Recognition of Handwritten Names II

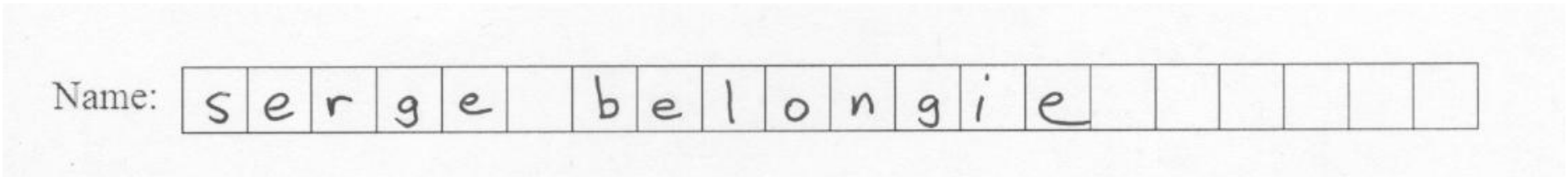
Dafna Bitton

Winter 2009, CSE 190a with Professor Serge Belongie

Problem Statement

Given a class roster, match an image of a handwritten name in character boxes to the string representation of the name.

For example, match:



to the string “serge belongie”.

Outline

Outline

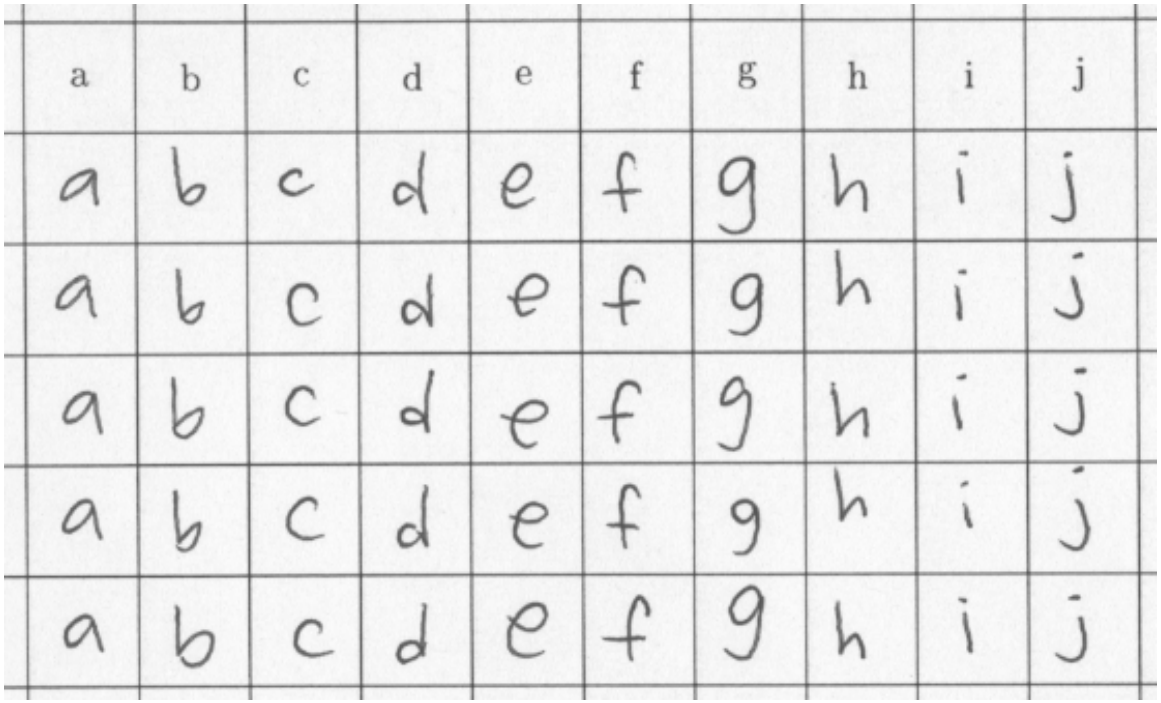
How it will work

- Students turn in work.
- TAs scan work after they grade it.
- Isolate the characters with normalized cross correlation.
- Perform OCR on cut out letters using a machine learning technique.
- Predict the string representation of the name based on the OCR results and the roster.

