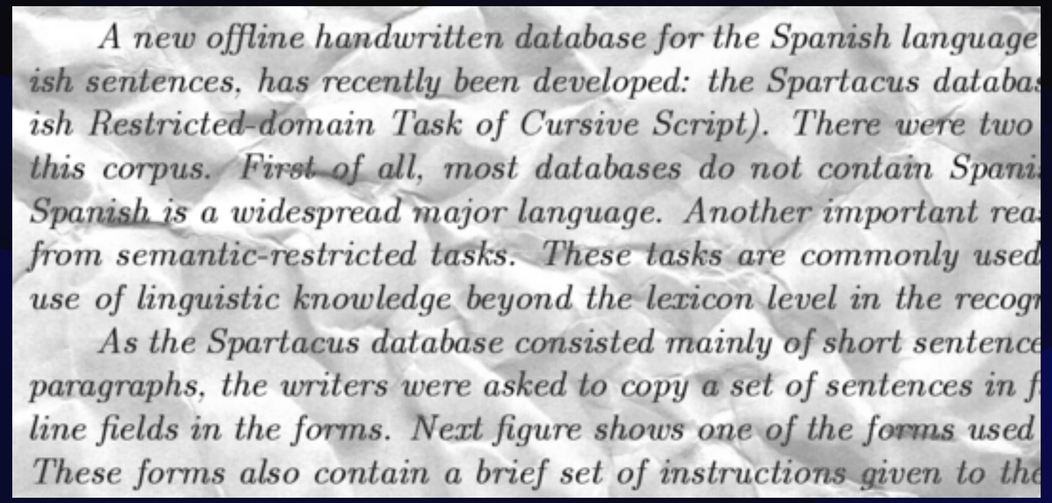
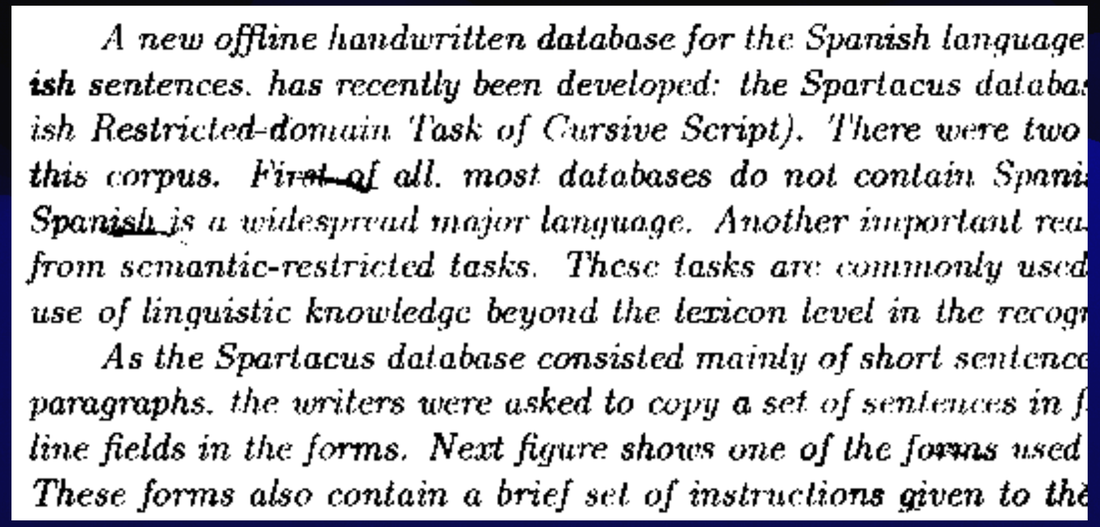
Text Document Enhancer

CSC 474 Project



A new offline handwritten database for the Spanish language
ish sentences, has recently been developed: the Spartacus databa
ish Restricted-domain Task of Cursive Script). There were two
this corpus. First of all, most databases do not contain Spani
Spanish is a widespread major language. Another important rea
from semantic-restricted tasks. These tasks are commonly used
use of linguistic knowledge beyond the lexicon level in the recogn
As the Spartacus database consisted mainly of short sentence
paragraphs, the writers were asked to copy a set of sentences in f
line fields in the forms. Next figure shows one of the forms used
These forms also contain a brief set of instructions given to the

(Espaa-Boquera, Pastor-Pellicer, Castro-Bleda, & Zamora-Martinez, 2015)



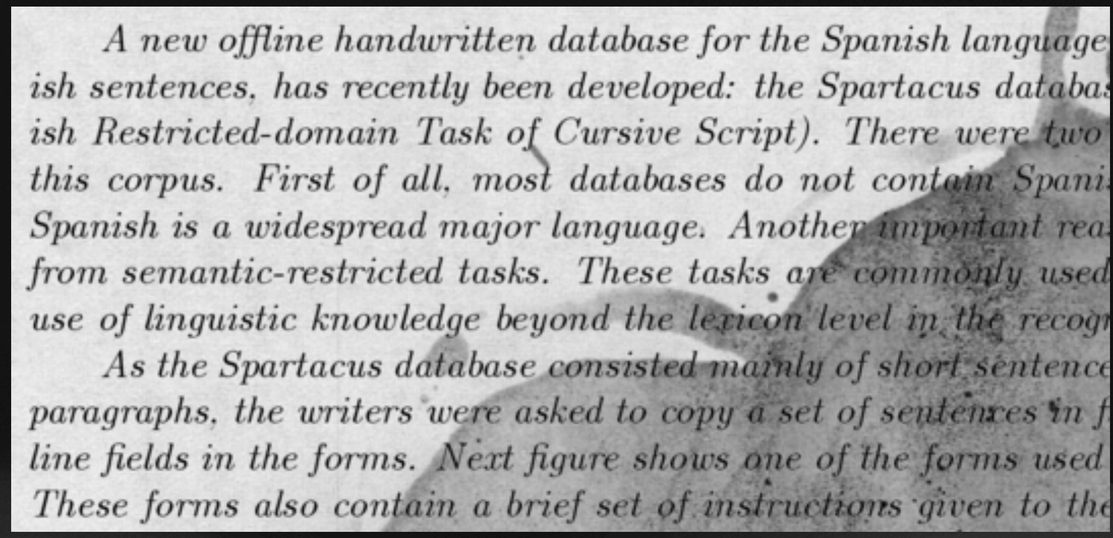
A new offline handwritten database for the Spanish language
ish sentences. has recently been developed: the Spartacus databa
ish Restricted-domain Task of Cursive Script). There were two
this corpus. First of all. most databases do not contain Spani
Spanish is a widespread major language. Another important rea
from semantic-restricted tasks. These tasks are commonly used
use of linguistic knowledge beyond the lexicon level in the recogn
As the Spartacus database consisted mainly of short sentence
paragraphs. the writers were asked to copy a set of sentences in f
line fields in the forms. Next figure shows one of the forms used
These forms also contain a brief set of instructions given to the

Project and Presentation Created by: Justyce Countryman

May 7, 2024

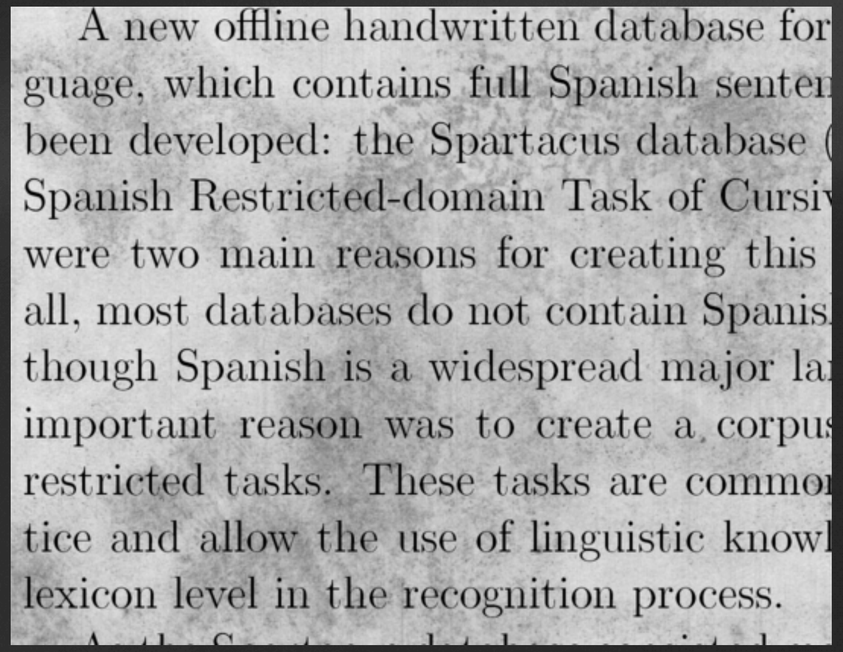
Project Vision & Purpose

- Text documents may consist of a noisy background that are difficult for computers and people to interpret.
- Need to consider noise variants commonly found in text documents.
- Reducing text document noise would greatly assist with:
  - Restoring old, distorted, and/or historical documents.
  - Increasing text visibility, which is especially advantageous for visually impaired individuals.
  - Useful for researching ideal image preprocessing techniques for all kinds of noise.



