Investigation on Handwritten Mathematical Symbol Recognition Based on the Combination of CNN and KNN Method

Yisong Zhang

Computer Science and Technology, Wuhan University of Technology, Wuhan, 430070, China
zhangyisong@whut.edu.cn

Abstract. Recognizing handwritten mathematical symbols presents a significant obstacle due to the inherent variability in individuals' writing styles. In order to enhance the accuracy of symbol recognition, this scholarly article introduces a pioneering methodology that synergistically merges the capabilities of the Convolutional Neural Network (CNN) and the K-nearest Neighbors algorithm (KNN). This approach endeavors to leverage the respective advantages offered by both CNN and KNN, with the ultimate objective of advancing the accuracy of symbol identification. Primarily, the CNN model undergoes multiple rounds of training to augment its feature extraction capabilities. Subsequently, the extracted features are employed for training and classification predictions within the KNN framework, yielding the final predicted results. To evaluate the performance of this approach, tests are conducted on the Handwritten Math Symbols dataset from Kaggle, and comparisons are made with methods that solely employ CNN or KNN. All three models are evaluated using identical training and testing datasets. The results demonstrate that the combined CNN and KNN approach outperforms in various performance indicators, achieving an ultimate accuracy of 98.7%. This evidences the superior performance of this method in the task of handwritten mathematical symbol recognition.

Keywords: CNN, KNN, Machine learning, Handwritten mathematical symbols recognition

1 Introduction

Mathematical symbols are markers and notations used to represent mathematical concepts, relationships, and operations. Their utilization extends across various domains, including education and scientific research, particularly in online teaching and scientific documentation, where mathematical formulas play a crucial role. However, the process of generating conventional mathematical formulas can pose significant challenges. While using tools such as LaTex for mathematical type-setting is a widely known method, in online teaching scenarios where formulas need frequent modification, it still incurs inconvenience to users. Furthermore, this method also requires a certain learning curve. Therefore, recognizing hand-written mathematical

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_66

formulas and converting them into their respective standardized expressions can greatly improve work efficiency and convenience.

Drawing upon feature selection, the existing approaches for recognizing hand-written formula symbols can be classified into online and offline methods. The online mode relies mainly on structural and non-structural features of symbols for analysis and processing. For instance, Mouchere et al. outlined a methodology called NCGF to extract feature vectors from online sequences and then used a generalized vector quantization classifier for symbol classification [1]. The offline mode primarily processes images, as demonstrated by Fang et al. [2], who used a convolutional neural network algorithm for recognizing hand-written formula symbols, and Nazemi et al. used the SqueezeNet model for symbol classification [3].

Currently, various machine learning algorithms have been widely used for image recognition and classification problems. The main types of machine learning algorithms can be divided into traditional algorithms such as Random Forest (RF) [4], KNN and Support Vector Machine (SVM) [5, 6], and deep learning algorithms such as Recurrent Neural Network (RNN) and CNN [7, 8]. Depending on the nature of the problem, a range of data preprocessing techniques can be employed. For instance, Principal Component Analysis (PCA) can be utilized to reduce dimensionality, or algorithmic adjustments can be implemented to optimize performance. For example, Lu et al. proposed an enhanced model combining particle swarm optimization with support vector machines [9], leading to a remarkable recognition rate of 96% in detecting rice blast. Their approach effectively addressed the limitations of conventional discrimination methods. Shen et al. [10], on the other hand, employed convolutional neural networks and channel attention mechanism to achieve an accuracy rate of 86.38% for melanoma skin disease recognition. These studies demonstrate that machine learning algorithms have been widely applied to image classification and recognition in many fields and have achieved significant results.

To explore a more accurate method for predicting mathematical symbols based on given images, this paper proposes a method that combines traditional machine learning algorithm KNN with deep learning algorithm CNN. CNN extracts features from images, and these features are used as inputs for KNN for classification and recognition. The Handwritten Math Symbols dataset on Kaggle was mainly used as the sample, which contains 82 categories with a total of 375,974 image samples. The dataset was randomly divided into training and testing sets, following an 8:2 ratio, to assess and compare the performance of three algorithms. The conclusive outcome reveals an exceptional accuracy rate of 98.7% for the proposed algorithm, which combines CNN and KNN, showcasing an impressive performance. Compared to other baseline algorithms, this algorithm showed a higher performance and faster training speed, and with the same data volume, it can more effectively perform image recognition and classification. This research provides new ideas and methods for recognizing hand-written mathematical symbols, with significant implications for improving work efficiency and convenience in education and scientific research fields.

2 Method

2.1 Dataset Preparation

The dataset called Handwritten math symbols dataset from Kaggle was employed in this study [11]. It consists of 375,974 images covering 82 different categories. These categories include common Greek alphabet symbols, mathematical operators, set operators, basic predefined mathematical functions, and other mathematical symbols. Each image in the dataset is saved in the jpg format, with a fixed size of 45×45 pixels and RGB color channels. The sample images of them can be found in Fig. 1.

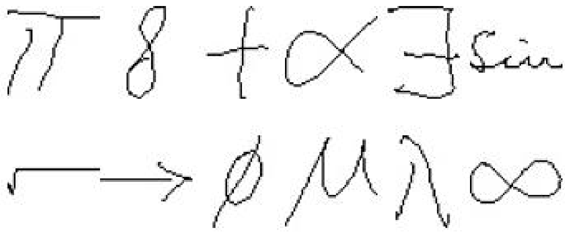

Fig. 1. some simple examples from the dataset [11].

In terms of the data obtained from the dataset, a series of preprocessing operations were applied. Firstly, to address the issue of inconsistent formatting of symbol category names, a dictionary was defined to map the class labels to unique numerical codes. Secondly, the image data was normalized by scaling the pixel values to the range of [0, 1], achieved by dividing them by 255 to convert pixel values from the range of 0-255 to 0-1.

Additionally, when using only the KNN algorithm, a data preprocessing operation called PCA was introduced to avoid the curse of dimensionality caused by high-dimensional data. PCA was used to minimize the number of dimensions in the data while keeping the key characteristics and enhancing the accuracy and efficiency of the method.

2.2 Developed CNN Combined with KNN Model

Introduction of CNN. Convolutional, pooling, and fully connected layers make up the foundation of CNN architecture shown in Fig. 2. When employing convolution operations to extract features from input data, convolutional layers are essential. Downsampling, a technique that decreases the size and quantity of parameters in feature maps to lessen computational complexity while maintaining the most crucial data from each local region, is done using pooling layers. The fully connected layers flatten the features obtained from convolution and pooling, and classify or regress predictions through multiple neurons. By learning the weights and biases, the extracted features are mapped to the final output categories or values.

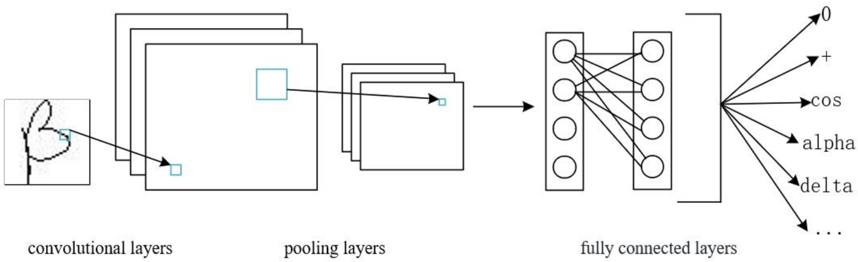

Fig. 2. the structure of the CNN (Photo/Picture credit: Original).

The convolutional and pooling layers are effective in extracting feature information from images, so multiple convolutional and pooling operations are employed to extract the most important features. This CNN model consists of a total of nine layers, which include an input layer, three convolutional layers, three max pooling layers, a flatten layer, and a fully connected layer. Among them, the 3 convolutional layers have different numbers and sizes of convolutional kernels. Convolutional layer 1 employed 32 3×3 convolutional kernels, convolutional layer 2 uses 64 3×3 convolutional kernels, and convolutional layer 3 uses 32 3×3 convolutional kernels. The three max pooling layers perform max pooling operations with a window size of 2×2, reducing the dimensionality of feature maps by half. This facilitates reducing computational complexity and overfitting risks while preserving significant features. Through the stacking of multiple convolution and pooling operations, the constructed CNN model gradually extracts abstract features from the input images, which are then connected to a fully connected layer with 82 output nodes. In order to generate a probability distribution that reflects the likelihood of each category, the softmax activation function is employed in the final fully connected layer.

Introduction of KNN. In KNN, each sample is accompanied by its corresponding class label. When there is a new unknown sample that needs to be classified, the specific steps include distance calculation, selecting the nearest neighbors, and making a voting decision.

