HANDWRITTEN EQUATION VERIFICATION

Indian Institute of Information Technology, Allahabad

Kushagr Garg IIT2018107 | Aditya IIT2018161 | Sushant Singh IIT2018171 | Vijit Jain IEC2018086

Abstract

In this era of modernity and development in science and technology each sector related to human activity has seen tremendous growth and advancements in automation. There has been various attempts in Ed-Tech as well to make it more intuitive to the learners meanwhile reducing the efforts of teachers so that they can use their mental faculties on the growth of learners in a better way. There have been several ideas which has revolutionized the Ed-Tech but still there is a long way ahead. But, there is one particular area which has received a lack of attention from technology and automation perspective, more on that in next paragraph.

We propose a auto-checking mechanism for mathematical equations. It can save precious time of educators from checking the students' submissions themselves. There hasn't been any substantial strides in this domain. The only major advancement in the domain of auto-checking was Optical Mark Recognition which is very old, restricted to only multiple choice questions and tightly bound in a restricted format. Additionally, multiple candidates face various problems in OMR sheets in exams. So to fix these issues we have built a Deep Learning model that can auto-check the handwritten the freehand documents containing mathematical equations. It will allow learners to freely express their know-how on the problem meanwhile reducing the effort of the educators considerably. More on that below the report.

I. INTRODUCTION

In spite of the ever-growing place of computers and other digital devices in our lives, pen and paper still remains the most convenient way for communicating or recording information or making small calculations. The linear input of a keyboard or a point and click device such as a mouse or a trackball is not very convenient for the preparation of complex documents with graphs, figures, tables and mathematical expressions. In the recent decade, deep learning with it's state-of the-art performance for many machine learning applications such as classification, detection, identification, and many more have made tremendous strides in all aspects related to human activities.

Among several recognition methods, Deep Learning with it's multiple layers of non-linear information processing is the need of the hour, it has played a major role in automation of various complex problems even for humans to be performed efficiently, fastly and correctly. These approaches automatically learn and solve problems without using any prior knowledge. The idea behind this project is to develop an algorithm using Deep Neural Architectures consisting of Convolutional Neural Networks along with various machine learning and deep learning algorithms for recognizing and solving mathematical equations written by free hand on a paper. In this project we mainly focus on the recognition of handwritten mathematical equations and after successful detection of the equations we apply Optical Character Recognition and then evaluate those equations.

II. PROBLEM STATEMENT

As discussed above, the motive behind this project is to develop an computer vision algorithm which uses deep learning to provide state-of-the-art performance for recognizing text and automating steps of evaluating a mathematical equation written by freehand on a paper, validating the steps and final answer of the recognized handwritten lines by maintaining the context. We have discussed the approach of our method at length in the methodology section.

III. MOTIVATION

Due to the swift improvement in computer technology and internet technology, most of the documents, books and literatures in the area of computer science as well as others are increasingly being digitized. Mathematics is the back bone of almost all areas of science, such as physics, engineering, medicine, economics, etc. In particular, mathematical expressions are most conveniently entered by handwriting. Computer understanding and recognizing of handwritten text (handwritten/freehand mathematical formulas and equations)is an ongoing research area. Mathematical expressions have a greater number of symbols to distinguish in comparison to handwritten text, and more importantly, the meaning of symbols in mathematical expression differs according to the spatial relations between them. Mathematical expression verification would greatly simplify the task of writing scientific articles that contain mathematical formulas. There are large databases of scientific and technical papers in offline form in archives. Without a mathematical expression checker tool, verifying these documents will be costly. Thus, a system that is capable of recognizing mathematical expressions is of great use for verifying tasks.

IV. Description of Past Studies

We have studies the past studies and researches in this domain and have reached to this conclusion.

- Work on mathematical expression recognition has been conducted especially in the last 10-15 years, with the advances in tablet technologies.

- We have studied various publications and found that most of them aimed at only interpreting expressions that were valid mathematical expressions according to some grammar.

- These grammars enforced a number of reasonable constraints, requiring that parentheses match, binary operators have valid operands, and integrals have differentials etc.

