



RESEARCH ARTICLE

Automated Handwritten Equation Solver

Shereen A. Hussein; Ahmed M. Abd el Azim; Ahmed Hagag

Published online: 25 March 2025

Abstract

Mathematics has an important role in person's life, so solving the mathematical equations is an essential. Solving mathematical expressions is not restricted to just students but also for mathematicians, physicists and scientists. Solving the mathematical equations is an interesting process. The traditional method of solving math expressions is unsatisfactory as the user should learn different rules and approaches for each mathematical equation. Also, these methods may take long time in complex or obscure problems which makes them subject to user errors and mistakes. The challenging in mathematical expressions must be written in a specific format, users prefer to write them on paper as an easy entering way than other computerized tools. This paper used the technology to introduce a new method over the traditional one using pen and paper. The equation handwriting easiness is blended (merge/integrate) with the advanced computer technologies speed to solve the equations with flexible robust way. An interface introduces that allows capturing the equations contained in an image then solving it without making the user dive into the complex rules. Various types of equations could be entered to this application (linear/nonlinear/quadratic) with achieving a convenient accuracy 95.7%.

Keyword: Handwritten Equations Pre-processing, Mathematical Expressions Validity, Building a Solver Model, Convolutional Neural Network (CNN), Sympy

Introduction

Mathematical expressions approximately are used in most scientific fields like Physics, Math, chemistry, etc. So solving mathematical expressions is not restricted to just students but also for mathematicians, physicists and scientists. And it has become necessary to find an easy way to solve these equations and mathematical expressions.

The technological approaches face the hardness of in writing the mathematical expressions special format. Most users prefer to draw expressions by hand as they do with pen and paper [1-3]. Hence, a new system is proposed to allow user capturing image that contains the mathematical expression and trying to solve it precisely. This will reduce the time required for solving any mathematical problem that used old fashioned techniques.

Related Work

Other existing systems will be reviewed to get more information about their methods and techniques in solving the problem and their results. In 2019, Shiyin Wei, Xiaowei Jin and Hui Li [4] have proposed a novel rule-based deep reinforcement learning (DRL) approach for solving differential equations without any built-in knowledge about the equations, and the equation is embedded in the network as the critic, which makes sense from a physical point of view. In 2016, Sagar Shinde and Rajendra Waghulade [5] have used the forward back propagation neural network technique or recognition of handwritten quadratic mathematical equations by considering rectangle with respect to position, edge color, line width then finding region properties bounding box and centroid for every component. In 2013, Amey Chavan, Asad Naik [6] have introduced an Android-based system for linear mathematical equations using OCR. Image goes through many stages for enhancement, such as binarization by Otsu algorithm then text recognition is done by Google Tesseract. After a confirmation from the user, the detected equation text is solved with parser and grammar rule.

Proposed Solution

The system is divided to four stages, each of them is a prerequisite for the next one. The detailed architecture of the system of solving handwritten equations is shown in Figure 1.

Computer Science Department, Faculty of Computers & Information, Fayoum University, Fayoum, Egypt

**) corresponding author*

Email: sam26@fayoum.edu.eg

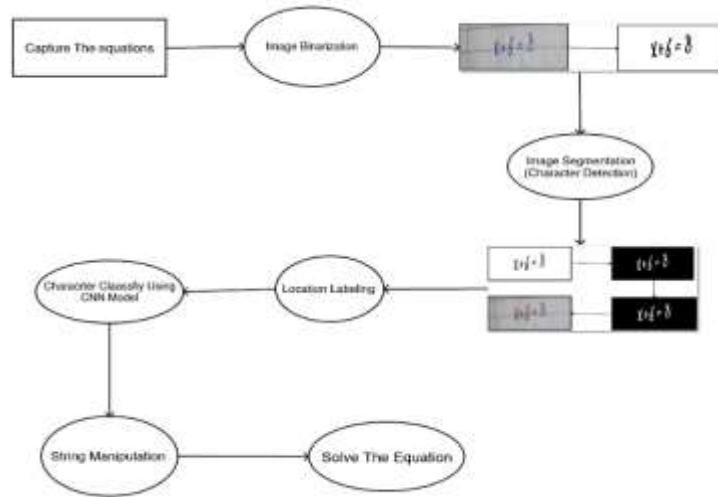


Fig 1. The Proposed System Architecture.

