Converting Handwritten Mathematical Expressions into LaTeX Representations

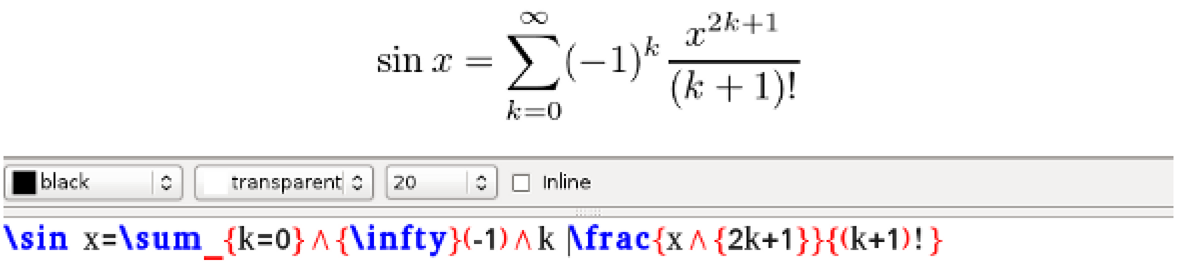
Number of Words: 1393

APS360
June 14, 2020

Team 41
Neha Basra
Ellen Chen
Ruqhia Frozaan
Cassey Shao

Introduction

LaTeX offers the ability to display mathematical expressions in a well-formatted manner. The goal of this project is to create a model that converts handwritten mathematical expressions into these LaTeX representations. As seen in Figure 1, coding the mathematical representation of the sin(x) Taylor series is tedious, error-prone, and time-consuming; this only becomes more difficult for longer expressions [1][2]. This project is useful because it addresses this; it reduces the time needed to code mathematical expressions manually. Machine learning is appropriate because this project requires training a model on a large dataset to recognize and label images of various characters. That way, it is able to convert the recognized symbols into LaTeX code for digital display. Success of the model will be measured by the accuracy of the converted test samples.



$$\sin x = \sum_{k=0}^{\infty}(-1)^k \frac{x^{2k+1}}{(k+1)!}$$

■ black | transparent | 20 | □ Inline

```
\sin x=\sum_{k=0}^{\infty}(-1)^k \frac{x^{2k+1}}{(k+1)!}
```

Figure 1: The LaTeX code and display of the Taylor series of sin(x) [3].

Illustration / Figure

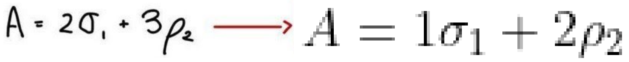


Figure 2: Example handwritten expression converted into LaTeX code

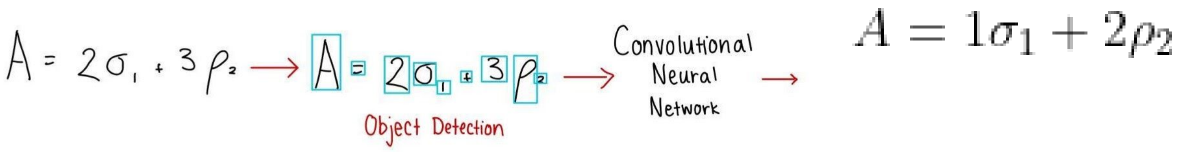


Figure 3: Breakdown of steps for model to convert mathematical expressions to LaTeX form.

