ISSN: 2582-7219

International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Impact Factor: 8.206

Volume 8, Issue 5, May 2025

© 2025 IJMRSET | Volume 8, Issue 5, May 2025| DOI:10.15680/IJMRSET.2025.0805215

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Building a Handwritten Multi-digit Calculator using CNN

Yenumula Pavani

Department of ECE, RVR&JC College of Engineering, Guntur, India

Thottempudi Triveni

Department of ECE, RVR&JC College of Engineering,Guntur,India

Shaik Dilruba

Department of ECE, RVR&JC College of Engineering, Guntur, India

ABSTRACT: This project aims to develop a handwritten multi-digit calculator that leverages the power of Convolutional Neural Networks (CNNs) to recognize and process handwritten digits and arithmetic operators. The CNN model is trained on an extended version of the MNIST dataset, which includes not only digits but also symbols representing basic arithmetic operations such as addition, subtraction, multiplication, and division. The system captures and interprets sequences of handwritten characters to form complete mathematical expressions. These expressions are then evaluated using a custom expression parser to return accurate results. The project involves multiple stages including data collection, preprocessing, model training, evaluation, and integration into a user-friendly interface. By combining digit and operator recognition with logical computation, this work provides a simple yet effective solution for performing calculations directly from handwritten input. A web interface has been developed to allow users to easily interact with the system and perform handwritten calculations online. The interface enables users to draw mathematical expressions using a mouse or touchscreen, which are then processed in real-time.

KEYWORDS: MNIST dataset, Convolutional Neural Network (CNN), handwritten digit recognition, expression evaluation, deep learning

I. INTRODUCTION

The ability of machines to interpret handwritten input has become increasingly important in the fields of computer vision and artificial intelligence. One of the most well-known benchmarks for this task is handwritten digit recognition, which has seen significant progress with the advent of Convolutional Neural Networks (CNNs). CNNs are particularly effective in learning spatial hierarchies and extracting local patterns, making them ideal for classifying handwritten characters with high accuracy.

In this paper, we propose a Handwritten Multi-Digit Calculator that leverages the power of CNNs to recognize and evaluate handwritten arithmetic expressions. Unlike traditional models that classify isolated digits, our system is designed to interpret entire expressions composed of digits and basic arithmetic operators—addition, subtraction, multiplication, and division. The calculator takes an image of a handwritten expression as input, segments it into individual components, classifies each using a CNN, and computes the result. This approach focuses on extending the capabilities of CNN-based classification from single-character recognition to multi-character sequences with embedded mathematical semantics. The work emphasizes challenges such as segmentation of concatenated digits and operators, accurate classification under varying handwriting styles, and the correct parsing of the recognized sequence for evaluation. By addressing these aspects, the study demonstrates the adaptability of CNNs in handling more complex structures.

© 2025 IJMRSET | Volume 8, Issue 5, May 2025| DOI:10.15680/IJMRSET.2025.0805215

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

II. LITERATURE REVIEW

Wang et al. [1] proposed a CNN-based system capable of recognizing handwritten digits alongside arithmetic operators, enabling the execution of basic arithmetic expressions. Their approach involved constructing a custom dataset comprising digits and symbols, designing multiple CNN architectures, and evaluating expression results using rule-based logic. This study aligns closely with the objectives of a handwritten multi-digit calculator, providing valuable insights into integrating operator recognition within CNN frameworks.

Praneeth et al. [2] explored the application of multiple machine learning models, including Convolutional Neural Networks (CNN), Multi-Layer Perceptron (MLP), and Support Vector Machines (SVM), for handwritten digit recognition. Using the MNIST dataset, their study demonstrated that CNNs outperform traditional models in both accuracy and learning efficiency. This work laid a foundational understanding of digit classification and highlighted the importance of selecting appropriate model architectures for reliable recognition

Yang et al. [3] introduced a deep learning-based handwriting recognition system that combines Faster R-CNN for character segmentation with CNN for classification. Their two-stage approach effectively handles complex handwriting by first isolating individual characters and then recognizing them with high accuracy. This modular design is especially useful for calculator applications that require precise extraction of each digit and symbol from multi-character inputs.