Method

Image Pre-processing

The first phase of the proposed system to solve the handwritten equations is the pre-processing. The image pre-processing is where the binarization and segmentation have been occurred.

1) Image Binarization

In binarization process as Figure 2, As HSV color system has more control on the image colors than RGB. it is used to detect the range of colors of the expression and apply an in-range mask to get the binary image in both complex and non-complex backgrounds [7]. Then some dilation / erosion filters to connect gaps between the characters.

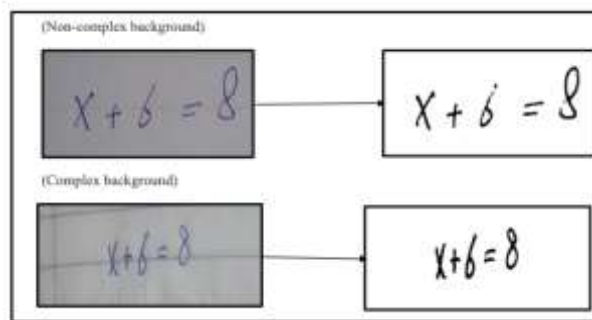


Fig 2. Equation Binarization Process in HSV Color Model

2) Image Segmentation

In segmentation as in Figure 3, find contours functions are applied to get the minimum bounding box for every contour of the binarized characters [8]. Then label the segmented characters of the expression corresponds to the position of the character in the image if it's superscript, subscript, main coefficient or variable.

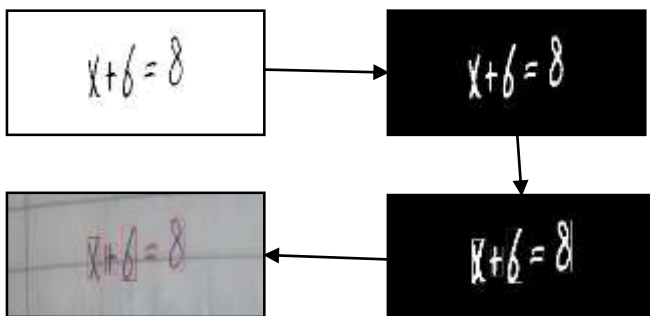


Fig 3. Equation Segmentation Process

Classification Modelling

The second phase is recognizing the components of the handwritten equation. Either, classifying of handwritten digits as integer values in range 0 – 9 in 10 classes, or symbols used in the mathematical expressions [11-16].

Datasets

Two datasets of handwritten mathematical characters are used in the classification model; Data division into training and testing sets to adjust the algorithm parameters with overcoming the overfitting problem

1. CHROME (Competition on Recognition of Online Handwritten Mathematical Expressions) [9] provides more than 10,000 expressions handwritten by hundreds of writers from different countries.
2. The **MNIST dataset** (Modified National Institute of Standards and Technology) [8] [10] is consist of 60,000 small square images of handwritten single digits between 0 and 9.

Digits Recognition

The convolutional neural network (CNN) is used to classify and recognize the equation characters in the given image as digits and symbols [16-20]. The CNN input layer in both training and testing phase is a gray scale image with size 32×32 and the convolutional layer is 5×5 filter. After the convolution of the input image with the filter for each input image a 28×28 feature vector is produced [20-24]. As it is shown in Figure 4, the architecture of CNN has different types of layers, the description of these layers is as follows:

1. *Input layer*: it is used to feed the model with the handwritten expression character.
2. *Convolution layer*: it consists of a set of learnable filters with responsibility of the task of character recognition.
- 3) *Pooling layer*: it is a layer between the convolutional layers to reduce the dimensionality of images by reducing the number of pixels in the output from the previous convolutional layer without losing important features or patterns.
- 4) *Fully connected layer*: The system may have multiple fully connected layers, this layer has fully connected to ReLU activation function all activations in the previous layer.

