Handwritten Mathematical Expressions

Team: Runting Shao, Ningji Shen, Chang Xu, Joyce Zhang, Summer Ai

**Background**

As one of the most important subject in school, math plays a crucial role in helping students develop logical thinking and strengthening their understandings of many other subjects. However, people often find it difficult to calculate math by hand, especially when facing complicated expressions. Although there are many calculators available, formatting math into text can be boring and a minor mistake will lead to a big difference in the result.

In this project, we aim to alleviate people’s pain of doing mathematically heavy calculations by building a full stack web application system that takes users’ handwritten image (.png format) as input and return the calculated result as output. Specifically, we will first use machine learning and computer vision to build a model that can recognize numbers and operators in the expression. The expression will then be calculated in the background and the returned result will be displayed on the web page.

**Data**

**Source:** <https://www.isical.ac.in/~crohme/CROHME_data.html>

We choose to use training and test data from competition CROHME 2012 and 2013, which aims at bringing the researchers under a common platform so that they share the same dataset of handwritten mathematical expressions for their respective research and report performance of their systems on a common test data.

The datasets include the following items:

CROHME2012\_data : all data from the CROHME 2012 competition:

* testData : inkml test files without ground truth
* testDataGT : inkml test files with ground truth
* trainData : inkml train files with ground truth
* gram : xml grammars and symbol lists for parts I, II and III
* lists : lists of inkml files and latex expressions for parts I, II and III

CROHME2013\_data : all data from the CROHME 2013 competition:

* TrainINKML : all training inkml files sorted by origin
* TestINKML : inkml test files without ground-truth, used to run the participants' systems.
* TestINKMLGT : inkml test files with ground-truth, used to evaluate the participants systems with the evalinkml tool.
* Test\_LG/Test2012LG Test\_LG/Test2013LG: label graph version of the test files for 2012 and 2013 dataset, using inherited edges (so the graphs are DAGs).
* Test\_LG/Test2012LG\_TREE Test\_LG/Test2013LG\_TREE: label graph version of the test files for 2012 and 2013 data set, without inherited edges (so the graphs are trees).

**User Profile**

Our target users are STEM students of all grades and faculties in the education industry who are interested in:

1. Checking a previous math expression that they calculated by hand
2. Getting the result of a handwritten math expression from the calculator

Computer experience requirement:

1. The users should have basic knowledge of browsing a website
2. The users should be able to convert their image into .png format if the original one doesn’t meet the requirement
3. The users should have basic knowledge of uploading images from local computer
4. The users should be able to follow the instruction to click corresponding buttons ()

Domain knowledge requirement:

1. The users should have basic knowledge of algebra in case they want to check the results returned by the page
2. The users should have some experience in machine learning and computer vision if they are interested in understanding how we recognize the expression

**Use Cases**

Use Case #1:

John is a seventh-year-old student who is doing his math homework. He is confident about his answers but just want to double check one of them for the homework question. To interact with our web application, John need to first write down his expression and make it into a .png file. Then he will need to go to the website and click on the ‘Choose File’ button to upload his file. If the file is uploaded successfully, the application will start to recognize the expression and run the calculation in the background. Once the process is done, the web will notify John that calculation is finished and he will be able to see the returned result after clicking on ‘OK’. Lastly, he needs to check if the returned result is the same as his answer.

Use Case #2:

Jennifer is an elementary school teacher who is trying to calculate the average score of each student across ten quizzes. She has all the scores by hand and hopes to calculate them in an easy way. She will need to write down an expression for the first student and upload it to the web as John does. If the file is uploaded successfully, the application will start to recognize the expression and run the calculation in the background. Once the process is done, the web will notify Jennifer that calculation is finished and he will be able to see the returned result after clicking on ‘OK’. Because she needs to calculate the average score for each student, she needs to click on the ‘Start Over’ button to repeat this process iteratively until finishing calculating the score for the last student.