Bharadwaj et al. [4] implemented a CNN architecture for digit recognition and evaluated its performance on both raster images and real-world handwritten digits. The system achieved an accuracy of 98.51%, demonstrating the robustness of CNNs in generalizing across varied handwriting styles. Their work also emphasized essential preprocessing steps such as grayscale conversion and image resizing, which are critical for standardizing inputs in practical calculator applications.

Yang, Chen, and Li [5] proposed an innovative handwriting recognition system using a combination of Faster R-CNN and Long Short-Term Memory (LSTM) networks. This architecture effectively captures spatial features and sequence dependencies, improving recognition accuracy for handwritten text and multi-character inputs.

Al-Taee et al. [6] provided a comprehensive survey on handwritten recognition techniques, highlighting advances in both traditional and deep learning approaches. Their survey emphasizes the increasing effectiveness of CNN-based methods and the importance of dataset quality and preprocessing techniques for robust handwriting recognition.

Pan et al. [7] introduced a discriminative cascade CNN model for offline handwritten digit recognition, which utilizes a series of CNN classifiers to progressively refine predictions. This method improves accuracy by combining the strengths of multiple CNNs in a hierarchical structure.

Niu and Suen [8] proposed a novel hybrid CNN-SVM classifier for handwritten digit recognition. By combining the feature extraction power of CNNs with the classification strengths of Support Vector Machines (SVM), their approach achieved competitive accuracy and demonstrated the benefit of hybrid models in complex recognition tasks

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

III. FLOWCHART OF PROPOSED ARCHITECTURE

The flowchart shown below in Fig.1 depicts the block diagram of a project.

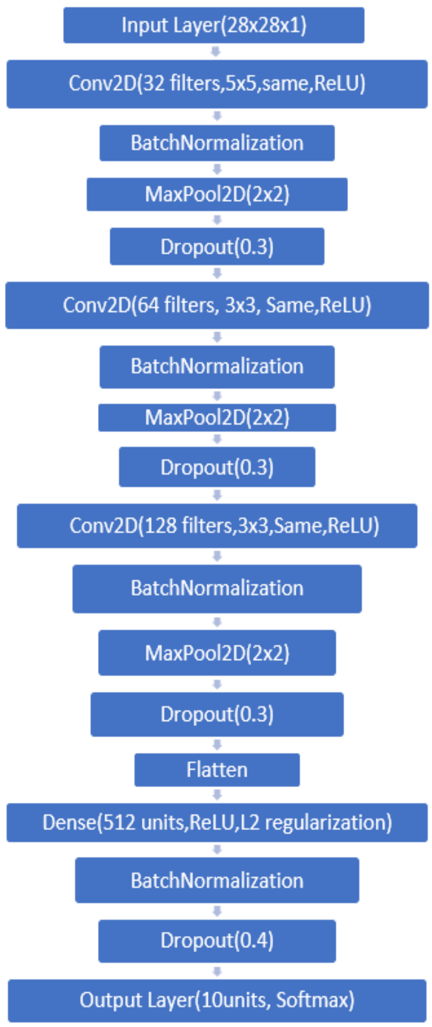

fig.1: CNN flowchart for digit recognition

The flowchart in Fig. 1 illustrates the CNN architecture used for handwritten digit recognition. The Convolutional Neural Network (CNN) model employed in this project is designed to recognize individual digits and arithmetic operators from segmented handwritten input. The architecture is composed of multiple convolutional and pooling layers, along with batch normalization and dropout to enhance generalization and prevent overfitting.

The model accepts grayscale images of size 28×28×1 as input. The first convolutional block consists of two convolutional layers with 32 filters of size 5×5 and ReLU activation, followed by batch normalization to stabilize learning. A 2×2 max-pooling layer reduces the spatial dimensions, and a dropout layer with a rate of 0.3 helps prevent overfitting.

The second convolutional block includes two convolutional layers with 64 filters of size 3×3, each followed by batch normalization. Another max-pooling and dropout layer are applied similarly.

In the third convolutional block, the model uses 128 filters of size 3×3 in two convolutional layers, again with batch normalization, followed by a 2×2 max-pooling layer and a dropout layer with a rate of 0.3.

© 2025 IJMRSET | Volume 8, Issue 5, May 2025| DOI:10.15680/IJMRSET.2025.0805215

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

After the convolutional layers, the feature maps are flattened into a one-dimensional vector. This is passed through a dense layer with 512 units and ReLU activation. L2 regularization is applied to this dense layer to reduce overfitting, followed by batch normalization and a dropout layer with a rate of 0.4.

Finally, the output layer consists of 10 units with a softmax activation function, corresponding to the digit classes 0 through 9. This configuration enables the model to classify each segmented input accurately, forming the basis for evaluating handwritten multi-digit expressions.

IV. METHODOLOGY

The proposed system is designed to recognize handwritten multi-digit arithmetic expressions and generate accurate results. The architecture involves several phases: dataset preparation, data preprocessing, model training using CNN, expression generation, and evaluation.

A. Data Collection

To train the handwritten multi-digit calculator, we utilized publicly available datasets for handwritten digits and symbols. The MNIST dataset was used for recognizing digits (0–9), as it contains 60,000 training images and 10,000 testing images of handwritten digits. For arithmetic operators (+, −, ×, ÷), we created a custom dataset by collecting handwritten samples from various individuals to capture diverse writing styles. These samples were labeled manually and preprocessed to match the input format of the digit dataset. The combined dataset ensured the model could recognize both digits and operators accurately in real-world handwritten expressions.

MNIST Dataset

The MNIST dataset was used as the primary source for handwritten digit recognition. It consists of 70,000 grayscale images of digits from 0 to 9, with 60,000 images used for training and 10,000 for testing. Each image is 28x28 pixels in size and represents a single handwritten digit. The dataset is widely used for benchmarking image classification models and provides a diverse range of handwriting styles, making it suitable for training a robust digit recognition model. The use of MNIST ensured that the calculator could accurately identify and process numerical inputs as part of multi-digit arithmetic expressions.

B. Data Preprocessing

Each image is resized to 28×28 pixels and converted to grayscale to reduce computational complexity. Pixel intensity values are normalized to the range [0, 1]. This normalization helps in faster convergence during training and improves model performance. Data augmentation techniques such as rotation, shifting, and zooming may also be applied to enhance the model's ability to generalize..

C. CNN Model Architecture

The CNN (Convolutional Neural Network) is employed as a feature extractor and classifier for recognizing handwritten digits and arithmetic operators. It is designed to learn spatial hierarchies of features through backpropagation, making it well-suited for image-based classification tasks. The architecture consists of the following layers:

Input Layer: Accepts 28×28pixel grayscale images, normalized to improve training stability and performance.

Convolutional Layers: Two to three convolutional layers are used, each with increasing filter sizes (e.g., 32, 64, 128). These layers use 3×3 kernels and ReLU (Rectified Linear Unit) activation functions to extract low- to high-level features such as edges, curves, and shapes from the input image.

Batch Normalization: In some configurations, batch normalization is added after convolution layers to stabilize and accelerate training by normalizing the output activations.

Pooling Layers: Each convolutional block is followed by a 2×2 max pooling layer to downsample the feature maps, reduce computational load, and introduce translation invariance.

Dropout Layers: Dropout is introduced after some layers (typically after pooling or fully connected layers) with a dropout rate (e.g., 0.25 or 0.5) to prevent overfitting by randomly deactivating a set of neurons during each training iteration.

Flatten Layer: The multi-dimensional output of the final convolutional block is flattened into a 1D vector to be fed into the dense layers.

Fully Connected (Dense) Layer: This layer integrates the extracted features and contributes to learning non-linear combinations. It acts as a decision-making layer before final classification.

Output Layer: A final dense layer with a softmax activation function outputs class probabilities over 14 categories (digits 0–9 and operators: +, −, ×, ÷), enabling multi-class classification.