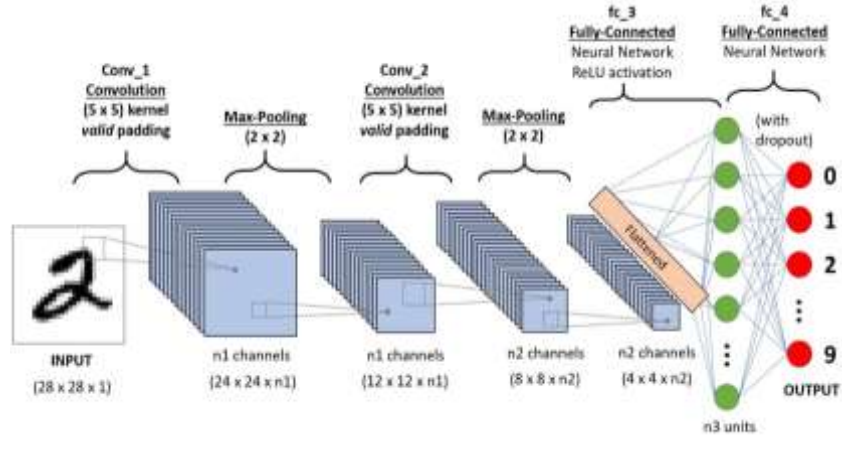


Fig 4. CNN Architecture

Equation Solving

The third phase of solving the equation is very important due to two reasons, the first is to identify the validity of the text string that was entered by the user and the second reason is to work on determining the type of mathematical equation and solve it then display its result [8]. It contains two main functions which are string manipulation and checking string validity.

String Manipulation

Manipulating the input text is needed to be compatible with the traditional human method in writing the equations. It is done by translating the special mathematical symbols to words by its name to detect the equation clearly, and deal with it. The normal handwritten equation actually is three-lines in a single line; main text, superscript and subscript text [25, 26]. In the previous segmentation phase, every segment in the image is labeled by number that determine its position in the line, 1 for superscript text and -1 for subscript text, so string manipulation class convert the three lines to one single line as this equation 1.

$$\text{e.g. } \sum_{i=1}^n (i^2) \rightarrow \text{Summation } (i^{**} (2), (i, 1, n)) \quad \text{equation 1}$$

Check String Validity

Before applying the solution method, the output string that represents the equation must be validated in many form such as it has No inputs, has unknown character [?, -] or the equation is wrong or has undefined function [sin(x)]. Figure 5 shows some of these invalid messages for the input equation.

```

enter an equation:
y=2x
*****
[['y', '2x']]
invalid input

enter an equation:
x?+5
*****
[['x?+5']]
invalid input

sum(x)=0
*****
[['sum(x)', '0']]
invalid input

[sum(x) = 0]
could not solve sum(x)
    
```

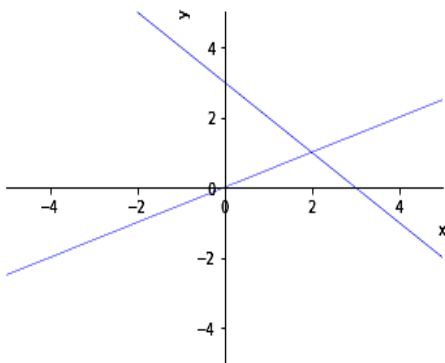
Fig 5. Invalid Equation Samples

After the equation validation comes the last step in the proposed system in identifying the appropriate solving method then visualizing the result. Solving the different equation using SymPy library [27] which is a computer algebra system written in pure python. SymPy allows the user to create and solve complicated math quickly and easily. As it is shown in figure 6 of solving several given equations.

```

enter an equation:
x=2*y;x+y=3
*****
[['x', '2*y'], ['x+y', '3']]

[X = 2y, x + y = 3]
sys= [Eq(x, 2*y), Eq(x + y, 3)]
    
```



{x:2, y:1}

Fig 6. Equation Solving Sample

Experimental Results

Based on the following table 1, the highest accuracy is 95 % and the lowest is 85%. The other algorithm yields an average accuracy of around 91.08% [6][8]. In fact, the experimental results reveal that proposed system outperforms the other systems in dealing with multiple kinds of mathematical equations. Also with handling the distortion that might be occurs in the image making the background complex. Figure 8, expresses by graph the relationship between the accuracy of recognizing the digits and symbols of the expressions and the number of complete passes through the training dataset (epochs).

Table 1. Comparison between proposed system and other competitor systems

| Features | Proposed System | Linear equation solver in Android using OCR | Recognition and solution of Handwritten Equation Using CNN |
|------------------|-----------------|---|--|
| Camera | Yes | Yes | Yes |
| Linear Equations | Yes | Yes | No |
| Non-linear | Yes | No | Quadratic only |

| | | | |
|------------|-------|-----|--------|
| Equations | | | |
| Complex | Yes | No | No |
| Background | | | |
| Calculator | Yes | No | No |
| Accuracy | 95.7% | 85% | 91.08% |

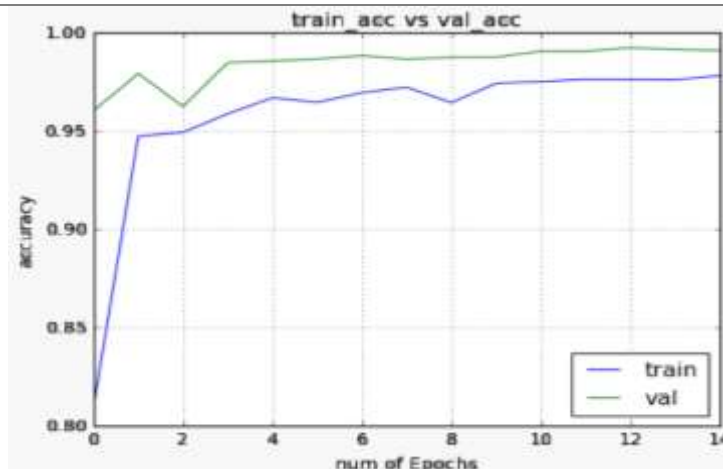


Fig 7. Training & validation accuracy vs. num of epochs

Conclusion

The main purpose of the system is facilitating the process of solving mathematical equations, and reduce the time consuming of solving it. As discussed above, there are three stages on the development process of the system. Image preprocess using most common and efficiently techniques and algorithms to prepare the image. The classification operation which uses a pre-trained model with Deep learning algorithms to provide a real time characters prediction. Finally, using a most resent and powerful python library to solve the equation which known as 'SymPy'. The system can solve different kinds of equations with complex background by dealing with each line individually. Nevertheless, the multiple line equation image might cause law accuracy in character detection and in the final result, The system looking forward to overcome such an issue by improving Character detection techniques. Also, it aims to solve equation of complex numbers.

References

- [1] L. Anthony, J. Yang and K. Koedinger, "Benefits of Handwritten Input for Students Learning Algebra Equation Solving", Proc. Artificial Intelligence in Education. Amsterdam, the Netherlands, IOS Press, 521-523, 2007.
- [2] L. Maftelii-Scai, "Solving Equations Systems Using Artificial Intelligence - a Survey", International Journal of New Computer Architectures and their Applications, vol. 8, no. 3, pp. 102-118, 2018.
- [3] S. Mori, C. Y. Suen, and K. Yamamoto, "Historical review of OCR research and development," Proc. IEEE, vol. 80, no. 7, pp. 1029-1058, 1992.
- [4] S. Wei, X. Jin, and H. Li, "General solutions for nonlinear differential equations: a rule-based self-learning approach using deep reinforcement learning," Computational Mechanics, vol. 64, no. 5, pp. 1361-1374, 2019.
- [5] S. Shinde and R. Waghulade, "Handwritten Mathematical Expressions Recognition using Back Propagation Artificial Neural Network," Communications on Applied Electronics, vol. 4, no. 7, pp. 1-6, 2016.
- [6] Chavan, "Linear equation solver in Android using OCR", IOSR Journal of Engineering, vol. 03, no. 05, pp. 42-44, 2013.
- [7] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation". Journal of Electronic Imaging vol 13, no.1, pp. 146-165, 2004.
- [8] H. Md Bipul, N. Feroza, J. Y.A., I. Md Zahidul, U. Md Jashim, "Recognition and Solution for Handwritten Equation Using Convolutional Neural Network", International Conference on Informatics, Electronics & Vision and International Conference on Imaging, Vision & Pattern Recognition (icIVPR), 2018.
- [9] H. Mouchere, C. Viard-Gaudin, R. Zanibbi, and U. Garain, "ICFHR2016 CROHME: Competition on Recognition of Online Handwritten Mathematical Expressions," 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), 2016.
- [10] E. Kussul and T. Baidyk, "Improved method of handwritten digit recognition tested on MNIST database", Image Vis. Compute., vol. 22, no. 12, pp. 971-981, 2004.
- [11] S. Shamim, M. Miah, A. Sarker, M. Rana and A. Jobair, "Handwritten Digit Recognition Using Machine Learning Algorithms", Indonesian Journal of Science and Technology, vol. 3, no. 1, p. 29, 2018.
- [12] M. Adnan, F. Rahman, M. Imrul, N. AL and S. Shabnam, "Handwritten Bangla Character Recognition using Inception Convolutional Neural Network", International Journal of Computer Applications, vol. 181, no. 17, pp. 48-59, 2018.
- [13] S. Sourav, S. Sudipta, C. Suhrid and S. M. Priya Ranjan, "A Machine Learning Framework for Recognizing Handwritten Digits Using Convexity-Based Feature Vector Encoding", Kolkata, India, eHaCON, 2018.
- [14] M. McDonnell, M. Tissera, T. Vladusich, A. van Schaik and J. Tapson, "Fast, Simple and Accurate Handwritten Digit Classification by Training Shallow Neural Network Classifiers with the 'Extreme Learning Machine' Algorithm", PLOS ONE, vol. 10, no. 8, p. e0134254, 2015.
- [15] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling", IEEE Trans. Pattern Anal. Mach. Intel., Vol. 35, no. 8, pp. 1915-1929, 2013.

- [16] J. Pradeep, E. Srinivasan and S. Himavathi. "Neural Network based Handwritten Character Recognition system without feature extraction", International Conference on Computer, Communication and Electrical Technology ICCET, 2011.
- [17] R. Jagan, R. A., "Recognition of Handwritten Characters using Deep Convolutional Neural Network," International Journal of Innovative Technology and Exploring Engineering Special Issue, vol. 8, no. 6S4, pp. 314–317, 2019.
- [18] P. Ravindra, "Training Deep Neural Networks", Towards Data Science, Sept. 2018.
- [19] R. Vaidya, D. Trivedi, S. Satra, M. Pimpale, "Handwritten Character Recognition Using Deep-Learning". Second International Conference on Inventive Communication and Computational Technologies (ICICCT), pp. 772-775, 2018.
- [20] R. Nijhawan, H. Sharma, H. Sahni, and A. Batra, "A deep learning hybrid CNN framework approach for vegetation cover mapping using deep features", 13th International Conference on Signal Image Technology & Internet-Based Systems (SITIS), pp. 192-196, 2017.
- [21] Y. Liang, J. Wang, S. Zhou, Y. Gong, and N. Zheng, "Incorporating image priors with deep convolutional neural networks for image super resolution", Neuro computing, Vol. 194, pp. 340- 347, 2016.
- [22] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning", Nature, Vol. 521, pp. 436-444, 2015.
- [23] Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Advances in neural information processing systems, pp. 1097–1105, 2012.
- [24] D. C. Ciresan, U. Meier, J. Masci, L. Maria Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification", Proceedings in 22nd International JointConference on Artificial Intelligence, Vol. 22, pp. 1237- 1242, 2011.
- [25] M. Celik and B. Yanikoglu, "Handwritten mathematical formula recognition using a statistical approach," 2011 IEEE 19th Signal Processing and Communications Applications Conference (SIU), 2011.
- [26] N. E. Matsakis, Recognition of handwritten mathematical expressions. 1999.
- [27] M. Rocklin and A. R. Terrel, "Symbolic Statistics with SymPy," Computing in Science & Engineering, vol. 14, no. 3, pp. 88–93, 2012