MANTECH Publications

Journal of Artificial Intelligence, Machine Learning and Soft Computing
Volume 7, Issue 1, January-June 2022

Recognition and Solution for Handwritten Equation Using Convolutional Neural Network

Asmita Ghosh[1], Vanshika[2], Harshvardhan Kharpate[3]

Students[1, 2], Professor[3]

Department of Computer Engineering

MKSSSS Cummins College of Engineering for Women

Email ID: *mailtoasmitaghosh@gmail.com[1], harshvardhan.kharpate@cummimscollege.edu.in[3]*

DOI: *https://doi.org/10.5281/zenodo.21620333*

Abstract

In recent years, the recognition of handwritten mathematical expressions has received an increasing amount of attention in pattern recognition research. The diversity of approaches to the problem and the lack of a commercially viable system, however, indicate that there is still much research to be done in this area. In this thesis, I will describe an on-line approach for solving a handwritten mathematical expression. For classification of specific characters we apply Convolutional Neural Network. Each of the correct detection, character string operations is used for the solution of the equation. Finally the experimental results show the great effectiveness of our proposed system.

Keywords: *Handwritten, Convolutional Neural Network, Pattern Recognition, proposed system*

INTRODUCTION

The problem of machine recognition of handwritten expressions has long been a focus of study in the field of pattern recognition. Research in this area has been driven by a desire to combine the natural advantages of handwritten input, including a simple interface and a well-established stylistic vocabulary, with the data processing capabilities of computers. Recently, the problem has been approached with increased vigor with the advent of palmtop computers with pen interfaces, which possess enough processing resources to handle the demands of machine recognition. As a result, a number of commercially successful products are available which

Page 33-44 © MANTECH PUBLICATIONS 2022. All Rights Reserved

recognize a user's natural handwriting and use this ability to perform simple tasks such as scheduling appointments and writing memos. Most scientists and engineers, however, are unable to take advantage of these computers for their technical work due to the lack of effective algorithms for interpreting more complex handwritten expressions, particularly diagrams, graphs, equations, and other mathematical forms. While computers can store these forms as "digital ink," the inability to work with the expressions in a meaningful way after they have been entered has prevented these systems from replacing pencil and paper. Compared to the effort put into the recognition of printed and cursive prose, the recognition of more complex forms has received only minor attention in pattern recognition research. In addition, the diversity of approaches to the problem and the lack of a commercially viable system indicate that there is still much research to be done in this area. As more powerful computers with better displays and input devices become available, demand will increase substantially for software systems which can work with the type of handwritten data that one would find in a research notebook or technical document. Mathematical expressions are a natural place to begin such research as they are critical to virtually all technical writing and there already exists a wide body of literature on recognizing handwritten letters and words, major subcomponents of these expressions. Combining mathematical expression recognition capabilities with existing algebra solving software, graphing programs and simulation systems would be a first step towards a superior user interface for doing technical work with a computer.

LITERATURE SURVEY

SolveIt: An Application for Automated Recognition and Processing of Handwritten Mathematical Equations

Solving mathematical equations is an integral part of most, if not all forms of scientific studies. Researchers usually go through an arduous process of learning the nuances and syntactic complexities of a mathematical tool in order to solve or process mathematical equations. In this paper, we present a mobile application that can process an image of a handwritten mathematical equation captured using the device's camera, recognise the equation, form the corresponding string that can be parsed by a computer algebraic system and display all possible solutions. We aim to make the whole experience of experimenting with equations very user friendly and to remove the hassle of

Page 33-44 © MANTECH PUBLICATIONS 2022. All Rights Reserved

learning a mathematical tool just for mathematical experimentation. We propose a novel machine learning approach to recognise handwritten mathematical symbols achieving a 99.2% cross validation percentage accuracy on the kaggle math symbol dataset with reduced symbols. The application covers useful features like simultaneous equation solving, graph plotting and simple arithmetic computations from images. Overall it is a very user friendly equation solver that can leverage the power of existing powerful math packages.

Handwritten Equatio Solver Using CNN

Using CNN to create a robust handwritten equation solver is a difficult task in image processing. One of the most difficult challenges in computer vision research is handwritten mathematical expression recognition. The work is made more difficult by the fact that certain characters are segmented and classified. a collection of quadratic equations created by hand This study looks at quadratic equations as well as a single quadratic equation. These equations must be recognized and solved. Horizontal compact projection analysis is used for segmentation. We use both connected component analysis and integrated connected component analysis methodologies. Convolution Neural Networks Characters are classified using a network. Each appropriate solution is required for the solution of the problem. Character string operation is used for detection. Finally, the results of the experiment show that the strategy we've described is quite effective. The goal of this project is to create a handwritten alphabet. Equation solver capable of dealing with a wide range of mathematical equations.