A .png text document file consisting of a coffee-stain like noise.

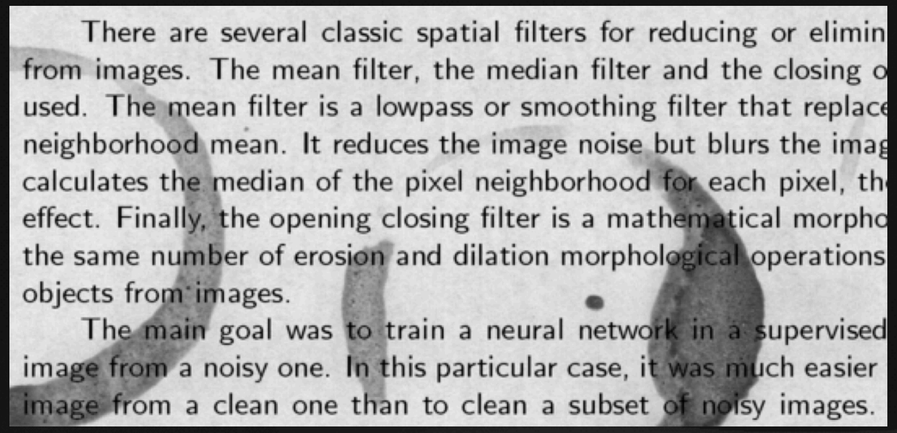
(Espaa-Boquera, Pastor-Pellicer, Castro-Bleda, & Zamora-Martinez, 2015)



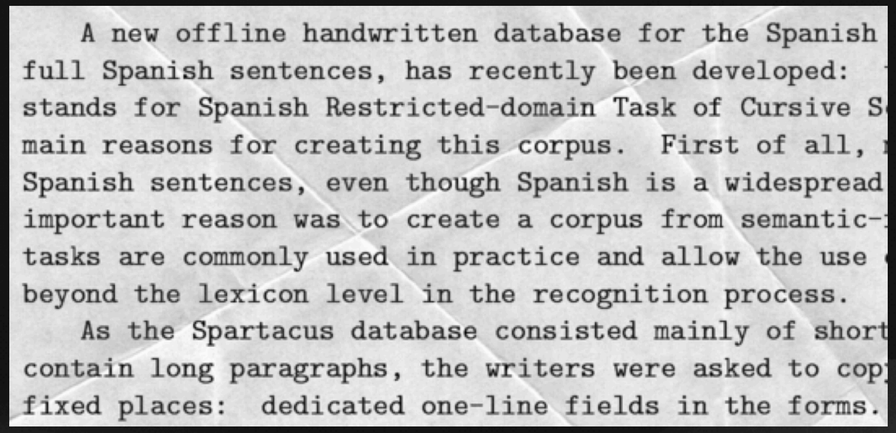
A .png text document file consisting of an eraser-mark like noise.

(Espaa-Boquera, Pastor-Pellicer, Castro-Bleda, & Zamora-Martinez, 2015)

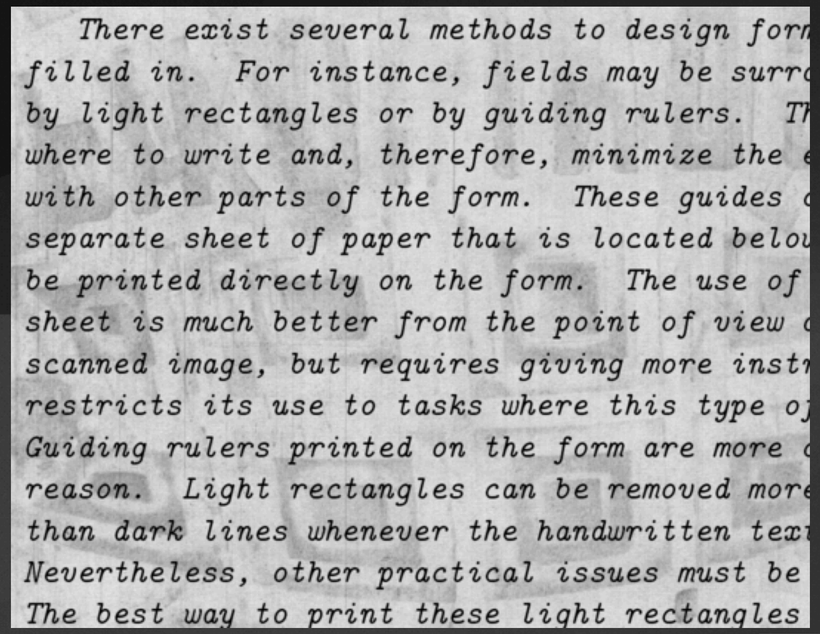
(1)



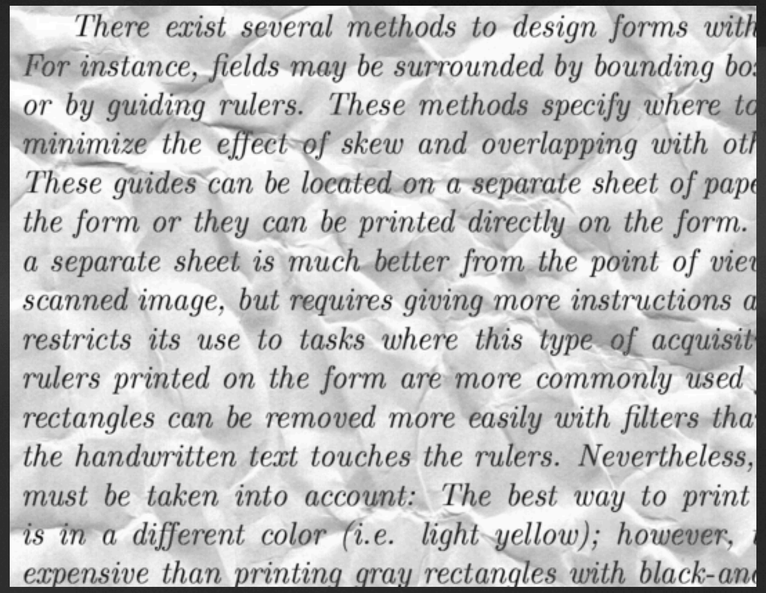
(2)



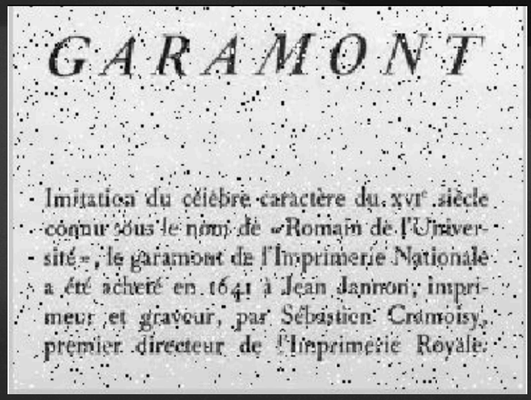
(3)



(4)



(5)



Noise Legend:

(1): Coffee-stained (2): Folded (3): Eraser-marked (4): Crumpled-up (5): Salt-and-pepper

(1), (2), (3), and (4): (Espaa-Boquera, Pastor-Pellicer, Castro-Bleda, & Zamora-Martinez, 2015)
(5): (Balamurugan, Sengottuvelan, & Sangeetha, 2013)

General Project Algorithm

While the user has not quit the program:

1: The user selects one of the subprograms.