- Almost none of the past study focused on end to end evaluation of handwritten mathematical equation in a step based manner.

V. METHODOLOGY

The overall idea behind this project is to develop a deep learning algorithm for recognizing a mathematical equation written by freehand on paper. Our model consist of 4 steps:

- Workspace detection.

- Detecting and localizing each single line.

- Performing an appropriate Optical Character Recognition technique on each detected line.

- Evaluating all the detected lines.

The approach can be subdivided into 2 parts: Workspace Detection(for detecting multiple workspaces in a single paper) and Analysis Module(for detecting characters in each workspace and analyzing each of them). We have used a pre-trained DCANN on two open source datasets, Kaggle and MNIST for mathematical symbols for Optical Character Recognition.

I. Workspace detection

The following assumptions have been made for our model to detect separate workspaces on a worksheet.

- There are valid rectangular boxes in the given scanned worksheet.
- The expressions to be evaluated are contained within the boxes.

The contours have been sorted according to their coordinates and appropriate contours, based on area have been selected

II. Line detection

Every identified workspace will contain some lines of mathematical equations. To separately identify each line, we have assumed that,

- A sufficient gap between lines
- There is some intersection between exponential characters and line.

The binary images of the detected workspaces is compressed in a single array to take forward derivative thereby detecting the coordinates of each line. Line detection is done using a Document Analysis program (OCRopus). A document analysis program performs Binarization, Page-Layout-Analysis and Text-Line-Recognition. The default parameters and settings of OCRopus assume 300dpi binary black-on-white images, so we have resized accordingly. After text line recognition each line in the workspace image is then extracted and saved as separate images.

III. Character recognition and evaluation

Any character whose bottom right corner point lies above the centre of the image is considered as exponent. A Deep columnar convolutional neural network to precisely identify all the digits and symbols. To overcome the problem of vanishing