Firstly, for the sample to be predicted, KNN calculates its distance to each sample in the training set. In this project, the Euclidean distance is employed as the measure of the distance between samples.

$$d(x,y) = \sqrt{(x1 - y1)^2 + \ldots + (xn - yn)^2} = \sqrt{\sum_{i=1}^{n} (xi - yi)^2} \quad (1)$$

The KNN algorithm frequently uses the Euclidean distance to calculate the separation of two points in a multidimensional space along a straight line. The KNN algorithm captures the distances between the sample to be predicted and each training sample, calculating the Euclidean distance between the forecasted sample and all samples in the training set. These distances are then stored in a distance matrix.

The KNN method then selects the sample's K nearest neighbors. In accordance with the selected value of K, the K samples with the smallest distances can be selected from the distance matrix. These nearest neighbor samples will serve as references for voting.

Finally, a voting decision is made based on the class labels of the nearest neighbor samples. Each nearest neighbor sample contributes a voting option for the classification prediction. By tallying the occurrences of each class label among the nearest neighbors, KNN selects the most frequently occurring class label as the classification result for the sample to be predicted, thus completing the classification prediction for new unknown samples.

Introduction of CNN combined with KNN method. This study combines CNN and KNN to extract features from pictures using the convolutional layers and pooling layers of CNN, followed by KNN classification shown in Fig. 3.

The structure of CNN is identical mentioned before. Subsequently, these extracted features are fed into a KNN model for training and prediction. The Euclidean distance formula is still employed as the distance calculation formula, finding the K nearest points and using voting to determine the predicted class.

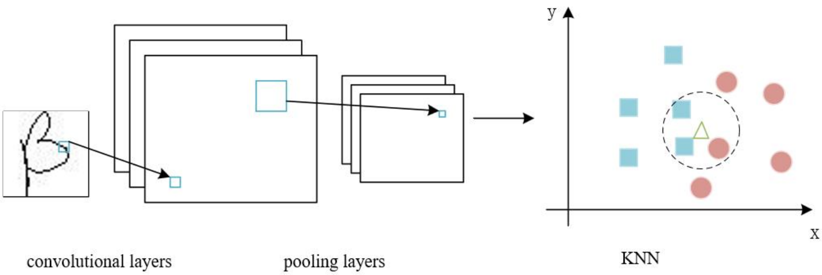

Fig. 3. CNN combined with KNN method process (Photo/Picture credit: Original).

2.3 Implementation Details

The construction of the three models all implemented based on the TensorFlow framework. The CNN model was compiled using the Adam optimizer, which is coupled with the sparse categorical cross-entropy loss function. Accuracy was utilized as the evaluation metric during the compilation process. The model was trained using stochastic gradient descent on 50 batches of data, each batch consisting of 20,000 samples.

After performing grid search for parameter tuning, the final selection for the KNN model was PCA with a parameter of 300 and K value of 2. In the model combining CNN and KNN, after comparison, the K value was also chosen as 2. To compare the performance of these three models, the classification_report method from the scikit-learn library was utilized to print the confusion matrix. Performance was assessed based on metrics such as accuracy, f1-score, precision, and recall.

3 Results and Discussion

3.1 The Performance of Models

Based on the findings, there exists a discernible trend in the accuracy of the KNN algorithm as K value varies, as depicted in Fig. 4. Furthermore, the accuracy of the CNN combined with KNN method also fluctuates with changes in K value, as observed in Fig. 5. Taking these trends into consideration, a K value that equals to 2 was chose as the optimal parameter selection.

Moreover, Fig. 6. illustrates the variations in the loss function and accuracy of the CNN algorithm throughout the training process. These results provide valuable insights into performance optimization during the training phase of the CNN algorithm.

