

# EE368 Project Proposal: Word Hunter

Droid Phone: Yes April 28, 2013

Lingren Zhang ([Lingren.zhang@gmail.com](mailto:Lingren.zhang@gmail.com))

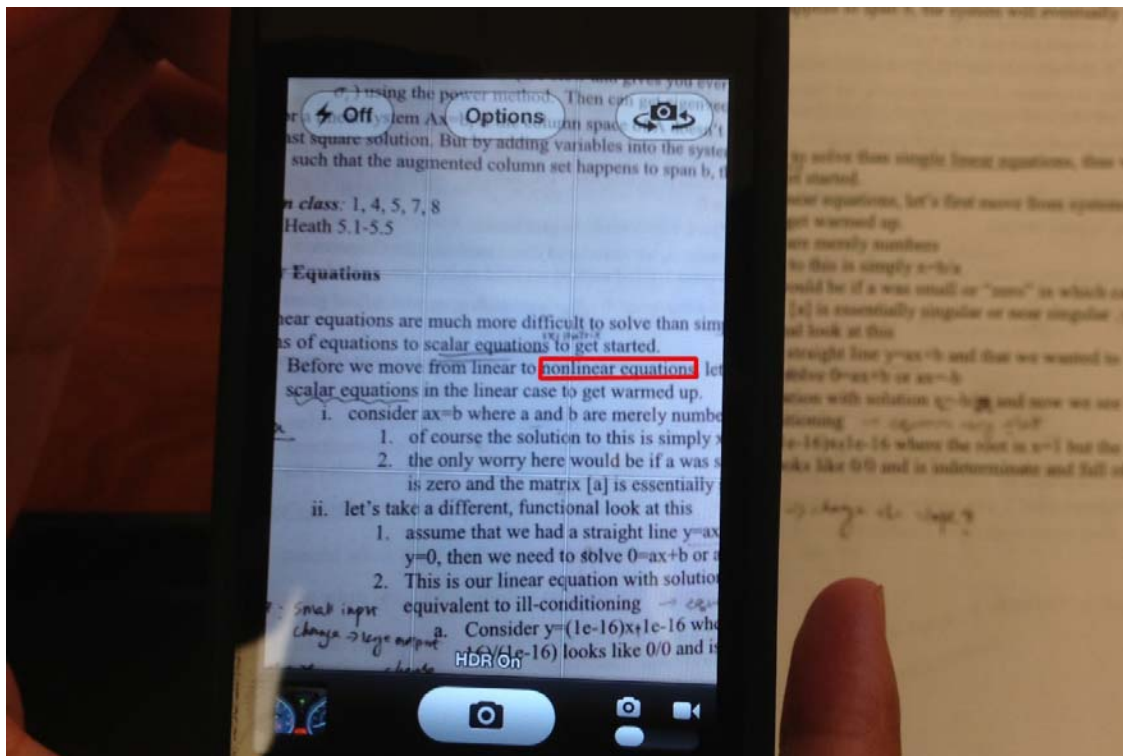
Shuo Liu ([shuol@stanford.edu](mailto:shuol@stanford.edu))

Yang Zhao ([yzhao3@stanford.edu](mailto:yzhao3@stanford.edu))

## Introduction and Background

Have you ever read a long paper based article and find yourself unable to locate keywords? With the advancement of digital media, sometimes we take basic word search for granted. But the world is still full of media printed on paper and it would make our lives much simpler if we can automate this basic reading tool for good old fashioned books. We propose a mobile application that will be able to find a word that the user specified through a phone's viewfinder. As soon as the phone detects the word it will highlight it, saving the user many minutes of looking for the word him/herself.

For example, we want to search for the phrase "nonlinear equation" in this page of paper. We only need to use our smart phone to scan over the paper. Whenever the phrase "nonlinear equation" appears on the phone screen, it will be immediately circled in red.