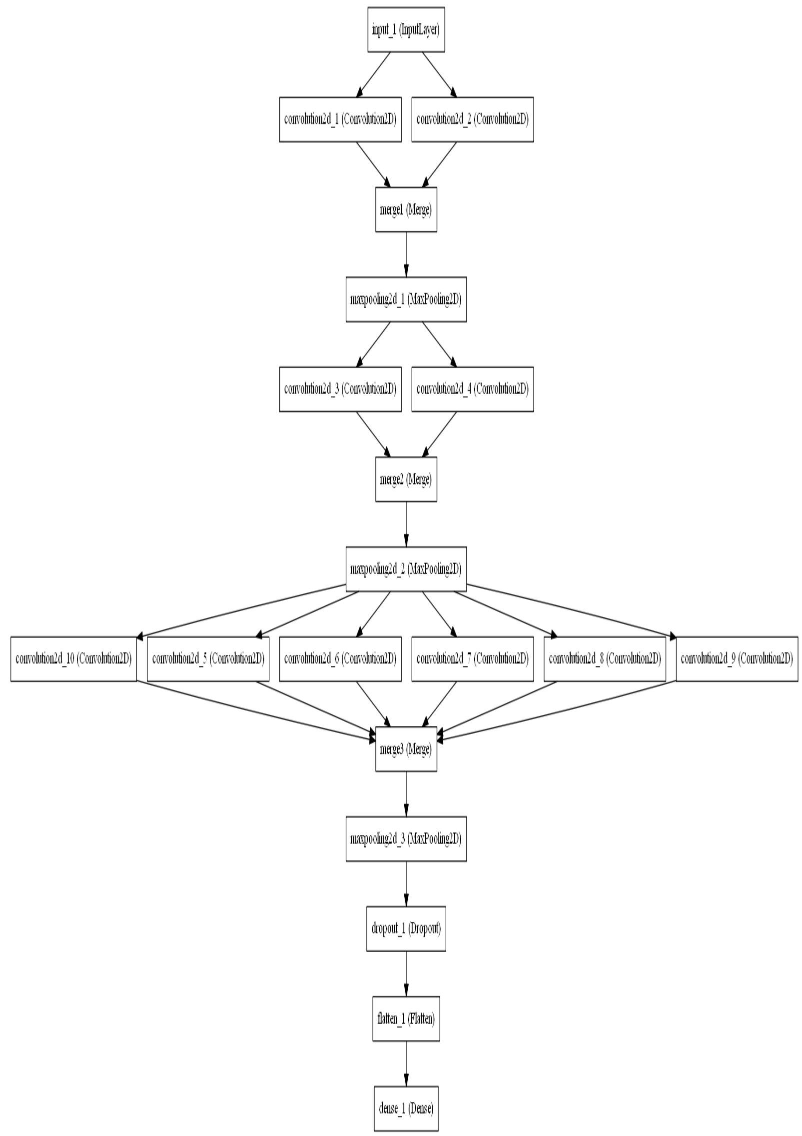

Figure 1. DCANN Architecture

gradient batch normalization is used. Once all the digits and symbols from the line are recognized, the expression is parsed and evaluated, the first line's evaluated value is assumed to be the ground truth and every following line is compared the to ground truth and marked.

VI. ARCHITECTURE

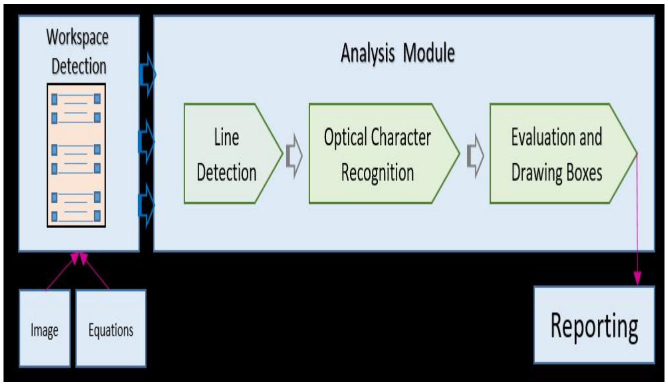

Figure 2. Block diagram

The primary/main functional blocks of the project can be outlined in the Architecture/block diagram shown below:

- As shown above, the overall solution can be divided into two parts, i.e 'Workspace Detection module and 'Analysis Module'.
- Workspace detection module is responsible for detecting multiple workspaces in a given sheet of paper using predefined markers as shown.
- Analysis module is responsible for detecting and localizing characters in any given single workspace, and mathematically analysing them and drawing red, green lines depending upon their correctness

I. DATASET DESCRIPTION

- We would primarily be focusing on two open source datasets from Kaggle and MNIST for mathematical symbols for Optical Character Recognition.
- The two datasets combined have over 60,000 images divided into many classes, like ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '+', '-', 'x', '(' , ')', '%', '/'].

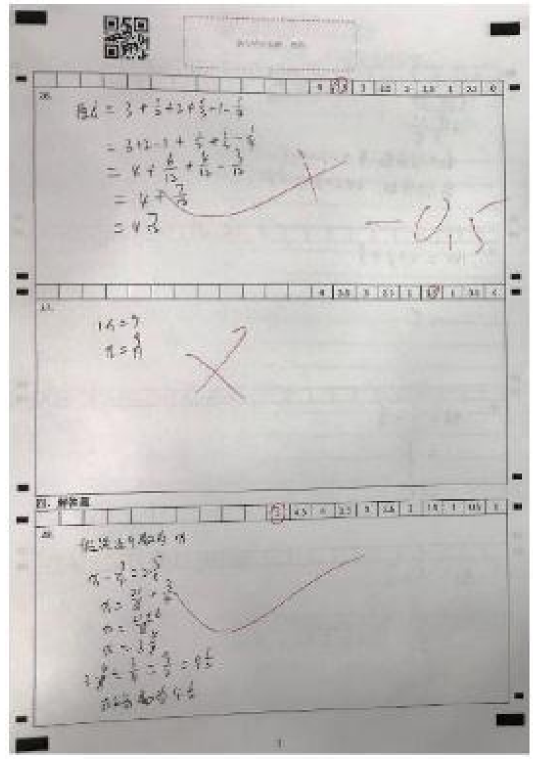

Figure 3. 1 Sheet

- The sample provided by MNIST Dataset consists of over 50,000 images of digits which will be normalized to a 20*20 pixel format with default aspect ratio.
- The kaggle dataset consists of handwritten symbols with a dataset of 4000 images each, from which few symbols will be selected and normalized.

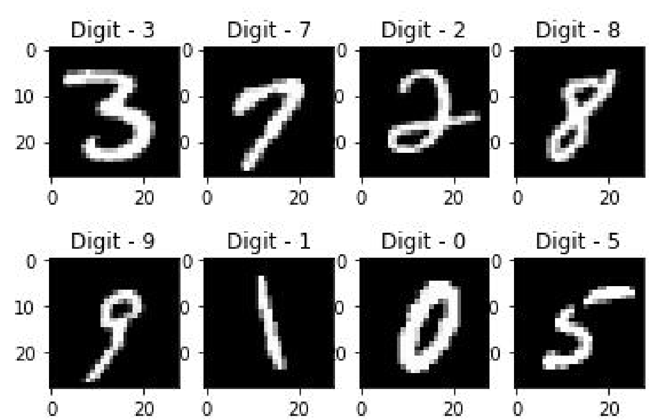

Figure 4. Sample images from MNIST

VII. TECH STACKS

We have used **Python** for creation/coding the entirety of the concept explained in methodology. We have used python 3.+ for coding purposes. We have used Linux Ubuntu for training purposes. We have also created a single check file which when executed using Google Colab will yield the results of single test. Apart from that we have also used various modules (of Python) and IDEs. The modules and IDEs used are most popular in field of deep learning research. The Python modules used are:

- h5py (2.10.0)
- Keras (2.3.1)
- Keras-Applications (1.0.8)
- Keras-Preprocessing (1.1.0)
- matplotlib (3.1.1)
- numpy (1.17.3)
- opencv-python (4.1.1.26)
- pandas (0.24.0)
- Pillow (6.2.0)
- tensorboard (1.14.0)
- tensorflow (1.14.0)
- tensorflow-estimator (1.14.0)

Also, the IDEs used are:

- Sublime Text
- Google Colab
- Jupyter Notebook

VIII. Results

- The input received is a image containing single/multiple lines of handwritten mathematical equations.
- The corresponding output generated will be having coloured contours of each line equation.
- It is evaluating the equation on topmost line and mark it to be a correct, and answer generated for that line is being saved for later evaluation of the lines of equations below.
- Then it check the answer generated for each line and compare it with answer generated for first line. If the both matches, it marks the line as correct, otherwise incorrect.
- However, the output will also be printed on used console.