While the user has not quit the subprogram:

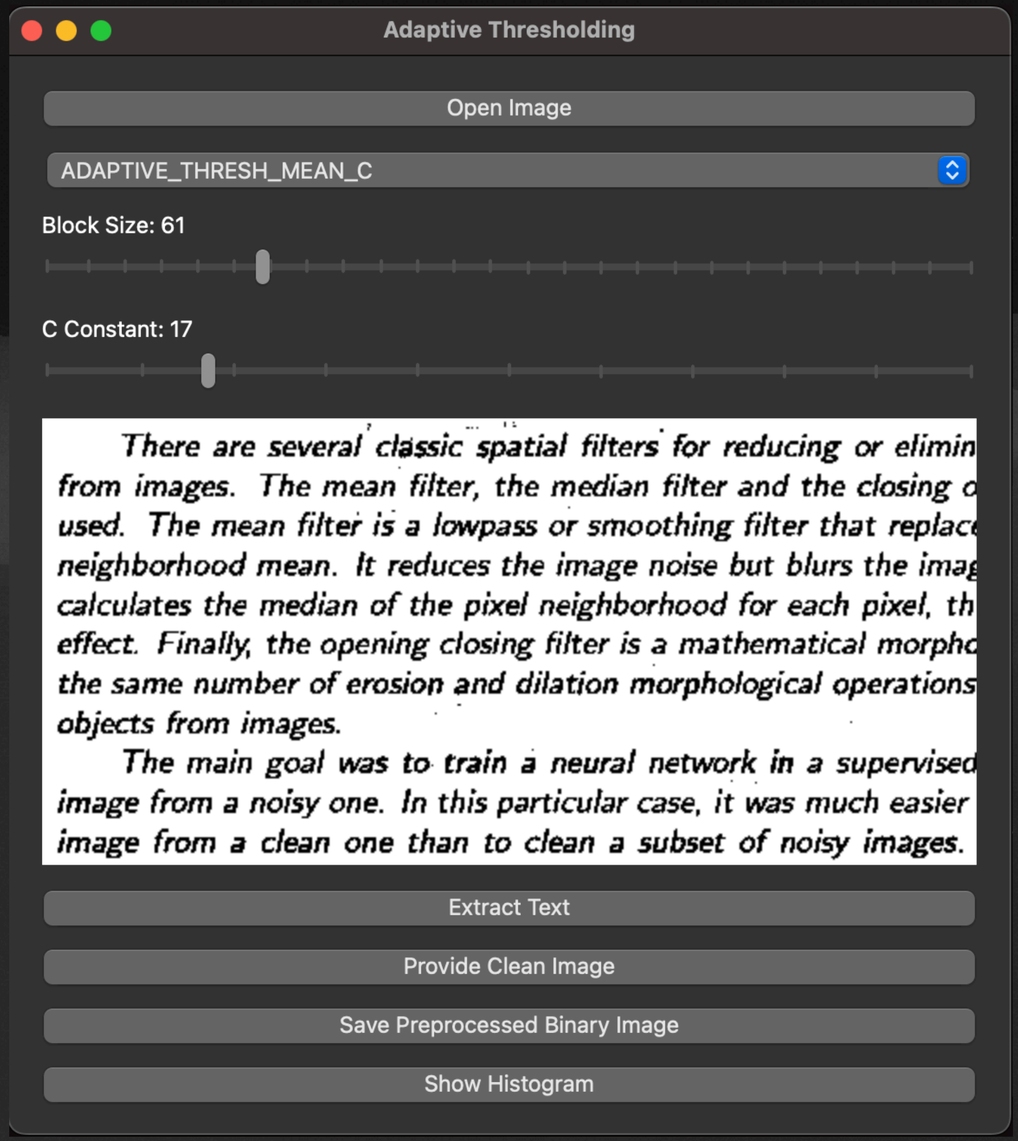
2: The user provides an image (preferably a text document).

3: The grayscale version of the image is then calculated.

4: Allow the user to change relevant parameters for the chosen subprogram, if any.

5: When requested by the user, perform the desired preprocessing operation, or a variant of the method if necessary, on the grayscale image with OpenCV, which will then be converted into a binary image to display to the user.

6: Dynamically update the displayed image based on changes to the parameters.



Text Extraction Accuracy Algorithm

Levenshtein Distance

1: Extract the text from the noisy and clean images.

2: Initialize a matrix with (number of words from the noisy image + 1) rows and (number of words from the clean image + 1) columns.

For each word in the noisy image:

    For each word in the clean image:

        If on the first row or the first column:

            3: Set the base distances that indicate the cost to get from a empty string to the current string.

        Otherwise, if the current noisy word is the same as the clean word:

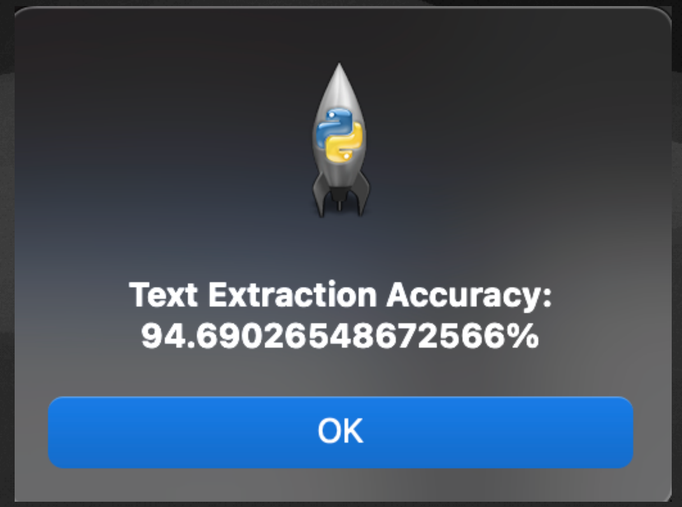
            4: Keep the distance the same.

        Otherwise:

            5: Check the distance for previously calculated neighbors and find the smallest distance + 1.

6: Set accuracy for the noisy image as ((max_distance - calculated_distance) / max_distance) * 100)

    max_distance: The length of the extracted text string from either the noisy or clean image, whichever is longer.
    calculated_distance: The cost calculated by the algorithm, which is found in the last row and column of the matrix.



“Get Best Image” Algorithm

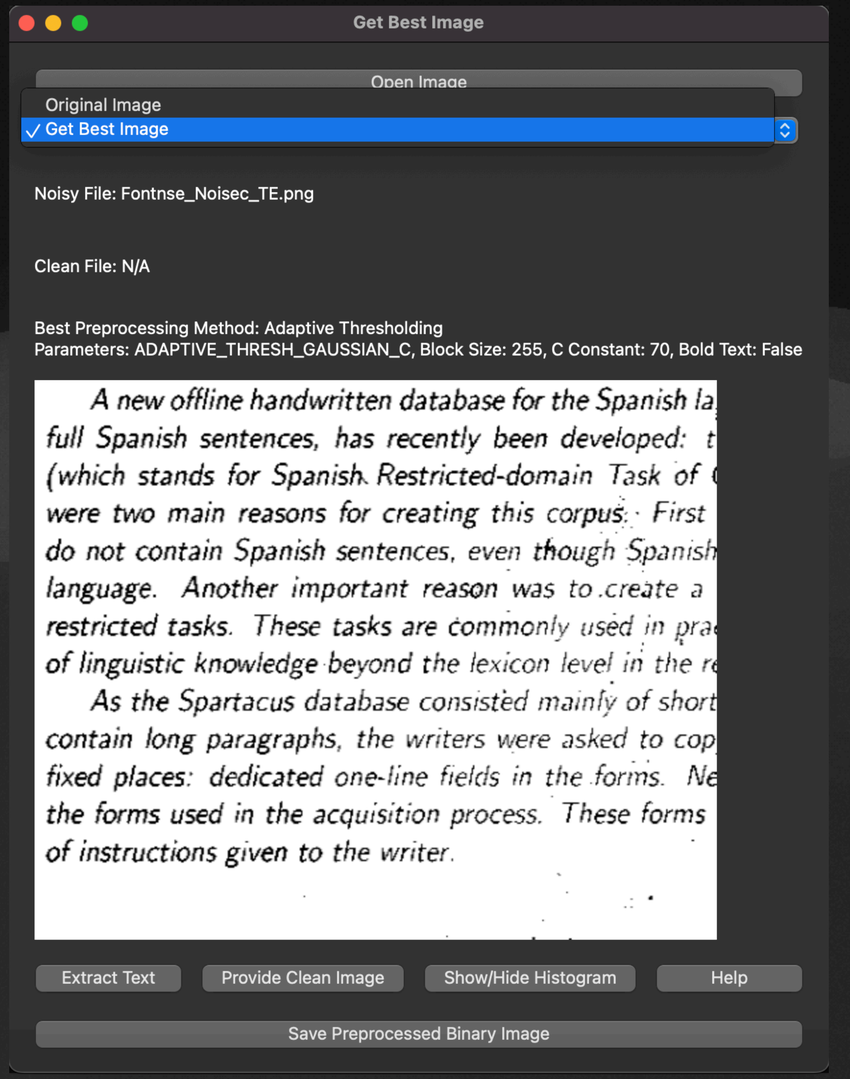
While the user has not quit the “Get Best Image” subprogram:

1: The user provides an image (preferably a text document).

