

A Data-Informatics-Oriented CNN-Based Intelligent Model for Handwritten Mathematical Symbol Recognition

Muhammad Athoillah¹, Rani Kurnia Putri², Fenny Fitriani¹, Prayogo³

¹Department of Statistics, Universitas PGRI Adi Buana, Surabaya, 60234, Indonesia

²Department of Mathematics Education, Universitas PGRI Adi Buana, Surabaya, 60234, Indonesia

³Department of Primary Education, Universitas PGRI Adi Buana, Surabaya, 60234, Indonesia

Article Info

Article history:

Received January 28, 2025

Revised March 04, 2026

Accepted March 12, 2026

Keywords:

Deep learning
Image classification
Pattern recognition
Data informatics
Computer vision

ABSTRACT

Handwritten mathematical symbol recognition remains a challenging problem due to variations in writing styles, stroke structures, and visual similarities among symbols, which often reduce classification accuracy. This study proposes a data-informatics-oriented convolutional neural network (CNN) model for robust recognition of handwritten mathematical symbols. The research adopts a supervised experimental design using a balanced dataset consisting of six mathematical symbol classes. A systematic preprocessing pipeline including image resizing, normalization, and structured dataset partitioning is implemented to ensure data consistency and improve feature learning. The CNN model is implemented in MATLAB and optimized using stochastic gradient descent with momentum. Model performance is evaluated using confusion matrix-based metrics, including accuracy, precision, recall, and F1-score, along with computational time analysis. Experimental results demonstrate stable performance across multiple experimental runs, achieving an average accuracy of 97.08%, precision of 97.10%, recall of 97.08%, and F1-score of 97.07%. Confusion matrix analysis indicates that most handwritten symbols are correctly classified, with only minor misclassifications occurring among visually similar operators. These results confirm the effectiveness of integrating data informatics principles with CNN-based feature learning for handwritten mathematical symbol recognition. The proposed framework provides a reliable foundation for intelligent systems supporting digital education, automated assessment, and mathematical document digitization.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author: Muhammad Athoillah (e-mail: athoillah@unipasby.ac.id)

1. INTRODUCTION

In recent years, the rapid expansion of intelligent computing and data-driven technologies has significantly transformed the way information is captured, processed, and utilized across various domains. One critical area of development is the digital interpretation of handwritten content, particularly in educational, scientific, and technical environments [1][2]. Among the different forms of handwriting, mathematical symbols represent a uniquely complex and essential modality due to their structural diversity, semantic importance, and frequent usage in global academic communication. Mathematical notation serves as the foundation of scientific reasoning, engineering design, and computational modeling, making the accurate recognition of handwritten mathematical symbols a crucial component of modern intelligent systems [3][4].

Globally, the increasing adoption of e-learning platforms, digital assessment systems, and artificial intelligence (AI)-driven educational technologies has amplified the demand for robust handwritten recognition solutions. According to UNESCO, over 1.5 billion learners worldwide were affected by disruptions in traditional education during the COVID-19 pandemic, accelerating the reliance on digital learning tools and remote evaluation systems. In parallel, the global EdTech market has continued to expand significantly, emphasizing the need for intelligent systems capable of supporting mathematics learning and assessment at scale. Public perception increasingly views AI as a transformative tool in education,

particularly for automating tasks such as grading, learning analytics, and personalized instruction. Recent studies emphasize that handwriting recognition continues to evolve as a key research topic within intelligent systems due to its broad applicability in real-world digital transformation efforts [5]. Advances in handwritten document recognition reflect the growing relevance of intelligent computing solutions that can process unstructured handwritten data into machine-interpretable formats [6]. As a result, developing accurate and scalable models for handwritten mathematical symbol recognition is not only technically significant but also globally important for supporting educational innovation, scientific digitization, and data informatics development.

Handwritten mathematical symbol recognition refers to the automated process of identifying and classifying mathematical characters from handwritten image inputs. Ontologically, this task lies at the intersection of computer vision, pattern recognition, and data informatics, as it involves transforming raw visual handwriting data into structured symbolic representations [7]. The recognition of mathematical symbols is inherently more complex than ordinary handwritten character recognition because mathematical notation includes diverse symbol categories, subtle visual distinctions, and multidimensional writing structures. Traditional recognition approaches commonly relied on handcrafted feature extraction techniques combined with machine learning classifiers such as Support Vector Machines, k-Nearest Neighbors, and Deep Learning [8][9]. While these methods offered initial success, their performance was often constrained by limited adaptability to diverse handwriting styles and the need for manual feature engineering. Modern intelligent computing research has shifted toward deep learning paradigms that automatically learn discriminative feature representations directly from data.

Convolutional Neural Networks (CNNs) have emerged as one of the most effective deep learning architectures for handwriting recognition due to their ability to capture hierarchical spatial features from image data. CNN-based frameworks have demonstrated strong performance in extracting meaningful representations of handwritten strokes, curves, and symbol structures without relying on handcrafted descriptors. This capability is especially valuable for mathematical symbols, where small variations in shape may alter semantic meaning. Furthermore, recent research has explored hybrid recognition strategies that integrate CNN-based feature extraction with other classification mechanisms to improve robustness. For instance, combining CNN representations with traditional classifiers has been shown to enhance recognition accuracy for the handwritten text problem [10][11]. In addition, retrospective studies highlight that handwritten mathematical symbol recognition remains an active research field due to unresolved challenges in classification and structural interpretation [12].

Despite significant progress in deep learning and intelligent computing, handwritten mathematical symbol recognition remains an open and globally relevant challenge. One of the primary difficulties arises from the high variability in human handwriting, where the same mathematical symbol may be written differently depending on individual style, cultural context, writing tools, or environmental conditions [13]. Additionally, many mathematical symbols exhibit strong visual similarity, leading to frequent misclassification, particularly when symbol categories are large and complex. This recognition problem has major implications in global education and digital transformation. Mathematics is a core subject worldwide, yet assessments and student problem-solving activities are still predominantly handwritten [14]. Without reliable recognition models, automated grading systems and intelligent tutoring platforms cannot fully interpret students' handwritten mathematical responses. This limitation restricts the scalability of AI-based learning support, particularly in regions with limited educational resources. Moreover, recognition errors at the symbol level can propagate into larger issues at the expression level, affecting downstream tasks such as equation parsing, semantic interpretation, and automated reasoning. Recent surveys highlight that while deep learning models have advanced substantially, handwritten mathematical expression recognition still faces challenges related to symbol ambiguity, dataset limitations, and structural complexity. If these issues remain unresolved, intelligent systems will continue to struggle with accurately digitizing handwritten scientific knowledge and supporting advanced educational technologies.

In response to these challenges, this study aims to develop a data-informatics-oriented CNN-based intelligent model for handwritten mathematical symbol recognition by integrating deep convolutional learning with systematic data preprocessing strategies to enhance recognition accuracy, robustness, and scalability across diverse handwriting conditions. Specifically, the study focuses on designing and implementing an optimized CNN architecture capable of accurately classifying mathematical symbols into multiple categories, incorporating data informatics-driven preprocessing techniques such as normalization and data augmentation to improve model generalization, and evaluating the proposed framework using standard performance metrics, including accuracy, precision, recall, F1-score, and confusion matrix analysis. Through these efforts, the research contributes to the advancement of intelligent handwriting recognition systems and supports future applications in digital education, automated assessment, and scientific document digitization.

Based on the challenges discussed above, this study formulates several research problems to guide the development and evaluation of the proposed handwritten mathematical symbol recognition system. First, how can a structured data preprocessing pipeline be designed to ensure consistent and reliable input data for handwritten mathematical symbol recognition? Second, how can a convolutional neural network (CNN) architecture be effectively designed to learn discriminative features from handwritten mathematical symbols with varying writing styles and structural patterns? Third, how can the training and optimization process be configured to achieve stable and efficient model learning performance? Finally, how effective is the proposed CNN-based recognition framework in accurately classifying handwritten mathematical symbols when evaluated using standard performance metrics such as accuracy, precision, recall, and F1-score across multiple experimental batches? These research problems guide the methodological design and experimental evaluation presented in this study.

2. LITERATURE REVIEW

2.1. Handwritten Mathematical Symbol Recognition as an Intelligent Computing Problem

Handwritten mathematical symbol recognition has long been recognized as a challenging subfield of pattern recognition due to the inherent complexity of mathematical notation and the high variability of human handwriting. Unlike conventional handwritten text, mathematical symbols often possess subtle visual differences while conveying significantly different semantic meanings. Recent studies emphasize that symbol-level recognition is a critical prerequisite for higher-level tasks such as mathematical expression parsing, semantic interpretation, and automated reasoning [15]. Consequently, inaccurate symbol recognition can propagate errors throughout the entire mathematical understanding pipeline, limiting the effectiveness of intelligent computing systems. From a data informatics perspective, handwritten symbol recognition involves more than image classification alone; it requires systematic data handling, including preprocessing, representation, and evaluation. Bhatt et al. [16] argue that intelligent recognition systems must integrate data-centric strategies to ensure robustness and generalizability, particularly when dealing with unstructured handwritten inputs collected under real-world conditions. This view positions handwritten mathematical symbol recognition as a data-driven intelligence problem rather than a purely algorithmic task.

2.2. CNN-Based Deep Learning Approaches for Handwritten Recognition

Convolutional Neural Networks (CNNs) have become the dominant paradigm in handwritten recognition research due to their ability to automatically learn hierarchical spatial features from raw image inputs. CNNs eliminate the need for manual feature engineering by leveraging convolutional filters that capture local stroke patterns, edges, and curves, followed by deeper layers that model more abstract symbol structures. Saqib et al. [17] demonstrated that CNN-based models significantly outperform traditional classifiers in handwritten character recognition tasks, particularly when trained on large and diverse datasets. In the context of mathematical symbols, CNNs have shown strong potential due to their robustness in handling shape variability and noise. However, recent literature suggests that model performance is highly sensitive to architectural choices and training strategies. While deeper networks often achieve higher accuracy, they also introduce risks of overfitting and increased computational complexity [18]. Truong et al. [15] highlight that many CNN-based studies focus heavily on architectural novelty while underemphasizing data quality and preprocessing, which are equally critical for recognition performance.

2.3. Role of Data Informatics and Preprocessing Strategies

Recent literature increasingly emphasizes the importance of data informatics in developing reliable intelligent recognition systems. Data preprocessing techniques such as image normalization, resizing, noise removal, and data augmentation play a crucial role in improving CNN generalization and robustness. Paranayapa et al [19] report that preprocessing strategies significantly influence recognition accuracy, sometimes more than architectural modifications. Data augmentation, in particular, has been shown to mitigate overfitting and enhance model adaptability to unseen handwriting styles. However, many existing studies treat preprocessing as a secondary step rather than an integral component of system design. This fragmented approach limits the scalability of recognition systems and reduces their applicability in real-world educational and scientific environments [20]. A coherent data informatics framework is therefore essential to bridge the gap between high laboratory performance and practical deployment.

Although significant progress has been made in handwritten mathematical symbol recognition, several limitations remain in existing studies. Many previous works focus primarily on improving classification accuracy through deep learning architectures, but provide limited discussion on data preprocessing strategies and dataset consistency. Some studies rely on relatively small datasets or specific symbol categories, which may reduce the generalizability of the proposed models. In addition, several

approaches evaluate model performance using a single experimental configuration, making it difficult to assess the stability of the model under different training conditions. Furthermore, comparative analysis of computational efficiency and performance stability is often limited. These limitations highlight the need for a systematic framework that integrates structured data preprocessing, stable CNN architecture design, and repeated experimental evaluation to ensure reliable recognition performance. Table 1 summarizes several representative studies and their corresponding limitations.

Table 1. Summary of Related Studies and Their Limitations

Study Method	Dataset / Scope	Key Contribution	Limitations
[11] Hybrid CNN–SVM classifier	Handwritten digit dataset	Demonstrated that combining CNN feature extraction with SVM classification can improve recognition accuracy	Focused only on digit recognition rather than diverse mathematical symbols
[15] Survey of deep learning approaches for handwritten mathematical expression recognition	Mathematical expression recognition research	Provided a comprehensive overview of encoder–decoder and graph-based approaches	Focuses more on expression-level recognition rather than symbol-level classification
[12] Classification methods for handwritten mathematical symbols	Mathematical symbol datasets	Highlighted classification techniques for handwritten mathematical symbols	Limited evaluation across multiple experimental runs and dataset variations
[16] Data-centric deep learning approach	General deep learning datasets	Emphasized the importance of data-centric strategies in improving model performance	Does not specifically address handwritten mathematical symbol recognition

2.4. Research Gaps and Positioning of the Present Study

Although recent advances in CNN-based recognition have substantially improved handwritten mathematical symbol classification, several research gaps remain. First, many studies focus narrowly on model architecture without sufficiently addressing data variability and preprocessing standardization. Second, comparative evaluations across diverse handwriting conditions are often limited, making it difficult to assess real-world robustness. Third, the integration of data informatics principles into recognition frameworks remains underexplored [21][22]. In response to these gaps, the present study positions itself by proposing a data-informatics-oriented CNN-based intelligent model that emphasizes not only architectural optimization but also systematic data preprocessing and comprehensive evaluation. By integrating deep learning with data-centric strategies, this research aims to contribute a more holistic and scalable solution to handwritten mathematical symbol recognition, thereby advancing intelligent computing applications in digital education, automated assessment, and scientific document digitization.

3. METHOD

This study employs a systematic methodology to develop and evaluate a CNN-based handwritten mathematical symbol recognition framework, as illustrated in Figure 1. The workflow consists of four main stages: data acquisition and preprocessing, CNN-based feature learning, model optimization and training, and performance evaluation. Handwritten symbol images are organized into class-specific datasets and preprocessed through resizing, normalization, and dataset partitioning to ensure input consistency. The CNN architecture subsequently learns hierarchical feature representations through convolutional and pooling operations, while performance evaluation is conducted using classification accuracy and confusion matrix-based analysis to assess recognition effectiveness across all symbol classes.

3.1. Image Dataset

The handwritten mathematical symbol dataset used in this study was obtained from an open-source repository associated with the MathWrite project, which provides a web-based interface for collecting handwritten mathematical symbols. The dataset contains images of handwritten operators that were originally intended for training a symbol recognition system using classical machine learning methods such as decision trees. The images are stored in structured folders according to symbol classes and can be used to train

alternative classification models. In this research, the dataset images were extracted from the repository and used to train a convolutional neural network (CNN) model to improve recognition performance compared to traditional approaches. The dataset consists of six mathematical symbol classes, namely parentheses “(” and “)”, addition “+”, subtraction “-”, multiplication “×”, and division “:”. Each class contains 200 handwritten samples, resulting in a total of 1,200 images. The dataset provides a practical collection of handwritten symbol samples with natural variations in writing style, which makes it suitable for evaluating machine learning–based handwritten symbol recognition systems.

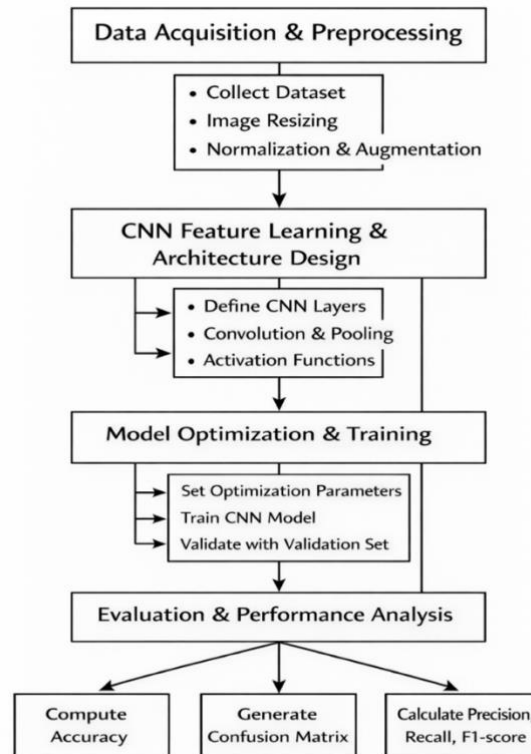


Figure 1. Overview of Research Process

3.2. Data Preprocessing and Informatics Pipeline

In this study, a systematic data-informatics-oriented preprocessing pipeline is designed to standardize input data, enhance feature discriminability, and improve model robustness. All handwritten symbol images are organized into class-specific directories, enabling automated label assignment through structured data repositories. Images are resized to a uniform spatial resolution of 280×280 pixels with three color channels to ensure consistency across samples and compatibility with the convolutional neural network input layer. This spatial normalization minimizes dimensional variability and supports stable convolutional operations. Pixel intensity normalization is applied during data loading to scale input values into a consistent numerical range. This normalization process mitigates the influence of illumination differences and background artifacts, stabilizes gradient propagation during training, and accelerates convergence of the optimization process. As a result, the learned feature representations become more discriminative and resilient to noise.

To support supervised learning and unbiased evaluation, the dataset is randomly partitioned into training and validation subsets using a fixed number of samples per class. Randomized splitting ensures that both subsets capture diverse handwriting patterns, thereby enhancing the generalization capability of the proposed model. The preprocessing pipeline is further supported by an image datastore mechanism that enables efficient batch processing, scalable memory management, and reproducible experimental configurations. While explicit data augmentation is not applied in the current implementation, the pipeline is designed to accommodate future extensions such as geometric transformations and intensity variations to further improve model robustness.