Step 1: Providing Input Image

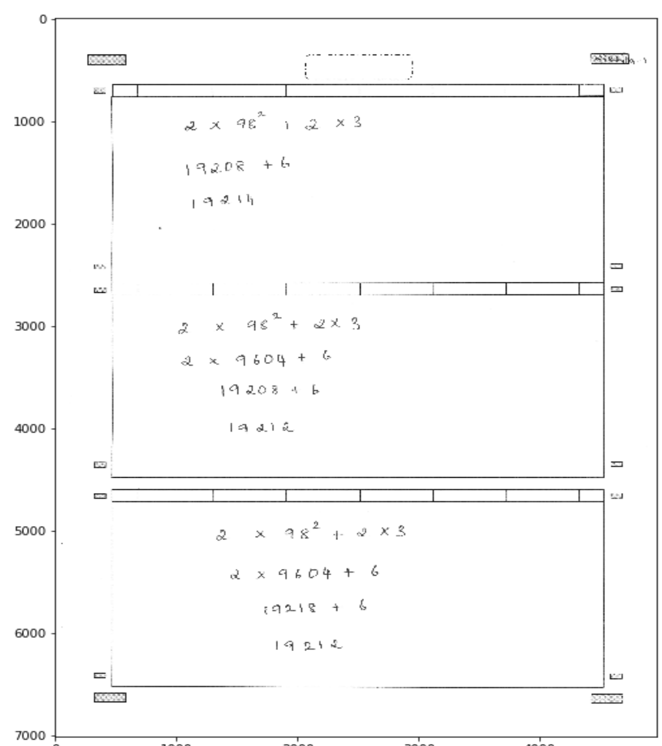

Figure 5. Input Image

Step 2: Segmenting each Workspace in the Input Image

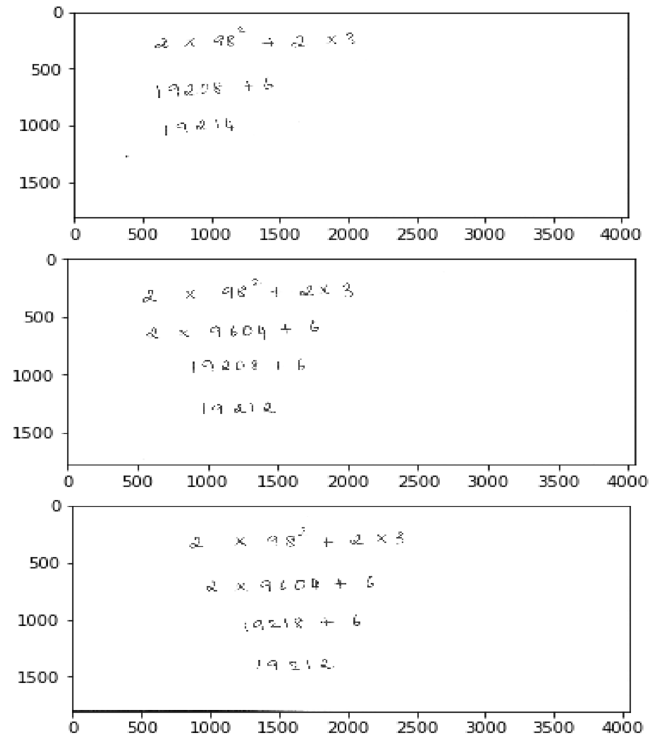

Figure 6. Output Image with workspace segmented

Step 3: Line Detection in Workspaces

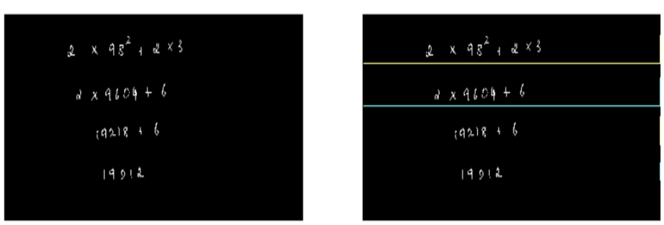

Figure 7. Line Detection

Step 4: Character Detection at Line Level

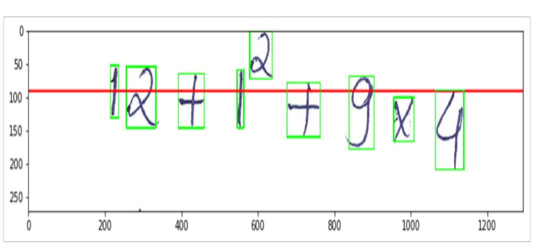

Figure 8. Character Detection

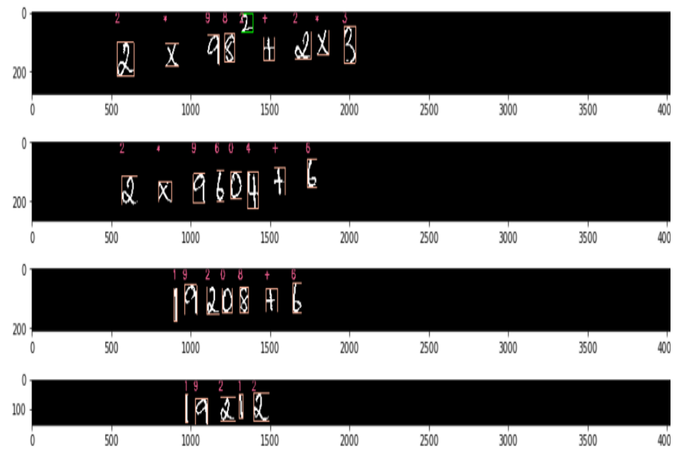

Figure 9. Evaluation of each line

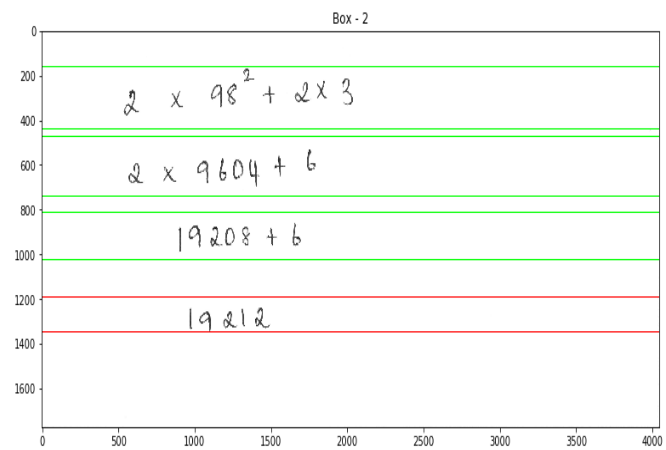

Figure 10. Final Output

Step 5: Character Recognition and Evaluating Each Line

I. Accuracy

We have trained our model using DCNN for character recognition. We have trained the model for 10 epochs, a learning rate of 1.0 and rho 0,95(Decay Factor). The finalized model achieved a accuracy of 96%.

IX. NOVELTY

- The presented work deals with evaluating handwritten numerical equations, which itself is quiet interesting.

- The study considers the existence of multiple questions solved in a single document and deals them accordingly, which almost no other study currently can do.

- The accuracy of the model is superior to currently existing models.

X. Conclusion

In this paper we mainly focused on verifying handwritten mathematical equations. The model is divided into following parts, namely : Workspace Detection, Line Segmentation, Character Recognition on line-by-line basis and Evaluation of each line and drawing contours. Depending on the Character Recognition method, the model can have trouble identifying decimal points. Due to the simple nature of the project, solving complex handwritten equations like ODEs/PDEs and algebraic equations will not be possible. In future we