Handwritten Math Symbol Recognition Based on Multiple Machine Learning Techniques

Dazhi Song

College of Arts and Sciences, The Ohio State University, OH, 43210-1132, USA
song.1700@osu.edu

Abstract. With the inherent complexity and variability of handwritten symbols, accurate recognition is crucial for various applications. This research aims to enhance the recognition of handwritten math symbols through a deep learning model named Convolutional Neural Network (CNN). In particular, the study utilizes a diverse dataset of approximately 370,000 images representing 82 math symbol categories, employing preprocessing and data augmentation techniques to enhance model performance. The implemented CNN model, built with TensorFlow, achieves an accuracy of 0.04. Comparative analysis with Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbor categorization algorithm (KNN) demonstrates the CNN model's inferior performance in terms of accuracy, recall, and F1-scores. This highlights the need for further refinement and optimization of symbol recognition models. Future research should focus on larger and more diverse datasets, exploring different CNN architectures, and optimizing hyperparameters to improve classification accuracy. Despite its limitations, this study contributes to the field of handwritten math symbol recognition, emphasizing the importance of developing reliable and effective solutions for automated interpretation of mathematical expressions.

Keywords: Machine Learning, Convolutional Neural Network, Deep Learning Model.

1 Introduction

Handwritten math symbol recognition is always a crucial component in the area of computer vision and pattern recognition, with the primary objective of developing an automated systems which is capable of understanding and interpreting handwritten mathematical expressions. The ability to accurately recognize handwritten math symbols holds immense potential for various applications, including digitizing handwritten mathematical documents, assisting in mathematics education, and facilitating interaction with digital devices through handwriting recognition.

In the contemporary digital era, the demand to efficiently process and interpret handwritten mathematical expressions has become increasingly significant. Despite the rapid progress achieved in Optical Character Recognition (OCR) technology, the accurate recognition of handwritten math symbols remains a challenging task due to

P. Kar et al. (eds.), *Proceedings of the 2023 International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2023)*, Advances in Computer Science Research 108,
https://doi.org/10.2991/978-94-6463-300-9_24

the inherent variability and complexity of handwriting styles. Consequently, there is a growing demand for robust and reliable handwritten math symbol recognition systems that can facilitate automated processing of mathematical content. By developing effective algorithms and models for handwritten math symbol recognition, this research aims to bridge the gap between manual processing and automated interpretation of mathematical expressions. The successful implementation of such systems can have a profound impact on numerous fields, including education, scientific research, and document analysis.

Numerous investigations have concentrated on the task of recognizing handwritten symbols, tackling various challenges encountered in digit recognition and character recognition across different domains. However, the specific domain of handwritten math symbol recognition poses unique challenges due to the complex nature of mathematical notation, the vast diversity of handwriting styles, and the need for precise recognition of symbols that convey critical mathematical concepts. Previous studies in handwritten math symbol recognition have primarily employed traditional machine learning approaches, such as template matching [1], feature extraction [2], and classification algorithms [3]. However, these approaches frequently rely on handcrafted features, thereby limiting their adaptability to different handwriting styles and hinder their ability to generalize well to unfamiliar data. As a result, these methods often fall short when faced with the intricacies of handwritten mathematical expressions.

Unlike previous studies, the core difference of this paper lies in the adoption of advanced techniques, including deep learning and neural networks, to overcome the limitations of previous studies in handwritten math symbol recognition. By leveraging the power of deep learning, which excels in capturing complex patterns and extracting hierarchical representations, this research project is calculated to develop a novel approach that can significantly enhance the accuracy and robustness of handwritten math symbol recognition systems. By building upon the strengths of deep learning algorithms, this study seeks to address the limitations of existing methods, which struggle to handle the diverse range of mathematical symbols and variations in handwriting styles. This research aims to provide a comprehensive and reliable solution that can accurately interpret handwritten math symbols, thereby facilitating the integration of handwritten mathematical content into digital workflows.

This research primarily contributes to implementing and investigating a Convolutional Neural Network (CNN) model for handwritten math symbol recognition. The study focused on exploring the potential of deep learning techniques in this domain, specifically leveraging the power of CNN. By evaluating how the CNN model performs on a diverse dataset of handwritten math symbols, a comprehensive understanding of its strengths and limitations was obtained. Notably, the comparison with classical machine learning models, including SVM, KNN, and Random Forest, highlighted the challenges faced by the CNN model in achieving accurate symbol recognition. The results revealed that the CNN model exhibited inferior performance compared to these traditional approaches, emphasizing the importance of considering alternative methods for handwritten math symbol recognition tasks. This investigation sheds light on the limitations and potential areas of improvement for CNN-based models in this particular domain, guiding future research efforts towards developing more effective and robust solutions for accurate recognition of handwritten math symbols.

2 Method

2.1 Data Preparation

The dataset used in this study was sourced from Kaggle [4]. The dataset was chosen as it provides a diverse collection of handwritten math symbols, making it suitable for training and evaluating handwritten math symbol recognition models. The dataset consists of approximately 370,000 images, each representing a handwritten math symbol. The images are of varying sizes; however, for this study, they were resized to 64×64 pixels to ensure consistency. To gain a comprehensive evaluation of the performance of the model, a total of 82 categories are cited. Fig. 1 presents three sample images of this dataset. An additional noteworthy aspect pertains to the presence of numerous intricate handwritten mathematical symbols within the employed dataset. These symbols encompass a wide range of complex notations, including differentiation, integration, sigma, infinity, and so on. The high level of complexity strengthens the credibility of this research.

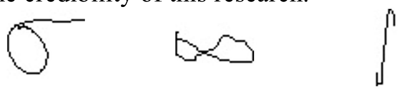

Fig. 1. The sample images of the collected handwritten math symbols [4].

2.2 Convolutional Neural Network

Convolutional Neural Networks (CNNs) have been demonstrated to be considerably practical in image recognition tasks, making them a suitable choice for handwritten math symbol recognition [5-8]. CNNs are a type of deep learning model that mimic the visual cortex's architecture in humans, enabling them to automatically learn and extract characteristics from images. They consist of multiple layers. In this study, three types of layers are included: convolutional layers, pooling layers, and fully connected layers.

The structure of a typical CNN begins with one or more convolutional layers. These layers apply a set of learnable filters to the input image, capturing local patterns and features. Each filter performs a convolution operation, followed by a non-linear activation function such as Rectified Linear Unit (ReLU) [9]. The outputs of the convolutional layers are then down sampled using pooling layers, which reduce the spatial dimensions while retaining important features. This hierarchical process of convolution and pooling allows CNN to learn complex visual representations. Following the convolutional and pooling layers, the feature maps are flattened into a 1D vector and passed through one or more fully connected layers. These layers integrate the learned features and perform highly developed reasoning to classify the input. The last fully connected layer outputs the probabilities of the different classes, which are then passed through a SoftMax activation function to obtain the final class probabilities.

2.3 Implementation Details

The implementation of the CNN model was carried out using the TensorFlow framework [10]. There are two convolutional layers included in the model architecture, each followed by a max pooling layer. The first convolutional layer has 32 filters with a filter size of 3x3, while the second convolutional layer has 64 filters with the same filter size. These layers capture and extract relevant features from the input images. The max pooling layers reduce the spatial dimensions, helping to improve computational efficiency and increase the model's translation invariance.

After the pooling layers, the feature maps are flattened into a 1D vector and passed through two fully connected layers. The first fully connected layer consists of 64 units with a ReLU activation function, allowing for the learning of more complex representations. The final fully connected layer has numerous units which is equivalent to the number of classes, with a SoftMax activation function to acquire class probabilities.

For training the model, the Adam optimizer was used with a default learning rate. The categorical cross-entropy loss function was chosen as it is well-suited for multi-class classification problems. The model was trained for a total of 10 epochs, with a batch size of 32.

After training, the model was evaluated on the test set. The predictions were obtained using the trained model, and classification metrics such as accuracy, precision, recall, and F1-score were calculated. The classification report provides a detailed evaluation of the model's performance for each class. Additionally, the macro average and weighted average metrics were calculated to make an assessment of the overall performance of the model.

3 Results and Discussion

3.1 The Performance of Models

As presented in Table 1, the performance of the CNN model in classifying handwritten math symbols was evaluated using various metrics. The obtained metrics indicate that the CNN model struggled to precisely classify the symbols, as evidenced by the low accuracy of 0.04. This implies that the model had difficulty generalizing well to unseen data and capturing the distinctive features necessary for precise classification.

Table 1. The performance of CNN evaluated by various metrics.

Metrics	Data
Accuracy	0.04
Macro-avg	0.01
Weighted avg	0.04
Recall (Macro Average)	0.0123
Recall (Weighted Average)	0.0403
F1-Score (Macro Average)	0.0121
F1-Score (Weighted Average)	0.0395

3.2 Discussion

Upon further analysis, it is evident that the recall-macro average and recall-weighted average were also considerably low, measuring at 0.012 and 0.040, respectively. These values reflect the model's limited ability to correctly identify instances of each class. Similarly, the F1-score-macro average and F1-score-weighted average were 0.012 and 0.039, respectively, indicating subpar performance in terms of both precision and recall.

To gain a comprehensive understanding of the CNN model's performance, a comparison was made with alternative classification methods, namely Random Forest, SVM, and KNN. The results revealed a stark contrast in performance between the CNN model and these methods.

Random Forest emerged as the most effective method, achieving an accuracy of 0.98, which greatly surpassed the CNN model's accuracy shown in Fig. 2. Additionally, Random Forest demonstrated superior recall values, with a recall-macro average of 0.99 and a recall-weighted average of 0.98. The F1-scores of Random Forest were also notably higher, with a macro average of 0.97 and a weighted average of 0.98.

Similarly, KNN showcased favorable results, achieving an accuracy of 0.95 and recall-macro average of 0.93, along with an F1-score-macro average of 0.80. SVM, although performing slightly lower than Random Forest and KNN, still outperformed the CNN model with an accuracy of 0.83, a recall-macro average of 0.95, and an F1-score-macro average of 0.90.

These comparisons emphasize the limitations of the CNN model in accurately classifying handwritten math symbols compared to other methods. The superior performance of Random Forest suggests that its ability to handle high-dimensional data and capture complex feature interactions contributed to its success in this classification task.

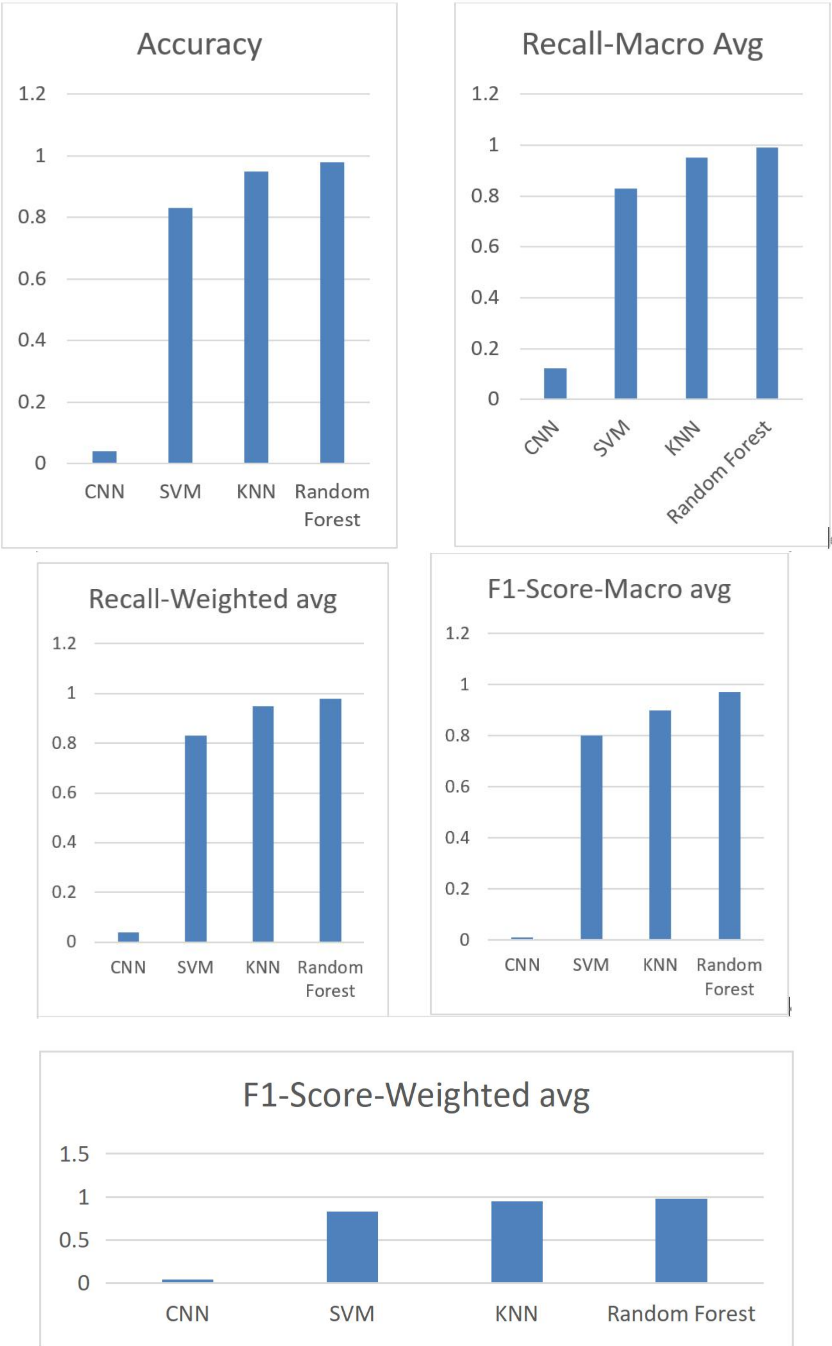

Fig. 2. The comparison of CNN, SVM, KNN and Random Forest evaluated by various metrics (Photo/Picture credit: Original).

4 Conclusion

This study aimed to assess the performance of the CNN model in accurately identifying the symbols and to compare it with alternative classification methods. The results obtained from the CNN model were modest, with an accuracy of 0.04 and relatively low recall and F1-score values. The model struggled to effectively capture the distinguishing features of the handwritten symbols, leading to limited classification accuracy. Furthermore, when compared to alternative methods such as Random Forest, SVM, and KNN, the CNN model fell behind in accordance with accuracy, recall, and F1-scores. However, the performance of the CNN model may have been influenced by various factors, such as the limited dataset's size and diversity, as well as the architecture and hyperparameters of the model itself. Future research could focus on addressing these limitations by incorporating larger and more diverse datasets, exploring different CNN architectures, and optimizing hyperparameters to enhance classification accuracy.

References

1. Simistira, F., Katsouros, V., Carayannis, G.: A template matching distance for recognition of on-line mathematical symbols. Proc. ICFHR, 415-420 (2008).
2. Ramírez-Piña, C., Sánchez, J. S., Valdovinos-Rosas, R. M., et al.: A hybrid feature extraction method for offline handwritten math symbol recognition. Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 23rd Iberoamerican Congress, CIARP 2018, Madrid, Spain, November 19-22, 2018, Proceedings 23. Springer International Publishing, 893-901 (2019).
3. Guan, S. K., Moh, M., Moh, T. S.: Context-based multi-stage offline handwritten mathematical symbol recognition using deep learning. 2019 International Conference on High Performance Computing & Simulation (HPCS). IEEE, 185-192 (2019).
4. Kaggle.: Handwritten Math Symbols. Available at: https://www.kaggle.com/datasets/xainano/handwrittenmathsymbols (2017).
5. Yu, Q., Wang, J., Jin, Z., et al.: Pose-guided matching based on deep learning for assessing quality of action on rehabilitation training. Biomedical Signal Processing and Control, 72: 103323 (2022).
6. Arena, P., Basile, A., Bucolo, M., et al.: Image processing for medical diagnosis using CNN. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 497(1): 174-178 (2003).
7. Yu, Q., Chang, C. S., Yan, J. L., et al.: Semantic segmentation of intracranial hemorrhages in head CT scans, 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS). IEEE, 112-115 (2019).
8. Salehi, A. W., Khan, S., Gupta, G., et al.: A Study of CNN and Transfer Learning in Medical Imaging: Advantages, Challenges, Future Scope. Sustainability, 15(7): 5930 (2023).
9. Hara, K., Saito, D., Shouno, H.: Analysis of function of rectified linear unit used in deep learning. 2015 international joint conference on neural networks (IJCNN). IEEE, 1-8 (2015).

10. Pang, B., Nijkamp, E., Wu, Y. N.: Deep learning with tensorflow: A review. Journal of Educational and Behavioral Statistics, 45(2): 227-248 (2020).