The CNN was trained using a categorical cross-entropy loss function and optimized with the Adam optimizer due to its adaptive learning rate capabilities. This architecture balances accuracy and computational efficiency, making it suitable for real-time handwritten digit and operator recognition in the multi-digit calculator application.
The CNN architecture shown in Fig.2

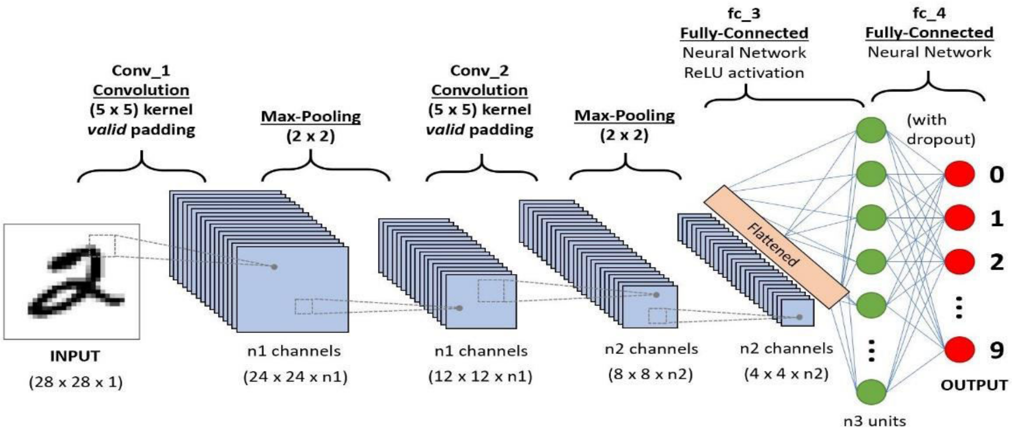

Fig.2: CNN architecture

TRAINING PHASE
During the training phase, the goal is to enable the Convolutional Neural Network (CNN) model to recognize handwritten digits and operators accurately from images. The model learns to classify each image as one of the defined classes, which may include digits (0–9) and encoded arithmetic operators.

Input: Each training sample consists of a grayscale image representing either a digit (0–9) or an arithmetic operator such as addition (+ as 10), subtraction (− as 11), multiplication (* as 12), less than (< as 13), greater than (> as 14), or not equal to (≠ as 15). These images are resized to 28×28 pixels and normalized. The preprocessed images are then passed into the CNN model for classification.

Loss Function: Categorical Cross-Entropy is used to compute the error between the predicted class probabilities and the actual labels. It serves as the objective function to be minimized during training.

Optimization: The Adam optimizer is employed to adjust the model weights based on the calculated loss. It provides efficient and adaptive learning by combining the benefits of momentum and RMSProp.

Batching: The training dataset is split into smaller batches to improve computational efficiency and model stability. This approach also allows for better memory management when training on large datasets.

© 2025 IJMRSET | Volume 8, Issue 5, May 2025| DOI:10.15680/IJMRSET.2025.0805215

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Epochs: The CNN model is trained over multiple epochs, with each epoch representing one full pass through the training data. After each epoch, performance is validated using a separate validation set to monitor generalization and prevent overfitting.

TESTING PHASE

The testing phase evaluates the trained CNN model's ability to recognize handwritten multi-digit expressions and compute accurate results for unseen images.

Image Preprocessing: The input handwritten image is converted to grayscale, resized to the CNN's required input dimensions, and normalized. This standardizes inputs for consistent model performance.

Segmentation and Character Extraction: The expression is segmented into individual characters (digits and operators) using image processing techniques such as contour detection. Each character image is extracted for classification.

Character Recognition: Each segmented character is fed into the trained CNN, which outputs a predicted label corresponding to digits (0-9) or arithmetic operators ('+', '-', '*', '/'). This step transforms the image data into a symbolic representation of the expression.

Expression Reconstruction and Evaluation: The recognized characters are combined to form the complete mathematical expression. The expression is then parsed and evaluated programmatically to produce the final calculation result.

Output Validation: The predicted result is compared against the ground truth to measure accuracy. Testing on diverse handwritten samples helps ensure the model's robustness and generalization ability.

V. RESULTS AND EVALUATION

This section presents the performance of the CNN-based handwritten multi-digit calculator. The evaluation includes training metrics such as accuracy and loss, along with classification analysis using a confusion matrix.

