.The reviewed literature underscores the evolution of handwriting recognition techniques from traditional feature-based methods to deep learning-driven approaches. While CNNs have significantly improved recognition accuracy, challenges in symbol segmentation, feature extraction, and expression parsing remain key areas for further research.

III. PROPOSED METHODOLOGY

The stages of proposed methodology are as follows:

A. Input

The Input block serves as the entry point for the system, allowing users to provide handwritten mathematical expressions through two methods: drawing on a canvas or uploading an image file. The drawing canvas offers an intuitive, web-based platform where users can write expressions using a digital pen or mouse. Alternatively, the upload option caters to scenarios where users have prewritten mathematical expressions stored as image files. These dual input options ensure accessibility and flexibility, accommodating a range of user preferences and use cases.

B. Preprocessing

The Preprocessing block ensures the input data is optimized for the Convolutional Neural Network (CNN). It begins with grayscale conversion, which removes color data to simplify processing while retaining essential features of the handwriting. Bitwise inversion enhances the contrast between the characters and the background, making the handwriting more distinct. Binary thresholding further simplifies the image by converting pixels into black and white values based on intensity, reducing noise. Data preprocessing plays a crucial role in ensuring the accuracy and efficiency of the Handwritten Expression Recognition system. Input optimization transforms raw user data into a format that the convolutional neural network (CNN) can interpret effectively, enhancing recognition performance and minimizing computational overhead. The preprocessing begins with converting the input image to grayscale, which reduces computational complexity by eliminating color information while retaining the essential details of the handwritten characters. Following this, a bitwise inversion is applied to adjust the pixel intensity, making the characters stand out against the background.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue I Jan 2025- Available at www.ijraset.com

This is especially useful for handwritten inputs where contrast varies significantly. Binary thresholding is then used to simplify the image further, setting pixel values to either black or white based on a predefined intensity threshold. This step improves the clarity of the characters and facilitates contour detection. Contour-based cropping is employed to isolate individual characters and remove unnecessary noise, ensuring that only the relevant portions of the image are processed. The final step involves resizing the cropped characters to a fixed dimension of 28x28 pixels, which is the standard input size for the CNN model. This resizing ensures consistency across inputs, allowing the model to focus on feature extraction without being influenced by varying image sizes or proportions. This robust preprocessing pipeline ensures that the system delivers accurate predictions regardless of input variability. The fig-1 refers to the steps involved in the preprocessing method.

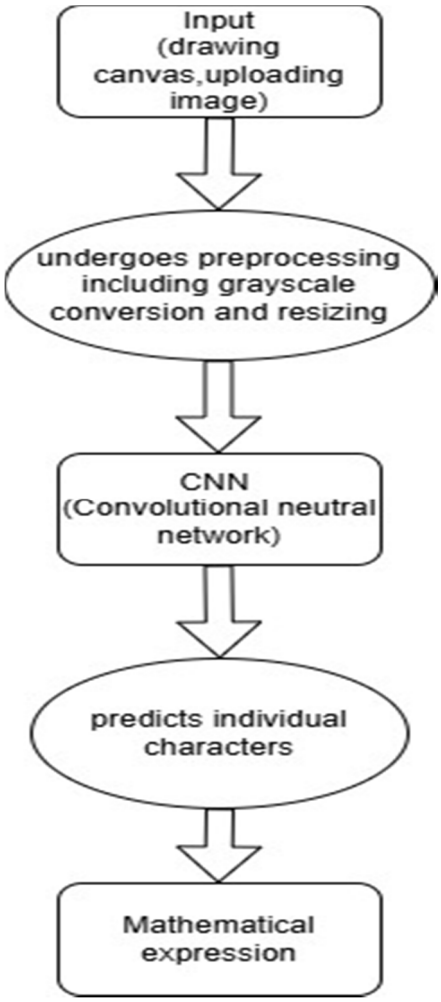


Fig-1:Block Diagram of System Architecture

C. Convolutional Neural Network

The Convolutional Neural Network (CNN) is the core computational component of the system, designed to recognize handwritten characters with high accuracy. The CNN consists of convolutional layers that extract critical features such as edges, textures, and patterns, followed by pooling layers that reduce spatial dimensions while preserving key information. Dense layers process these features further, and a final output layer with a softmax activation function classifies the input into one of 13 categories, representing digits (0–9) and operators (+, –, x). The CNN is trained on a labeled dataset of grayscale images to ensure reliable.