2: The grayscale version of the image is then calculated.

3: When the user wants to begin the preprocessing, send the grayscale image through several effective preprocessing methods that are the most likely to reduce background noise from the image.

4: Display the binary image that Tesseract believes is the most recognizable, not essentially the cleanest image.



Required Algorithm Adjustments for Implementation

- General Purpose
  - Limitations are set on which preprocessing parameters are user-adjustable, as well as what values they can be.
    - Example: Only odd kernel sizes allowed in adaptive thresholding, median filtering, and Gaussian blur.
- Text Extraction Accuracy
  - No changes required.
- “Get Best Image”
  - Only considers preprocessed images where its text string length is at least 90% of that of the original noisy text string.
    - The loss of text was occasionally preventing the project from achieving its intended purpose.
  - The effective (and tested) preprocessing methods used for this subprogram are Otsu’s, binary, and adaptive thresholding, as well as median filtering.

Additional Features

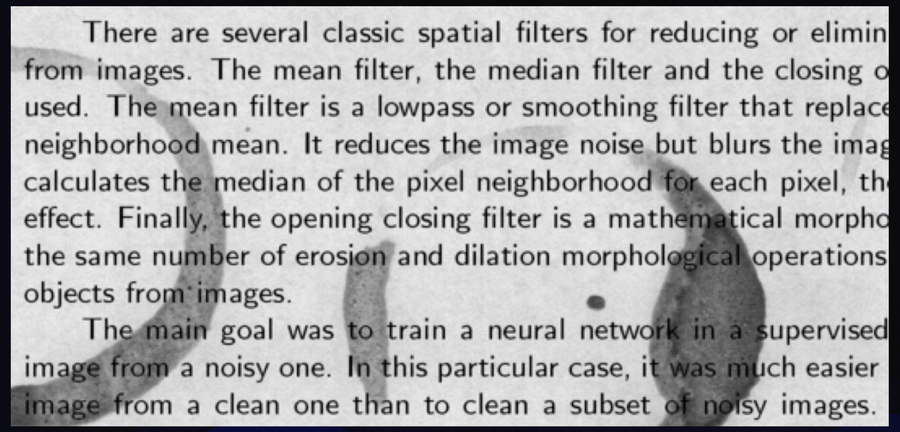
- In each of the subprograms, the user may also:
  - Save a preprocessed binary image.
  - Extract text with Tesseract.
  - Provide a clean version of the noisy image.
    - A text extraction accuracy can then be calculated between the noisy and clean text strings.
    - The “Get Best Image” subprogram will then rely on text extraction accuracies instead of Tesseract’s word confidence percentages.
  - Obtain and save the histogram for the current noisy image, whether preprocessed or not.
  - Bold the text of the current noisy image (Performs foreground dilation with a 2x2 matrix).
    - Not available in the “Get Best Image” subprogram.
  - If Otsu’s thresholding is selected, provide two directories, one clean and one noisy, and then retrieve a bulk text extraction accuracy.
  - Request help about when and how to use the subprogram.

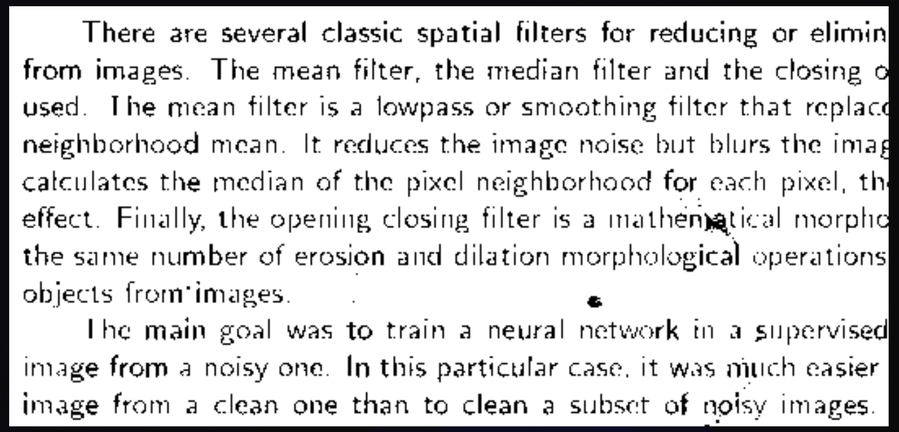
Error Handling & Quality Assurance

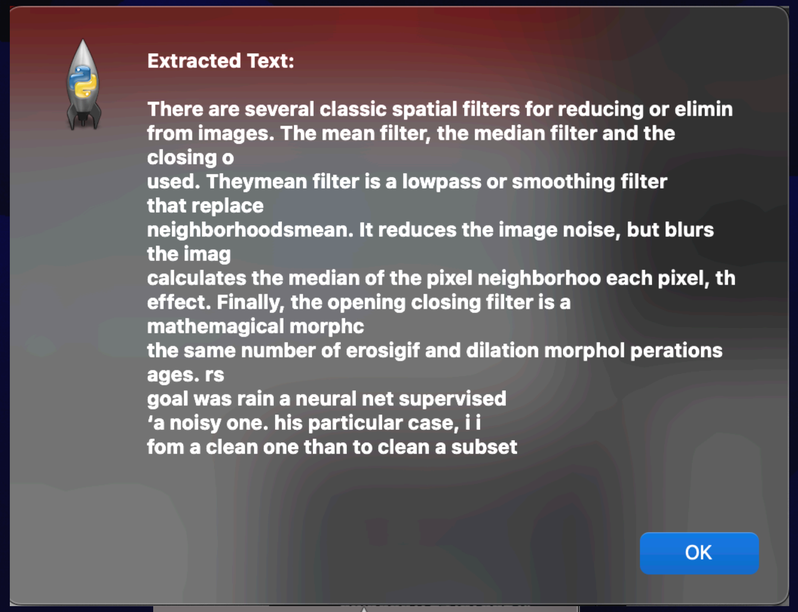
- Only OpenCV compatible formats allowed (.png, .tiff, .jpeg, .jpg, and .bmp).
- In the “Get Best Image” subprogram, the supplied image must have extractable text.
- GUI is dynamically updated with respect to the displayed images.
- Images that are too large are compressed to a size that allows for GUI reachability.
    - All preprocessing and calculations are done with the original image dimensions.

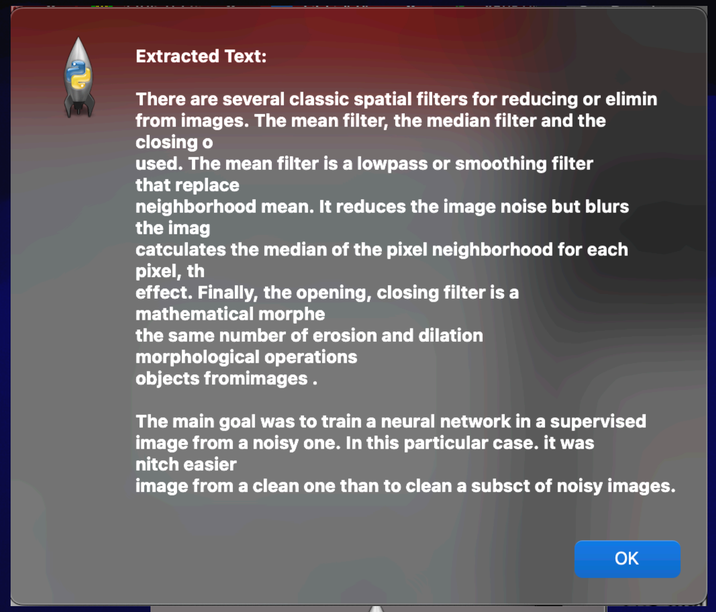
Project Performance!