A. Training and Validation Metrics

The model was trained for a fixed number of epochs, and both training and validation accuracy and loss were tracked. The plots demonstrate the learning behavior of the model over time.

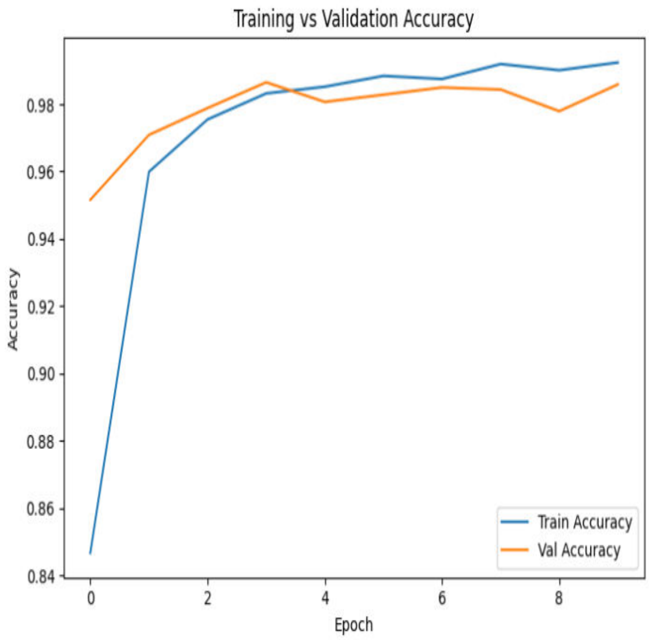

Fig.3: Training vs Validation Accuracy

© 2025 IJMRSET | Volume 8, Issue 5, May 2025| DOI:10.15680/IJMRSET.2025.0805215

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Fig.3 plot shows how accurately the model predicts the correct digit classes during training and validation phases. A steady increase in accuracy with minimal overfitting indicates successful learning.

Fig.4: Training vs Validation Loss

Fig.4 plot presents the loss values during training and validation. A consistent decrease in both curves suggests effective optimization.

B. Confusion matrix

The confusion matrix presents a detailed evaluation of the CNN model's performance in classifying handwritten digits and arithmetic/relational operators. The actual class labels are displayed along the rows, and the predicted labels are shown along the columns. The classes include digits from 0 to 9, and operators encoded as follows: '+' is 10, '−' is 11, '×' is 12, '<' is 13, '>' is 14, and '≠' is 15. This comprehensive labeling scheme enables the model to distinguish between numeric and operator symbols effectively. The confusion matrix confirms the model's high accuracy in recognizing all classes, demonstrating its capability for precise classification necessary for a handwritten multi-digit calculator.

