APS360 Project Proposal
A Web-Based Hand-Writing Equation Solver

Group 10:

Binyu Li

Jiaxing Li

Ruijie Xu

Haoran Zhang

2021.10.08

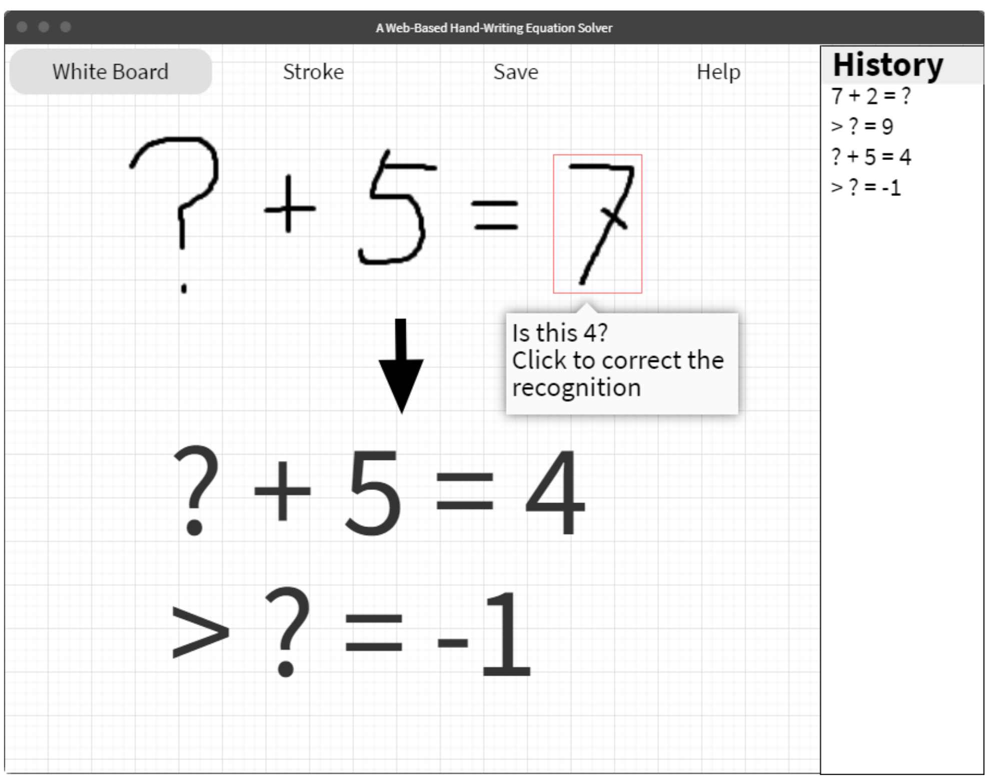
1.0 Introduction

An inevitable part of academic studies is solving mathematical equations. The effort of entering handwritten complex mathematical equations into an electronic device is frustrating to most teachers, students, and researchers [1]. It causes extension in learning period and also typing errors, leading to potentially miscalculated results.

Our proposed project aims to develop a web-based application that either receives user-captured pictures of handwritten mathematical equations, recognizes the equation through machine-learning models, and outputs the text form of the equation and its solutions.

The proposed application serves as a time-saver, reduces calculation errors, and can be accessed anywhere with internet connection.The major task of our project is to recognize mathematical symbols. Machine-learning has potent ability in image recognition especially with supervised deeping learning neural network models.The correct application of neural network model can yield high accuracy, precision, and recall rate. In addition, there are existing datasets, such as MNIST, that have an abundant amount of well-labeled handwritten equations training data available [2], which can be used in our proposed model to generalize the process of mathematical symbols recognition. Therefore, machine learning is a suitable approach to accommodate handwritten equations solver application.

2.0 Illustration / Figure



3.0 Background & Related Work

The following two articles are relevant to our project. The first one provides a training model for handwritten number recognition with a high accuracy rate of 99.45%, and the second offers a reference design for a mobile application, our project will be built upon the reference design with web-based implementation and higher accuracy rate.