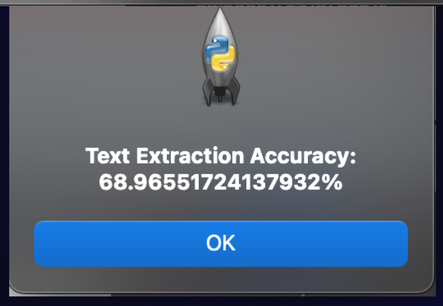
(1): Coffee-stained

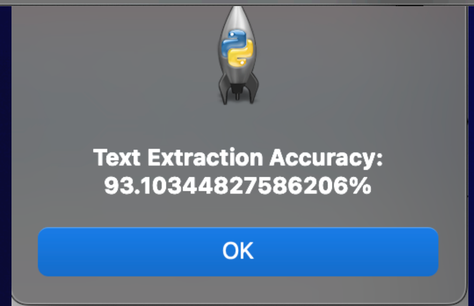




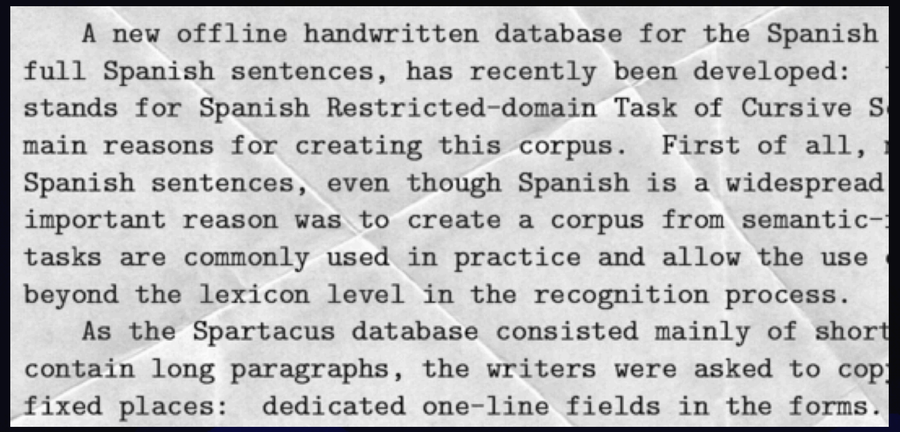


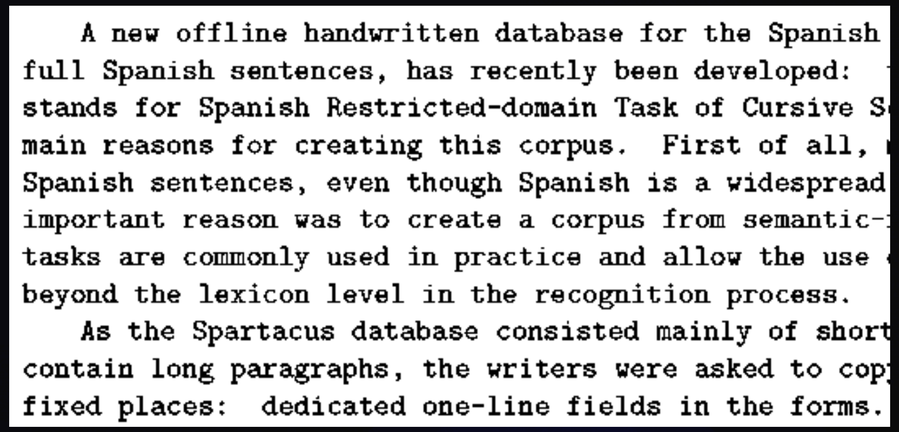


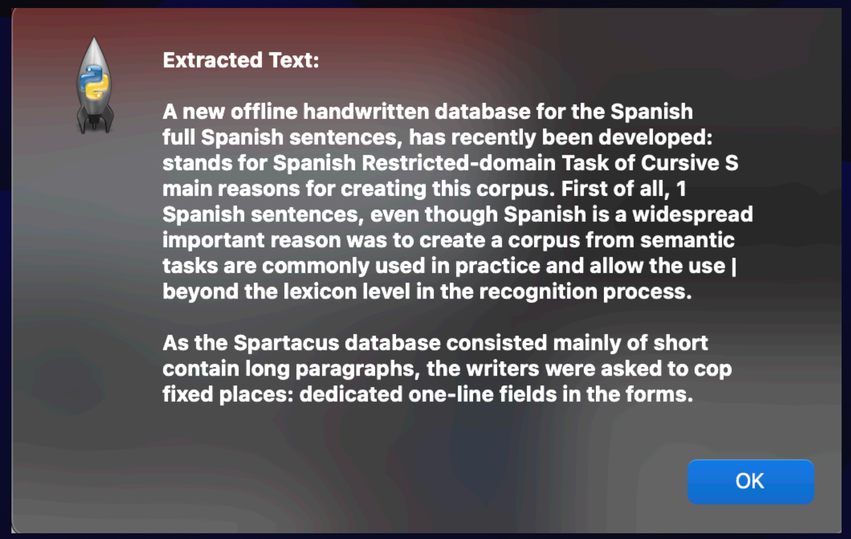


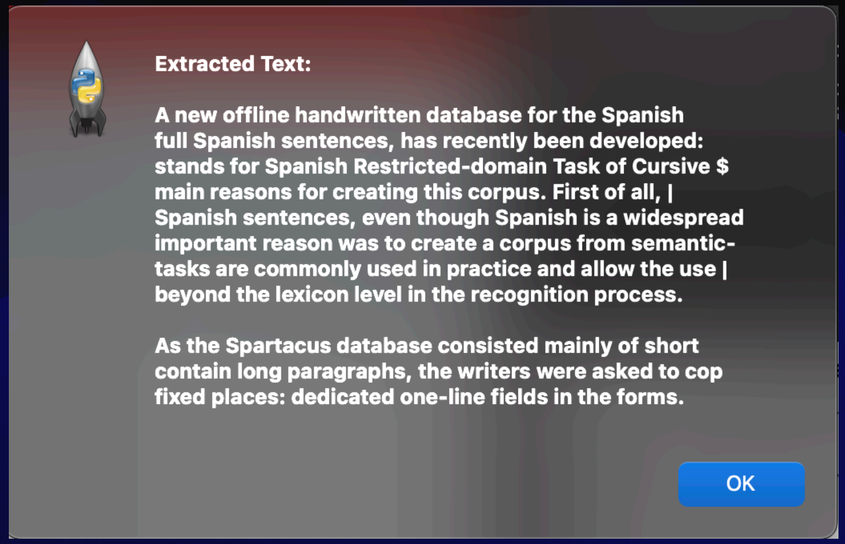


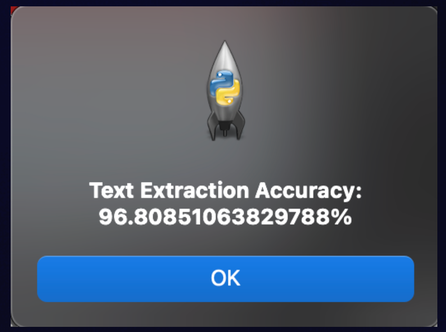
(2): Folded

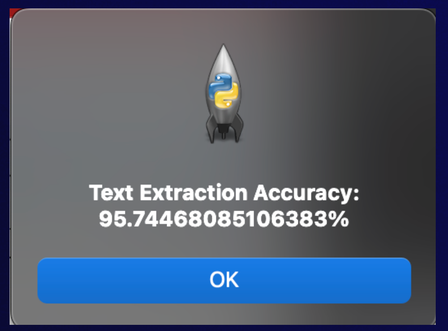

A new offline handwritten database for the Spanish
full Spanish sentences, has recently been developed:
stands for Spanish Restricted-domain Task of Cursive S
main reasons for creating this corpus. First of all,
Spanish sentences, even though Spanish is a widespread
important reason was to create a corpus from semantic-
tasks are commonly used in practice and allow the use
beyond the lexicon level in the recognition process.
As the Spartacus database consisted mainly of short
contain long paragraphs, the writers were asked to cop
fixed places: dedicated one-line fields in the forms.


A new offline handwritten database for the Spanish
full Spanish sentences, has recently been developed:
stands for Spanish Restricted-domain Task of Cursive S
main reasons for creating this corpus. First of all,
Spanish sentences, even though Spanish is a widespread
important reason was to create a corpus from semantic-
tasks are commonly used in practice and allow the use
beyond the lexicon level in the recognition process.
As the Spartacus database consisted mainly of short
contain long paragraphs, the writers were asked to cop
fixed places: dedicated one-line fields in the forms.

