International Conference on Recent Trends in Data Science and its Applications
DOI: rp-9788770040723.116

Handwritten Mathematical Equation Solver

ShubhankSargam
Department of Data Science and Business Systems
SRM Institute of Science and Technology
Kattankulathur, Chennai.
shubhank.official@gmail.com

J. Shobana
Assistant Professor, Department of Data Science and Business Systems
SRM Institute of Science and Technology
Kattankulathur, Chennai.
shobanaj1@srmist.edu.in

***Abstract*—Automatic Recognition and solution of handwritten mathematical text is quite an arduous task which can have multitude of use cases in the modern world. In this paper we have devised a model which can detect the characters, solve the handwritten equation and provide the result accordingly to the end user. This is of enormous use especially in the academic field where students as well as faculties can use the technique to solve the mathematical text having numerous equations with near to human accuracy with way less time consumption. The evaluation process employed in the schools and universities can also use the methodology to quickly correct the students' papers. The model can be further deployed in the form of an end to end application installed in the mobile devices and thus save countless hours of manual time. We have used the Deep Columnar Convolutional Neural Network to get the output which produces the state of the art results in handwritten text related applications. Convolutional networks with multiple layers are capable of approaching human level accuracy in tasks like handwritten digit categorization and object recognition, according to recent advancements in the field of deep learning. This model achieves at most 96% accuracy and is thus a suitable choice for handwritten mathematical text evaluation.**

Keywords—Deep Columnar Convolutional Neural Network, handwritten equation, academics, machine learning, digit recognition.

I. INTRODUCTION

As computers gradually catch up with the nuances, inconsistencies, and imprecisions of the actual world to enable better communication, the complexity associated with human- computer contact is being dramatically decreased. (HCI). There has been a great deal of research done on natural language processing, facial recognition, handwriting recognition, and other irregularities that occur in the real world. We have made significant progress in teaching computers to comprehend our inaccurate but natural inputs, interpret them, and produce results that are at least somewhat helpful. Input methods other than the time-consuming punched cards, such as mouse clicks and touch, are now accepted by computers. The day when computers can be programmed without the aid of specialized hardware is rapidly approaching. We are now able to train others using both natural words and visuals. Now, computers have the ability to "listen" and "see" like humans do while still having greater processing capacity. Huge amounts of research have been done on how to best employ these newly discovered computing capabilities and streamline them for certain use cases. In this research, one use case that we investigate is the solution of handwritten mathematical problems.

The process of grading is crucial to education. Most of the time, it is challenging to hand grade each response sheet while still providing a fair, impartial, and genuine mark. The process for developing a computer vision model is described in this paper, ensuring that the grades are completely determined by the students' performance. The answer sheets will be automatically graded as a result. Convolution Neural Network (CNN) technology is crucial for image processing. Deep learning methods and improvements in processing power have enabled CNN to do complex visual tasks because they call for a lot of training data.

Our goal is to identify a handwritten equation from a picture and to ascertain the equation's solution for each successful identification. Convolutional networks with multiple layers are capable of approaching human level accuracy in tasks like handwritten digit categorization and object recognition, according to recent advancements in the field of deep learning. It has been noted that model ensembles, in which multiple models are trained on the same data and their prediction probabilities are averaged or decided upon, yield the most cutting-edge performance.

Deep Columnar Convolutional Neural Network is used in the proposed model, which provides performance that is close to state-of-the-art on a huge range of image classification tasks, including the MNIST dataset and the CIFAR-10 and CIFAR-100 datasets.