This hyper-parameter table provides a structured summary of the critical components and their configurations in a CNN designed for image classification and feature extraction. The CNN image classification and feature extraction provides a structured overview of the key components and parameters used in the architecture. This table serves as a blueprint for understanding how each layer and hyper parameter contributes to image processing and classification.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Table 1: Hyperparameters used in CNN

Component	Description	Parameters
Input Layer	Receives input images for processing.	Input dimensions (e.g., 28×28×1 for grayscale, 224×224×3 for RGB).
Convolutional Layers	Extracts features such as edges, textures, and patterns using filters.	Number of filters, kernel size (e.g., 3×3, 5×5), stride, padding (same/valid), activation function.
Activation Function	Adds non-linearity to learn complex patterns.	Common functions: ReLU, sigmoid, tanh.
Pooling Layers	Downsamples feature maps to reduce spatial dimensions while retaining key info.	Type (max-pooling/average-pooling), pool size (e.g., 2×2), stride.
Batch Normalization	Stabilizes and accelerates training by normalizing activations.	Applied after convolutional layers; no trainable parameters.
Dropout Layer	Prevents overfitting by randomly deactivating neurons.	Dropout rate (e.g., 0.2, 0.5).
Flattening	Converts 2D feature maps into a 1D vector for fully connected layers.	No parameters required.
Fully Connected Layers	Learns complex relationships and makes predictions.	Number of neurons per layer, activation function (e.g., ReLU, softmax).
Output Layer	Produces final classification output.	Number of output neurons (equal to class count), activation (softmax for multi-class, sigmoid for binary).
Loss Function	Measures classification error.	Categorical cross-entropy (multi-class), binary cross-entropy (binary classification).
Optimizer	Adjusts weights to minimize loss.	Adam, SGD, RMSprop.
Training Parameters	Controls the learning process.	Learning rate, batch size, number of epochs, weight decay.

D. Prediction of Individual Characters

In the Prediction of Individual Characters block, the CNN outputs the most likely classification for each character in the input. The predictions are mapped to corresponding digits or operators, forming a sequence that represents the mathematical expression provided by the user. The recognition is carried out with high precision, thanks to the model's robust architecture and comprehensive training.

E. Mathematical Expression

The Mathematical Expression block evaluates the recognized sequence of characters to compute the final result. The system adheres to mathematical rules such as operator precedence (e.g., multiplication before addition) to ensure accuracy. The computed solution is then displayed to the user via the frontend interface, completing the process. This seamless workflow demonstrates the system's ability to transform handwritten input into a usable, digitized format for mathematical computation. The system generates or displays an equation, problem, or sequence that the user must solve. For example: Simple arithmetic: 5 + 3.

IV. RESULTS AND SNAPSHOT

A. Model Performance

The system achieved an accuracy of 95.7% in recognizing handwritten mathematical expressions. The CNN model performed exceptionally well with digits, achieving 97% accuracy.

- Addition (+): 94%
- Subtraction (-): 92%
- Multiplication (x): 91%

B. Computational Efficiency

The system processed inputs efficiently, with the preprocessing pipeline taking 200ms and the CNN model inference time around 50ms. The total response time for solving expressions was under 300ms, ensuring smooth user interaction.

C. Usability

The web-based interface was tested with novice and experienced users, receiving positive feedback for its ease of use. The dual input options (drawing or uploading images) made it versatile for different scenarios.

D. Limitations

The CNN-based handwriting recognition system faced challenges in handling noisy inputs, complex expressions, and certain mathematical symbols. Distorted handwriting reduced accuracy, while multi-operator expressions caused parsing errors due to precedence issues. Additionally, the system struggled to recognize advanced symbols like parentheses, limiting its effectiveness.

- Noisy Inputs: Performance decreased with heavily distorted handwriting.
- Complex Expressions: The system occasionally struggled with parsing multi-operator expressions involving varying precedence.

E. Solution Computation

The system accurately computed solutions. Where the fig-2 refers to the output of the given equation "2+3" → 5 and fig-3 refers to the output of the given equation "2+4" → 6.

