IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 3, May 2025
Impact Factor: 7.67

Mathematical Script Resolver using Convolutional Neural Network

Rahul Kumar Srivastava, Anirudh Som, Raman Nirwal, Sameer Gupta
Department of Computer Science And Engineering (Artificial Intelligence)
Meerut Institute of Engineering and Technology, Meerut

Abstract: *This research investigates the development of a robust equation solver for handwritten equations utilizing Convolutional Neural Networks (CNNs). Recognizing handwritten mathematical expressions presents a significant challenge in computer vision and machine learning due to the variability and complexity of handwritten symbols. The paper reviews various object and character recognition methods and their applications, emphasizing the deep learning architecture involving CNNs. A dataset of handwritten equations is employed to evaluate the proposed solution, focusing on accuracy, fault tolerance, and potential improvements. The results demonstrate the effectiveness of the CNN-based approach in accurately recognizing and solving handwritten equations, highlighting future directions for enhancing the model's performance and expanding its capabilities.*

Keywords: Convolutional Neural Networks

I. INTRODUCTION

In addition, image processing faces developing a robust equation solver for handwritten equations using CNNs. Handwritten math expression recognition is, however, the most challenging problem among many others in computer vision and machine learning. Other ways of developing methods for object and character recognition have found a wide range of applications in traffic monitoring, autonomous self-driving cars, weapon detection, and natural language processing. Deep learning is a branch of machine learning and uses neural networks to perform the process of feature extraction from complex data. The deep learning architecture therefore involves multi-layered data understanding wherein CNN is the core application that involves convolutions, activation functions, pooling, and classification layers. Deep learning has achieved great results in the field of image analysis problems and also finds its usefulness in handwriting recognition, robotics, artificial intelligence, and image processing.

The rest of the article is organized as: Section 2 provides a review of recent research works in handwritten character recognition, Section 3 describes the contribution of each component in CNN, and Section 4 presents the proposed algorithm of deep learning for recognition of handwritten equations and the dataset used. Then, Section 5 compares different approaches using technical metrics, and finally, Section 6 concludes the work by mentioning the scope for future work.

II. LITERATURE REVIEW

Handwritten digit recognition has been a topic of interest in various fields where techniques like SVM, Naive Bayes, CNN, and K-Nearest Neighbors have been employed. Among these, CNN proved to be brilliant in the subject area.

Several works have proved the authenticity of CNN in recognizing handwritten digits.

Agarwal et al. [4] proposed an offline handwritten character recognition system that utilized CNN and TensorFlow. It achieved an accuracy of more than 90%. Bharadwaj et al. [6] proposed a system in which handwritten digit detection was done using Deep Convolutional Neural Networks in the MNIST dataset with an accuracy of 98.51%. Their model, however, was unable to recognize characters if the image from which it had to read the characters was blurred or noisy.

Gawas et al. proposed a system to recognize handwritten digits and symbols to solve simple equations that contain those operations. The developed system created a front-end interface for users to write down equations, using CNN.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 3, May 2025
Impact Factor: 7.67

However, the trained model was limited to basic mathematical operations, while the model did not extend to linear equations.

III. CONVOLUTIONAL NEURAL NETWORK

A Convolutional Neural Network, sometimes referred to as a ConvNet, is a deep neural network developed and utilized for high accuracy in image processing, benefiting from a grid-like topology. Such a network is a feed-forward neural network that consists of several stacked layers. As depicted in Fig. 1, the CNN is trained based on the raw pixel data of images; thereafter, it extracts features to enhance the accuracy of classification.

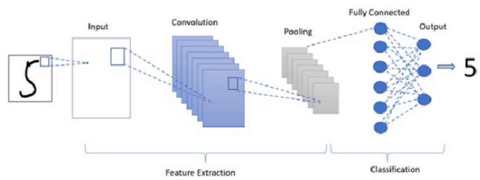

Fig 1: CNN architecture for handwritten images

Convolutional layers, fully connected layers, and pooling layers make up the three differ ent kinds of layers that are included in a deep neural network.

3.1 Convolution Layer

The Convolution Layer is the first step for any process of information extraction in input images. It performs a dot product between the input data and a two-dimensional array of weighted parameters, also called kernel or filter, as illustrated in Fig. 2

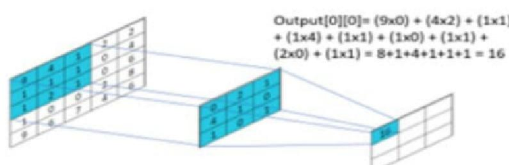

Fig 2 : Convolution Layer

3.2 Pooling layer

The Pooling Layer is utilized for downsampling the feature maps with the purpose of reducing the parameters involved. This, in turn, speeds up computation. There are three main types of pooling layers, including: (1) Max Pooling, which selects the maximum value from the feature map region shown in Fig. 3, computes (2) Average Pooling-the average value from the feature map region, and (3) Global Pooling-applies a filter on all the dimensions of the feature map.