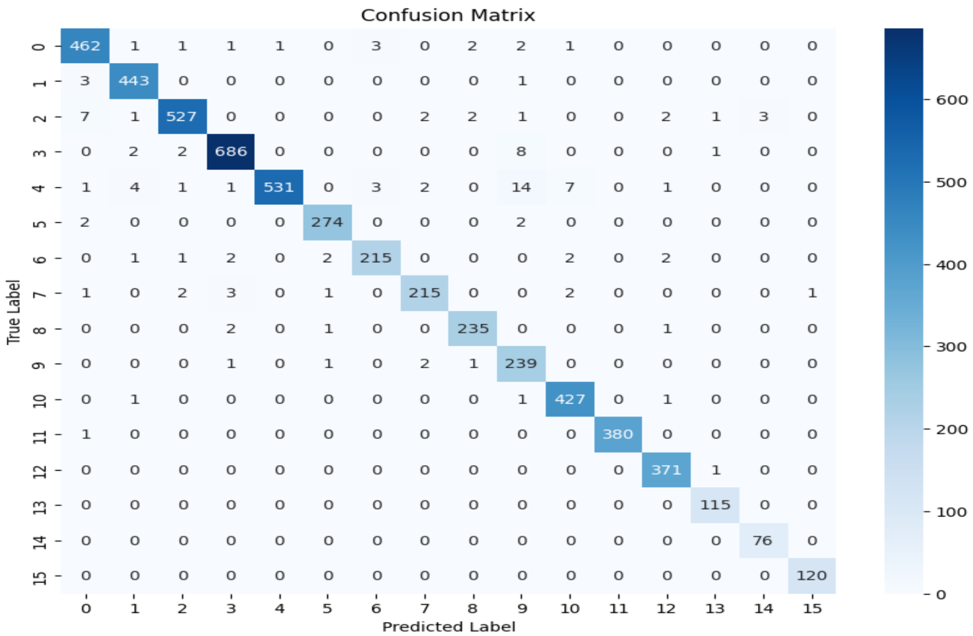

Fig.5: Confusion matrix

© 2025 IJMRSET | Volume 8, Issue 5, May 2025| DOI:10.15680/IJMRSET.2025.0805215

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

C. Quantitative Metrics

The performance of the handwritten multi-digit calculator was evaluated using precision, recall, F1-score, and accuracy metrics for both digits and arithmetic operators. Precision measures the proportion of correctly predicted instances among all predictions for a class, while recall reflects the proportion of actual instances correctly identified. The F1-score balances precision and recall to provide an overall performance measure. Accuracy indicates the percentage of correct predictions out of all predictions made.

For digits (0-9), the model achieved high precision and recall, indicating reliable recognition of individual numbers. The arithmetic operators (+, -, *, /) showed slightly lower but still strong performance, reflecting the model's ability to distinguish between different symbols accurately. These results demonstrate the effectiveness of the CNN-based approach in recognizing handwritten characters essential for multi-digit calculations.

Class label	Description	Precision	Recall	F1-Score	Accuracy
0	Digit 0	1.00	0.98	0.99	0.99
1	Digit 1	0.99	1.00	0.99	0.99
2	Digit 2	0.98	1.00	0.98	0.98
3	Digit 3	0.99	0.98	0.99	0.99
4	Digit 4	0.99	0.98	0.99	0.99
5	Digit 5	0.97	0.99	0.98	0.98
6	Digit 6	0.96	0.96	0.96	0.96
7	Digit 7	0.99	0.93	0.96	0.96
8	Digit 8	0.99	0.96	0.97	0.97
9	Digit 9	0.98	0.97	0.97	0.96
+	Addition (10)	0.98	1.00	0.99	0.99
-	Subtraction (11)	1.00	1.00	1.00	1.00
t	Multiplication (12)	0.99	1.00	0.99	0.99
l	Less than (13)	0.96	1.00	0.98	0.98
g	Greater than (14)	0.90	0.97	0.94	0.94
n	Not equal to (15)	0.98	1.00	0.99	0.99

Table.1: Class-wise Performance Metrics

A. Sample Outputs

1. Digits Recognition

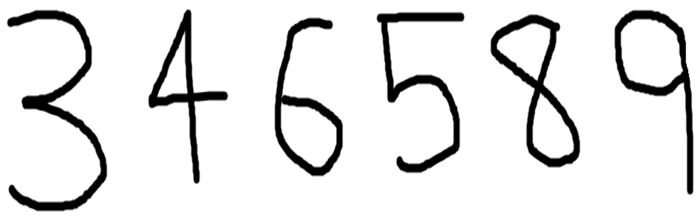

Fig.6: Actual image

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

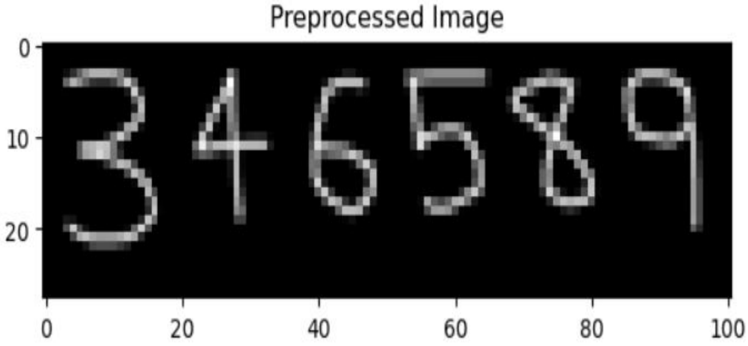

Fig.7. Pre-processed image

```
Predicted digits: ['3', '4', '6', '5', '8', '9']
Final number: 346589
```