will try to improve the accuracy and try to make the system workable for multiple mathematical formula simultaneously. To simplify the math, the main task done in the feature extraction from the image and recognition with the help of the CNN model. If the CNN model classifies correctly all of the segmented images then this will be generated the correct list of equations. Which will be better for the simplification part. This is a successful representation of the state of the art. Any person can easily use this as the process is easy and does not require any advance knowledge of the technology. In future days the main focus will be to try to raise the precision level and build a segmentation system that can successfully segment two connected digits, and also increase the performance level of the dataset.

XI. Reference

1. Hossain, Md Naznin, Feroza Joarder, Y. A. Islam, Md Zahidul Uddin, Md. (2018). Recognition and Solution for Handwritten Equation Using Convolutional Neural Network. 250-255. 10.1109/ICIEV.2018.8640991.

2. S. B. K.S., V. Bhat and A. S. Krishnan, ”SolveIt: An Application for Automated Recognition and Processing of Handwritten Mathematical Equations,” 2018 4th International Conference for Convergence in Technology (I2CT), 2018, pp. 1-8, doi: 10.1109/I2CT42659.2018.9058273.

3. Handwritten Math Problem Solver using Convolutional Neural Network

4. Hossain, Md Bipul et al. “Recognition and Solution for Handwritten Equation Using Convolutional Neural Network.” 2018 Joint 7th International Conference on Informatics, Electronics Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision Pattern Recognition (icIVPR) (2018): 250-255.

5. https://www.kaggle.com/clarencezhao/handwritten-math-symbol-dataset

6. http://yann.lecun.com/exdb/mnist/