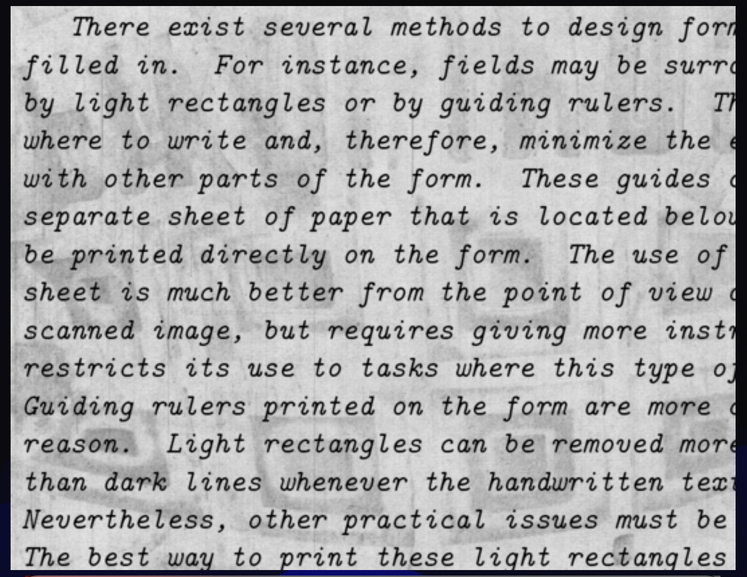

Extracted Text:
A new offline handwritten database for the Spanish
full Spanish sentences, has recently been developed:
stands for Spanish Restricted-domain Task of Cursive S
main reasons for creating this corpus. First of all, 1
Spanish sentences, even though Spanish is a widespread
important reason was to create a corpus from semantic
tasks are commonly used in practice and allow the use |
beyond the lexicon level in the recognition process.
As the Spartacus database consisted mainly of short
contain long paragraphs, the writers were asked to cop
fixed places: dedicated one-line fields in the forms.
OK

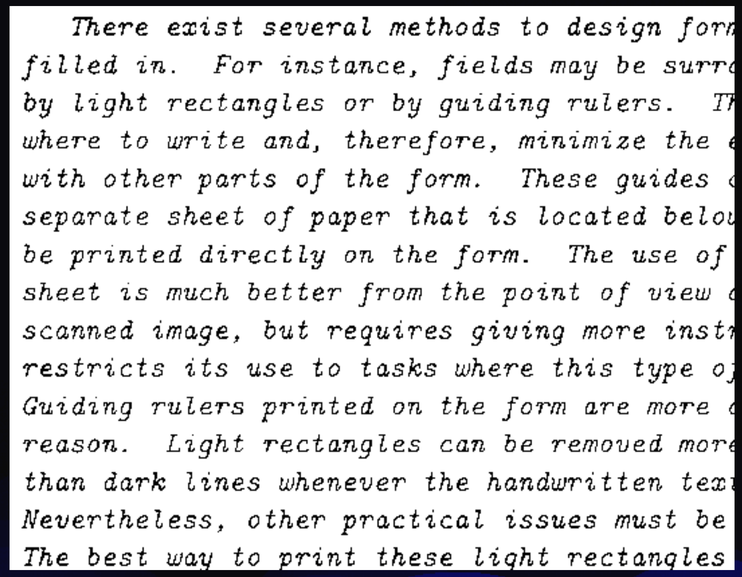

Extracted Text:
A new offline handwritten database for the Spanish
full Spanish sentences, has recently been developed:
stands for Spanish Restricted-domain Task of Cursive $
main reasons for creating this corpus. First of all, |
Spanish sentences, even though Spanish is a widespread
important reason was to create a corpus from semantic-
tasks are commonly used in practice and allow the use |
beyond the lexicon level in the recognition process.
As the Spartacus database consisted mainly of short
contain long paragraphs, the writers were asked to cop
fixed places: dedicated one-line fields in the forms.
OK

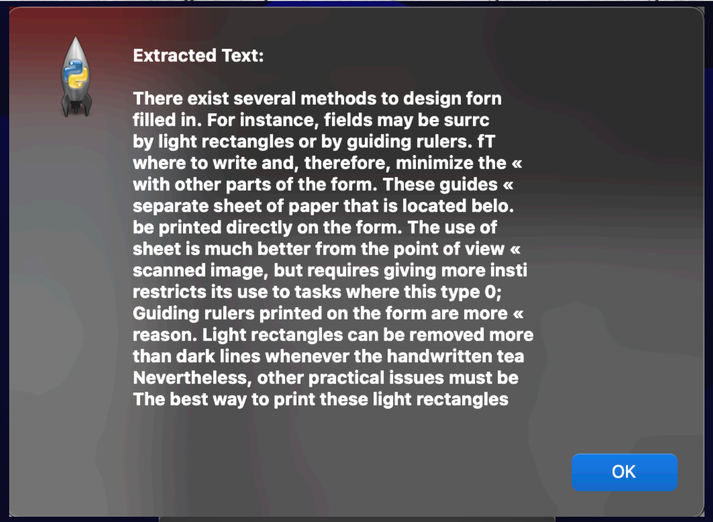

Text Extraction Accuracy:
96.80851063829788%
OK

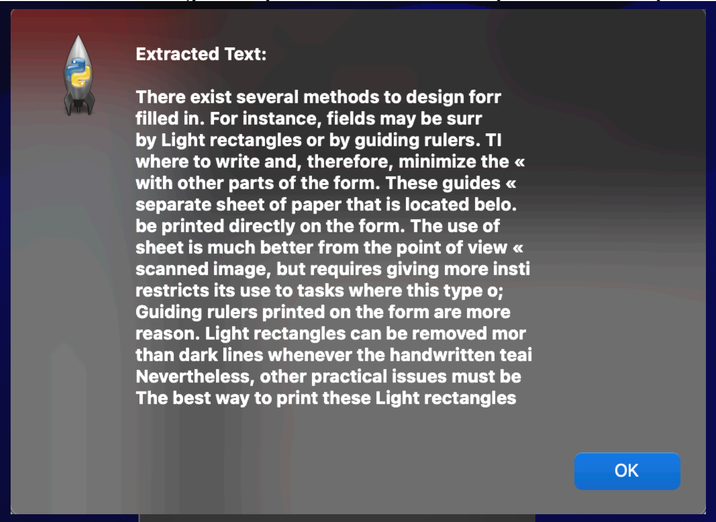

Text Extraction Accuracy:
95.74468085106383%
OK

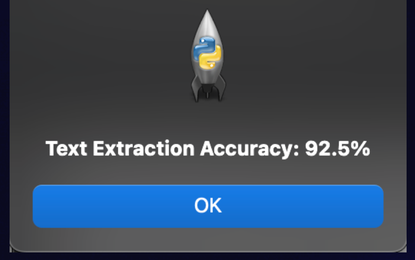
(3): Eraser-marked

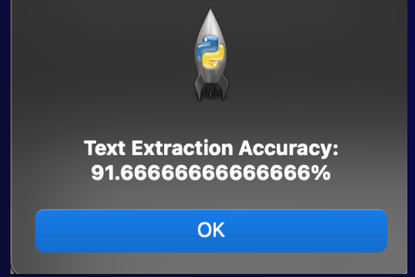




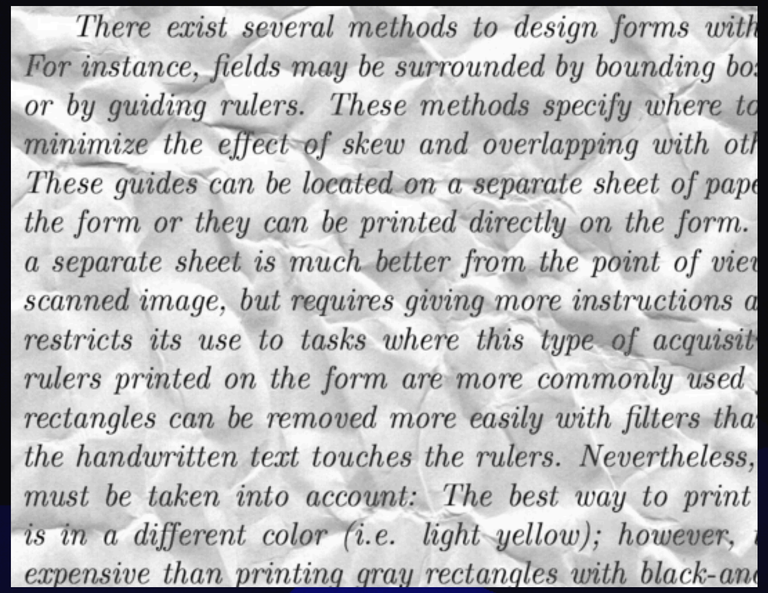


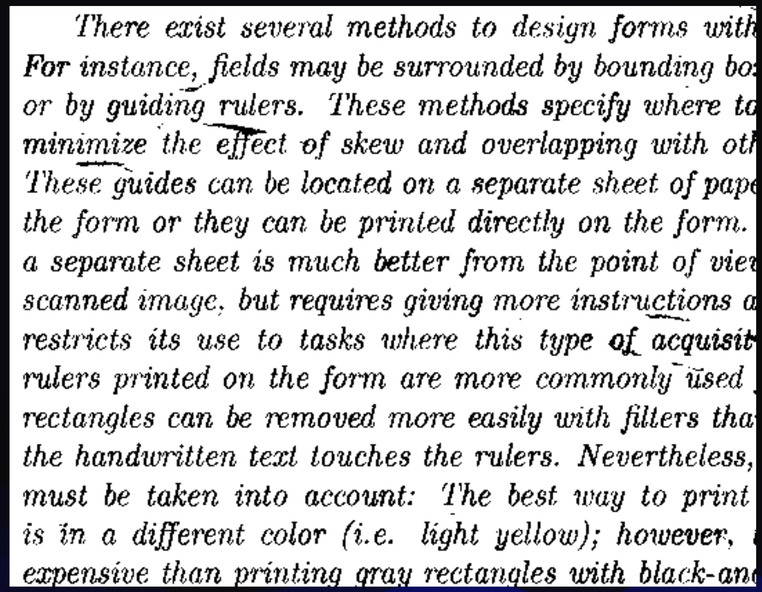


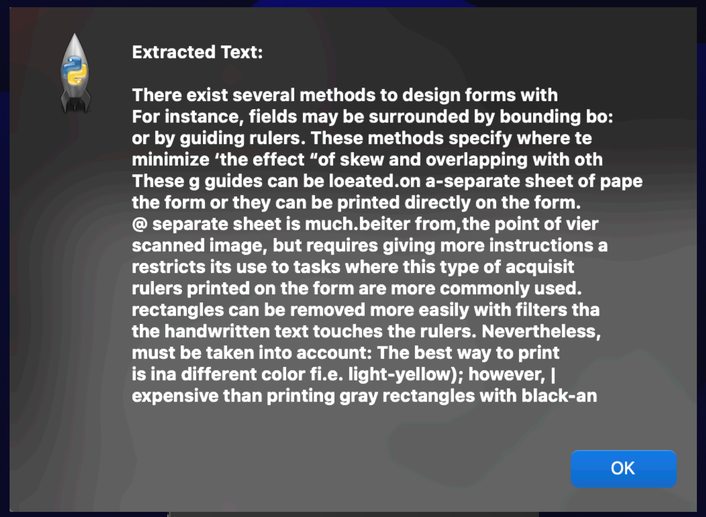


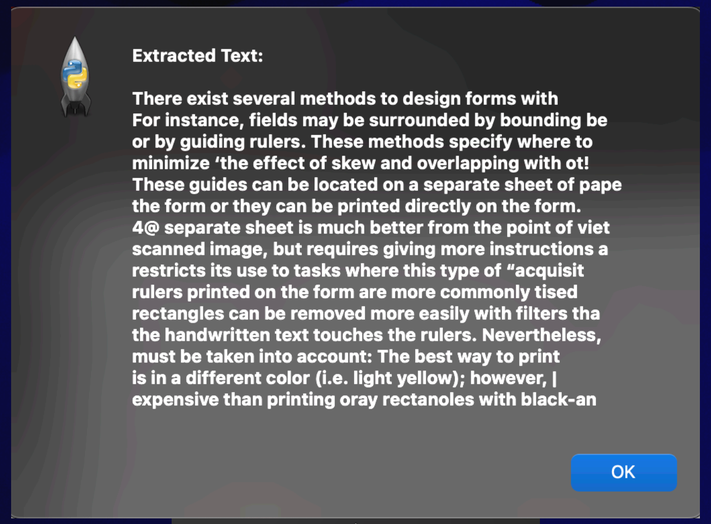


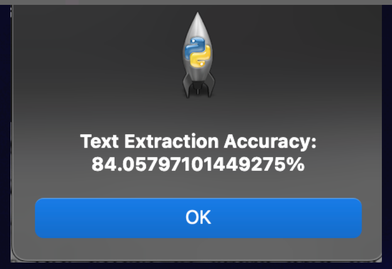
(4): Crumpled-up

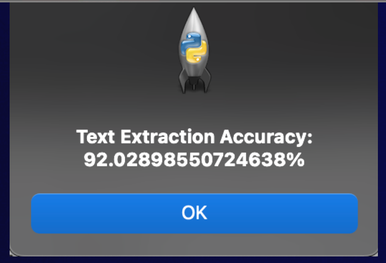




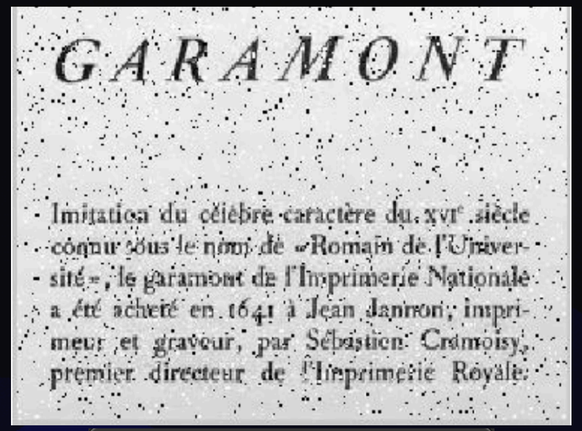


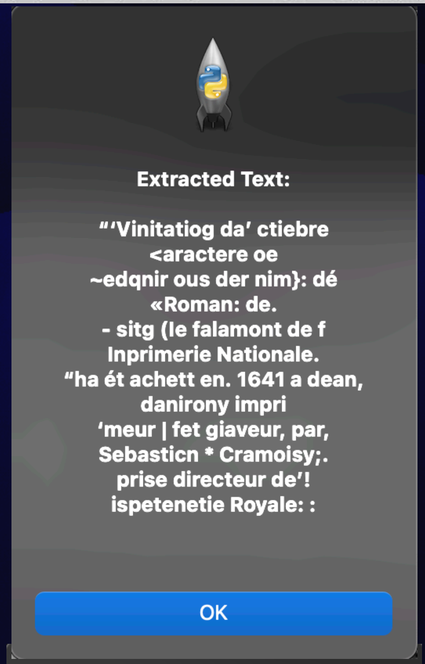


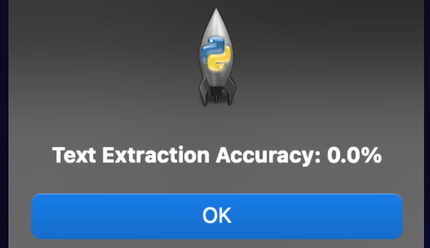


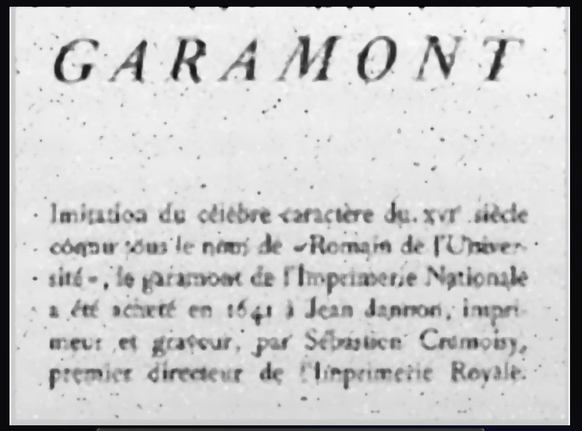


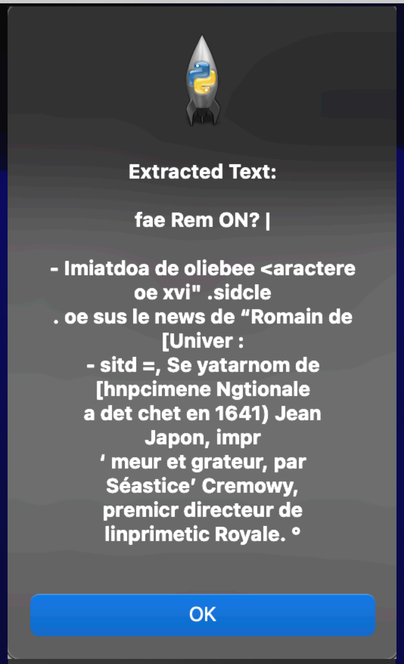
(5): Salt-and-pepper

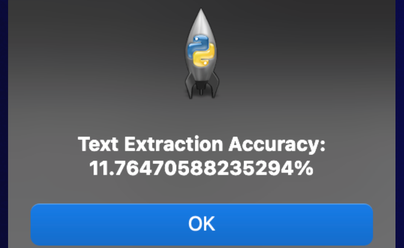


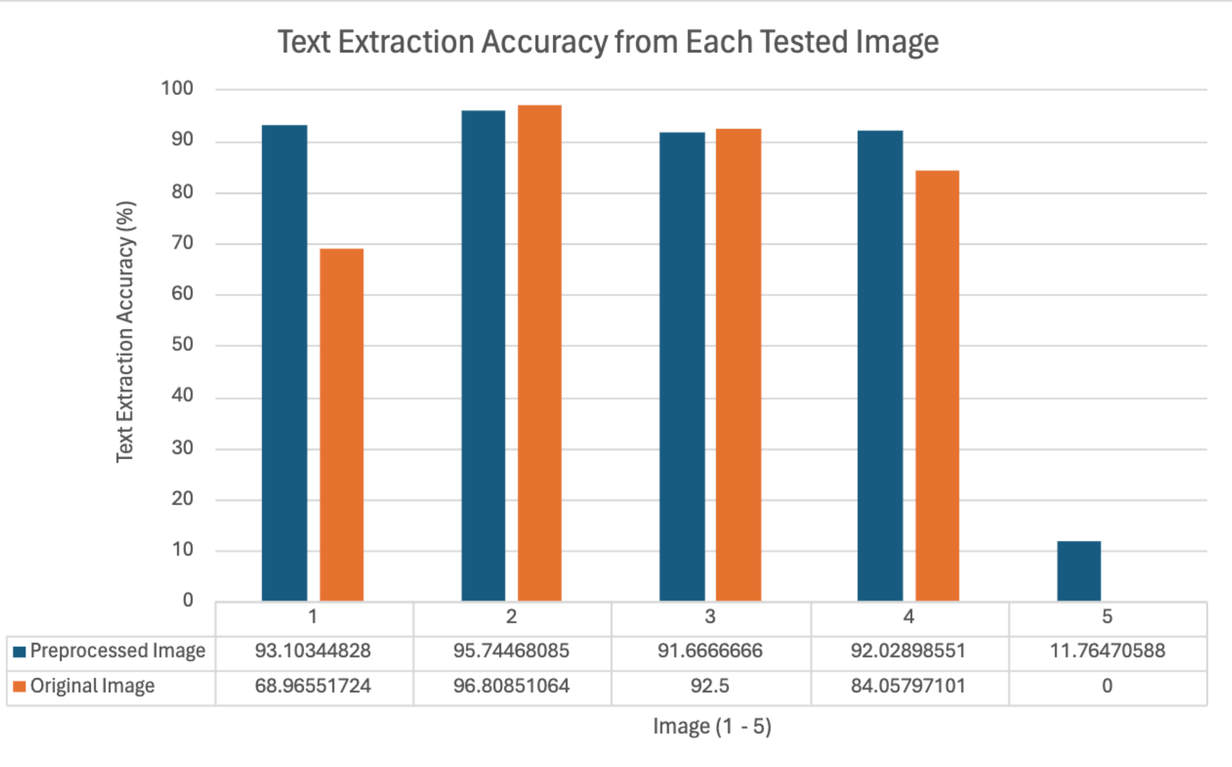




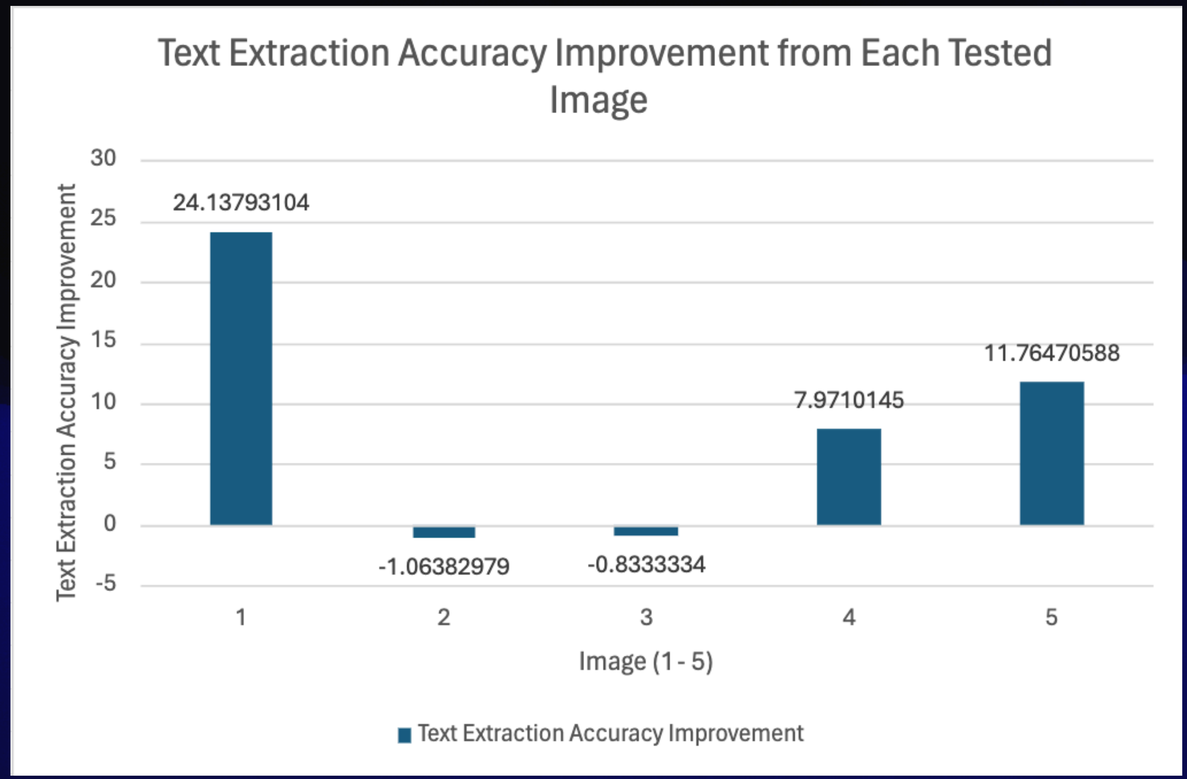








	1	2	3	4	5
Preprocessed Image	93.10344828	95.74468085	91.6666666	92.02898551	11.76470588
Original Image	68.96551724	96.80851064	92.5	84.05797101	0



Discussion of Results & What's Next

- Light noise types (Folding creases and eraser marks) resulted in Tesseract becoming less confident about the calculated preprocessed images, ultimately leading to a lower text extraction accuracy.
- On the other hand, heavy types of noise (Coffee stains, crumples, and salt-and-pepper) got Tesseract more confident in the resulting preprocessed images.
- Tesseract is not essentially grading on user-readability, but rather computer-readability.
- There is no "one-size-fits-all" solution to image processing, it's all about trial and error.
  - Encouraged me to combine individual solutions together to create a complete solution.