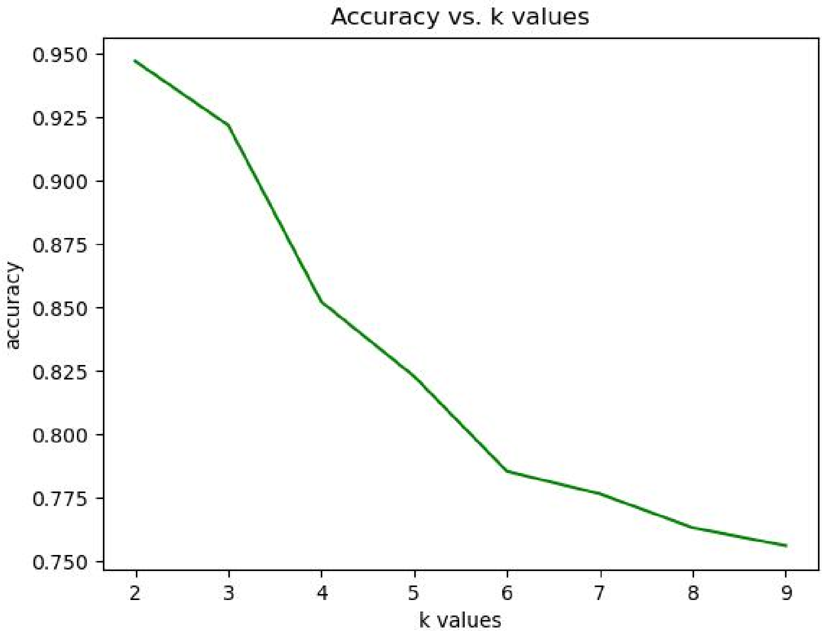

Fig. 4. Accuracy variation by KNN (Photo/Picture credit: Original).

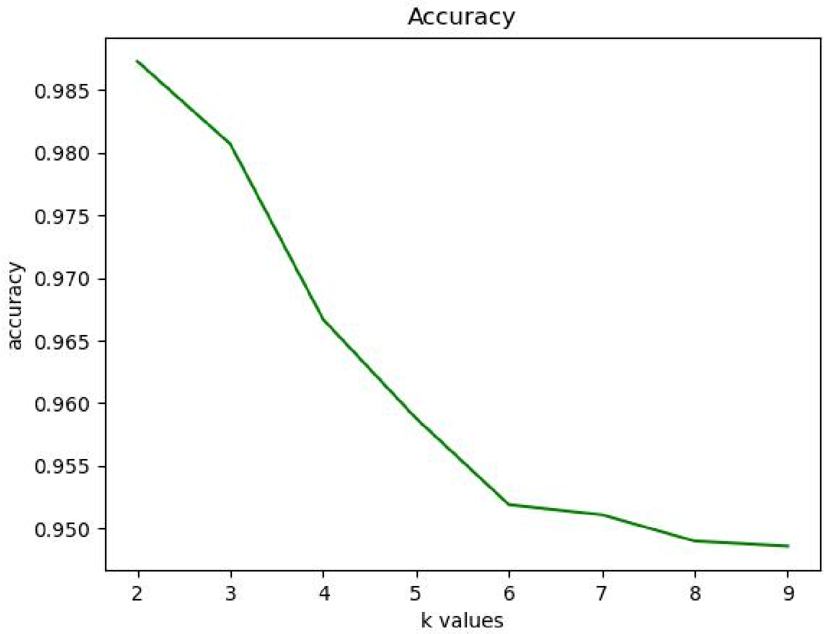

Fig. 5. Accuracy variation by CNN combined with KNN (Photo/Picture credit: Original).

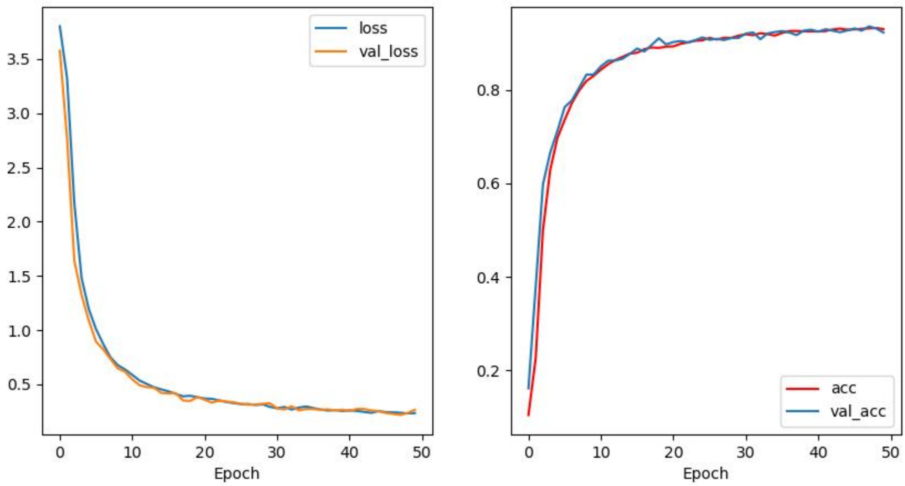

Fig. 6. The training curve of the CNN (Photo/Picture credit: Original).

