Introduction to Optical Character Recognition

Xin Huang, Raul Martinez, Leo Meister, Jonathan Nguyen

hy Make a Tutorial?

- Optical character recognition (OCR), has been extensively researched and documented
- Lack of a fully comprehensive step-by-step guide to using OpenCV
- Further expand our knowledge and contribute to the open source community
- Provide basic insight to beginners to OCR in the field of text-based character recognition

Algorithms

- K-Nearest Neighbors (k=1)
- Image filtering – convolve intensity matrix by filter

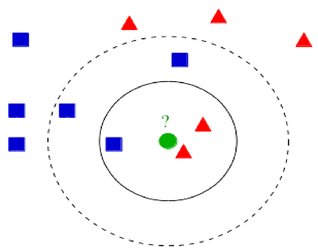


Illustration of KNN (k=3)

$$\frac{1}{115}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Gaussian filter

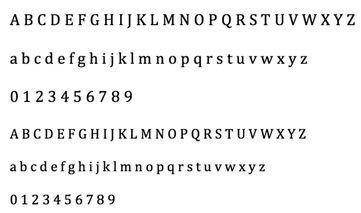
OpenCV

- Experimented with a variety of techniques
- Settled on bounding rectangles over Canny edge detection
- K-Nearest Neighbors (KNN) over Support Vector Machine (SVM)

OpenCV

Training Data

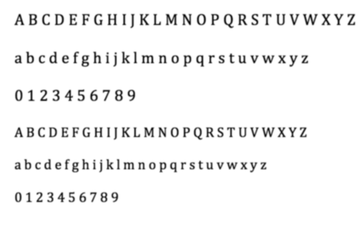
- Multiple fonts and sizes
- Intensity values as features
- Added more data for similar characters to improve recognition



Gaussian Blur

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

- Eliminates high-frequency noise that may result in false edges

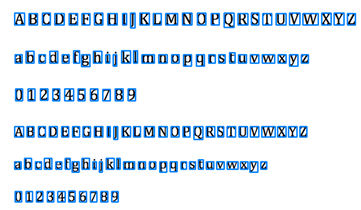


Thresholds

- Calculates gradient using adaptive thresholding
- Compares to threshold value

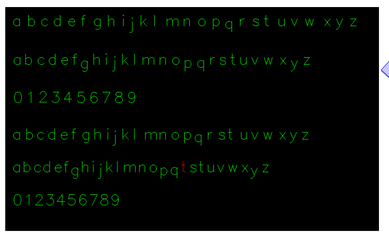
Contour

- Find edges based on gradient
- Outline the edges on input image
- Resize each bounding rectangle to a 10x10 matrix of intensity values



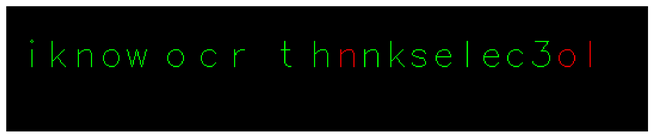
Recognition

- Compares intensity vectors against training results
- Uses k-nearest neighbors
- Green characters are matches
- Red characters are misses



Conclusion

- Achieved ~85% accuracy using simple features
- Achieved 10% greater accuracy with the Gaussian blur than without
- Produced an easy-to-follow tutorial providing experience in Python and OCR
- Soon to be released on Connexions



Future Work

- Include how to find and use a greater variety of features for better accuracy
- Incorporate more techniques (SVM instead of KNN, MSER regions)
- Use it on pictures of text instead of screenshots
- Learn more in ELEC 345 and improve upon algorithms

References

Open Source Computer Vision Library: opencv.org

Introduction to edge detection: http://dasl.mem.drexel.edu/alumni/bGreen/www.pages.drexel.edu/_weg22/can_tut.html

Acknowledgements

Special thanks to Dr. Richard Baraniuk, Chris Metzler, and Vivek Boominathan.

Rice University DSP