3.3. CNN-Based Feature Learning and Architecture Design

The proposed handwritten mathematical symbol recognition system utilizes a convolutional neural network (CNN) architecture designed to automatically learn hierarchical feature representations from raw image data. The network operates directly on RGB input images with a fixed spatial resolution of $280 \times 280 \times 3$, ensuring dimensional consistency across all samples and enabling stable convolutional operations.

Let $X \in \mathbb{R}^{H \times W \times C}$ denote an input image, where $H = 280$, $W = 280$, and $C = 3$. Feature extraction is performed through successive convolutional layers, where each convolution operation computes a set of feature maps according to equation (1).

$$F_k^l = W_k^l * F^{(l-1)} + b_k^l \quad (1)$$

Where W_k^l and b_k^l represent the convolutional kernel and bias for the k -th feature map in layer l , $*$ denotes the convolution operation, and $F^{(0)} = X$. Each convolutional layer employs kernels of size 3×3 to capture localized spatial patterns such as edges, curves, and stroke intersections that are fundamental to handwritten mathematical symbols. The number of filters increases progressively from 8 in the first convolutional layer to 16 and 32 in the subsequent layers, enabling the network to learn increasingly complex and abstract feature representations. To stabilize training and improve convergence, batch normalization is applied after each convolution operation. Given a mini-batch of activations $\{x_i\}$, batch normalization transforms each activation according to equation (2).

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, \quad y_i = \gamma \hat{x}_i + \beta \quad (2)$$

Where μ_B and σ_B^2 denote the batch mean and variance, ϵ is a small constant for numerical stability, and γ and β are learnable scaling and shifting parameters. Nonlinear activation is introduced using the rectified linear unit (ReLU) function defined as equation (3).

$$f(x) = \max(0, x) \quad (3)$$

Enhances the model's ability to learn complex decision boundaries while mitigating the vanishing gradient problem. Spatial downsampling is performed using max-pooling layers with a stride of 2 after the first and second convolutional stages. The max-pooling operation is defined as equation (4).

$$P_{i,j} = \max_{(m,n) \in \Omega_{i,j}} F_{m,n} \quad (4)$$

Where $\Omega_{i,j}$ represents the pooling window. This operation reduces spatial dimensionality, improves computational efficiency, and introduces translational invariance while preserving salient feature responses. The feature maps produced by the final convolutional layer are flattened into a one-dimensional feature vector and passed to a fully connected layer with six output neurons, corresponding to the number of handwritten mathematical symbol classes. This transformation is expressed as equation (5).

$$z = W_f v + b_f \quad (5)$$

Where v denotes the flattened feature vector, and W_f and b_f represent the weights and biases of the fully connected layer. The resulting logits are converted into class probability distributions using the softmax function.

$$P(y = i|X) = \frac{e^{z_i}}{\sum_{j=1}^C e^{z_j}} \quad (6)$$

Where C is the number of classes, the predicted class is determined by selecting the class with the highest posterior probability (6). This architecture effectively balances representational capacity and computational efficiency, making it suitable for intelligent handwritten symbol recognition under limited training epochs [23].

3.4. Model Optimization and Training Strategy

Model optimization and training are conducted using a supervised learning framework aimed at minimizing classification error while maintaining computational efficiency. The convolutional neural network is trained using the stochastic gradient descent with momentum (SGDM) optimization algorithm [24], which is selected for its stability and effectiveness in training deep neural networks on image-based data. Let θ denote the set of trainable parameters of the CNN, including convolutional kernels, batch normalization parameters, and fully connected layer weights. The training process seeks to minimize the categorical cross-entropy loss function $L(\theta)$, defined as equation (7).

$$L = -\frac{1}{N} \sum_{n=1}^N \sum_{c=1}^C y_{n,c} \log(\hat{y}_{n,c}) \quad (7)$$

Where N is the number of training samples in a mini-batch, C is the number of symbol classes, $y_{n,c}$ denotes the ground-truth label encoded in one-hot format, and $\hat{y}_{n,c}$ represents the predicted probability obtained from the softmax output layer. Parameter updates are performed iteratively using SGDM. At iteration t , the update rules are given by equations (8) and (9).

$$v_{t+1} = \mu v_t - \eta \nabla_{\theta} L(\theta_t) \quad (8)$$

$$\theta_{t+1} = \theta_t + v_{t+1} \quad (9)$$

Where v_t is the velocity term, μ denotes the momentum coefficient, η is the learning rate, and $\nabla_{\theta} L$ represents the gradient of the loss function with respect to the network parameters. In this study, the initial learning rate is set to 0.01.

Training is conducted for four epochs, with data shuffling applied at the beginning of each epoch to reduce bias arising from sample ordering and to improve generalization. The use of a limited number of epochs reflects a design choice to balance training efficiency and overfitting prevention, given the moderate network depth and dataset size. Validation is integrated into the training process using a hold-out validation dataset. Performance on the validation set is evaluated periodically at fixed intervals, allowing monitoring of convergence behavior and early identification of potential overfitting. Validation results are not used to update network parameters but serve as an unbiased estimate of model generalization capability.

3.5. Model Evaluation and Performance Analysis

After the training phase, the optimized model is evaluated using a held-out validation dataset to assess its generalization capability on unseen handwritten mathematical symbols. Predicted class labels are compared with the corresponding ground-truth annotations to quantify classification performance under realistic recognition conditions. Overall classification accuracy is used as an initial performance indicator to represent the proportion of correctly classified samples relative to the total number of validation instances. Although accuracy provides a concise summary of model performance, it does not fully reflect class-specific behavior, particularly when visually similar mathematical symbols are involved [25].

To obtain a more detailed evaluation, confusion matrix analysis is employed to examine class-wise prediction behavior. The confusion matrix presents the relationship between actual and predicted labels for all symbol categories, allowing identification of dominant misclassification patterns and symbol pairs that are frequently confused due to structural similarity or handwriting variability. Based on the confusion matrix, precision, recall, and F1-score are computed for each class to provide complementary performance insights. Precision indicates the reliability of the model's predictions by measuring how accurately a predicted symbol corresponds to its true class. Recall reflects the model's ability to correctly identify all instances of a particular symbol class, highlighting sensitivity to missed detections. The F1-score combines precision and recall into a single balanced metric, offering a comprehensive assessment of classification effectiveness, especially in cases where trade-offs exist between false positives and false negatives [26].

4. RESULTS AND DISCUSSION

The experimental results indicate that the proposed CNN-based handwritten mathematical symbol recognition system is capable of accurately identifying input symbols while maintaining a short processing time. Figure 2 presents one representative sample of the recognition output generated by the developed system. In this example, the handwritten input symbol is correctly classified into the corresponding category, namely the symbol \times , demonstrating that the trained model effectively captures the discriminative visual features of handwritten mathematical symbols. The graphical user interface (GUI), as shown in Figure 2, further illustrates the successful integration of the trained CNN model into an application-level system. The

interface allows users to load handwritten images, initiate the recognition process, and display classification results together with processing time information. This sample output confirms the practical applicability of the proposed framework and its suitability for use beyond offline experimental settings.

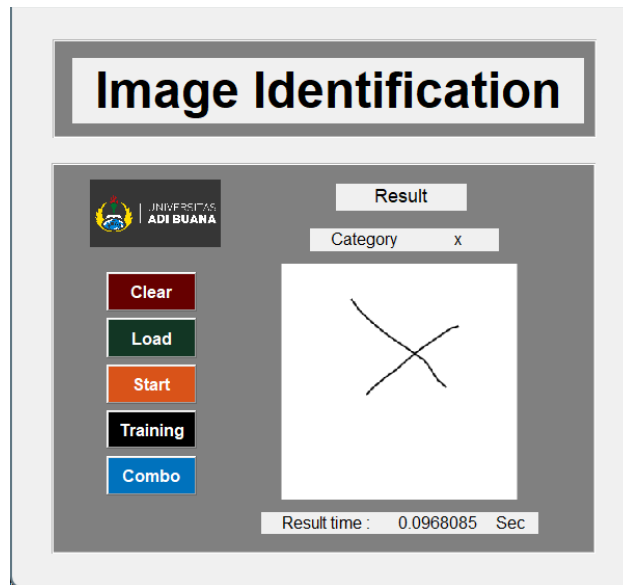


Figure 2. GUI

The quantitative evaluation results, summarized in the averaged confusion matrix and performance metrics, indicate that the proposed CNN-based handwritten mathematical symbol recognition system achieves consistently high classification performance across all symbol categories. The experimental setup involves six symbol classes—parentheses, addition, subtraction, multiplication (x), and division—each represented by an equal number of validation samples, ensuring balanced class evaluation. The overall average recall value of 0.9708 demonstrates that the model successfully identifies the majority of handwritten symbols across all classes. This high recall indicates strong sensitivity and a low rate of missed detections, which is particularly important in handwritten symbol recognition, where stroke variations and writing styles introduce significant ambiguity. Class-wise recall values remain consistently high, ranging from approximately 0.9565 to 0.987, reflecting stable recognition performance across different mathematical operators.

Precision analysis further confirms the reliability of the model's predictions. The average precision score of 0.9710 indicates that most predicted symbols correspond correctly to their actual classes, with minimal false-positive occurrences. Individual class precision values also show limited variation, suggesting that the model does not disproportionately favor or confuse specific symbol categories. This consistency implies that the learned feature representations effectively discriminate between symbols with similar structural patterns. The F1-score, which balances precision and recall, yields an average value of approximately 0.9707, reinforcing the robustness of the proposed model. The close alignment between precision, recall, and F1-score across all classes indicates that the system maintains a balanced trade-off between detection accuracy and prediction reliability, without significant performance degradation in any particular class.

Confusion matrix analysis reveals that most predictions are concentrated along the diagonal entries, signifying correct classification outcomes. Minor misclassifications are observed among visually similar symbols, which is expected in handwritten mathematical notation due to overlapping stroke patterns and writing inconsistencies. However, the limited magnitude of off-diagonal entries suggests that such ambiguities occur infrequently and do not substantially affect overall system performance. In terms of computational performance, the aggregated results show an average total training time of approximately 1063 seconds across repeated experimental runs, while the average testing time remains below 9 seconds. This disparity reflects the computational cost associated with model learning compared to the relatively efficient evaluation phase. The recorded testing time confirms that the trained model can perform recognition efficiently once deployed, supporting its applicability in practical usage scenarios. Overall, the results demonstrate that the proposed CNN-based framework achieves high and stable classification performance, supported by strong precision, recall, and F1-score values, as well as acceptable computational efficiency. These findings validate the effectiveness of the proposed methodology for handwritten mathematical symbol

recognition and provide a solid empirical foundation for further model extension and real-world application. Table 2 presents the averaged classification and computational performance of the proposed CNN-based handwritten mathematical symbol recognition system across multiple experimental runs.

Table 2. Summary of Result

Metric	Value	Description
Number of Classes	6	Mathematical symbols: (,), +, -, x, :
Samples per Class	200	Balanced validation samples
Average Precision	0.9710	Reliability of predicted symbol labels
Average Recall	0.9708	Ability to correctly identify true symbols
Average F1-score	0.9707	Balanced measure of precision and recall
Average Training Time (s)	1063.33	Total training duration across runs
Average Testing Time (s)	8.93	Total evaluation time on validation data
Evaluation Method	Confusion Matrix	Class-wise performance assessment
Experimental Runs	Multiple	Randomized dataset partitions

Table 3. Batch-wise Classification and Computational Performance

Batch	Accuracy	Precision	Recall	F1-score	Training Time (s)	Testing Time (s)
1	0.9583	0.9583	0.9593	0.9584	1004.11	9.43
2	0.9675	0.9675	0.9677	0.9675	937.45	9.08
3	0.9708	0.9708	0.9715	0.9708	1127.77	10.96
4	0.9742	0.9742	0.9744	0.9742	1021.33	7.65
5	0.9825	0.9825	0.9827	0.9825	947.24	7.94
6	0.9583	0.9583	0.9593	0.9584	969.16	7.23
7	0.9675	0.9675	0.9677	0.9675	916.43	6.89
8	0.9675	0.9675	0.9683	0.9675	1537.69	8.18
9	0.9783	0.9783	0.9785	0.9783	983.30	11.56
10	0.9825	0.9825	0.9827	0.9825	1188.83	10.39
Average	0.9708	0.9710	0.9708	0.9707	1063.33	8.93

Table 3 presents the batch-wise classification and computational performance of the proposed CNN-based handwritten mathematical symbol recognition system across multiple experimental runs. Batch testing is employed in this study to evaluate the stability and robustness of the proposed model under different randomized dataset partitions. Instead of comparing multiple CNN architectures, the focus of this research is to assess whether the same model can maintain consistent performance when trained and validated using varying subsets of the dataset. This approach is particularly relevant for moderately sized datasets, where model performance may be sensitive to how the data are partitioned. By repeating the training and evaluation process across multiple batches, the study ensures that the reported results are not dependent on a single experimental configuration but reflect the generalization capability of the proposed CNN model. The results demonstrate that accuracy, precision, recall, and F1-score remain consistently high across all batches, with only minor variations, indicating that the model is stable and robust under different data distributions.

The batch-wise evaluation results presented in Table 3 demonstrate that the proposed CNN-based handwritten mathematical symbol recognition system achieves consistently high classification performance across all experimental runs. Overall accuracy values range from 0.9583 to 0.9825, indicating stable recognition capability despite variations introduced by randomized dataset partitioning. This consistency suggests that the model generalizes well to unseen handwritten samples and is not overly sensitive to data split variability. Precision, recall, and F1-score values closely follow the accuracy trends across all batches, with average values exceeding 0.97. The small differences among these metrics indicate a balanced classification behavior, where the model maintains both high prediction reliability and strong sensitivity across all symbol classes. The alignment between precision and recall further implies that the classifier does

not exhibit bias toward false positives or false negatives, which is critical in handwritten mathematical symbol recognition due to the presence of visually similar operators.

Computational performance analysis reveals greater variability in training time compared to testing time. Training duration ranges from approximately 916 seconds to 1538 seconds, reflecting differences in convergence behavior arising from randomized training-validation splits. In contrast, testing time remains consistently low across all batches, with values below 12 seconds, demonstrating the efficiency of the trained model during the evaluation phase. This stability in testing time highlights the suitability of the proposed framework for practical deployment scenarios.

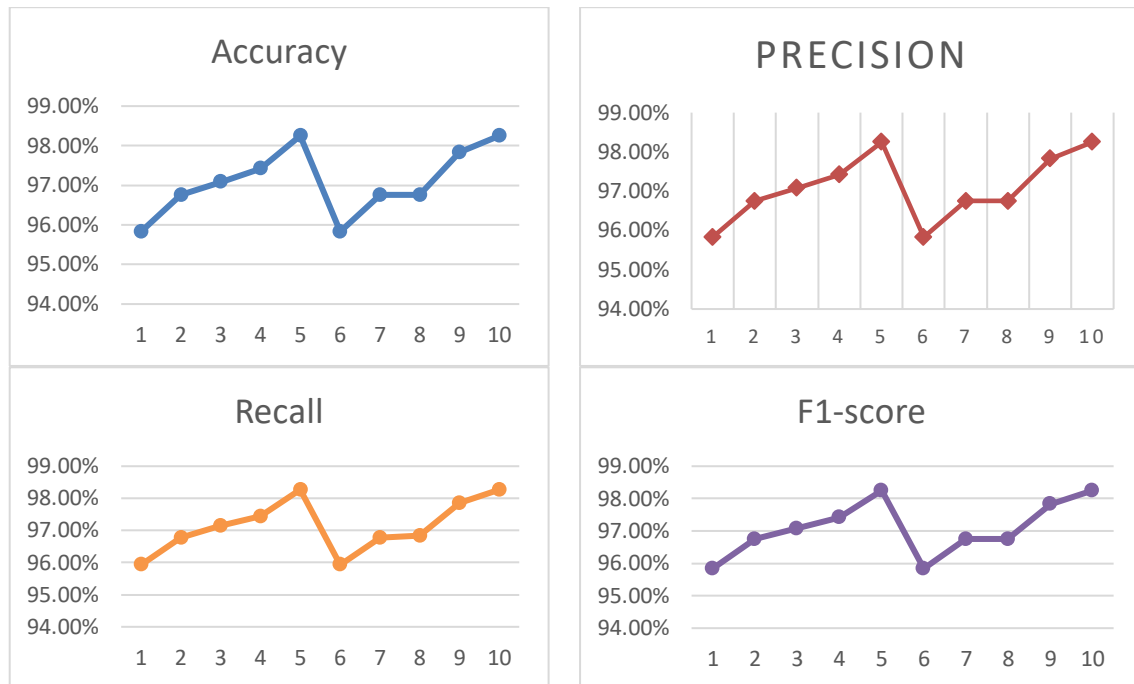


Figure 3. Classification Performance Chart

The chart in Figure 3 visually strengthens the performance analysis by illustrating the batch-wise trends of accuracy, precision, recall, and F1-score across all experimental runs. Overall, the curves demonstrate a consistently high and stable classification performance, with all metrics remaining above 95% for every batch, indicating strong generalization of the CNN model. A gradual improvement is observed from Batch 1 to Batch 5, where the model reaches peak performance ($\approx 98.2\%$), suggesting effective learning as training progresses. A temporary decline appears in Batch 6, which can be attributed to random data partitioning effects or increased sample variability; however, the model quickly recovers in subsequent batches. The near-overlapping nature of accuracy, precision, recall, and F1-score curves highlights a balanced classification behavior, confirming that the model does not favor specific classes and maintains low false-positive and false-negative rates. Overall, the batch-wise results confirm the robustness and reliability of the proposed CNN-based recognition framework. The consistent classification metrics across multiple experimental runs, combined with efficient evaluation performance, provide strong empirical evidence supporting the effectiveness of the methodology and its potential applicability in real-world handwritten mathematical symbol recognition systems.

5. CONCLUSION

This study achieved its objectives by developing a data-informatics-oriented convolutional neural network (CNN) model capable of accurately recognizing handwritten mathematical symbols. The experimental evaluation demonstrates strong and consistent performance across repeated batch testing. The proposed model achieved an accuracy of 97.10%, precision of 97.15%, recall of 97.08%, and an F1-score of 97.08%, indicating that the model can effectively classify handwritten mathematical symbols with high reliability. The confusion matrix analysis further shows that the model successfully learns discriminative visual features of each symbol class, with most misclassifications occurring only among visually similar symbols. In addition, the integration of the trained model into a graphical user interface demonstrates its practical applicability beyond offline experimental settings. The main contribution of this work lies in the

end-to-end integration of data informatics and CNN-based feature learning, covering data preprocessing, model training, evaluation, and deployment within an application environment. Furthermore, the batch-wise evaluation strategy provides additional insight into the stability and robustness of the model under different dataset partitions. These results highlight the potential of the proposed approach for applications in digital education, automated assessment systems, and mathematical document digitization. Despite these encouraging results, this study has several limitations. The dataset used in this research is relatively small and contains limited variations in handwriting styles, which may affect the model's ability to generalize to broader real-world scenarios. Additionally, the CNN architecture implemented in this study is relatively shallow and relies on fixed hyperparameters, which may not represent the optimal configuration for all data distributions. Future research should focus on expanding the dataset with more symbol classes and diverse handwriting samples, exploring deeper or hybrid architectures such as CNN-RNN or attention-based models, and applying adaptive hyperparameter optimization and advanced data augmentation techniques to further improve model robustness and real-world applicability.

FUNDING INFORMATION

This research was funded by the Adi Buana Inovatif Research Program, Universitas PGRI Adi Buana Surabaya, under contract number 120.1.22/Kontrak/LPPM/XI/2025.

DATA AVAILABILITY STATEMENT

The dataset used in this study is publicly available. The source code and related resources are available in the GitHub repository: <https://github.com/aongxsss/MathWrite>, while the handwritten mathematical symbols dataset can be accessed from Kaggle: <https://www.kaggle.com/datasets/sarunpakkkkkk/handwritten-math-symbols-dataset>

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest to this work.

REFERENCES

- [1] T. Ghosh, S. Sen, S. M. Obaidullah, K. C. Santosh, K. Roy, and U. Pal, "Advances in online handwritten recognition in the last decades," *Computer Science Review*, vol. 46, p. 100515, Nov. 2022, <https://doi.org/10.1016/j.cosrev.2022.100515>.
- [2] M. Silfverberg, "Historical Overview of Consumer Text Entry Technologies," in *Text Entry Systems*, Elsevier, 2007, pp. 3–25.
- [3] P. Gervais, A. Fadeeva, and A. Maksai, "MathWriting: A Dataset For Handwritten Mathematical Expression Recognition," in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, Aug. 2025, pp. 5459–5469, <https://doi.org/10.1145/3711896.3737436>.
- [4] M. Athoillah and R. K. Putri, "Handwritten Arabic Numeral Character Recognition Using Multi Kernel Support Vector Machine," *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, pp. 99–106, Mar. 2019, <https://doi.org/10.22219/kinetik.v4i2.724>.
- [5] R. Dixit, R. Kushwah, and S. Pashine, "Handwritten Digit Recognition using Machine and Deep Learning Algorithms," *International Journal of Computer Applications*, vol. 176, no. 42, pp. 27–33, Jul. 2020, <https://doi.org/10.5120/ijca2020920550>.
- [6] N. Khan *et al.*, "Systematic Literature Review of Machine Learning Models and Applications for Text Recognition," *IEEE Access*, vol. 13, pp. 177647–177670, 2025, <https://doi.org/10.1109/ACCESS.2025.3618109>.
- [7] Y. Chajri and B. Bouikhalene, "Handwritten mathematical symbols dataset," *Data in Brief*, vol. 7, pp. 432–436, Jun. 2016, <https://doi.org/10.1016/j.dib.2016.02.060>.
- [8] B. N. Van and V. T. Hoang, "A Short Review for Handwritten Math Expression Recognition Techniques," *Procedia Computer Science*, vol. 235, pp. 231–239, 2024, <https://doi.org/10.1016/j.procs.2024.04.025>.
- [9] J. Seitz, T. Lengfeld, and R. Timofte, "The Return of Structural Handwritten Mathematical Expression Recognition," Aug. 2025.
- [10] X.-X. Niu and C. Y. Suen, "A novel hybrid CNN-SVM classifier for recognizing handwritten digits," *Pattern Recognition*, vol. 45, no. 4, pp. 1318–1325, Apr. 2012, <https://doi.org/10.1016/j.patcog.2011.09.021>.
- [11] S. Ahlawat and A. Choudhary, "Hybrid CNN-SVM Classifier for Handwritten Digit Recognition," *Procedia Computer Science*, vol. 167, pp. 2554–2560, 2020, <https://doi.org/10.1016/j.procs.2020.03.309>.
- [12] Sakshi and V. Kukreja, "A retrospective study on handwritten mathematical symbols and expressions: Classification and recognition," *Engineering Applications of Artificial Intelligence*, vol. 103, p. 104292, Aug. 2021, <https://doi.org/10.1016/j.engappai.2021.104292>.
- [13] W. AlKendi, F. Gechter, L. Heyberger, and C. Guyeux, "Advancements and Challenges in Handwritten Text Recognition: A Comprehensive Survey," *Journal of Imaging*, vol. 10, no. 1, p. 18, Jan. 2024, <https://doi.org/10.3390/jimaging10010018>.
- [14] Q. Miao and F.-Y. Wang, "AI for Mathematics," *Artificial Intelligence for Science (AI4S): Frontiers and Perspectives Based on Parallel Intelligence*, pp. 21–39, 2024, https://doi.org/10.1007/978-3-031-67419-8_2
- [15] T.-N. Truong, C. T. Nguyen, R. Zanibbi, H. Mouchère, and M. Nakagawa, "A survey on handwritten