Outline

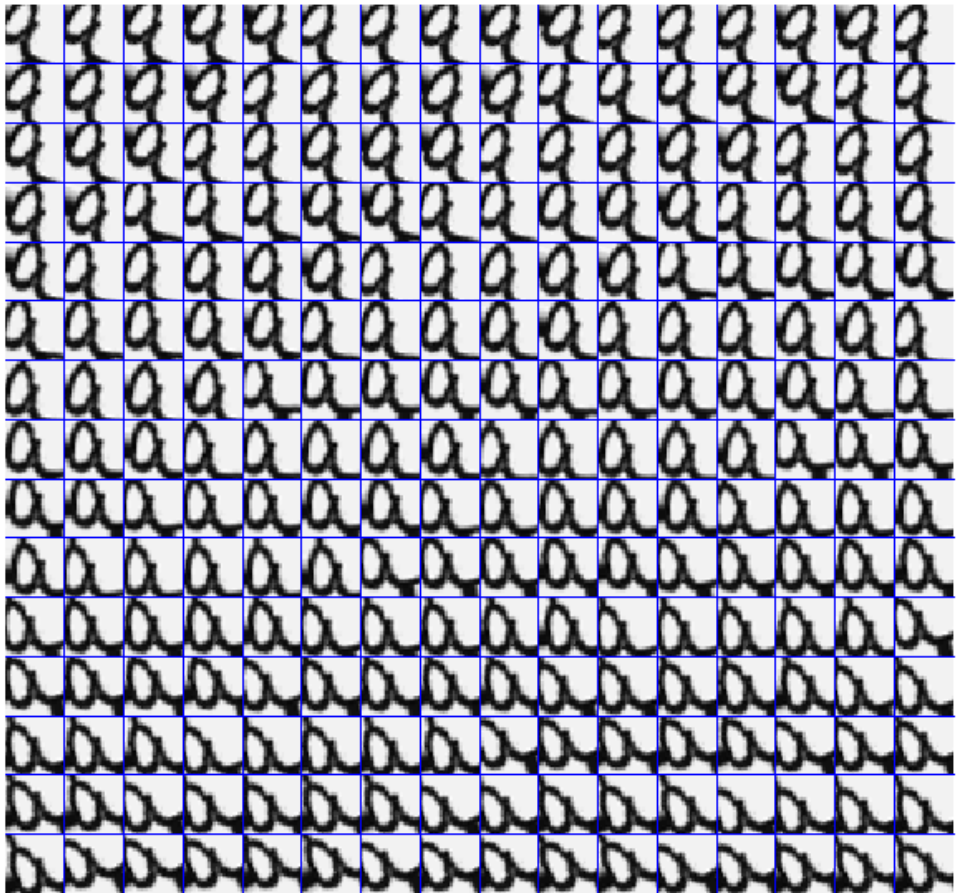
Training Data

- In order for the project to work, we need a large number of examples of each character.
- Used ABCDETC template for character examples.
- Scanned in filled-out sheets and used the Hough Transform to cut out letters.
- Sized all characters to 24 by 24 pixels.