Fig8. Final result

2. Expression Calculation

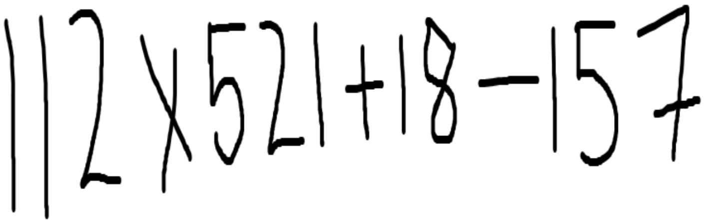

Fig9. Actual image

Fig.10: Preprocessed image

```
Evaluating Expression: 112*521+18-157
Result: 58213
```

Fig.11: Expression final result

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

3. Web Interface

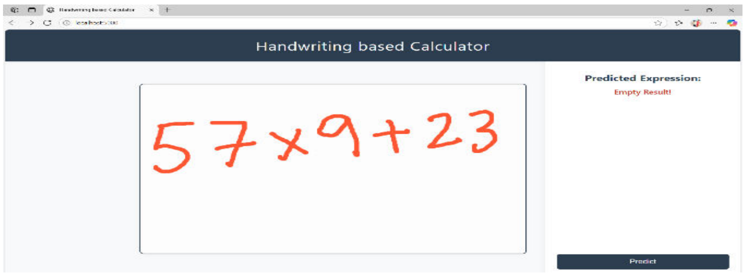

Fig.12: Input image

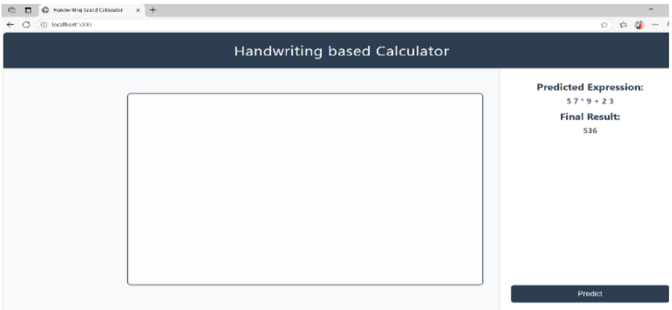

Fig.13: Expression result image

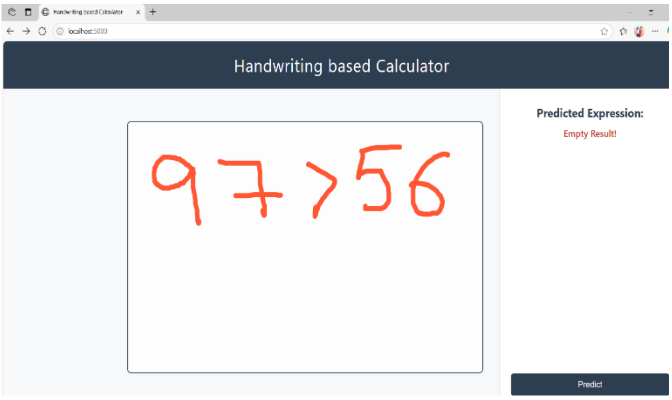

Fig.14:Input image

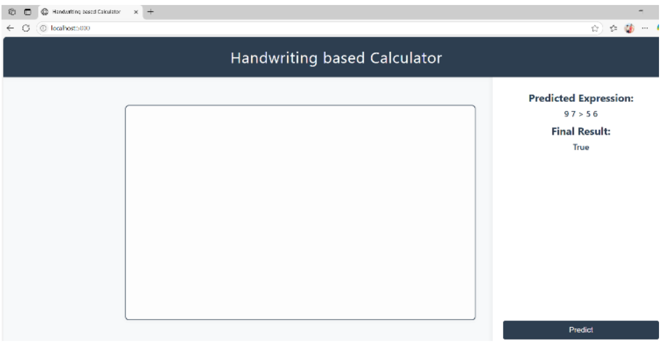

Fig.15: Result image

© 2025 IJMRSET | Volume 8, Issue 5, May 2025| DOI:10.15680/IJMRSET.2025.0805215

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

VI. CONCLUSION

In this project, we developed a handwritten multi-digit calculator that utilizes Convolutional Neural Networks (CNN) for accurate recognition of handwritten digits and arithmetic operators. The system successfully detects and interprets multi-digit inputs and performs the corresponding arithmetic operations, providing a user-friendly solution for handwritten mathematical calculations. The model demonstrated robust performance in recognizing handwritten characters with high accuracy, making the calculator reliable for practical use. However, there remains scope for improvement, such as enhancing recognition accuracy for complex or noisy inputs, and expanding the range of supported mathematical operations.

Future work will focus on integrating more sophisticated neural network architectures, like attention mechanisms, to improve recognition speed and accuracy. Additionally, extending the system to support real-time input and voice commands could make the calculator more versatile.

Overall, this project illustrates the effectiveness of CNN-based models in handwritten character recognition and arithmetic computation, offering a promising tool that combines computer vision and machine learning for educational and practical applications.

REFERENCES

[1] K.Wang, J.Deng, L.Xu, C.Tang, Z.Pei, and H.Wang, “The Four Arithmetic Opera tions for Handwritten Digit Recognition Based on Convolutional Neural Network,” 2020 IEEE International Conference on Document Analysis and Recognition (IC DAR),pp.567572,2020. DOI: 10.23919/ICDAR50203.2020.00098