Table 1 showcases a comparative analysis of performance metrics for three algorithms, encompassing accuracy, precision, f1-score, and recall. It is noteworthy that the latter three metrics incorporate both macro and weighted averages. Upon perusing Table 1, it becomes apparent that the amalgamation of CNN with KNN (K=2) algorithm outshines the other two algorithms in all aspects.

Table 1. Comparison of three algorithms

Heading level	CNN	KNN(K=2)	CNN combined with KNN(K=2)
accuracy	0.93	0.95	0.99
precision (macro avg)	0.89	0.95	0.98
recall (macro avg)	0.86	0.90	0.96
f1-score (macro avg)	0.86	0.92	0.97
precision (weighted avg)	0.93	0.95	0.99
recall (weighted avg)	0.93	0.95	0.99
f1-score (weighted avg)	0.93	0.95	0.99

3.2 Discussion

When it comes to image feature extraction, CNN demonstrates superior capability compared to PCA. CNN not only reduces the dimensionality of the data but also learns to extract advanced features from the images. By employing multiple convolutional and pooling layers for hierarchical abstraction, CNN is able to learn local and global features from the training set images. These learned features possess stronger discriminative power, enabling better differentiation between different categories of images. In contrast, PCA solely achieves dimensionality reduction through linear transformations and fails to capture the complex features present in images. Due to the discriminative nature of the features provided by CNN, the KNN algorithm can effectively leverage these features to form nonlinear decision boundaries. KNN classifies based on the proximity of samples in the feature space, and when CNN-extracted features better differentiate between different image categories, KNN can more accurately utilize these features for classification.

4 Conclusion

This study introduces a novel approach that combines CNN and KNN for accurately predicting mathematical symbols in images. The experimental results showcase the algorithm's exceptional performance, achieving an impressive accuracy rate of 98.7% on a dataset comprising handwritten math symbols. The approach utilizes CNN to extract features from images, which are then used as inputs for KNN classification and recognition. Compared to other algorithms, this method exhibits higher performance, effectively improving the efficiency of image recognition and classification. In the future, further study will combine Natural Language Processing (NLP), edge detection algorithms, and other methods to better understand the meaning of handwritten formulas.

References

1. Mouchere, H., Viard-Gaudin, C., Zanibbi, R., et al.: ICFHR 2014 competition on recognition of on-line handwritten mathematical expressions (CROHME 2014). 2014 14th International Conference on Frontiers in Handwriting Recognition. IEEE, 791-796 (2014).
2. Dingbang, F.: Algorithm research on handwritten mathematical formula character recognition based on convolutional neural network. Huaqiao University. DOI: 10.27155/d.cnki.ghqiu.2020.001042 (2020).
3. Nazemi, A., Tavakolian, N., Fitzpatrick, D., et al.: Offline handwritten mathematical symbol recognition utilizing deep learning. arXiv preprint arXiv:1910.07395 (2019).
4. Speiser, J. L., Miller, M. E., Tooze, J., et al.: A comparison of random forest variable selection methods for classification prediction modeling. Expert systems with applications, 134: 93-101 (2019).
5. Zhang, S., Li, X., Zong, M., et al.: Learning k for knn classification. ACM Transactions on Intelligent Systems and Technology (TIST), 8(3): 1-19 (2017).
6. Huang, S., Cai, N., Pacheco, P. P., et al.: Applications of support vector machine (SVM) learning in cancer genomics. Cancer genomics & proteomics, 15(1): 41-51 (2018).
7. Qiu, Y., 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).
8. Li, S., Li, W., Cook, C., et al.: Independently recurrent neural network (indrnn): Building a longer and deeper rnn, Proceedings of the IEEE conference on computer vision and pattern recognition. 5457-5466 (2018).
9. Yang, L., Hualong, Y., Yu,C. et al.: Rice blast disease image recognition based on TSDPSO-SVM. Jiangsu Agricultural Sciences, 50(23): 164-170. DOI: 10.15889/j.issn.1002-1302.2022.23.025 (2022).
10. Haojie, S., Jili, H., Qingyu, X. et al.: Melanoma skin disease image recognition algorithm based on deep learning. Electronic Technology and Software Engineering, No. 249(07): 194-199 (2023).
11. Kaggle, Handwritten math symbols dataset https://www.kaggle.com/datasets/xainano/handwrittenmathsymbols (2017).