NEW YORK STATE CSTEP — COLLEGIATE SCIENCE & TECHNOLOGY ENTRY PROGRAM

preparing science, technology & licensed professionals since 1986

Text Document Enhancement with Image Processing

Justyce Countryman | SUNY Oswego

Project Vision & Purpose

• Text documents may consist of a noisy background that are difficult for computers and people to interpret:
  - Stains
  - Eraser Marks
  - Fold Creases
  - Salt and Pepper Noise
• Significant real-world benefits include:
  - Restoring old, distorted, and/or historical documents.
  - Increasing text visibility, which is especially advantageous for visually impaired individuals
  - Useful for researching ideal preprocessing techniques for all kinds of noise.

Project Algorithm

- Allow the user to select an image preprocessing technique
  - Practical options currently include Otsu's, binary, and adaptive thresholding, opening/closing morphology, and bilateral filtering
- Obtain an image (preferably a text document)
- Calculate the grayscale version of the image
- Perform the user-selected preprocessing technique with the OpenCV library on Python and display the resulting binary image.
  - Dynamically update the image based on additional user-input
- Allow the user to extract text via Tesseract.

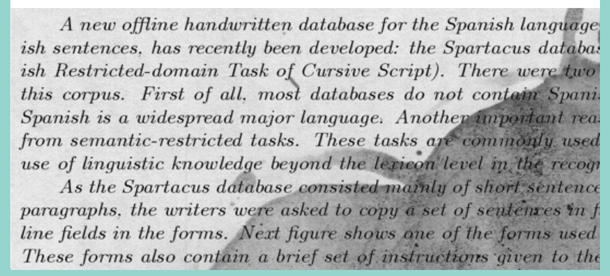


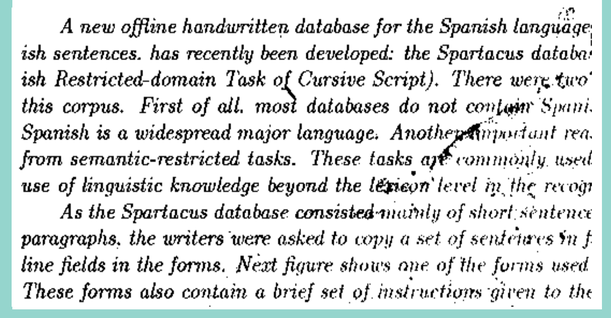
Image Preprocessing Techniques

- Thresholding
  - Otsu's Thresholding
    - Automatically calculates an optimal threshold value that separates the foreground from the background
  - Binary Thresholding
    - Allows the user to manually input a threshold value.
    - Convenient because Otsu's method could worsen the noise.
  - Adaptive Thresholding
    - Divides an image into regions and calculates thresholds for each region.
    - Computed by either taking the mean of the region or through a weighted sum, and then subtracted by a constant
- Mathematical Morphology
  - Opening
    - Dilate and then erode image regions
    - Useful to remove small objects, lines, and bridges
  - Closing
    - Erode and then dilate image regions
    - Helps with eliminating holes and filling gaps
- Filtering
  - Median Filtering
    - For every pixel, set that pixel to the median value based on a certain size of neighboring pixels
    - Preserves edges
  - Gaussian Blur
    - Same idea as median filtering, except instead of the median, it's a weighted sum
  - Bilateral Filtering
    - Considers neighboring pixels and their intensity differences.
    - Results in only nearby and similar pixels being blurred
    - Reduces "salt-and-pepper" noise

Determining The Best Method

- There is no "one-size-fits-all" solution for enhancing text documents.
  - However, there are algorithms that find the best image that a computer understands.
- A "text extraction accuracy" is calculated through the "Levenshtein Distance" algorithm
  - Calculates the distance required to go from one list of characters to another.
    - Ex: Going from "You are sitting" to "You are a kitten" requires two words to be modified.
  - The accuracy is calculated as ((max_cost - calculated_cost) / max_cost) * 100
  - Useful for when there is a noisy and clean version of the same image.