[2] A.V.S.R.Praneeth, P.Karthik, Y.Hari Krishna, and B.S.Sindhuja, “Handwritten Digit Recognition Using Machine Learning,” International Journal of Research in Engineer ing, Science and Management, vol.5,no.1, pp.31-35, Jan 2022.

[3] J.Yang, P.Ren, and X.Kong, “Handwriting Text Recognition Based on Faster R-CNN,” Proceedings of the IEEE International Conference on Computer Vision, 2019. DOI: 10.1109/CAC48633.2019.8997382

[4] S.S.Bharadwaj, V.V.S.R.Karthik, and M.V.S.S.Ram, “Effective Handwritten Digit Recognition Using Deep Convolutional Neural Network,” International Journal of Re search in Engineering, Science and Management,vol.3 no.5,pp.123–126,2020.

[5] J.Yang, L.Chen, and X.Li, ”Handwritten Text Recognition Using Faster R-CNN and LSTM,” 2020 IEEE International Conference on Image Processing (ICIP),pp.789-793,2020. DOI: 10.1109/ICIP40778.2020.9191277

[6] M.M.Al-Taee, S.B.H.Neji and M.Frikha, ”Handwritten Recognition: A survey,” 2020 IEEE 4th International Conference on Image Processing, Applications and Systems (IPAS), 2020, pp. 199-205, 2020. DOI: 10.1109/ IPAS50080 .2020 .933 4936

[7] S.Pan,Y.Wang, C.Liu, and X.Ding, “A discriminative cascade cnn model for offline handwritten digit recognition,” in 2015 14th IAPR International Conference on Ma chine Vision Applications (MVA), pp. 501–504, IEEE, May 2015.DOI: 10.1109/MVA.2015.7153240

[8] Tang,J.,Han,P., & Liu, D. (2020). Adhesive Handwritten Digit Recognition Algorithm Based on Improved Convolutional Neural Network, IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS), pp. 388–392,2020. DOI: 10.1109/ICAIIS49377.2020.9194797

[9] K. Pranit Patil, "Handwritten Digit Recognition Using Various Machine Learning Algorithms and Models," International Journal of Innovative Research in Computer Science & Technology (IJIRCST), vol. 8, no. 4, p. 337 – 340, July 2020.DOI: 10.21276/ijircst.2020.8.4.16

[10] Xiao-Xiao Niu n, Ching Y.Suen “A novel hybrid CNN–SVM,IEEE classifier for recog nizing handwritten digits” Pattern Recognition 45(2012) 1318–1325,2012. DOI: 10.1016/j.patcog.2011.09.021