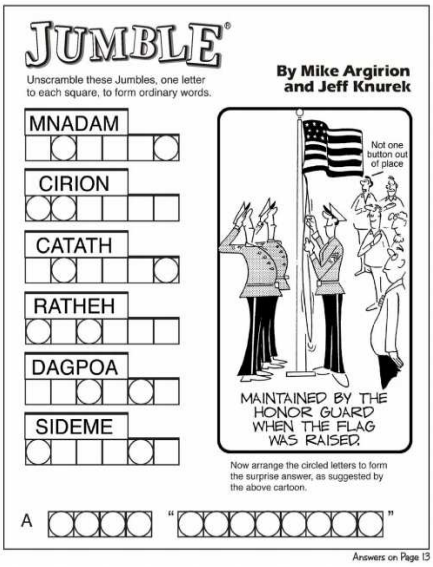
EE368 PROJECT PROPOSAL

Debabrata Sengupta, Abhishek Sharma

{dsgupta, abhisheksharma} @ stanford.edu

Solving Word Jumbles

We propose to develop an **android based** application that can **assist the user in solving word jumbles** commonly found in newspapers and magazines. An example of a typical word jumble which might be found in a book of puzzles is shown in the figure below.



Goals :

We envisage that the application should be able to do the following:

(1) Identify the (scrambled) letters of each word from an image of this puzzle taken using a droid phone.

(2) Unscramble these letters to create valid english words.

(3) Identify the “circled” letters which can further be shuffled to form the “mystery answer”.

(4) Provide either a complete solution to the entire puzzle or provide hints to the user in the form of revealing one letter at a time for each word.

Challenges :

At a first glance, the challenges that we might encounter while designing this application are as follows :

(1) Making the application rotation and scale invariant : The tilt and the magnification of the image should not affect the performance of the application.

(2) Tackling perspective skew that might be present in the image.

(3) Making the application invariant to illumination conditions, slight blurring of letters while taking the image, etc.

(4) Identifying and recognizing only those letters that belong to scrambled words : As can be seen from the image, a typical puzzle has quite a lot of text, not all of which is necessary for our application. Hence we need to isolate and recognize only those letters that are important to us.

Assumptions :

We propose to make the following assumptions about the general structure of our word jumbles, at least in the initial phase of our project. We might explore how to get rid of a few assumptions and generalize our application even more, if we have time in hand after our application is up and working.

(1) The scrambled letters of each word is placed inside a rectangular box. We do not insist on each individual letter be separately boxed out in the image. However, we require that all the letters be capitalized and there be some spacing between adjacent letters.

(2) The “important letters” which further need to be rearranged to get the “mystery answer” must be shown by circles and not by any special color or any other symbol.

(3) The letters, when unscrambled, should produce valid words belonging to the English Dictionary, i.e. proper nouns are not allowed.

(4) In case the letters can be arranged to form more than one valid word, the application will try to guess the “mystery answer” using all possibilities and will populate the results using those combinations which successfully yield a valid “mystery answer”. In case there are multiple valid “mystery answers” possible, the application will display all choices to the user and let him choose one.

(5) We do not intend to use the picture clue given on the right of the puzzle since even if we could use OCR to successfully read the clue, it would be difficult to *interpret* it and use it to obtain a solution. Rather, we propose to use a rudimentary brute-force approach to come up with all possible solutions and let the user choose the best one (in case there are many).

(6) The “mystery answer” should again be composed of valid english words.

Algorithm

Very broadly, our application will probably consist of performing the following steps. We might add or remove a few steps later on as required.

(1) Using Hough Transform to detect every box containing the scrambled letters and then designing a segmentation algorithm to separate each letter within that box. These segmented letters will then be fed into the OCR engine.

(2) Identifying the circles in the image, and their relative positions in the boxes, thereby detecting the positions of the “important letters”.

(3) Identifying the number of words in the “mystery answer” and number of letters in each word from the number of rectangular boxes right at the bottom of the image.

(3) Using an OCR engine to identify each letter.

(4) Writing code to rearrange the letters of every word to form valid words.

(5) Coming up with the mystery answer.

References

[1] "Use of the Hough Transformation to Detect Lines and Curves in Pictures," R. O. Duda, and P. E. Hart, Comm. ACM, Vol. 15 , pp. 11 – 15 (January, 1972)

[2] "An Overview of the Tesseract OCR Engine" , R. Smith ,9th International Conference on Document Analysis and Recognition

[3] “Adaptive Deblurring for Camera-Based Document Image Processing”, Yibin Tian and Wei Ming, ICSV 2009