Recognition and Solution for Handwritten Equation Using Convolutional Neural Network

This paper basically focus on numerous technique used for feature extraction and recognition. An effective and robust system for recognition of printed and handwritten mathematical characters has been proposed by Zanibbi. This technique uses diagonal, horizontal and vertical features for the classification purpose. A few papers are available online for recognition of mathematical expression using convolution neural networks.

For feature extraction, the shape of the character is considered. The proposed system is based on relative study of feature extraction methods.

Page 33-44 © MANTECH PUBLICATIONS 2022. All Rights Reserved

Recognition of Handwritten Mathematical Expressions

The problem of interpreting an expression can be divided into three modular subproblems called isolated symbol classification, expression partitioning and parsing. This division has the advantage that each subproblem can be solved and its performance evaluated essentially independent of the others, so that improvements can be made in each area while still maintaining the integrity of the entire system. In processing an expression, it is first partitioned into symbols in a process called expression partitioning. The symbol classifier is used in this process to evaluate the likelihood that particular strokes should be combined into symbols.

System Architecture

Convolutional layer, pooling layer, completely connected input layer, fully connected layer, and fully connected output layer are all layers in iNdesign.

- *Convolutional layer:* The backbone of any CNN working model is the convolutional layer. This layer is where the images are scanned pixel by pixel and a feature map is created to define future classifications.

- *Layer for pooling:* Pooling is also known as data down sampling, in which the total dimensions of the photos are reduced. Each feature's information from each convolutional layer is condensed to only include the most essential data. The creation of convolutional layers and the use of pooling is a continuous process that may require multiple iterations.

- *Fully connected input layer:* The flattening of the images is also known as the fully linked input layer. The previous layer's outputs are flattened into a single vector.

- *Fully connected layer:* When it's time to compute after the feature analysis, this layer applies random weights to the inputs and predicts a suitable label. Fully connected

- *Output layer:* The CNN model's final layer stores the results of the labels determined for classification and assigns a class to the images.

Proposed System

In our proposed method at first noise from the original input image is removed by applying binarization to it. After that we use compact horizontal projection for segmenting each line of the equation from the input image. Then we consider each part of the segmented image as a full image for further process. For each line of

Page 33-44 © MANTECH PUBLICATIONS 2022. All Rights Reserved

the equation image we then find specific characters in the form of a connected component. Each segmented character is then provided as input to the convolutional neural network model for classification of the character. The resulting character that is the output of CNN is then used for making a character string which is similar to the original equation.

For each correct detection we finally find the solution of the quadratics. In Figure 1 the workflow diagram of our proposed method is shown.

Dataset Preparation

Preparation of the dataset is the fundamental concern for this work. Characters such as the English digit, alphabet and mathematical symbol all can be well defined by their edges. For this reason we first prepare the dataset as the most precedence given to its edges that enlighten the edges. We prepared some dataset by ourselves and we also used some modified version of NIST dataset which is similar to the popular MINIST dataset for digit. For each category we use 2000 data items for the training of the network. And most of the time our network training we gained more than 98.5% training accuracy. Image size we used in the dataset is 32×32 gray level image.

B. *Pre-Processing*

Pre-processing of the input image is the procedure which encompasses changes and modifications to the image to make it fit for recognition. The following techniques may be used for image enhancement.

1) Conversion of RGB to Gray-Scale

First of all this coloured image is transformed into a typical gray-scale image and is represented through a single matrix because the detection of characters on a coloured image is more challenging than on a gray-scale image. IF the gray bitmap Y and color bitmap is R,G and B then the formula we Used is

$$Y = 0.299R + 0.587G + 0.114B$$

2) Binarization

Binarization is the procedure of choosing a threshold value for adaptation of pixel values into 0's and 1's. In This research for horizontal projection calculation 1's represent the black pixels while 0's characterize the white pixels. The threshold choice of binarization can be approved in two ways: overall threshold and partial threshold. Otsu's method is an overall threshold method, based on image

Page 33-44 © MANTECH PUBLICATIONS 2022. All Rights Reserved

statistical characteristics. This method can make the computer automatically select the threshold.