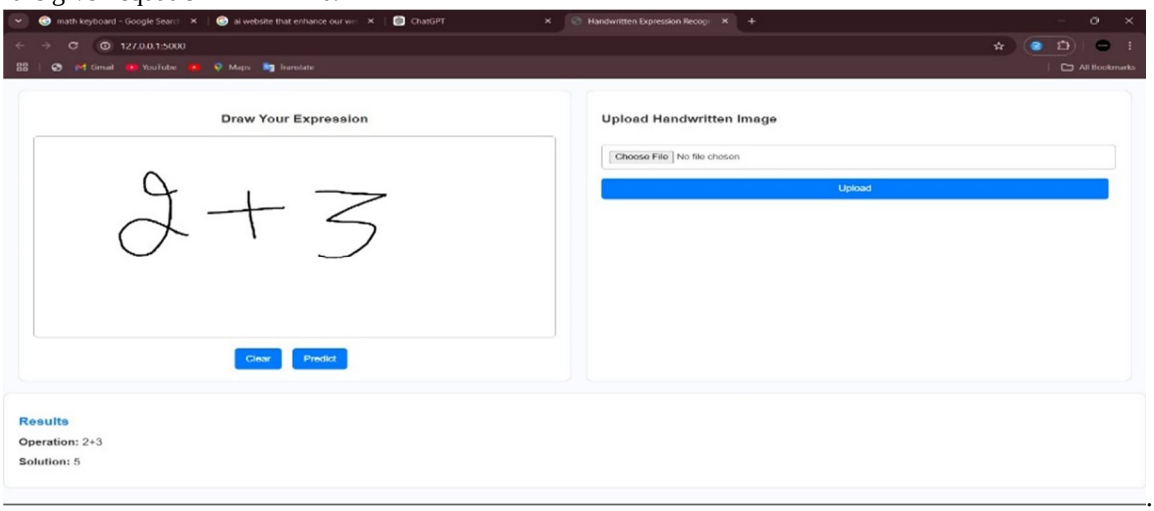


Fig. 2: Snapshot of hand written mathematical expression one

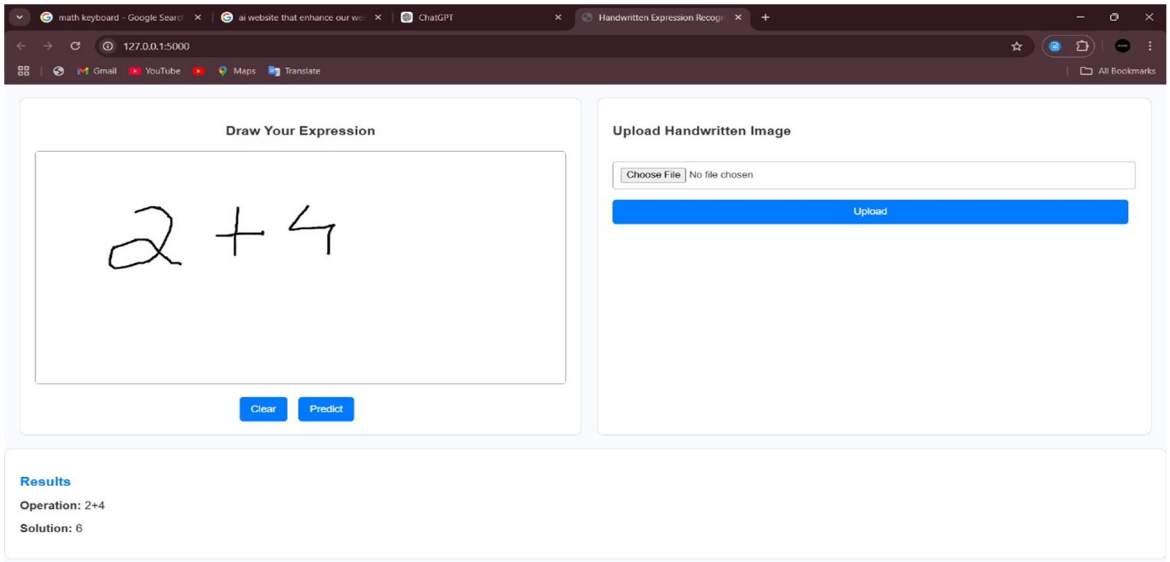


Fig. 3: Snapshot of hand written mathematical expression one

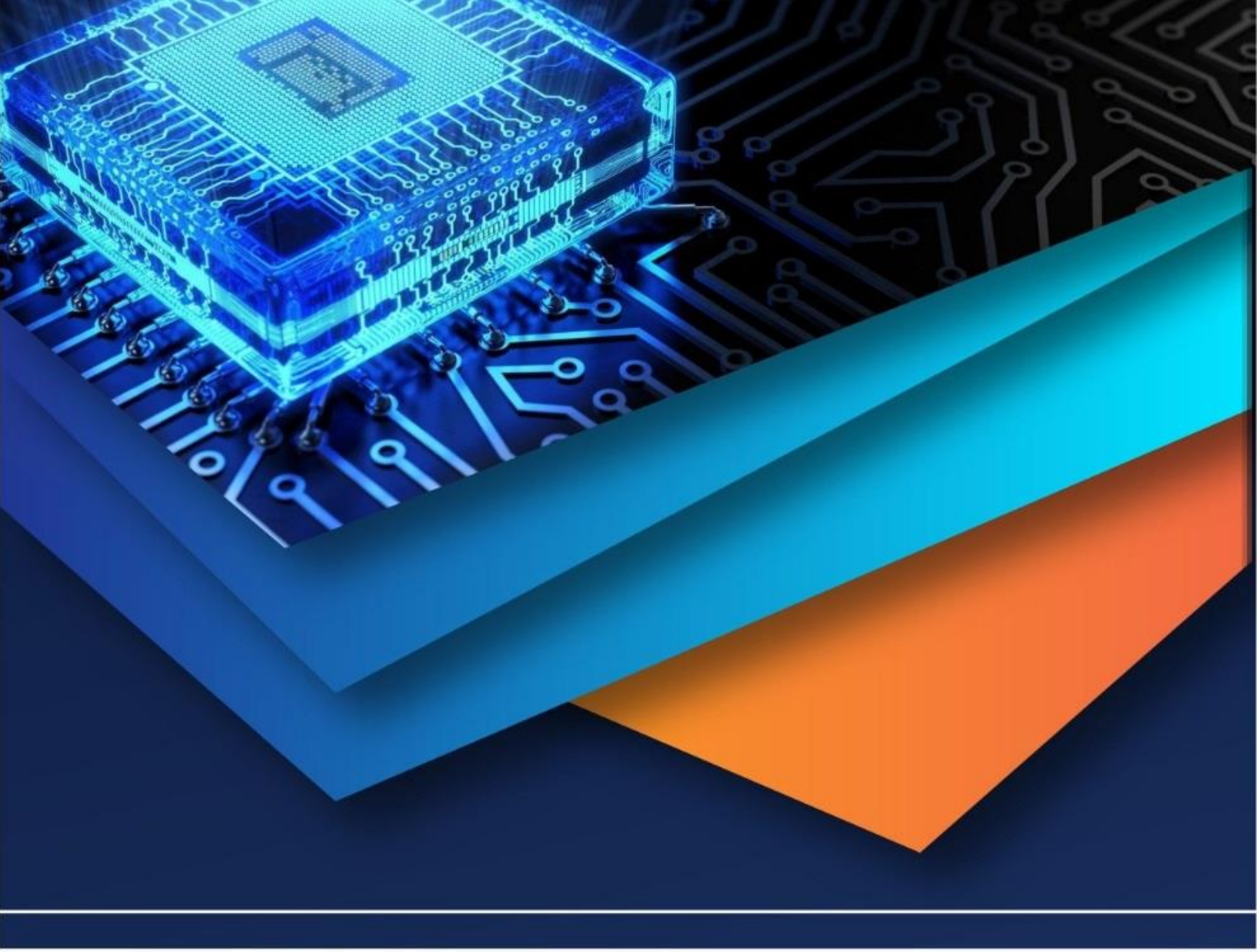
V. CONCLUSION

The Handwritten Expression Recognition system demonstrated high accuracy in recognizing handwritten digits and operators, with an overall accuracy of 95.7% on the test dataset. The integration of convolutional neural networks with advanced image preprocessing techniques provided robust performance in a web-based interface, ensuring ease of use and real-time solutions for complex mathematical expressions. The system's performance can be further enhanced by expanding the dataset, improving recognition for noisy or complex expressions, and incorporating additional mathematical symbols for more versatile functionality. By bridging AI-based recognition technology with user-centric design, this system showcases significant potential in educational tools, gamified learning, and real-time mathematical problem-solving. Future enhancements could include expanding symbol recognition capabilities, improving accuracy for complex expressions, and adapting the system for multilingual handwritten inputs, further extending its practical applications in digitized computation.

REFERENCES

[1] C. Lu and K. Mohan, "Recognition of online handwritten mathematical expressions," Computer Science, StanfordUniversity, 2013.