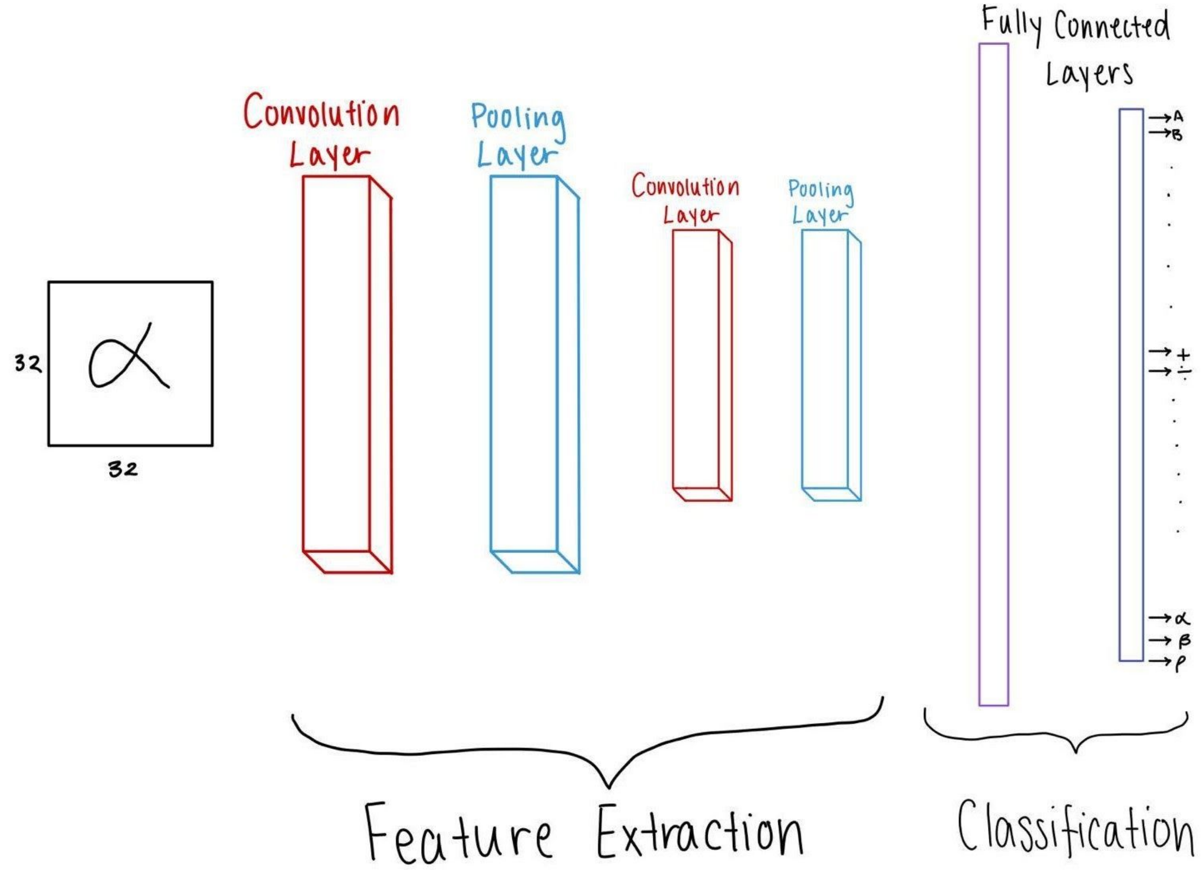


Figure 4: Generalized CNN model for classifying single digits, letters and symbols.

Background & Related Work

In the field of handwritten mathematical expressions recognition (HMER), two common approaches used are online and offline recognition. Online recognition refers to writing done on a digital device, whereas offline is on paper [1]. Online recognition tends to have a higher accuracy as more information can be obtained (e.g., order of the characters)[1].

Here are some state of the art in HMER:

HMER Research Paper

Using online recognition, the team achieved an accuracy of 94% on test/validation samples [3]. The dataset this team used is the Handwritten Math Symbols dataset [5]. Below is the methodology the team used:

Finding Contours

This team starts by finding the contours of the different numbers in the expression. The team used OpenCV library–specifically the contained cv2.findContours–to find individual character contours and place around it a bounding box.

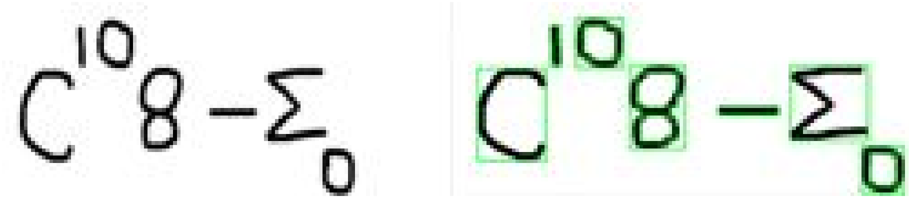


Figure 5: Original image vs Image with bounded boxes [3]

Processing Boxes

All the characters were cropped to a standard size of 45 by 45 pixels to feed into the model for classification.

Model

The cropped images were inputted into a CNN model individually to classify each character. The flow of their architecture is show in Figure 5:

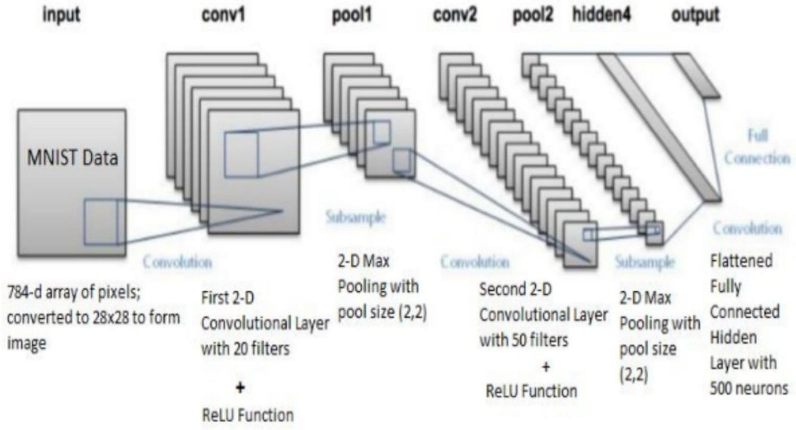

input
conv1
pool1
conv2
pool2
hidden4
output
MNIST Data
Convolution
Subsample
Convolution
Subsample
Full Connection
Convolution
784-d array of pixels; converted to 28x28 to form image
First 2-D Convolutional Layer with 20 filters + ReLU Function
2-D Max Pooling with pool size (2,2)
Second 2-D Convolutional Layer with 50 filters + ReLU Function
2-D Max Pooling with pool size (2,2)
Flattened Fully Connected Hidden Layer with 500 neurons

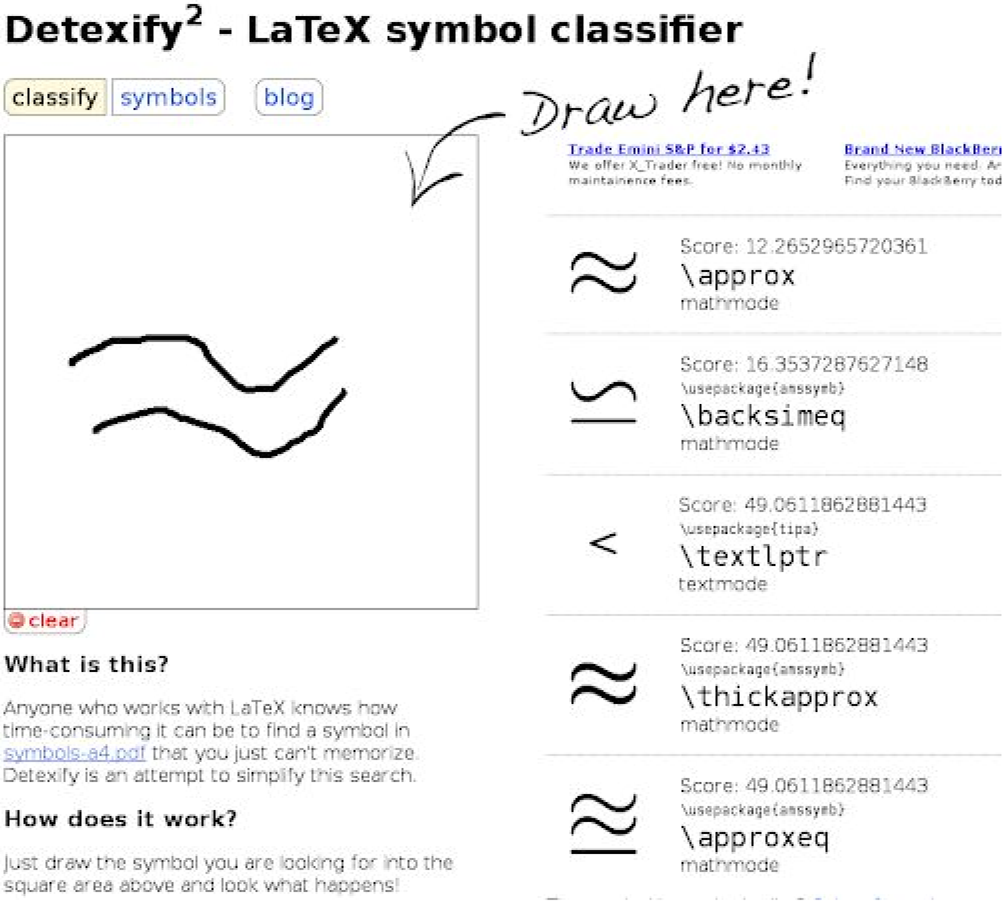
Figure 5: CNN architecture [3]

Latexification

In math expressions, characters can have many positions(e.g, subscript, superscript etc). The team used a recursive approach to determine the location of the next character, and saved the order of the characters in a spanning-tree structure. After all the characters were classified, they were converted to LaTeX.

Detexify

Detexify is an online software that allows users to draw characters which are then converted to LaTeX format. The problem with this software is that users can only classify single characters at a time.



Detexify² - LaTeX symbol classifier

Trade Emini S&P for $2.43
We offer X_Trader free! No monthly maintainence fees.

Brand New BlackBerr
Everything you need. Ar
Find your BlackBerry tod

Score: 12.2652965720361
\approx
mathmode

Score: 16.3537287627148
\usepackage{amssymb}
\backsimeq
mathmode

Score: 49.0611862881443
\usepackage{tipa}
\textlptr
textmode

Score: 49.0611862881443
\usepackage{amssymb}
\thickapprox
mathmode

Score: 49.0611862881443
\usepackage{amssymb}
\approxeq
mathmode

What is this?

Anyone who works with LaTeX knows how time-consuming it can be to find a symbol in symbols-a4.pdf that you just can't memorize. Detexify is an attempt to simplify this search.

How does it work?

Just draw the symbol you are looking for into the square area above and look what happens!

Figure 6: Screenshot of Detexify application [4].

Data Processing

The datasets that we will use to train the model are Handwritten math symbols [5] and HASYv2 [6] datasets. We are using two databases because the distribution of characters is not uniform (see Figure 8):

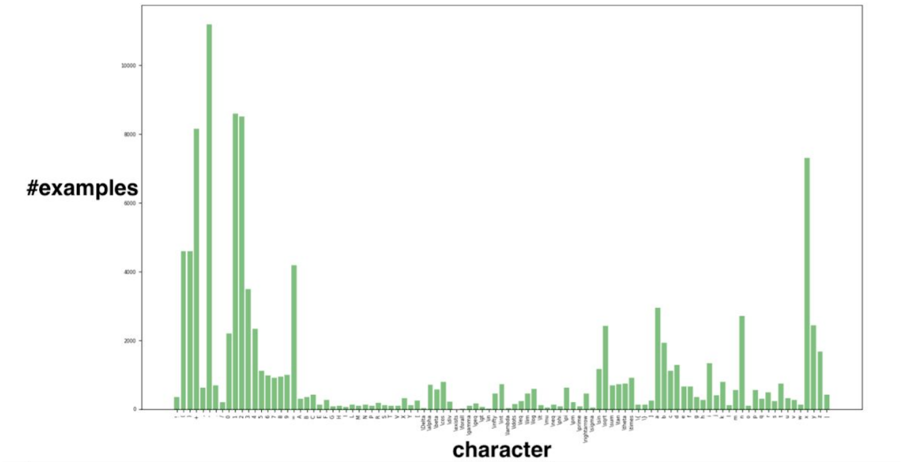


Figure 8: Distribution of characters in Handwritten math symbols dataset [6].

The images from the first dataset are 45 by 45 pixels, whereas the images from the second dataset are 32 by 32 pixels so we will resize the images from the first dataset to 32 by 32. Additionally, the second dataset contains characters which are the same but are placed under two labels, such as sigma and sum; we will remove duplicate labels.

Finally, the dataset of full equations which we will use to test our model is CROHME's dataset [7]. For this dataset, we will categorize the equations depending on their difficulty, in order to make our project more modular. We will start off with simpler equations (e.g., with no subscripts/superscripts), and then move onto more complex examples.

Architecture

In order to detect each character in a given image of an expression, we will utilize an existing object detection model such as YOLO. YOLO is a real-time object detection algorithm that is currently one of the most effective CNNs in the world with quick processing capabilities [8]. Once we detect the image of a single character, that image will be cropped to a determined dimension, saved, and classified.

For image classification for symbols, letters, and numbers, we will use a convolutional neural network (CNN). It will contain convolution and max pooling layers to make our model robust and accurate. It will also utilize the ReLU activation after every convolution layer. The architecture will feature fully connected layers that will use the ReLU activation.

Baseline Model

The main purpose of the baseline is to compare its performance with our project's neural network. Since this project revolves around image classification, a convolutional architecture should be used due to its effective feature extraction algorithms.

The baseline model for this project will be the VGG model. Its general architecture is simple, modular, yet proven to be accurate. This VGG model is split into two parts: feature extraction and classification (as illustrated in Figure 4). Feature extraction: involves convolutional layers with 3x3 filters followed by a pooling layer. Each convolutional layer uses a ReLU activation function. Classification: flatten the output from feature extraction and add one or two fully-connected layers. The output is a prediction, and it will go through the softmax activation function before being outputted [9].

Ethical Considerations

The model is limited by a training dataset of relatively neat handwriting examples. As such, if the handwritten characters are not fairly smooth or straight, the model may produce inaccurate results in LaTeX. This raises ethical issues in terms of accessibility, as a user with a writing disability may not see their desired results. In addition, the training data used are characters common in North American mathematical expressions. This means that the model is limited to recognizing only these symbols, thus, making the model inaccurate in labelling other characters.

Project Plan

Logistics:

- Meet Saturdays at 10 AM through audio call.
- Communication: group chat messaging and regular audio calls.
- Work method: Work will be done on Google Colab, which allows members to see others' edits and be on the same page. No one will be allowed to delete work that is not their own.

Tasks and Deadlines:

Week/Deadline	Task	Assigned Members
Week 7	Research Document ● Gather available relevant information (models, datasets, methodologies, limitations)	Cassey, Ellen

	Processing and Visualizing Data ● Load data onto Google Colab, pre-process, visualize, randomize ● Split data into training, validation, testing	Neha, Ruqhia
	Constructing Baseline Model ● Build baseline, train it, test it	Neha, Ellen
Week 8	Creating the Model ● Build neural network model	Ruqhia, Cassey
Week 9	Write Progress Report	All
	Train Model ● Also tune hyperparameters	Ellen, Ruqhia
Week 10	Evaluate the Model ● Evaluate model on *testing* set. Compare performance to baseline model	Neha, Cassey
Week 11	Improve and Finalize Model	All
Week 12	Videography (ongoing/throughout project) ● Create detailed video plan ● Edit using iMovie	Cassey, Ruqhia Neha, Ellen

	Final Deliverable ● Write final report	All

Risk Register

The following are 4 risks associated with this project, the likelihood of the risks, as well as mitigation strategies.

1. Project is too difficult to complete in the time frame - Likely.
   We will switch to a simpler version of the project, such as one requiring less time to train. We can simplify the project by finding an alternate final form for our expression, or by stating limitations on certain expressions the model can classify. We will also create an alternative plan to further modularize our project.

2. One of our team members drops the course - Not likely.
   We will have a meeting to discuss next steps, and redistribute the work to the remaining group members. We will also discuss with the TAs and Professor Colic about how we can still maintain our standards for our final product.

3. Our dataset does not align well with our model - Likely.
   We will continue to do research online to find datasets that fit our needs. Depending on the time, we can further clean the dataset.

4. Our team falls behind on the project and training takes longer than expected - Likely.
   We will reorganize the plan to either simplify our project, or create an arrangement to organize our workflow more efficiently.

Link to Google Colab Notebook

https://colab.research.google.com/drive/1VXAhNTkZuy8yztSCprxjEfpsYGcIcC47?usp=sharing

References

[1] A. Schechter, *Pdfs.semanticscholar.org*, 2017. [Online]. Available: https://pdfs.semanticscholar.org/fd25/9c0d70368a50b52755d4e33590566ef5d373.pdf?_ga=2.12415604.1536964691.1591985627-78053362.1591985627. [Accessed: 14- Jun- 2020].

[2] N. Phan, "WYSIWYG LaTeX editor for maths", *TeX - LaTeX Stack Exchange*, 2019. [Online]. Available: https://tex.stackexchange.com/questions/57068/wysiwyg-latex-editor-for-maths. [Accessed: 14- Jun- 2020].

[3] A. Gunda, *Aravindreddy.org*, 2020. [Online]. Available: https://aravindreddy.org/HMRE_Report.pdf. [Accessed: 14- Jun- 2020].

[4] “Detexify,” *Detexify LaTeX handwritten symbol recognition*. [Online]. Available: https://detexify.kirelabs.org/classify.html. [Accessed: 14-Jun-2020].

[5] X. Nano, *Handwritten math symbols dataset,* Kaggle: Kaggle, 2017. [Dataset]. Available: https://www.kaggle.com/xainano/handwrittenmathsymbols?select=data.rar. [Accessed: 14-Jun-2020].

[6] *HASY v2,* Kaggle: Kaggle, 2019. [Dataset]. Available: https://www.kaggle.com/guru001/hasyv2?fbclid=IwAR3001xLNRRzVPyb7xjYjWYDqzW_RTRO-GvwOpEQpGNqB_sw7sqwqKlEcR0. [Accessed: 14-Jun-2020].

[7] *Crohme Task 1 and Task 2*, Icdar 2019, 2019. [Online]. Available: https://www.cs.rit.edu/~crohme2019/downloads/zipped_CROHME2019_testData.zip. [Accessed: 14-Jun-2020]

[8] J. Hui, “Real-time Object Detection with YOLO, YOLOv2 and now YOLOv3,” *Medium*, 27-Aug-2019. [Online]. Available: https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088. [Accessed: 14-Jun-2020].

[9] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *Computer Vision and Pattern Recognition*, 2014.