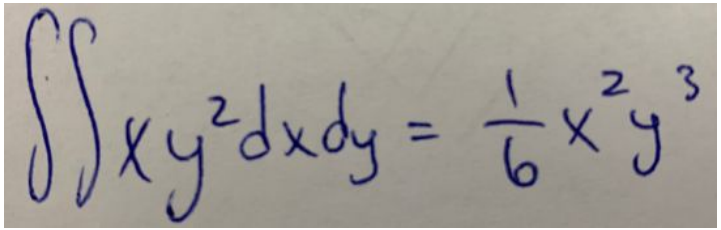
Team LaTeX Math

Members: Rachel Breshears, Casey Duncan, Levi Petty, Jason Pickett, Matthew Stanley

Problem Statement

- Some students prefer to hand-write assignments



- Some classes require assignments in LaTeX

```
\begin{math}
    \iint xy^2\,dx\,dy
    =\frac{1}{6}x^2y^3
\end{math}
```

$$\iint xy^2\,dx\,dy = \frac{1}{6}x^2y^3$$

- Solution: Take photo and use ML to put into LaTeX code

Solution

- Everyone has a different handwriting style, so we need ML to extract common features to best predict what is written. Our solution is to:
  1. Have the student take photo of math equation
  2. Upload photo to user interface (UI)
  3. Use Computer Vision techniques to separate individual characters
  4. Send characters through trained CNN model which outputs typed up equation in LaTeX code
  5. Copy and paste LaTeX code into LaTeX file

Assumptions, Constraints, and Implications

- We are only doing basic algebraic equations
  - Ex: X+Y=Z
- Time constraint - were not able to implement everything we wanted (RNN)
- Knowledge constraint - learned about solutions a little too late

