Handwritten Expression Recognition

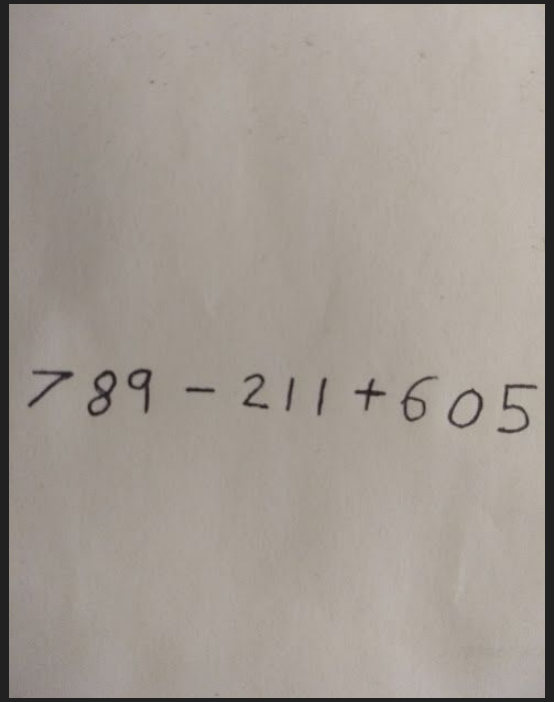
David DeFazio

Goal:

Find what a basic math expression evaluates to

Assumptions:

- White paper, fully connected black digits
- Only support + and - binary operators



Procedure

3 parts: segment digits/operators, digit recognizer, evaluation.

Segmentation:

- From left to right, find a bounding box (subimage) for each digit or operator.
- Resize each digit or operator to 28x28.

Digit Recognizer:

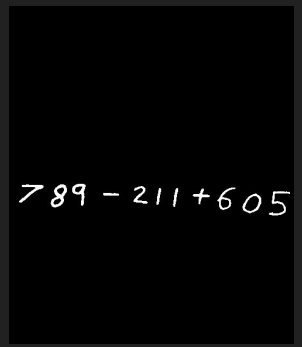
- Using the MNIST dataset (42,000 labeled 28x28 digits), find a model that can accurately classify digits (0-9).
- Extend MNIST to support operators.

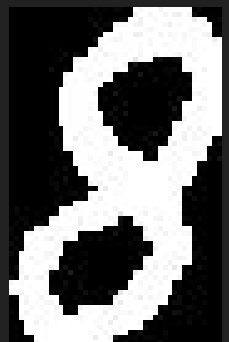
Evaluation:

- Feed subimages one by one into model, calculate result of the expression

Segmentation

- First converted to binary image with black background and white numbers (MNIST has this format).
- Dilated image by 2x2 structuring element.
- Used connected component algorithm to find the leftmost, rightmost, topmost, and bottommost point of each digit.
- Used these bounds to construct subimages of each digit and operator.
- Detected all minus signs and some 1's, if white pixels took 60% or more of subimage.



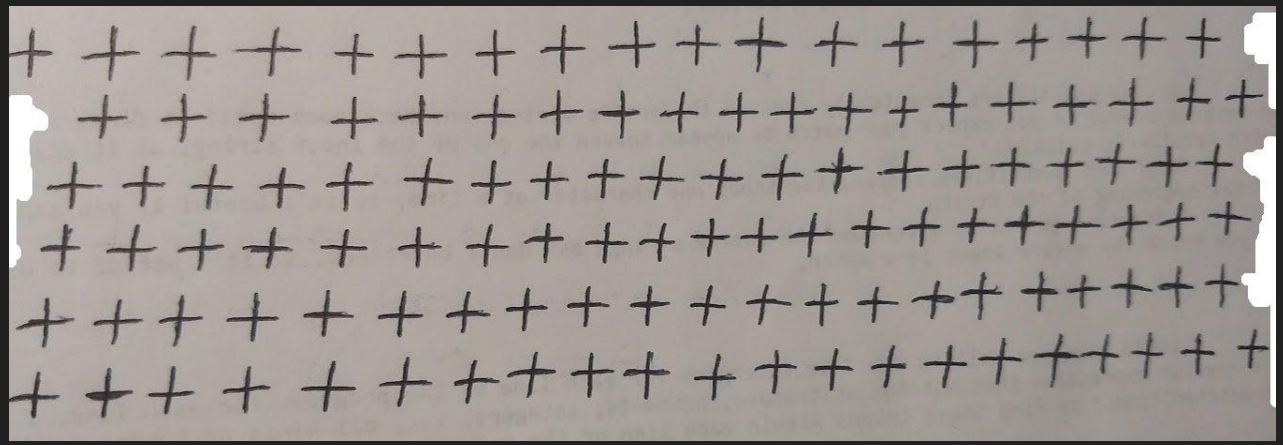


Segmentation Continued

- Of the non minus signs and classified 1’s, I padded the bounded figure with background, so it looks similar to MNIST.
- Next I scaled the image to 28x28 (average nearby pixel values)
- Next I eroded the resulting image by a 2x2 structuring element.
- Digits are now ready to be classified by a model trained from MNIST, but not + yet.

Add + to MNIST

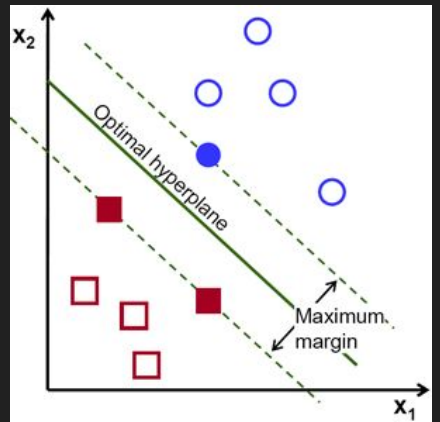
- Following the same segmentation process as earlier, generate 28x28 images of + from this handwritten image:



- Break down each subimage to a row of pixel values, and append to MNIST with a unique label (10)

Digit Recognizer - SVM

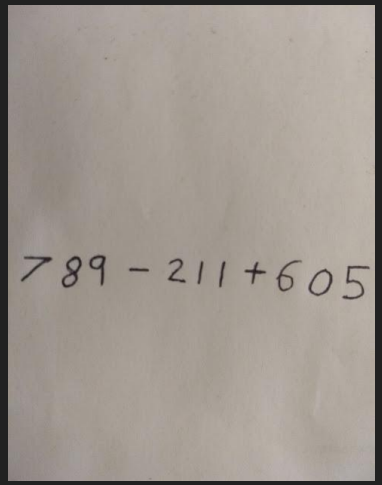
- Find hyperplanes that maximize the margin between classes
- Data is sparse (784 dimensions)
- Divide each pixel by 255, in order to reduce the distance between each point and make it separable.
- One against one classification.
- Used 80% of data to train, in order to find model parameters (slopes and intercepts).
- Accuracy: 98%, against the remaining 20% of dataset.



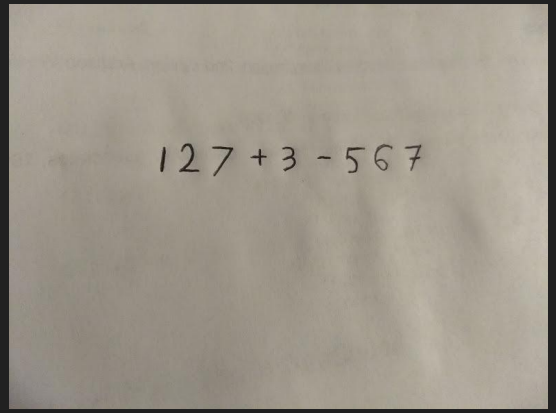
Evaluation of Expression

- Save trained SVM model to file.
- After segmentation, feed digits one by one into model, and retrieve its predicted class.
- Combine adjacent digits to form numbers.
- Apply classified operations to the numbers as appropriate to obtain the result.