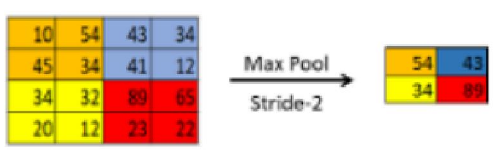

Fig 3: Max Pooling

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 3, May 2025
Impact Factor: 7.67

3.3 Fully Connected Layer

The Fully Connected Layers represent the last layers within the neural network. It could be seen from Fig. 4, that in case a preceding layer is fully connected this means that each neuron of it connects to each neuron in the layer below. According to the approach proposed, there are two fully connected layers involved followed by a classification layer.

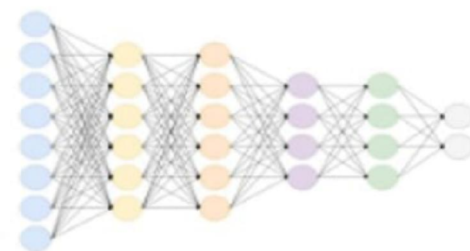

Fig 4 : Full Connected Layer

3.4 Activation Function

Activation Function is used as shown in Fig. 5 in order to activate neurons, it tells whether neurons fire or not and it sets the output of the convolution layer. Some common activation functions are Sigmoid, ReLU, Leaky ReLU, and Softmax.

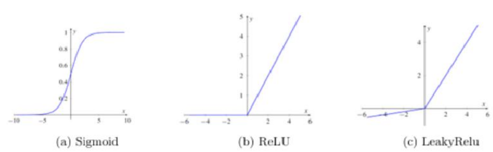

Fig 5 : Several different kind of activation functions

The mathematical representations for Sigmoid and Softmax are presented in Equations 1 and 2, respectively.

$$\sigma(x) = \frac{1}{1+e^{-x}} \qquad \text{---------------Eq(1)}$$

$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \qquad \text{-----Eq(2)}$$

IV. HANDWRITTEN EQUATION RECOGNITION

4.1 Dataset Preparation

Obtaining a dataset is the first step in research. Obtaining data containing numerals, operations, and characters from Kaggle, augmentation was done to increase the dataset. The dataset consists of about 24,000 images across 16 categories: the digits from 0 through 9, variables and five basic mathematical operators-addition, subtraction, multiplication, equals and division-presented in Fig. 6.

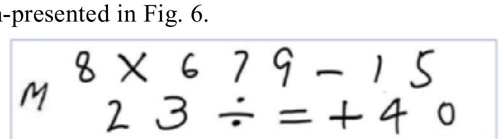

Fig 6 : sample images in the data set

V. RESEARCH METHODOLOGY

The designed CNN model will be able to identify in complex equations: a simple arithmetic equation such as addition, subtraction, multiplication, and division; a linear equation such as x + a = b, where x is a variable and a and b are constants. The block diagram of the implemented model has been described with the following description as shown in Fig. 7.

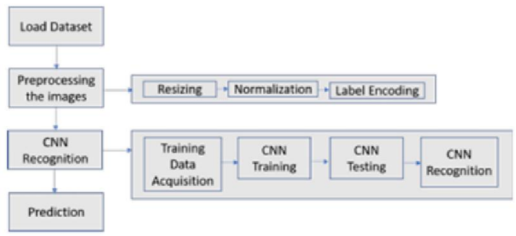

Fig 7 : Block Diagram of the implemented model

5.1 Dataset Acquisition

This contains approximately 24,000 images of handwritten characters, which are further divided into the training set and the test set. This training set includes around 1,300 images, while roughly 50 images represent every class in the test set. All images were preprocessed by resizing, cropping, and padding for standardization of the dataset, so the resolution is 95 x 84 for digits and 94 x 89 for the character 'M'. Images were further resized to 100 x 100 to improve the training and results.

5.2 Preprocessing

Preprocessing of images increases their suitability for analysis. Some of the preprocessing techniques used include: (i) Image Augmentation: This artificially increases the number of datasets using rotation, shearing, shifting, and flipping. Image augmentation techniques allow one artificially to increase the size of a dataset through rotation, shearing, shifting, and flipping. Then, the process proceeds with: (iii) Normalization: it normalizes the range of pixel intensities to lie between 0 and 1 for computational convenience. (iv) Label encoding: this step translates categorical labels into numeric values readable by machines.

5.3 Recognition through CNN Model

It was trained using handwritten datasets with different augmentations, such as shearing, rotation, and shifting. The total number of images used is approximately 23,000 for training and 950 for testing. For added variability, data augmentation techniques were performed to help the CNN model recognize handwritten digits more effectively.

5.4 Processing inside CNN Model

The model in this script, developed with a sequential approach, considers seven Conv2D layers, four MaxPooling2D, and six dropout layers with a dropout rate of 0.2. The choice of activation function evolved from Sigmoid to ReLU to Leaky ReLU for faster training. Softmax activation is used in the final dense layer as it deals with multi-class classification. The model uses the Adam optimizer and embeds the architecture of InceptionV3 supported by several convolution layers and pooling layers.

VI. SOLUTION APPROACH

The solution approach described in Fig. 8, by its flowchart. The flowchart has in input a hand written math equation provided by the user.

IJARSCT

International Journal of Advanced Research in Science, Communication and Technology

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Volume 5, Issue 3, May 2025

ISSN: 2581-9429

Impact Factor: 7.67

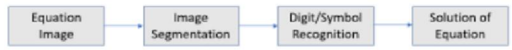

Fig 8 : Pipeline of the proposed model

Image Segmentation: This involves dividing the image into segments, which reduces computational complexity and helps analyze it easily. It segregates the areas of interest that are to be classified or detected. The image, as shown in Fig. 9, segmented into fixed proportions commonly in two numerals and one operator. Here, the segmentation is of a 1 : 3 ratio. The model here works under the constraint of having the middle segment as an operator and the outer segments as the numerals.

The steps of the segmentation algorithm are presented in Fig. 10. These segmented images were first thresholded using Otsu's Algorithm and then normalized before feeding them into the model for training. Each of these segments is resized to 100x100. Later on, the model will extract and recognize characters or operators in these segments. The final goal was to classify each segment, solve an equation using a mathematical formula based on a trained model.

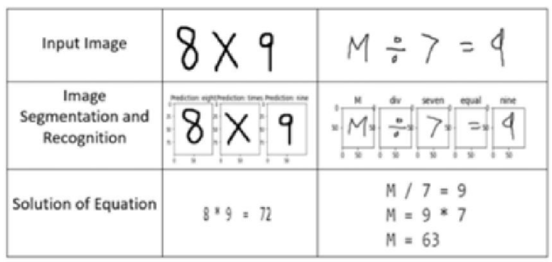

Fig 9: Practical implementation of proposed model

Algorithm 1 Algorithm for Handwritten Equation Solver using CNN Model

```
Input: Image of Handwritten Linear Equations
Image Cropping: Crop the input image into the required segments; n is number of segments.
for i = 1 to n do
    height, width = size(img)
    div = int( width / n)
    left = i * div
    top = 0
    right = (i+1) * div
    bottom = height  segs = img.crop(left, top, right, bottom)
Image Resizing: Resized to 100 X 100 for uniform training set size
cropped_segs = segs.resize(100,100)
Thresholding:
bin_segs = cv2.threshold(cropped_segs)
Normalization: By dividing all pixel values by the highest pixel value
bin_segs = bin_segs / 255
Output: Each blocks are recognized seperately containing characters or operators.
```

Fig 10 : Algorithm for handwritten equation solver

REFERENCES

[1]. Tian, X. D., Li, H. Y., Li, X. F., & Zhang, L. P. (n.d.). Research on Symbol Recognition for Mathematical Expressions. Proceedings of the First International Conference on Innovative Computing, Information, and Control, ICICIC'06, doi: 10.1109/icicic.2006.506

[2]. Okamoto, M. (1991). Recognition of Mathematical Expressions by Using the Layout Structure of Symbols. Proceedings of the 1st International Conference on Document Analysis and Recognition, pp. 242-250.

[3]. Faure, C., & Wang, Z. X. (1990). Automatic perception of the structure of handwritten mathematical expressions. In Computer Processing of Handwriting, pp. 337-361.

IJARSCT

International Journal of Advanced Research in Science, Communication and Technology

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

ISSN: 2581-9429

Volume 5, Issue 3, May 2025

Impact Factor: 7.67

[4]. Okamoto, M., & Miyazawa, A. (1992). An experimental implementation of a document recognition system for papers containing mathematical expressions. In H. S. Baird, H. Bunke, & K. Yamamoto, Structured Document Image Analysis, pp. 365-373, Springer-Verlag, Berlin.

[5]. Ha, J., Haralick, R. M., & Phillips, I. T. (1995). Understanding mathematical expressions from document images. In Proceedings of the 3rd International Conference on Document Analysis and Recognition (Vol. 2, pp. 956-959). IEEE.

[6]. Chou, P. A. (1989). Recognition of equations using a two-dimensional stochastic context-free grammar. In Visual Communications and Image Processing IV (Vol. 1199, pp. 852-865). International Society for Optics and Photonics.

[7]. Nakayama, Y. (1989). Mathematical formula editor for CAI. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '89) (pp. 387-392). Association for Computing Machinery, New York, NY, USA. DOI: https://doi.org/10.1145/67449.67523

[8]. Belaid, A., & Haton, J. P. (1984). A syntactic approach for handwritten mathematical formula recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-6(1), 105-111. doi: 10.1109/TPAMI.1984.4767483

[9]. Chan, K. F., & Yeung, D. Y. (1999). Recognizing on-line handwritten alphanumeric characters through flexible structural matching. Pattern Recognition, 32(7), 1099-1114.

[10]. Chen, L. H., & Yin, P. Y. (1992). A system for on-line recognition of handwritten mathematical expressions. Computer Processing of Chinese and Oriental Languages, 6(1), 1939