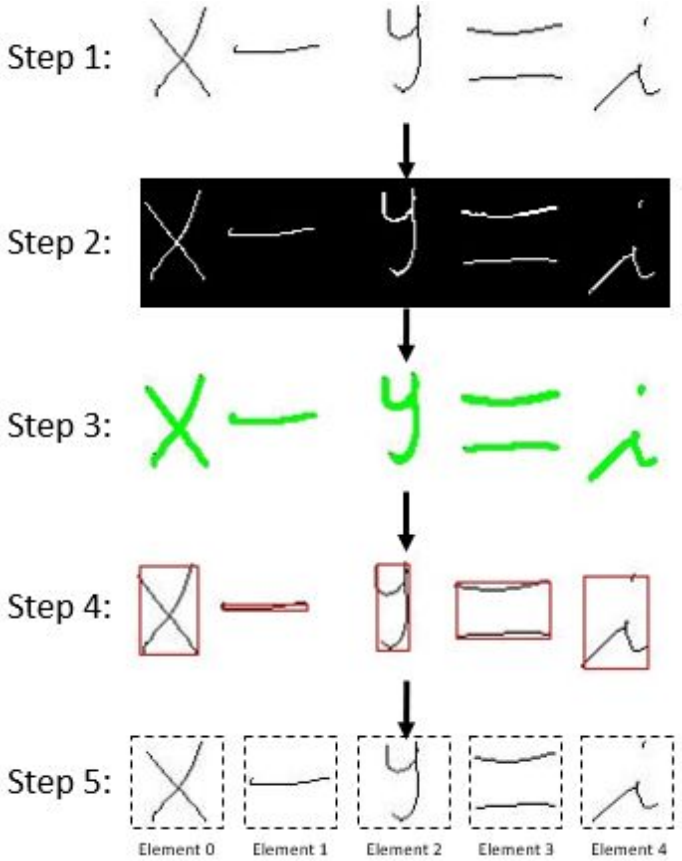
How Solution is Built

- Computer Vision
- Data Manipulation
- CNN Model
- LaTeX Output

Computer Vision

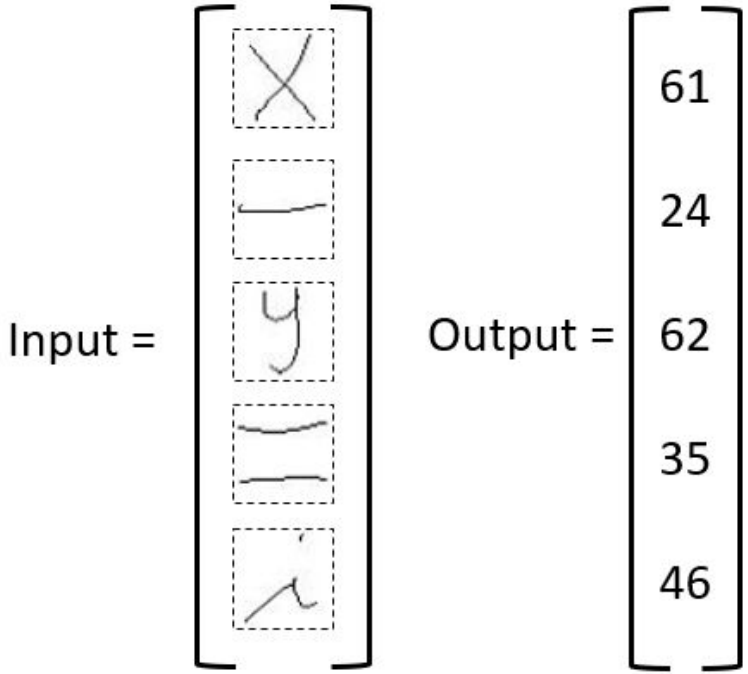
Used OpenCV in Python

1. Import Image and convert to Black & White
2. Bolden Characters using Thresholding Techniques
   a. Change pixel values to 0 (white) or 1 (black)
   b. Invert white & black for boldening
   c. Bolden characters through morphologically closing
3. Find Contours (drawn in green)
4. Find Bounding Box for each character
   a. Minimum x & y contour value is top left corner
   b. Maximum x & y contour value is bottom right corner
5. Save each character image as matrix in 3D array
   a. Each image is 2D matrix
   b. Array element 0 is “X”, element 1 is “-”, etc.



Data Manipulation

1. Delete duplicates in data
    a. +300,000 images → +80,000 images
2. Order data into a dictionary for labeling
    a. 66 labels total (0-65)
    b. Ex: “x” is label “61”, “=” is label “35”, etc
3. Split into train (80%) and test (20%) data
    a. Train → +64,000 Image Matrix & Labels
    b. Test → +16,000 Image Matrix & Labels
4. Pickle file for training and testing



CNN Model

- Architecture
  - 3 Convolutions of size 3 by 3 with 32, 64, then 128 layers
  - Each Convolution followed by a 2 by 2 max pooling
  - Flattened to dense of 500, 250, 120, then 66 layers
  - Used relu activation except for softmax on last layer
  - Drop before each dense layer of 0.15, then 0.2 for the rest
  - Using Sparse Categorical Cross-Entropy loss

CNN Model

- Best model after 16 epoch
- Validation accuracy of 89.5%

LaTeX Output

- Equation object created from outputs of CNN
- Stores character and state as array of integers
    - State differentiates between normal characters, superscripts, and subscripts
- Print function returns string of LaTeX code

Demo

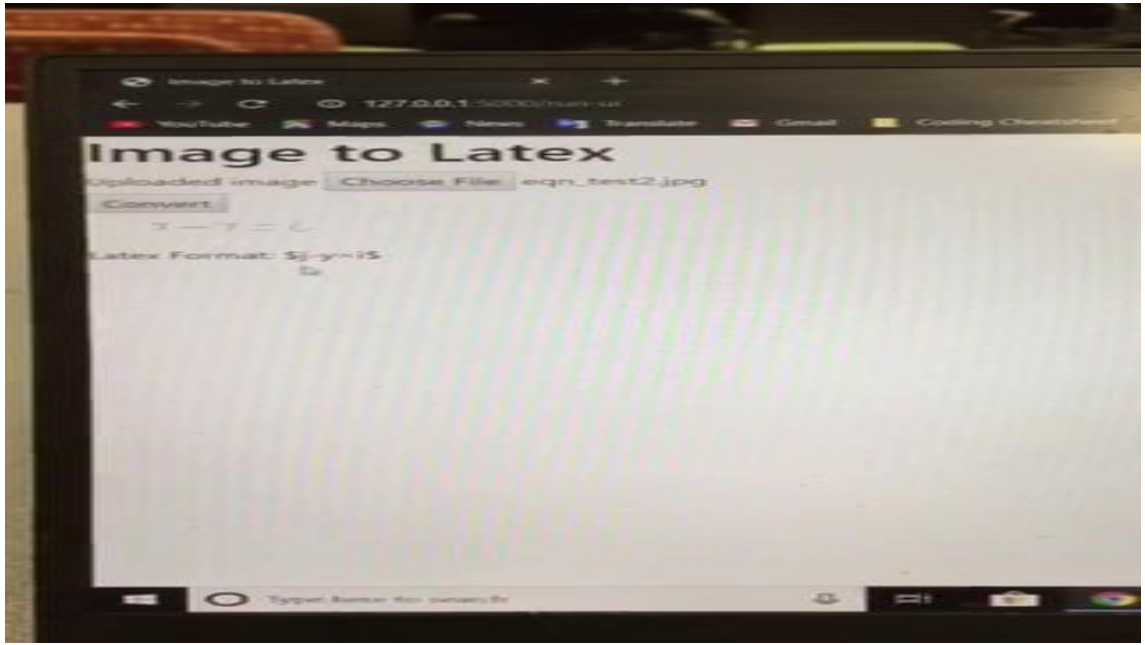


Image to Latex
Uploaded image:
Choose File
eqn_test2.jpg
Convert
Latex Format: $j-y=i$

Summary

- We want to convert handwritten equations to LaTeX code.
- Our solution is to run an image of an equation through a CNN to identify each individual character, convert it to LaTeX code in a python script, and output the line of code to the user.
- Some issues we ran into were duplicated data, steep learning curve, and time constraint.

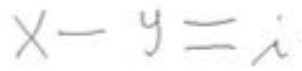
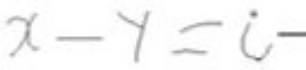
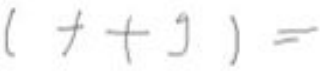
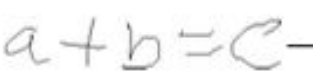
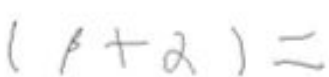
Next Steps

- General improvement of CNN accuracy
- Recognizing fractions
- Recognizing superscripts and subscripts
- Addition of RNN for grammar induction

Questions?

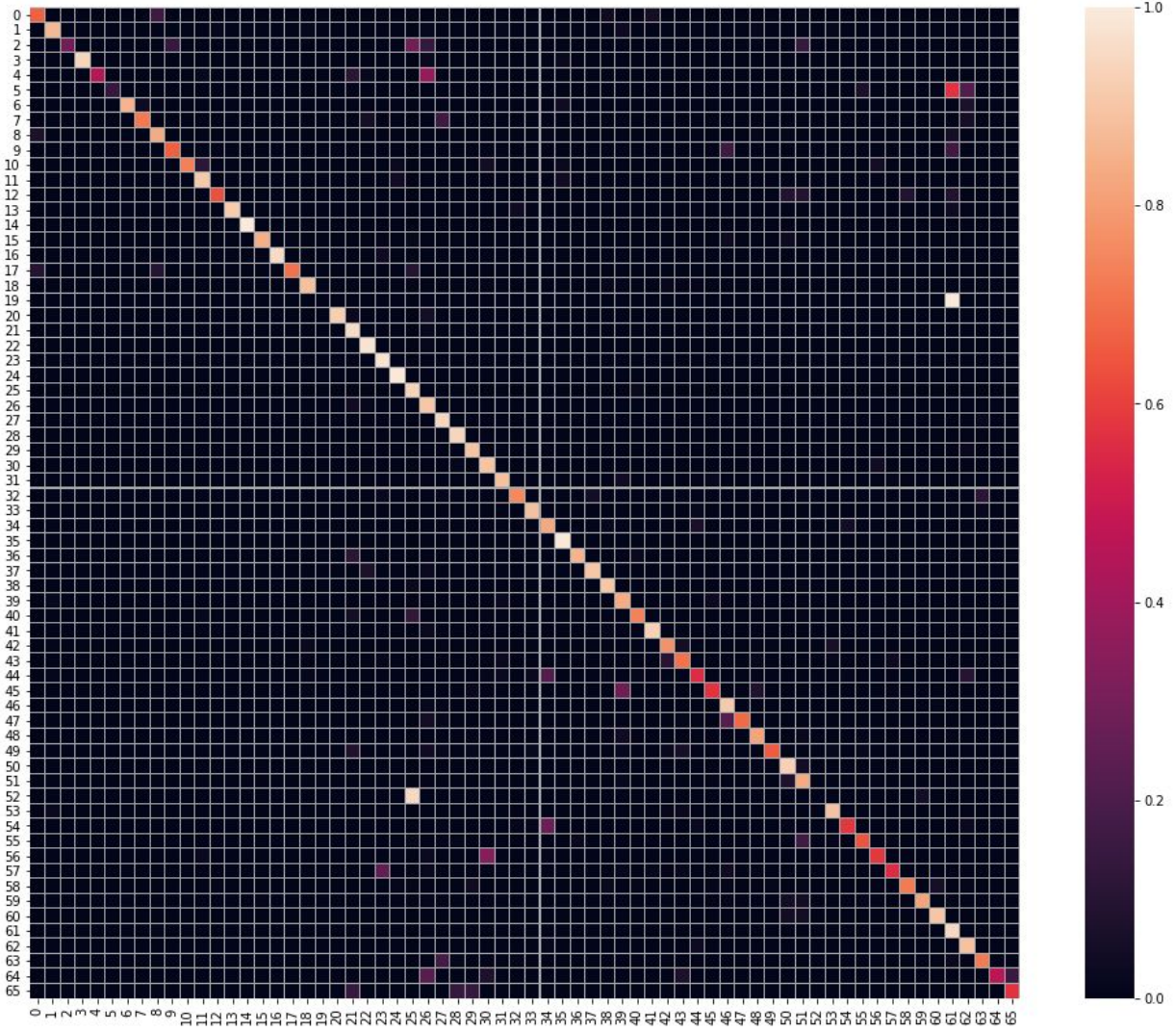
Performance & Analysis

- Success rate of 41%
  - 12 of 29 characters
- Possible Problems
  - Duplicates potentially still exist
  - Character size in training different from testing

Input Equation	Expected Output	Actual Output
	$x–y=i$	$1-1=j$
	$x–y=i$	$j–y=i$
	$(t+g)=j$	$=++11=i$
	$a+b=c$	$=-b==$
	$(β+2)=R$	$(1-\divi==$

Heatmap of Confusion Matrix

- 19 and 61 mixed up
  - Times symbol and x
- 25 and 52 mixed up
  - 0 and o



Timeline from Progress Report

1. Data Sets
    a. Collecting Initial Data Set (CD)
    b. Collecting Additional Data Set (if required) (ALL)
    c. Generating Testing Data (MS)
2. Character Recognition
    a. Research CV Libraries (RB, CD, MS)
    b. Implement Computer Vision (LP, MS)
    c. Extract Features using CNN (LP, MS)
3. Supervised Learning (Repeat for simple strings, exponents, and then fractions)
    a. Train Algorithm to Distinguish Characters (ALL)
    b. Validate Algorithm (ALL)
    c. Test Algorithm (ALL)
4. LaTeX Code Generation
    a. Research LaTeX Shortcuts (JP)
    b. Decide which Shortcuts we want to implement (ALL)
    c. Program LaTeX Shortcuts (JP)
5. Documentation
    a. Make Progress Report Presentation (ALL)
    b. Present Progress Report (RB, CD, MS)
    c. Make Project Final Report Presentation (ALL)
    d. Present Project Final Report (ALL)

Updated Final Timeline

1. Data Sets
    a. Collecting Initial Data Set (CD)
    b. Remove Duplicates (RB, MS)
    c. Label Data (RB, MS)
    d. Generating Testing Data (RB, MS)
2. Character Recognition
    a. Research CV Libraries (CD, MS)
    b. Implement Computer Vision (CD)
    c. Extract Features using CNN (MS)
3. Supervised Learning (Repeat for simple strings, exponents, and then fractions)
    a. Train Algorithm to Distinguish Characters (RB, CD, JP, MS)
    b. Validate Algorithm (RB, CD, JP, MS)
    c. Test Algorithm (RB, CD, JP, MS)
4. LaTeX Code Generation
    a. Research LaTeX Shortcuts (JP)
    b. Decide which Shortcuts we want to implement (ALL)
    c. Program LaTeX Shortcuts (JP)
5. Documentation
    a. Make Progress Report Presentation (RB, CD, JP, MS)
    b. Present Progress Report (RB, CD, MS)
    c. Make Project Final Report Presentation (RB, CD, JP, MS)
    d. Present Project Final Report (ALL)