A subproblem of handwritten mathematical equations recognition is number recognition. Several traditional methods have been proved useful to classify handwritten digits, such as K-nearest neighbour, artificial neural network, and support vector machine classifiers [3]. The article proposed a Convolutional Neural Network (CNN) that yields a higher precision rate of 99.45% compared to 99.3% in traditional methods [3]. The CNN also performs well in English letter recognition, which is another subproblem of mathematical equations recognition due to trigonometric functions (sin, cos) and variables (x, y).

For practical applications, there is a mobile application that focuses on providing a user-friendly interface that can reduce the learning curve for professional mathematical tools [4]. It also includes a few additional features such as graph plotting and simultaneous equation solving. The cross validation percentage accuracy on the kaggle math symbol dataset is 99.2% with reduced symbols.

4.0 Data Processing

The training and validation datasets for our project will be retrieved from MNIST, which includes mathematical symbols and handwritten alphanumerical datasets [5]. Our raw input data for testing will mainly be captured from online videos on mathematics lectures, such as Khan Academy [6]. We are going to take screenshots of the video frames and crop it such that only the equation portion remains. Furthermore, we also support user input, which allows the user to write dynamically on a whiteboard, and each character would be fed into the model as inputs, which are then put together as the equation input. Additionally, we will transform the input data through editing to make them all the same size and convert them to grayscale, to reduce the dimension of our data. Since most of the typical mathematical equations has a larger width than height, we will keep the input image size to be 320*180 pixels, and all input images will be in JPEG format. In order to feed the actual image to the model, we will flatten the image to a 1D array. (i.e. 1 x (320*180) arrays).

5.0 Architecture

The neural network model we will be using in this project is CNN. The application of neural networks in this project will mainly be mathematical formula recognition, with the user hand-writing the text. There are other good options such as Artificial Neural Networks (ANN) and Recurrent Neural Networks (RNN). ANN will be more suitable for tabular data [7], while the input to the system will be images. RNN will be good for sequential data [7], while mathematics formulas are not usually sequential, meaning the relevance between each character is not as strong as natural language. CNN has the advantage of being good at handling images. In addition, from our background CNN has already been proven efficient in optical character recognition (OCR) [3]. Overall, CNN is a good fit for our hand-written mathematical formulas recognition.

6.0 Baseline Model

A baseline model we will use to compare with our neural network is Support Vector Machines (SVM). SVM is a machine learning algorithm without utilizing neural network models. Both CNN and SVM "can equally tackle all types of classification problems; hence, the decision to use one over the other doesn't depend on the problem itself." [8]. SVM shows comparable accuracy against CNN on image classification [9]. Since they both can tackle the hand-written mathematical formulas task and have comparable accuracy on similar topics, we are expecting this SVM baseline model will help us learn the advantages and disadvantages of our neural network.

7.0 Ethical Considerations

The proposed model will be capable of recognizing and solving mathematical equations that are written by human beings. The training process naturally will involve the collection of handwritings from different individuals. Researchers at the University College London have developed a program that is capable of mimicking individuals' handwriting by studying their various characteristics (style, irregulations, spacing habit, etc) [10]. This poses potential risks to identity theft from the forgery of legal documents and signatures. To prevent it, the data collected must be completely unlinked to the original author or potentially generated and labeled by another program by utilizing established handwritten datasets like the MINIST library that are used in our labs [2].

8.0 Project Plan

Our team will meet every Thursday at 6 p.m. via Zoom. In-person meetings will be arranged for the purpose of code review and documentary specification. A Git repository is created for the team to write code simultaneously. In addition to clearly divided task assignment, a pull request will be conducted to ensure no overlapping in the code base. All team members are expected to attend a team meeting if any merging conflict occurs.