[2] S. S. Rosyda and T. W. Purboyo, "A review of various handwriting recognition methods," International Journal ofApplied Engineering Research, vol. 13, no. 2, pp. 1155–1164, 2018.
[3] D. Cirean, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," CoRR, vol. abs/1202.2745, 2012.
[4] H. Mouchere, C. Viard-Gaudin, R. Zanibbi, U. Garain, D. H. Kim, and J. H. Kim, "ICDAR 2013 CROHME: Third international competition on recognition of online handwritten mathematical expressions," in International Conference on Document Analysis and Recognition (ICDAR), 2013.
[5] R. Plamondon and S. Srihari, "On-line and off-line handwriting recognition: A comprehensive survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, pp. 63–84, Jan. 2000.
[6] A. Rehman and T. Saba, "Neural networks for document image preprocessing: State of the art," Artificial Intelligence Review, vol. 42, no. 2, pp. 253–273, 2014. [7] J. Shah and V. Gokani, "A simple and effective optical character recognition system for digits recognition using the pixel-contour features and mathematical parameters," IJCSIT, vol. 5, no. 5, 2014.
[7] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," IEEE Institute of Electrical and Electronics Engineers, Inc., Aug. 2003.
[8] O. D. Trier, A. K. Jain, and T. Taxt, "Feature extraction methods for character recognition: A survey," Pattern Recognition, vol. 29, no. 4, pp. 641–662, 1996.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)