## Proposed Work and Timeline

### Pre-processing:

1. Binarization: Use Otsu's method (minimize intra-class variance/maximize between-class variance) to perform image thresholding on the gray-scale image. It should work reasonably well with black text on white background (or uni-colored text on uni-colored background).
2. De-skew: We use the method described in Chen and Haralick. Define text skew angle to be the angle that the image is rotated counter-clockwise by. We detect the directions of each text line, with which we perform a Bayesian estimate of the document text skew angle. Finally we can perform a correctional rotation of the image.
3. Layout Analysis: Pages will have different layouts (columns, blocks, etc.), and we would like to identify those layouts, as a word can be wrapped-around from the end of one text line to the beginning of the next text line. We will study and implement the algorithm described in Diem, Kleber and Sablatnig. The idea is to identify word blobs first and then group word blobs together with a clustering algorithm.

### Word Recognition -- OCR:

- We propose to use the tesseract OCR engine for this project.  
Approximate tesseract algorithm:
  - the engine first detects component outlines, which are then grouped into blobs.
  - Blobs are organized into lines and lines are analyzed for fixed pitch.
  - The pitch is then used to separate out the words.
  - During the first pass, each satisfactory word is entered into an adaptive classifier which will enable the engine to more easily recognize the words further down the page. A second pass will be made to reanalyze the words at the beginning of the page since the classifier was not trained during the first pass.
  - A final phase resolves fuzzy spaces, and uses x-height normalization to detect lower case letters.

### Post-processing:

- Spell-check backed by a lexicon

Since the correctness of words detected by our algorithm may be influenced by various factors, such as skewed view, unbalanced illumination, the spell-checking will also be included in our project. The basic idea is to improve OCR accuracy by constraining the word by a lexicon – a list of “legal” words that are allowed to occur in a document. The available lexicon such as Hunspell dictionary and UNIX shared dictionary will be helpful in our projects. Also, by leveraging the technique of dynamical programming to find the edit distance, spell-check could be done based on edit distance analysis.

- Other Possible Extension Services

- Word translator:

Once the particular word is circled out, we could use Google Translate API to translate the word.

- Word counter:

If each word could be spotted by our algorithm, the application could also easily be modified to become a word counter software

### Time Line

- proposal write-up and background reading (4/25 - 5/1)
- static image word recognition (5/1 - 5/12)
- mobile platform set-up and build something that can work(5/12 - 5/19)
- Apply to real-time video and modify the extension (5/20 - 5/31)
- Report write-up (6/1 - 6/4)

### Possible Difficulties:

- The segmentation has to be as clean as possible. Any background images must be removed. Pitch between words need to be uniform.
- De-skew: When we take a picture/video of a piece paper, two types of skew can occur: First, misalignment of horizontal direction (horizontal rotation) of the phone camera and the paper, which will potentially be taken care of by the proposed de-skew algorithm. Second, the camera might be tip-tilt, so one side/corner of the image will appear larger/closer than the opposite side/corner.

There are limited literature addressing this issue, especially in the text recognition context, and it is much more complicated compared to the first type of skew as it requires two variables to describe (both horizontal and vertical rotation). For the moment, we will leave this part as an optional feature.

- Efficiency/speed: Performing each of the task above frame-by-frame is going to computationally heavy, it is highly unlikely that we will be able to finish all the steps described above for the previous frame before the next frame comes in. We may have to compromise by speeding up the algorithm or performing the computation once every few frames.

For instance, since spell-checking needs to go through the lexicon and do comparison for each word, a high efficient spell-check algorithm is crucial to our real-

time word hunter. We may use C++ to implement this part of algorithm, and import it as a package in MATLAB.

## References:

- [1] "An Overview of the Tesseract OCR Engine" , R. Smith ,9th International Conference on Document Analysis and Recognition.  
[http://static.googleusercontent.com/external\\_content/untrusted\\_dlcp/research.google.com/en/us/pubs/archive/33418.pdf](http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en/us/pubs/archive/33418.pdf)
- [2] An Automatic Algorithm For Text Skew Estimation In Document Images Using Recursive Morphological Transforms, Su Chen and Robert M. Haralick, Image Processing, 1994  
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=413291>
- [3] Text Classification and Document Layout Analysis of Paper Fragments, Diem, Kleber and Sablatnig, International Conference on Document Analysis and Recognition (ICDAR), 2011  
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6065432>