Group Member	Task	Internal Deadline
Binyu	Home page interface	Oct. 22
Haoran	Backend setup	Oct. 22
Ruijie	Frontend write board UI	Oct. 29
Jiaxing	Backend image upload API	Oct. 29
Binyu	Data processing	Nov. 5
Jiaxing	Build CNN Model	Nov. 12
Haoran	CNN model training	Nov. 14
Ruijie	Build SVM model	Nov. 12
Binyu	SVM model training	Nov. 14
Haoran	CNN Hyperparameter tuning	Nov. 17
Ruijie	Testing/comparing both model	Nov. 19
Jiaxing	More tuning and accuracy report	Nov. 22
All	Presentation Preparation	Nov. 25
All	Final report	Dec. 1

9.0 Risk Register

Group members opting out of the course poses a major threat to the overall success of the project. The solution to this potential risk is to create a welcoming and positive environment within the team and have tasks and responsibilities clearly divided among team members. Team members will help each other if someone were to fall behind and will prevent conflicts through positive communication. These actions will provide a safe environment within the team to help members develop trust, preventing the risk from the root. Having clearly divided tasks and responsibilities will help members manage their time better, lowering the chance of people dropping out. If a member were to drop out, this action can ensure that the team can pick up where the member left off. The team members all know each other outside of school and are all extremely driven individuals so the overall possibility of this risk is very low.

Not finishing the model's training is also a major threat that the team faces. Training a model with high accuracy is a very time and resource consuming process. Having a thorough background research is important, allowing the team to have a good estimate of time required so the team can plan accordingly. Starting early is also a great way to prevent the incompleteness of tasks. It allows us to have more time to work on things and provides us with buffer time to handle unexpected delays if they were to arise. With an assertive and responsible leader in the team who is always tracking the team's progress, the risk of our model not completing its training is low.

Lastly, not having enough data with diversity and high quality can also negatively impact the project. Good training data is critical for the accuracy of the model on unseen data. The team can resolve this risk by utilizing existing high quality and well labeled handwritten character datasets to form our own training data. With many industry-level datasets available, the risk of not having enough high quality data is low.

10.0 Link to github

https://github.com/leojiaxingli/APS360-Project

11.0 References

[1] K. Chang, “Why mathematicians hate that viral equation,” *The New York Times*, 06-Aug-2019. [Online]. Available: https://www.nytimes.com/2019/08/06/science/math-equation-pemdas.html. [Accessed: 08-Oct-2021].

[2] “The mnist database,” *MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges*. [Online]. Available: http://yann.lecun.com/exdb/mnist/. [Accessed: 05-Oct-2021].

[3] Y. A. Nanehkaran, D. Zhang, S. Salimi, J. Chen, Y. Tian, and N. Al-Nabhan, “Analysis and comparison of machine learning classifiers and deep neural networks techniques for recognition of Farsi handwritten digits,” *The Journal of Supercomputing*, 28-Jul-2020. [Online]. Available: https://link.springer.com/article/10.1007%2Fs11227-020-03388-7. [Accessed: 05-Oct-2021].

[4] 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.

[5] “Handwritten equations to latex,” *OpenSourc.ES*. [Online]. Available: https://opensourc.es/blog/he2latex/. [Accessed: 05-Oct-2021].

[6] “Khan Academy,” *YouTube*. [Online]. Available: https://www.youtube.com/user/khanacademy. [Accessed: 05-Oct-2021].

[7] “Ann vs CNN vs RNN: Types of neural networks,” *Analytics Vidhya*, 19-Oct-2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/. [Accessed: 08-Oct-2021].

[8] “SVM Vs Neural Network,” *Baeldung, Gabriele De Luca,* 25-Aug-2021. [Online]. Available: https://www.baeldung.com/cs/svm-vs-neural-network [Accessed: 08-Oct-2021].

[9] “A Comparison Between Support Vector Machine (SVM) and Convolutional Neural Network (CNN) Models For Hyperspectral Image Classification,” *Hayder Hasan et al,* 2019. [Online]. Available: https://iopscience.iop.org/article/10.1088/1755-1315/357/1/012035/pdf. [Accessed: 08-Oct-2021].

[10] “This program can mimic your handwriting with shocking accuracy-what could go wrong?,” *The Daily Dot*, 26-May-2021. [Online]. Available: https://www.dailydot.com/debug/handwriting-algorithm-privacy-security/. [Accessed: 05-Oct-2021].