Competitive Analysis

Since my program has primarily two components, I will be independently investigating two kinds of products, the first, OCR as applied to mathematical formulas, the second, the processing of more intuitive forms into computer legible inputs and the solving of them.

A thorough web search indicated that the field of OCR recognition applying to mathematical formulas is still extremely nascent with only a single semi-commercial product and a handful of proof of concept software. The most complete of the softwares is the InftyProject, a research project that produces software that can analyze scanned scientific and mathematic documents and output LATEX format mathematical formulas. Its features include the preservation of original layout via LATEX formatting as well as fairly good accuracy on less complicated expressions. This software and mine share the same core feature, the OCR of mathematical symbols into computer legible form. The largest strength of this software, and a feature I would like to include, is the ability to preserve the layout of the original document, enabling scanning of large documents with multiline components. Despite being the most complete of the software packages, Infty is still quite incomplete, in particular it struggles with more complex expressions, often missing variables and misinterpreting when the expressions grow long. Its largest drawback by far however, is its inability to process handwritten formulas, Infty is only capable of recognizing scanned documents, not handwritten symbols.

The next software, JMathNotes, is a proof of concept software that attempts to recognize digital handwritten mathematical formulas. Although it is incomplete the software nonetheless manages to impress as it recognizes distorted handwritten expressions. This software, however, along with most other softwares that recognize handwritten symbols, only allow for digital input and is incapable of reading physical handwritten inputs.

The second set of software, the ones that process and solve mathematical inputs, is far more established and is represented by its most famous example, WolframAlpha. WolframAlpha is capable of taking in a staggering variety of digital mathematical inputs as well as natural language writing and convert them into computer legible form before solving and displaying the results. The features of WolframAlpha are too many to enumerate but the one feature that I would like to implement in my program is some basic commands that can be handwritten on paper that can be parsed by the program. For example if the command “plot” is beside an equation then the program would plot the equation, and if the command is “solve”, then the program would solve and display the equation.