- Most effective preprocessing methods to remove specific types of noise:
  - Stains -> Adaptive thresholding
  - Eraser marks -> Otsu's or binary thresholding
  - Fold creases -> Otsu's or binary thresholding
  - Salt and pepper noise -> Bilateral filtering



Where Do We Go From Here?

- Custom-made image preprocessing algorithms could be created based on which ones were the most successful
  - Additional real-world applications include training machine learning algorithms, medical imaging, object detection/recognition
- The Levenshtein distance will help in determining the best preprocessing technique, as well as the best parameters when given a noisy image
- Advantageous features to add to software that utilizes binary images, like PDF scanning apps.

References

Chinnasarn, K. (n.d.). *Removing Salt-and-Pepper Noise in Text/Graphics Images. Faculty of Science, Burapa University*. angsila.cs.buu.ac.th. https://angsila.cs.buu.ac.th/~kirsteen/cv/paper/Apccas98.pdf
Coman, I. (2024, February 8). *Image Analysis - Part 1: Binary Image Analysis* [PowerPoint slides]. *College of Liberal Arts and Sciences, SUNY Oswego*. *https://mylearning.suny.edu/d2l/le/content/1083058/viewContent/33296009/View*
Coman, I. (2024, February 15). *Texture* [PowerPoint slides]. College of Liberal Arts and Sciences, SUNY Oswego. https://mylearning.suny.edu/d2l/le/content/1083058/viewContent/33414578/View
Cukierski, W. (2015). *Denoising Dirty Documents*. Kaggle.com. https://kaggle.com/competitions/denoising-dirty-documents
Espua-Boquera, S., Pastor-Pellicer, J., Castro-Bleda, M. J., & Zamora-Martinez, F. (2015, January 2). *NoisyOffice*. archive.ics.uci.edu. https://archive.ics.uci.edu/dataset/318/noisyoffice
Huamin, A. (n.d.). *Histogram Equalization*. Docs.OpenCV.org. *https://docs.opencv.org/3.4/d4/d1b/tutorial_histogram_equalization.html*
Jayasetti, P. (2020, October 16). *Remove Salt and Pepper noise with Median Filtering*. Medium.com. https://medium.com/analytics-vidhya/remove-salt-and-pepper-noise-with-median-filtering-b739614fe9db
Kumar, A. (2019, August 22). *Denoising Documents with background noise*. Medium.com. https://medium.com/@amardeepkumar_25731/denoising-documents-with-background-noise-f449e1fd92d2
North Code. (2020, July 13). *Enhance a Document Scan using Python and OpenCV*. YouTube.com. https://www.youtube.com/watch?v=tYF3HBkvYOO&t=1s
Nurlkri, F. (2022, November 1). *How to Build Optical Character Recognition (OCR) in Python*. *Builtin.com. https://builtin.com/data-science/python-ocr*
Prasad, S. (2020, December 10). *Python for Character Recognition – Tesseract*. Topcoder.com. *https://www.topcoder.com/thrive/articles/python-for-character-recognition-tesseract*
The Qt Company. (2024). *QFileDialog Class*. *doc.qt.io*. https://doc.qt.io/qt-6/qfiledialog.html
Sanchhaya Education Private Limited. (2023, February 15). *Image Enhancement Techniques* using OpenCV – Python. GeeksforGeeks.org. https://www.geeksforgeeks.org/image-enhancement-techniques-using-opencv-python/
Schuck, J. (2020a, July 12). *Enhance a Document Scan using Python and OpenCV*. Medium.com. https://medium.com/analytics-vidhya/enhance-a-document-scan-using-python-and-opencv-9934a0c2da3d
Schuck, J. (2020b, July 12). *OpenCV-with-Python-Series/LICENSE*. GitHub.com. https://github.com/joschuck/OpenCV-with-Python-Series/blob/master/LICENSE
zaidkhan15. (2024, January 31). *Introduction to Levenshtein distance*. GeeksforGeeks.org. https://www.geeksforgeeks.org/introduction-to-levenshtein-distance/
https://www.skyillabs.com/blog/innovative-image-processing-based-final-year-projects