Bananagrams Helper for Android

Members: Steve Leung (stevenleung@stanford.edu), Steffi Perkins (slperkins@stanford.edu), Colleen Rhoades (rhoades@stanford.edu)

Motivation:
Bananagrams is a highly competitive game in which players use their letter tiles to add words to a Scrabble-style game board. Those with a strong ability to recognize words in their assortment of random tiles have an advantage over those that struggle to spot favorable combinations of letters. In order to level the playing field when the opponent is a highly skilled Bananagrams player, we propose an image processing mobile application that is able to process a player’s letters through the viewfinder and inform the player of possible words he or she could play. The application would also take into account additional letters on the game board (inputted by the user if desired).

Goals:
The Bananagrams Helper will be able to:
- receive images of Bananagrams tiles as inputs
- perform local adaptive thresholding on the input images
- reorient the tiles to upright positions
- extract letters from the tiles and combine them into an anagram string
- accept user input for additional letters
- query a server to solve the anagram
- display a list of words for the user

Image Processing Pipeline:
User Input
The image processing pipeline will work as follows. A user captures lettered tiles in the viewfinder using the Bananagrams Helper Android application. An option to input an
additional letter will be available (in order for the user to connect his or her tiles to the existing board).

Tile Reorientation/Letter Detection
We are considering several possible algorithms for tile reorientation and letter extraction:
1. The application will compute MSERs on the image and isolate the tiles. Using the region properties of the MSERs, the letters may be aligned in upright positions.
2. Tiles will be deskewed based on the projective transform method for aligning playing cards upright described by Brinks and White [1].
3. Tiles will be aligned by estimating the skew angle, using a method that first isolates the card within a bounding box, and then compares the distances from the edge of the bounding box to the nearest foreground pixel at different points along the length of the box [2].
4. Morphological gradient detection uses image edges to detect letters. It is insensitive to noise, skew, and text orientation [3].
5. Stroke width transform detects contiguous parts of images that form a band of nearly constant width [4]. These "strokes" of constant width are likely to form letters.

Letter extraction will be performed using a hit-or-miss detector. The hit-or-miss detector could fail if the letters are unable to be sufficiently de-skewed or if there are other distortions in the image.

Server Communication
The recognized characters as well as the additional character inputed by the user will be sent to a server. We plan on using an anagram solver to produce a list of possible words that can be formed with the user’s tiles. This list of words will then be sent back to the phone and displayed for the user on the viewfinder.

References: