Mobile Image-Based Equation Solver

Members:
Will Harvey (wharvey@stanford.edu)
Scott Harvey (sfharvey@stanford.edu)

Problem:
A major source of frustration when dealing with complex calculations is the time and effort cost of entering equations into a calculator or computer. When doing computational work, an individual must go through the laborious and time-consuming process of entering equations into a calculator or scientific computing program even after writing it all out by hand. Also, many potential users of these resources are unfamiliar with the syntax of these programs and cannot take advantage of the more advanced functions. Mobile apps such as ‘MyScript’ have proven the consumer desire for automated mobile calculators. However, these apps require the user to enter information on the screen instead of on handwritten paper and do not support more rigorous uses.

Solution:
We would like to continue efforts towards developing a robust mobile application where users take a picture of handwritten equations and the result is returned. The initial version will resolve an equation from a picture of the handwritten text. If time allows, we will improve this design by incorporating equation-recognition to allow users to solve equations in real-time by simply hovering live video-capture. With respect to initial success, other potential features would include more advanced functions.

We will not be using a DROID camera phone. Instead, we plan to develop our project for iOS (specifically the iPhone, and extended to the iPad if time allows). Since the objective is to recognize handwritten text, KNN and SVM classifiers will be used instead of Tesseract (which performs poorly on non-typed text). Ideally, we would like as much of the program to run locally on the iPhone (instead of relaying data to and from a server).

Basic Program Flow:
1) Pre-processing (morphological operations, transform and characters/equation to adjust for angled-capturing of images, extract each individual character)
2) Optical Character Recognition (classify each character with KNN/SVM and possibly taking into account position to other classified characters as well as format of legal equations)
3) Construct entire equation from each individual character based on position/knowledge of format of legal equations)
4) Solve equation either locally or by sending to equation-resolver

References: