

## IMRESTAURANT()

Oftentimes, one would like to have the benefit of easily accessible reviews for restaurants while deciding where to eat. We propose a Droid application called `imrestaurant()` that will, in one step, provide [yelp.com](http://yelp.com) reviews from a photograph of a restaurant's sign. `imrestaurant()` will extract the name of the restaurant from the photograph using an amalgam of optical character recognition techniques. Thus, we save the user the tedious task of opening a web browser and typing the restaurant name on a tiny smartphone keyboard or touchscreen. There will be four major processing steps in our pipeline: preprocessing, feature extraction, classification, postprocessing.

To begin, we will need to do preprocessing to segregate individual characters. To compensate for camera shortcomings, we will first perform non-linear noise reduction and sharpening on the image. Thresholding based on color histograms will be used to produce a binary image. This new image will be eroded to help segment any connected letters and aid in feature extraction. Horizontal projection histograms will be used to determine how many lines of text are in the image, and vertical projection histograms of each line will be used to locate words. Then, region counting and labeling will identify individual characters. Finally, skew and slant removal techniques will also help reduce variation among letters of different fonts.

Now that we have isolated, aligned characters, we can perform feature extraction on each character. Useful information can be found through vertical and horizontal projection histograms, profiling, zoning, and counting crossings. Projection histograms will count the number of pixels in vertical or horizontal buckets. Profiling presents the information of the number of pixels between the character and the bounding rectangle. Zoning determines direction features and also distinguishes line segments. Counting crossings determines transitions from text to background.

All of this information will be compared to the respective properties of a reference character set. The reference character set will include letters and numbers of various fonts. If regions in the photograph cannot be classified, they will be ignored as extraneous logo elements. After classification, we can use the results to perform a preliminary search on Yelp. If no restaurants are found in the surrounding area, we repeat the algorithm with different preprocessing parameters and classification requirements. Otherwise, we present the review to the user.

Due to the limited scope of the project, we place the following criteria on the photograph:

1. The only text contained in the photograph is the restaurant's name.
2. Characters in a word are in a straight line.
3. Characters are read from left to right, not top to bottom.
4. No cursive or connected characters.
5. Individual letters may only consist of contiguous segments.

## References:

Chellapilla, Simard, Nickolov. "Fast Optical Character Recognition through Glyph Hashing for Document Conversion."

<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1575661>

Kahan, Pavlidis, Baird. "On the Recognition of Printed Characters of Any Font and Size."

<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4767901>

Kopec. "Least-Squares Font Metric Estimation from Images."

<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=242359>

Trier, Jain, Taxt. "Feature Extraction Methods for Character Recognition- A Survey."

<http://www.cs.sfu.ca/fas-info/cs/CC/414/li/material/refs/OCR-survey-96.pdf>

Vamvakas. "Optical Character Recognition for Handwritten Characters."

[http://www.iit.demokritos.gr/IIT\\_SS/Presentations/Off-Line%20Handwritten%20OCR.ppt](http://www.iit.demokritos.gr/IIT_SS/Presentations/Off-Line%20Handwritten%20OCR.ppt)

Xie. "Optical Character Recognition Based on Least Square Support Vector Machine."

<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5370427>

Zramdini, Ingold. "Optical Font Recognition Using Typographical Features."

<https://www.cs.drexel.edu/classes/CS/cs680-502/proj4/fontrecog.pdf>