II. LITERATURE REVIEW

Mathematical expression recognition has been the topic of a sizable amount of research since the 1990s. The available mathematical expression recognition approaches are summarized and compared by Chan et al. The structure analysis and symbol recognition phases of mathematical expression recognition are the two main phases. Recognizing segmented symbols and segmenting symbols are both included in symbol segmentation. The former can be done by a number of methods, such as connected component discovery, recursive horizontal and vertical projection profile cutting, recursive X-Y cut, and progressive grouping algorithm, as described in Chan et al. Various methods, such as template matching, structural approaches, and statistical techniques, are used to achieve the latter. Online and offline are the two main categories of mathematical expression recognition. First, when the user enters symbols, and then after they have been entered, symbol recognition. The tree transformation method, which creates a tree with symbols inside the nodes to recognize the mathematical expressions,as successful according to Zanibbi et al. Garain et al. employ a strict feature extraction process to identify symbols by taking advantage of the neuromotor properties of handwriting.

Using computer software, LaViola et al. developed a technique for producing and analysing mathematical doodles. Through 2011 , most of the most recent research in this area conducted by Zanibbi et al. concentrated on online recognition of computer-sketched mathematical

DOI: rp-9788770040723.116

formulations. Since the advent of deep learning, more research has been done on mathematical symbol recognition. Despite this, Mouchere et al.'s extremely encouraging results in the recognition of online handwritten mathematical expressions also demonstrate that handwritten equations continue to be a challenging structural pattern recognition task.

III. METHODOLOGY

Workspace detection and analysis modules are the two modules that make up the workflow.

The workspace detection module identifies many work areas on a given sheet of paper.

In each given workspace, the analysis module is in charge of locating and identifying characters in lines, numerically evaluating them, and then visually indicating whether they are accurate by painting red or green boxes.

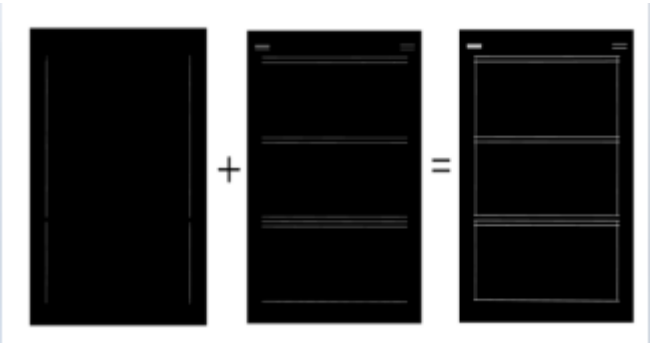
Detecting of the Workspace

The workspace detection module assumes that the given scanned spreadsheet contains legitimate rectangular boxes. The worksheet layout is shown in the picture below. The work-spaces are the 3 biggest boxes in this worksheet.

With openCV, the workspace detection module is completed. We'll locate the rectangular boxes first, then sort them according to where they are on the worksheet. As the worksheet contains a lot of rectangles, we must choose which of the other rectangles appropriate work spaces are. Let's examine each procedure in turn.

Step 1: Finding Rectangular Boxes

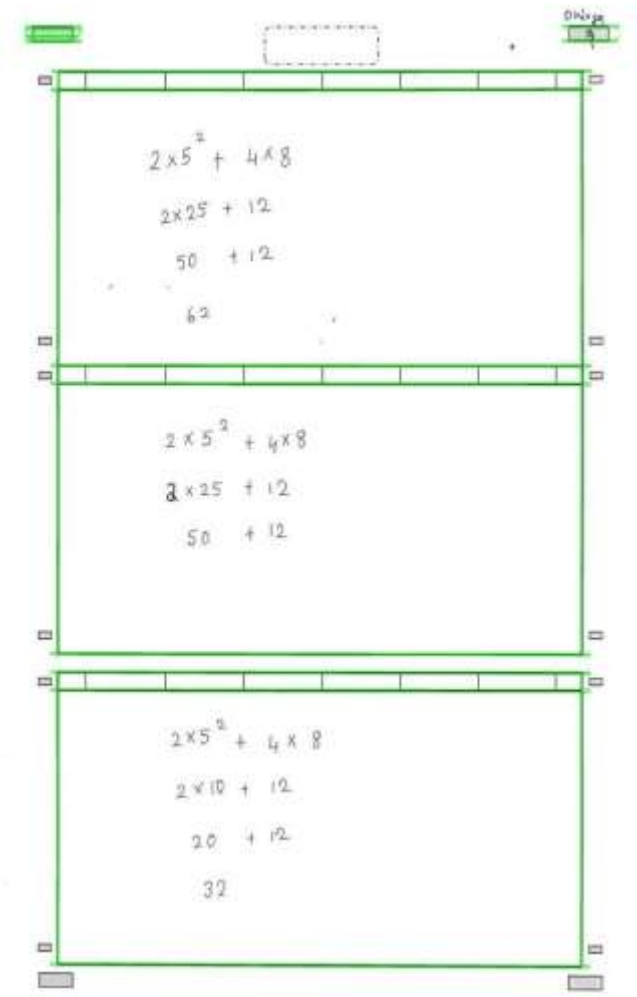
Therefore, the initial step is the identification of all of the horizontal and vertical lines on the spreadsheet, ignoring any textual information such as the digits, symbols, and other lines.



Adding vertical and horizontal line

The contour is the line that connects all of the identically intense points along an image's edge. OpenCV'sfindContour() method assists in extracting contours from the image. A numpy array stores the (x,y) coordinates for each contour. That will enable us to identify every object in the final image. (Rectangles are the only objects which are present in the final image).

Rectangles in the final image have the same coordinates as se in the original image because it is simply the binaryrepresentationof the original image.

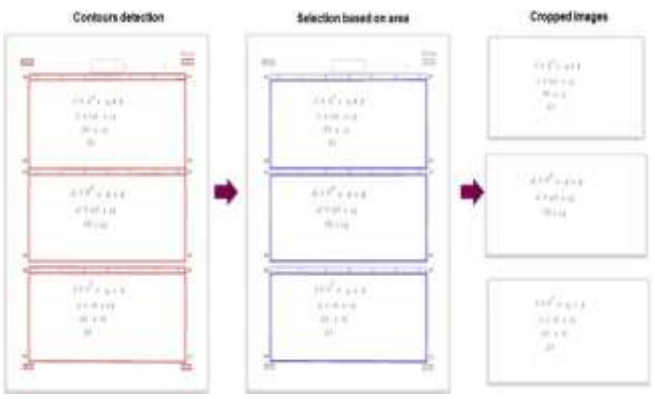


Step 2: Contour's sorting

We can now arrange the rectangles in ascending order by their coordinates after we have located each one. The code below accomplishes it for us.

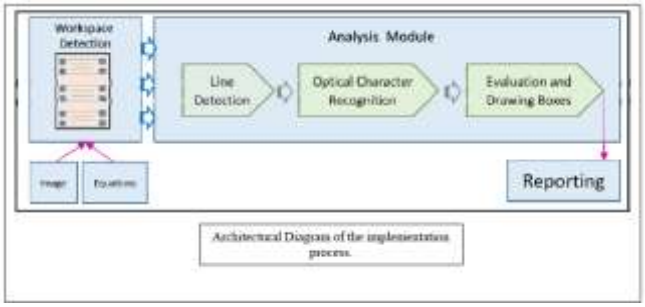
The sort contours method returns sorted bounding boxes and contours in the top-left and bottom-right locations that we gave.

Step 3: Selection on the basis of area



Theareaselection

DOI: rp-9788770040723.116



We only need the three largest rectangles out of the numerous that are. We wonder which of the three rectangles is the largest. Calculating the rectangles' areas and selecting the top three with the largest areas is one possibility.

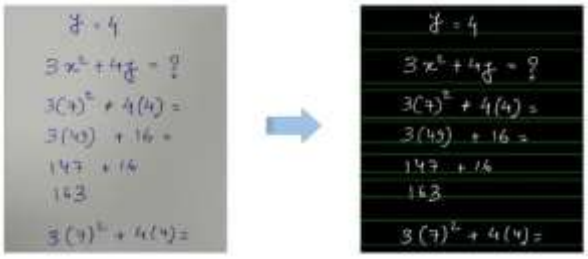
Workspaces are then removed from the worksheet using these chosen rectangles and sent to the next module for further analysis.

Module for Analysis

The analysis module will carry out the aforementioned procedures: it will first recognise the lines, forecast the characters in each line, create an equation using the forecasted characters, and finally assess the characters by placing boxes next to them.

Detection of Line

Everybody has a different method for solving equations; some people can do it step-by-step, while others can do it in a single line, while still others may write out their steps over several pages. Additionally, some people may write exponents that are far from the equation in order to trick the module into treating those exponents as a separate line.

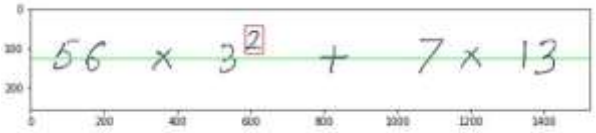


Our system for recognising lines makes the assumption that there is a definite space between them and that lines and exponential symbols occasionally cross paths. Before calculating the forward derivative, the selected work-spaces are transformed to binary pictures and then compressed into a single array. The derivative will alter each time a line is present.

Character Segmentation and Exponential detection

In order to sort the characters using the function sort_contours described above, where method is now set to left-to-right, we must submit the extracted line images to the text_segment function after detecting all the lines. This function will use openCV's locate contours to segment the characters.

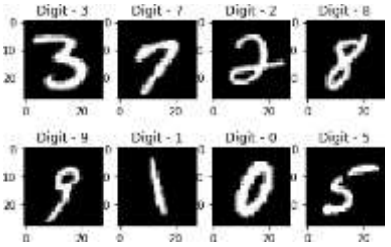
Although it is simple for us to determine if a given number is an exponent or not, the model finds it more difficult. We can draw a baseline in the image's center, and any character that is above the baseline is regarded as an exponent, presuming that the exponents are at least halfway up the line.



Exponential detection

Optical Character Recognition

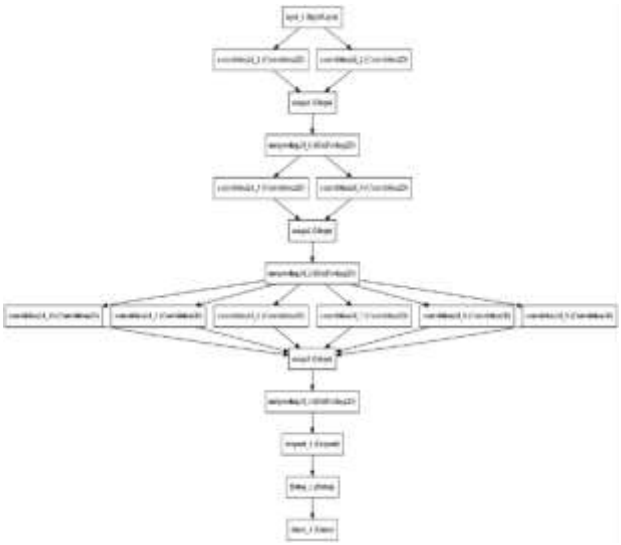
We can use the Handwritten Mathematical Symbols dataset from Kaggle (45*45 pixels) for symbols and the MNIST dataset for digits (28*28 pixels) to train the model.



MNIST IMAGES

How was MNIST truly made?

1. A collection of 500 distinct writers' 128 * 128 handwritten pixels.
2. To soften the edges, we apply Gaussian filter to the image.
3. After that, aspect ratio's maintained while the digit is positioned and centered within a square image.
4. Using bi-cubic interpolation, the image is then down sampled to a resolution of 28 by 28 pixels.



Deep Columnar Convolutional Architecture (DCCNN)

DOI: rp-9788770040723.116

Evaluation and drawing boxes

The final and most vital step is evaluation. Python's evaluation method can be used to answer any equation.

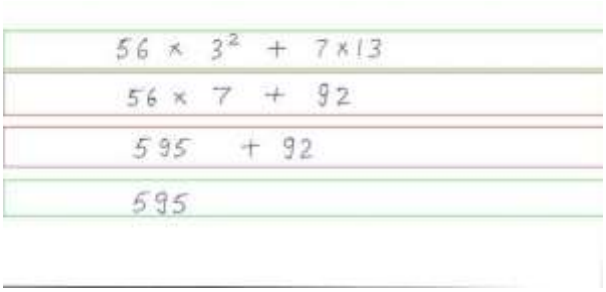
The eval method analyses the expression that is supplied to it before executing the program's python expression.

Parts of the mathematical material that have been evaluated

1. Complete the maths problem and save the solution as part of the evaluation procedure.
2. Compute each handwritten line's value and compare it to the saved solution.

If the line is accurate, a green box is drawn bounding the text; if the line is incorrect, draw a red box.

The same preprocessing is applied to symbol images as it is to MNIST digits before to training. The lengths, thicknesses, and line widths of the two data sets we choose make it difficult for the deep learning model to detect patterns. Preprocessing is therefore necessary. Preprocessing allows for decreased digit-to-symbol discrepancies. A single deep and broad neural network architecture, the Deep Columnar Convolutional Neural Network (DCCNN), was trained using over 60,000 images of preprocessed symbols and numbers. On a variety of image classification problems, including the MNIST, CIFAR-10, and CIFAR-100 datasets, DCCNN gives performance that is nearly state-of-the-art and comparable to ensemble models. The maximum accuracy of this model was 96%.



Sampleworkspace

Think about the following example: In the equation A*x2 +B*y, where A is 56 and B is 7, you must find the answer if x is 3, and y is 13, and the solution is 595 (56*32 + 7*13).

If the box is green, the line is right; if it is red, the line is incorrect.

When the first and last lines are combined, we obtain 595, which is the right answer. Therefore, the first and last lines are correct.

The second line is incorrect. When we arrive at 584, which is not the same as 595, we have solved 32 (9, although it is written as 7).

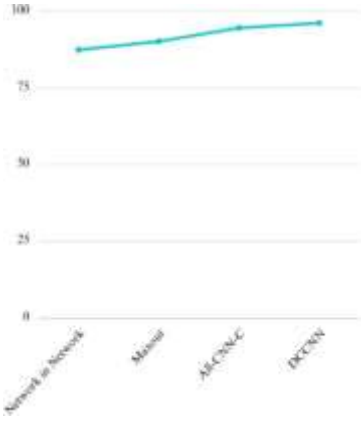
Similar errors occur in the third line (595 + 92), which when solved yields 684, which is once more not the same as 595.

IV. ResultsandDiscussions

METHOD	ACCURACY %
Network in Network	87.54
Maxout	90.28
All-CNN-C	94.65
DCCNN	96.21

Comparison of accuracy percentage ofdifferenttypesofmodelsontheMNISTd ataset

On careful analysis of all the different competing models trained on the MNIST dataset, it is found that Deep Columnar Convolutional Neural Network outperforms the other competing algorithms, i.e. Network in Network, Maxout, All- CNN-C and DCCNN when evaluated on the MNIST dataset with a much more human like accuracy of 96.21%. The table above demonstrates this comparison pretty accurately. The adjoining line graph plots this variation to further illustrate the superiority of DCCNN while trained and tested on the MNIST dataset. This provides a solid background to choose this algorithm for the functioning of our handwritten equation solver to produce the best results.



Accuracyofdifferent models plotted onthegraph.

V. Conclusion And Future Works

In end, the work-spaces detection module receives the scanned spreadsheet. The line extraction module gets the recognized work-spaces and extracts all the lines in order to return all the rectangular work-spaces in the provided worksheet. While the deep learning model DCCNN predicts the number or symbol, the character segmentation module separates the characters after receiving the extracted lines. The evaluation module will then draw a red/green bounding box after evaluating the line.

The Deep Columnar Convolutional Neural Network's success can be further exploited by using it to solve harder mathematical problems like differential integral equations, the identification of chemical equations, the recognition of cursive handwriting with uninterrupted characters, and the detection of plagiarism

REFERENCES

[1] F. Alvaro, J.A.S´anchez, and J.M. Bened´I, "Recognition of on-line handwritten mathematical expressions using 2d stochastic context-free grammars and hidden markov models," Pattern Recognition Letters, vol. 35, pp.58–67, 2014.

[2] Y. Bengio, P.Frasconi, P. and Simard, "The problem of learning long-term dependencies in recurrent networks," In: IEEE international conference on neural networks, IEEE, pp. 1183–1188, 1993.

[3] K.F.Chan,and D.Y.Yeung,"An efficient syntactic approach to structural analysis of on-line handwritten mathematical expressions," Pattern recognition, vol. 33, no. 3, pp. 375–384, 2000.

[4] K.F. Chan, and D.Y. Yeung, D.Y. "Mathematical expression recognition: a survey," International Journal on Document Analysis and Recognition, Vol. 3, no.1, pp. 3–15, 2000.

[5] K.F. Chan, D.Y.and Yeung, "Error detection, error correction and performance evaluation in on-line mathematical expression recognition," Pattern Recognition, vol. 34, no. 8, pp. 1671–1684, 2001.

[6] Pazhani. A, A. J., Gunasekaran, P., Shanmuganathan, V., Lim, S., Madasamy, K., Manoharan, R., &Verma, A. (2022).Peer–Peer Communication Using Novel Slice Handover Algorithm for 5G Wireless Networks.Journal of Sensor and Actuator Networks, 11(4), 82

[7] J. MDevlin, M.W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding,"arXiv preprint arXiv:1810.04805, 2018.

[8] Majumdar, Somshubra, and Ishaan Jain,"Deep columnar convolutional neural network," International Journal of Computer Applications, vol. 975, p. 8887, 2016.

[9] T. N. Truong, C.T. Nguyen, K.M.Phan, and M. Nakagawa, "Improvement of end-to-end offline handwritten mathematical expression recognition by weakly supervised learning," In: 2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR), IEEE,pp. 181–186, 2020.

[10] J.W. Wu, F. Yin, Y. Zhang, X.Y. Zhang, and C.L. Liu, "Graph-to-graph: Towards accurate and interpretable online handwritten mathematical expression recognition," AAAI, 2021.

[11] Z. Yan, X. Zhang, L. Gao, K. Yuan, Z. and Tang, "Convmath: A convolutional sequence network for mathematical expression recognition," In: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, pp. 4566–4572, 2021.

[12] Z. Yan, T. Ma, L. Gao, Z. Tang, and C. Chen, "Persistence homology for link prediction: An interactive view,"arXiv preprint arXiv:2102.10255, 2021.

[13] K. Yuan, D. He, Z. Jiang, L. Gao, Z. Tang, and C.L. Giles, "Automatic generation of headlines for online math questions," In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 9490– 9497. 2020.

[14] R. Zanibbi, and D. Blostein, "Recognition and retrieval of mathematical expressions," International Journal on Document Analysis and Recognition (IJDAR), vol. 15, no. 4, p. 331357, 2012.

[15] Rajesh, M., &Sitharthan, R. (2022). Image fusion and enhancement based on energy of the pixel using Deep Convolutional Neural Network. Multimedia Tools and Applications, 81(1), 873-885.

[16] J. Zhang, J. Du, Y. Yang, Y.Z. Song, S. Wei, and L. Dai, " A tree-structured decoder for image-to-markup generation.," In: ICML. p. In Press, 2020.