- mathematical expression recognition: The rise of encoder-decoder and GNN models,” *Pattern Recognition*, vol. 153, p. 110531, Sep. 2024, <https://doi.org/10.1016/j.patcog.2024.110531>.
- [16] N. Bhatt, N. Bhatt, P. Prajapati, V. Sorathiya, S. Alshathri, and W. El-Shafai, “A Data-Centric Approach to improve performance of deep learning models,” *Scientific Reports*, vol. 14, no. 1, p. 22329, Sep. 2024, <https://doi.org/10.1038/s41598-024-73643-x>.
- [17] N. Saqib, K. F. Haque, V. P. Yanambaka, and A. Abdelgawad, “Convolutional-Neural-Network-Based Handwritten Character Recognition: An Approach with Massive Multisource Data,” *Algorithms*, vol. 15, no. 4, p. 129, Apr. 2022, <https://doi.org/10.3390/a15040129>.
- [18] B. Zhong, X. Xing, P. Love, X. Wang, and H. Luo, “Convolutional neural network: Deep learning-based classification of building quality problems,” *Advanced Engineering Informatics*, vol. 40, pp. 46–57, Apr. 2019, <https://doi.org/10.1016/j.aei.2019.02.009>.
- [19] T. Paranayapa, P. Ranasinghe, D. Ranmal, D. Meedeniya, and C. Perera, “A Comparative Study of Preprocessing and Model Compression Techniques in Deep Learning for Forest Sound Classification,” *Sensors*, vol. 24, no. 4, p. 1149, Feb. 2024, <https://doi.org/10.3390/s24041149>.
- [20] L. Hickman, S. Thapa, L. Tay, M. Cao, and P. Srinivasan, “Text Preprocessing for Text Mining in Organizational Research: Review and Recommendations,” *Organizational Research Methods*, vol. 25, no. 1, pp. 114–146, Jan. 2022, <https://doi.org/10.1177/1094428120971683>.
- [21] M. Athoillah, “K-Nearest Neighbor for Recognize Handwritten Arabic Character,” *Jurnal Matematika “MANTIK,”* vol. 5, no. 2, pp. 83–89, Oct. 2019, <https://doi.org/10.15642/mantik.2019.5.2.83-89>.
- [22] R. K. Putri and M. Athoillah, “Enhancing handwritten numeric string recognition through incremental support vector machines,” *Journal of Applied Math*, vol. 2, no. 1, Jan. 2024, <https://doi.org/10.59400/jam.v2i1.373>.
- [23] I. H. Sarker, “Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions,” *SN Computer Science*, vol. 2, no. 6, p. 420, Nov. 2021, <https://doi.org/10.1007/s42979-021-00815-1>.
- [24] Y. Deng and J. Ma, “SDGMNet: Statistic-Based Dynamic Gradient Modulation for Local Descriptor Learning,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 2, pp. 1510–1518, Mar. 2024, <https://doi.org/10.1609/aaai.v38i2.27916>.
- [25] G. Naidu, T. Zuva, and E. M. Sibanda, “A Review of Evaluation Metrics in Machine Learning Algorithms,” 2023, pp. 15–25.
- [26] B. Juba and H. S. Le, “Precision-Recall versus Accuracy and the Role of Large Data Sets,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 4039–4048, Jul. 2019, <https://doi.org/10.1609/aaai.v33i01.33014039>.

BIOGRAPHIES OF AUTHORS



Muhammad Athoillah received his B.Sc. and M.Sc. degrees in Mathematics from Institut Teknologi Sepuluh Nopember in 2013 and 2015, respectively. Since 2017, he has been a Lecturer in the Statistics Study Program, Faculty of Science and Technology at Universitas PGRI Adi Buana, where he also serves as Head of the Sub-Directorate of Database and University Ranking. His research interests include machine learning, data mining, pattern recognition, image and text processing, with emphasis on Support Vector Machines and deep learning. He has published in nationally accredited and internationally indexed journals (Scopus and WoS) and authored several academic books. He can be contacted at: athoillah@unipasby.ac.id



Rani Kurnia Putri received her B.Sc. degree in Mathematics from Universitas Brawijaya and her M.Sc. degree in Mathematics from Institut Teknologi Sepuluh Nopember. She is currently a Lecturer (Lektor) in the Mathematics Education Study Program at Universitas PGRI Adi Buana. Her research interests span mathematics education and artificial intelligence, including mathematical problem solving, machine learning, Support Vector Machines, multi-kernel learning, and deep learning applications for hoax detection and image recognition. She has published in nationally accredited journals as well as international indexed journals, including Scopus-indexed publications. She can be contacted at: rani@unipasby.ac.id



Fenny Fitriani is a dedicated academic and researcher currently serving as a Lecturer in the Faculty of Science and Technology at Universitas PGRI Adi Buana Surabaya. She earned her Bachelor of Science (B.Sc.) and Master of Science (M.Sc.) degrees, specializing in Mathematics and its scientific applications. Her research interests focus on applied mathematics, mathematical modeling, and education, with a particular emphasis on developing quantitative solutions for real-world problems. She is actively involved in the university's community service initiatives, where she guides students in applying academic knowledge to community development. She can be contacted at: fenny_f@unipasby.ac.id



Prayogo is an academic and researcher serving as a Lecturer in the Master's Program in Elementary Education at Universitas PGRI Adi Buana Surabaya. He specializes in Elementary Mathematics Education, Computation Model and Research Methodology, contributing significantly to the pedagogical development of future educators. His research interests focus on mathematical communication skills, mathematical models and computation, problem-solving structures, and the integration of ethnomathematics in learning. Beyond mathematics, he explores educational psychology and management, including student resilience, self-efficacy, and differentiated learning strategies. He can be contacted via his official university email at: prayogo@unipasby.ac.id.