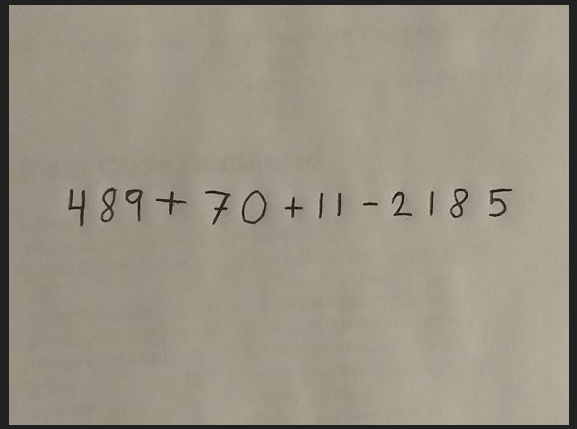
Results



789-211+505 = 1083

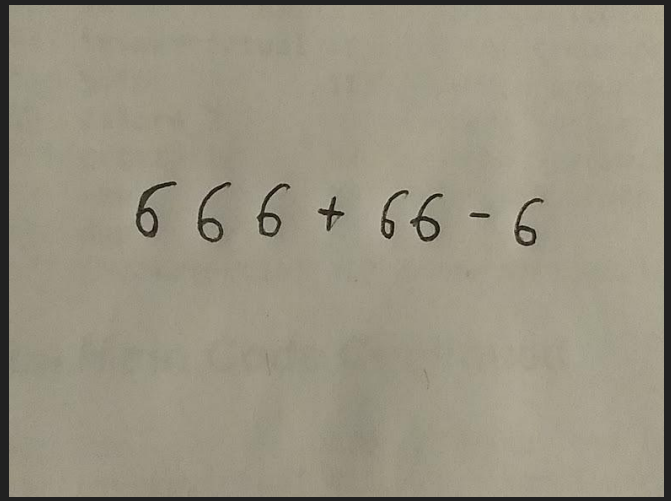


127+3-557 = -427

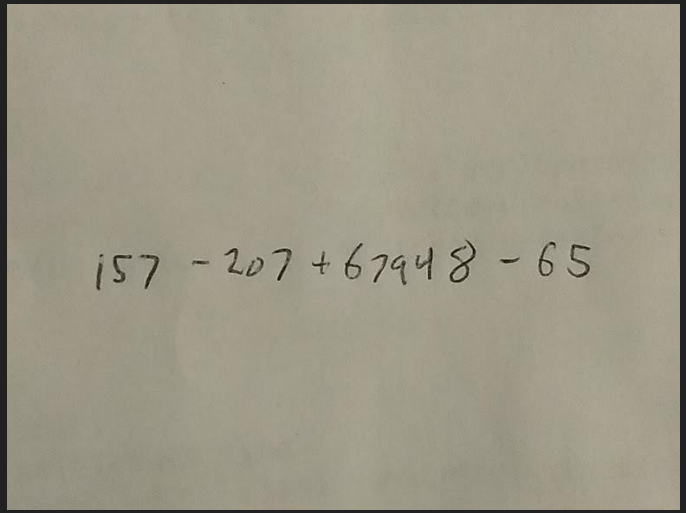


489+70+11-2185 = -1615

Results Continued



555+55-5 = 605



Nonsense

Future Work

- Be able to distinguish 5’s from 6’s.
  - Maybe my handwriting was not similar enough to MNIST
  - Maybe morphological operations and padding made my 6’s too different from MNIST
  - Difficult, because 5’s and 6’s look pretty similar
- Implement another method of finding bounding boxes, to evaluate more realistic images.
- Add support for more operations, and equations.
- Replace SVM with a deep learning model (over 99% accuracy has been achieved).