3) Noise Reduction

Excessive pixels that are present in an image is called noise. Noise may be in the form of Salt and Pepper noise or Gaussian noise. Low pass filtering is used to remove the Gaussian noise from the image [22] and there is no need to filter Salt and Pepper noise as it is very low as compared to the Gaussian noise. In our proposed method we removed all components which are less than 5 pixels for simplicity of small unwanted pixel noise.

C. ***Segmentation***

Identifying the objects or other significant information in digital images segmentation is mostly used in image processing and computer vision application. Which is the process of dividing an image into multiple parts[19]. The segmentation involves two main steps in our proposed method.

a) Equation line Segmentation

Equation segmentation is the split-up of the different lines of characters existing in the image. Each line is well-defined by a minimum vertical gap between the characters existing on a line and on the line overhead and below it. This gap can be used for the detection and separation of not the same lines of characters. For the solution of square term in quadratics as mentioned above we use compact horizontal projection. Combining two curves into one that is representing one equation line with two curves is done by comparing with a threshold value that is obtained by averaging the maximum value of each horizontal projection vector. If any curve's maximum value is less than the threshold this curve is then considered as a part of the next projection curve.

Algorithm 1: Compact horizontal projection

Step 1. Calculate N horizontal projection vectors, X.

Step 2. Find the maximum of each vector.

Step 3. Calculate Threshold,

$$T_h = \frac{\sum_{i=1}^{N} max(Xi)}{N}$$

Step 4. For each curve if max(X)< Th , Combine the curve with next one for segmentation

b) Character Segmentation

Character segmentation is a procedure that looks to decompose an image of a series of characters into sub images of individual symbols [19]. In our proposed method we

Page 33-44 © MANTECH PUBLICATIONS 2022. All Rights Reserved

use a connected component analysis method for the segmentation of specific characters from the image. Problem arises at the point of extracting the math symbol '=' which is a combination of two connected components. If two consecutive components have the minimum height and are in a same horizontal direction we can consider this two into a single component. If an outside rectangle of two symbols region with connected regions is set to X(x1, y1, x2, y2) and Y(x3, y3, x4, y4). Now if these two symbols X and Y have minimum height and belong to a whole horizontal direction then the combination of X and Y produces a third component Z that is our desire.

The position of Z can be obtained as

Z.x 5=min(x1,x3)

Z.x6=max(x2,x4)

Z.y5=max (y2, y4)

Z.y6 =min(y1 ,y3)

Where the event x and y variable represent the right and top position of the rectangle respectively, odd x and y variable represent respectively the left and bottom position of the rectangle.

D. Classification Model

It is a simple way to prevent overfitting in the neural network. Overfitting is a stern difficulty in such networks. Large networks are also sluggish to use, making it more challenging to contract with overfitting by combining the predictions of many different large neural nets at the test time. The technique for addressing this problem is Dropout. The core idea is to randomly drop units (along with their connections) from the neural network during training. In our proposed method we use the probability 0.3 at the dropout layer during the training. After the convolutional and down- sampling layers one or more fully connected layers are used. The layer in which the neurons associate to all the neurons in the previous layer is a fully connected layer. Fully connected layers combine all the features learned by the previous layers through the image to recognize the larger patterns. To classify the images the last fully connected layer combines the features. Hence, the Output parameter in the last fully connected layer is equal to the number of classes in the target data. In our work, the output size is 14, corresponding to the 14 classes. An activation function softmax which normalizes the output of the fully connected layer is also used in the convolutional neural network model. The softmax layer output comprises positive numbers that sum to one, in the nest that can be used as the classification

Page 33-44 © MANTECH PUBLICATIONS 2022. All Rights Reserved

probabilities at the classification layer. At the final layer, the classification layer uses the probabilities given by the softmax activation function to find the input image classes and find the loss by comparing it with the pre assigned ground truth classes.

E. Solution of Quadratics

After successful detection of the quadratics we organized the character string in the form of '+ay^2+by+c=0'. Now for solving these quadratics our job is to find the value of a, b, c. where a,b,c can be a single digit number or a multiple digit number. To find the value of a, b, c we apply a character string operation algorithm as Algorithm 2. After finding the parameter of the quadratics we apply the formula for the solution of the quadratic equation.

Algorithm 2 Quadratics parameter calculation

Step 1. Find the position i of y^2, y and '=' in the character string s. Initialize sum=0.

Step 2. [For each of a, b, c]From j= i-1 to till s(j)!= '+' or s(j)!= '-'decrease j by 1. Sum =10*[s(j)-48]+sum.

Step 3. If s(j)== '-' then sum= - sum problem by considering two consecutive components with the smallest width and also considering two components with the same horizontal direction as shown in figure 3 (d). Most complex part is the classification part; generally several characters are distinct in shape. But for the different handwritten strategy some the character creates ambiguity to each other. For example 'y' and '2' are two separate characters with different structures. But when '2' is used as a square term it shape is then something like as the figure 3 which is sometimes similar to the structure of the 'y'

METHODOLOGY

- **Methodologies Explanation:** First, noise from the original input image is reduced using our proposed method by binarizing it. Then, from the input image, we utilize compact horizontal projection to segment each line of the equation. Then, for subsequent processing, we treat each segment of the segmented image as a full image. We then look for certain characteristics in the form of related components for each line of the equation image. After that, each segmented character is fed into a convolutional neural network model for character categorization. The resulting character, which is CNN's output, is then utilized to create a

Page 33-44 © MANTECH PUBLICATIONS 2022. All Rights Reserved

character string that looks like the original equation.

- Dataset preparation the preparation of the dataset is the most important aspect of this project. The borders of characters like the English numeral, alphabet, and mathematical symbol can all be accurately defined. As a result, we begin by preparing the dataset with the highest priority given to its edges, i.e., illumine the edges. We created our own datasets and used a modified version of the NIST dataset for digits, which is similar to the popular MINIST dataset. For the training of the network, we use 2000 data items for each category. And in the majority of cases, our network training was accurate to the tune of 98.5 percent. We used a 32x32 gray level image in our dataset.

- Pre-processing the procedure of changing and modifying the input image to make it suitable for recognition is known as preprocessing. Image enhancing techniques include the ones listed below. 1) Conversion of RGB to Gray-Scale because character detection on a colored image is more difficult than a grayscale image, this colored image is first turned into a conventional gray-scale image and represented through a single matrix. If the gray bitmap is Y and the color bitmaps areR, G, andB,the formula is Y = 0.299R + 0.587G+0.114B.

2) Binarization- Binarization is the process of converting pixel data into 0s and 1s by selecting a threshold value. In this study, 1s indicate black pixels and 0s represent white pixels in the horizontal projection computation. Binarization thresholds can be approved in two ways: overall threshold and partial threshold. Otsu's method is based on picture statistical properties and is 11 an overall threshold method. This strategy allows the computer to choose a threshold on its own. 3) Noise Reduction Noise refers to the presence of too many pixels in an image. Salt and pepper noise and Gaussian noise are two types of noise. Low pass filtering is used to remove Gaussian noise from the image, and Salt and Pepper noise does not need to be filtered because it is relatively low in comparison to Gaussian noise. For the sake of simplicity, we deleted all components that are less than 5 pixels in our proposed solution.

Page 33-44 © MANTECH PUBLICATIONS 2022. All Rights Reserved

- Segmentation Image processing and computer vision applications frequently use segmentation to identify objects or other key information in digital images which is the division of one image into several parts. In our proposed method, segmentation is divided into two parts.

Design

Data is collected first in our implemented approach. The data was then standardized. Normalization is divided into two parts: training data and testing data. The training data is then fed into a Convolution Neural Network, which treats each portion of the image as a complete image for processing.

We then look for certain characteristics in the form of related components for each line of the equation image. Each segmented character is then sent into a Convolutional neural network model, which is used to classify the character.

The character obtained as a result of CNN is then utilized to create a character string that is similar to the original equation. The correctness of these characters is then checked. The answer is predicted by the prediction model, which is then delivered as an output. We eventually get the quadratics' solution for each correct detection.

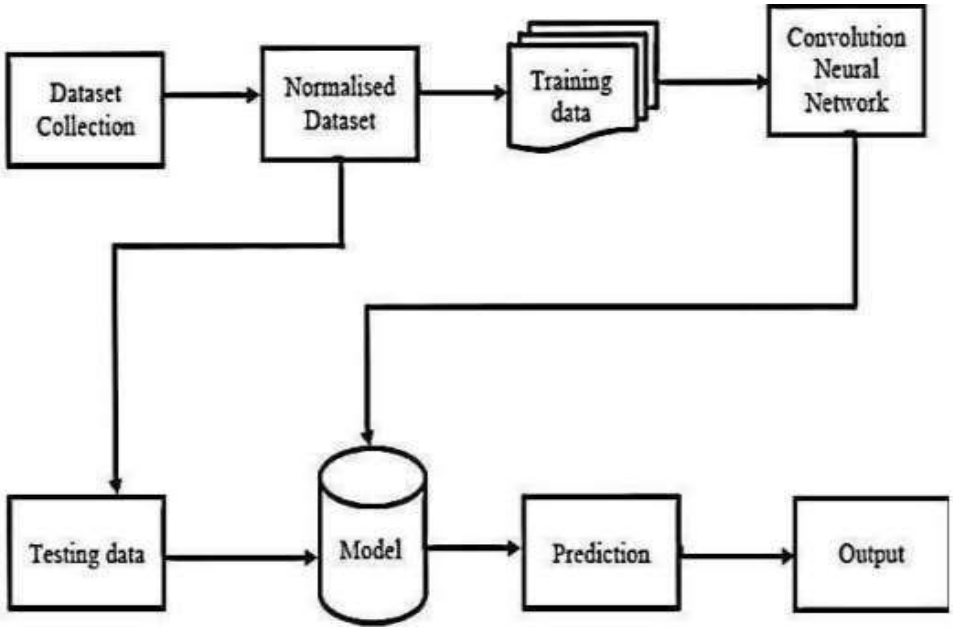


Figure: 1

Page 33-44 © MANTECH PUBLICATIONS 2022. All Rights Reserved

RESULT

We discovered the MNIST handwritten digit recognition problem and deep learning models developed in Python using the Keras library that are capable of achieving excellent results.

Working through this tutorial you learned:

- To load the MNIST dataset in Keras and generate plots of the dataset.
- To reshape the MNIST dataset and develop a simple but well performing multi-layer perceptron model on the problem.
- To use Keras to create convolutional neural network models for MNIST.
- To develop and evaluate larger CNN models for MNIST capable of near world class results.

CONCLUSION

In this paper we mainly focused on recognizing handwritten mathematical quadratics. Connected components which have a very high success rate are used for character segmentation. Improved version of the connected component is used for the symbol like '=' detection which is a single symbol combined with two distinct connected components Feature extraction is the most complicated part of classification. Moreover with some predefined features about handwriting it is difficult to recognize handwritten. Convolutional Neural Network the most powerful classification model is used in the classification part. Once successful recognition of the quadratics in any combination we further process the detected equation for finding the solution of the quadratics. In this part we apply an string operation algorithm for finding the value of a, b, c of each quadratics in the form of ay2+by+c=0. Finally we obtained a state of art performance in the detection phases and also in the solution phases. In future we will try to improve the accuracy and try to make the system workable for multiple mathematical formulae simultaneously. Through the use of convolutional neural network we will try to make a vision based mathematical formula dictionary in future.

FUTURE SCOPE

Automating the grading process not only helps the teachers and also creates a comfortable learning experience for the students. This solution can be made even cooler by recognizing more complex mathematical equations like differential

Page 33-44 © MANTECH PUBLICATIONS 2022. All Rights Reserved

integral equations, recognition of cursive handwriting where the character is not separated, detecting plagiarism and recognizing chemical equations.

REFERENCES

1. Handwritten Digit Recognition using Convolutional Neural Networks in Python with Keras
2. https://machinelearningmastery.com/handwritten-digit-recognition-using-convolutional-neural-networks-python-keras/
3. Computer Vision — Auto grading Handwritten Mathematical Answer Sheets
4. https://towardsdatascience.com/computer-vision-auto-grading-handwritten-mathematical-answersheets-8974744f72dd
5. Handwritten equations to LateX
6. https://opensourc.es/blog/he2latex/
7. SolveIt: An Application for Automated Recognition and Processing of Handwritten Mathematical Equations
8. https://ieeexplore.ieee.org/abstract/document/9058273
9. Handwritten Equation Solver Using Cnn
10. Recognition and Solution for Handwritten Equation Using Convolutional Neural Network
11. Recognition of Handwritten Mathematical Expressions
12. https://youtu.be/u3FLVbNn9Os
13. https://youtu.be/eyKwPyOqMg4
14. https://youtu.be/uMKF3bpH9Ec
15. https://youtu.be/1VBuDk21J44
16. https://github.com/Handwritten-Equation-Solver

Page 33-44 © MANTECH PUBLICATIONS 2022. All Rights Reserved