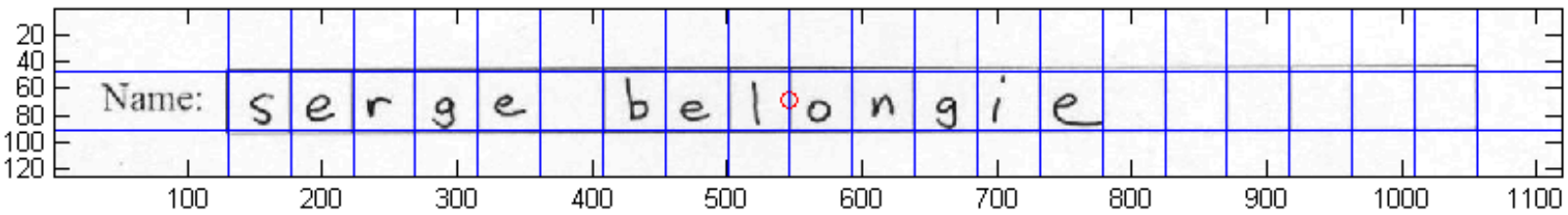
Jittering

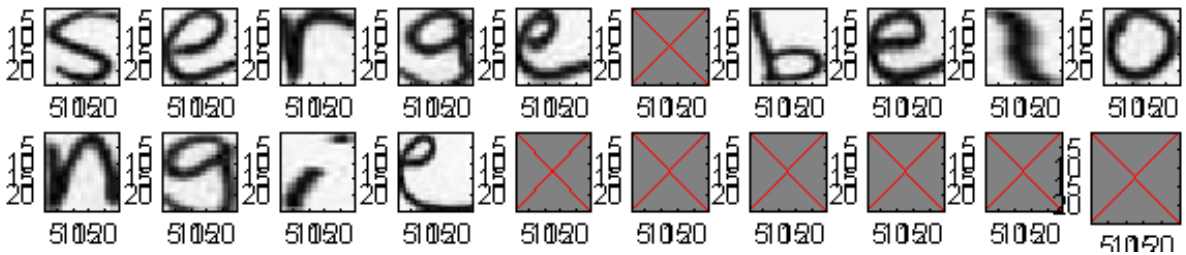
- In order to product more training data, we apply rotation and translation to the images we already have, to cover more bases.



Test Data

- Had people fill in their names in character boxes.
- Used normalized cross correlation to cut out characters.





Outline

First Attempt: Nearest Neighbor

- Have a large amount of examples of each type of letter 'a' through 'z'.
- Compute the Euclidean distance between an unknown letter and all training examples.
- Whichever letter is closest is what we classify the unknown letter as.

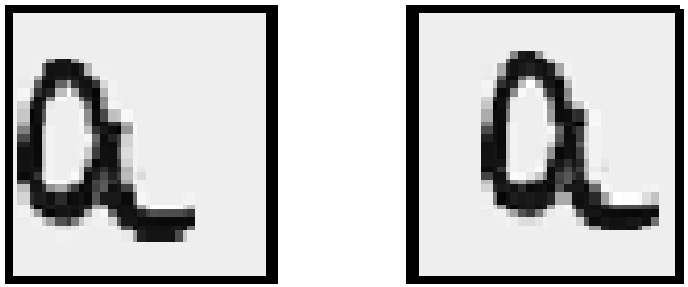
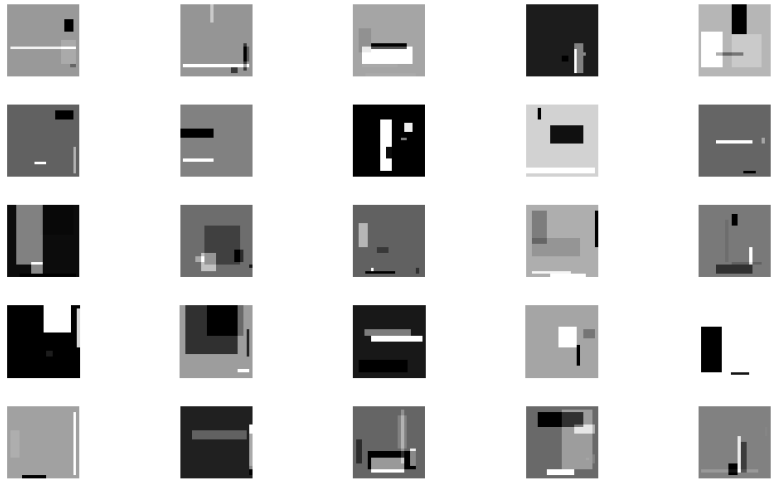


Figure: Example of when the Nearest Neighbor method would perform poorly.

Second Attempt: AdaBoost with Haar-like features

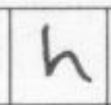
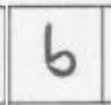
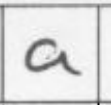
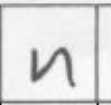
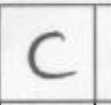
We create 26 classifiers - 1 for each letter. The 'a' classifier gives a confidence of whether or not an image is an 'a' or not.
Haar-like features:



- In the beginning, an obscene number of Haar-like features are created, but the number is cut down to the 200 most defining features.
- Instead of representing the image as a bitmap, we represent it as a 200 x 1 vector of the best Haar-like features.

Taking the Roster into Account

- Given an unknown handwritten name, we consider all names in the roster that have the same number of characters.
- For each of these possibilities, we find the confidence of each letter being the corresponding letter in the possibility.
- Whichever possibility yields the highest confidences is the winner.

unknown handwritten name													
confidences	.1	.2	.3	.2	.4	.6	.3	.1	.2	.5	.3	.2	.4
possibility	s	t	e	p	h	a	n	i	e	m	a	r	k

Outline

Results

- The results are not good. The classifiers do not perform as well as they should.

Outline

Improvements to be Made

- Debugging
- I would like to redo the training phase. If the training data is not good, then I can't expect to get good results.
- In the training phase, I want to be as close to the conditions that I will be in during testing.