- The nature of this project, along with the skills that were learned, will be appreciated in many other image processing fields and projects.

References

Balamurugan, E., Sengottuvelan, P., & Sangeetha, K. (2013, November). *An Empirical Evaluation of Salt and Pepper Noise Removal for Document Images using Median Filter*. research.ijcaonline.org. https://research.ijcaonline.org/volume82/number4/pxc3892139.pdf

Brownlee, J. (2022, September 12). *Number of CPUs in Python*. SuperFastPython.com. https://superfastpython.com/number-of-cpus-python/#:~:text=We%20can%20get%20the%20count%20of%20the%20number%20of%20CPUs,will%20return%20the%20value%20None.

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/~krisana/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

Espaa-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

Huamán, A. (n.d.). *Histogram Equalization*. Docs.OpenCV.org. *https://docs.opencv.org/3.4/d4/d1b/tutorial_histogram_equalization.html*

Hoffstaetter, S. et al., (2022, August 16). *pytesseract 0.3.10*. PyPI.org. https://pypi.org/project/pytesseract/

Jayanetti, 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=tYF3EBkvYO0&t=1s

Nurfikri, 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*

Python Software Foundation. (2024, April 16). threading - Thread-based parallelism. docs.python.org. https://docs.python.org/3/library/threading.html#lock-objects

The Qt Company. (2024). *QFileDialog Class*. doc.qt.io. https://doc.qt.io/qt-6/qfiledialog.html

Rosebrock, A. (2021, April 28). *OpenCV Image Histograms (cv2.calcHist)*. PyImageSearch.com. https://pyimagesearch.com/2021/04/28/opencv-image-histograms-cv2-calchist/

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/