CZ4003 Computer Vision

Project Report: Text Image Segmentation for Optimal Optical Character Recognition

Chulpaibul Jiraporn, U1822666H

Deng Jinyang, U1820375G

7 November 2020

Table of Contents

Member	Contributions
Chulpaibul Jiraporn	Adaptive thresholding, Background removal, Bounding box adjustment
Deng Jinyang	Otsu algorithm, Background removal, Location-aware correction, A search-based re-processing approach

Part 1

Implement the Ostu global thresholding algorithm for binarizing the sample text images and feed the binarized images to the OCR software to evaluate the OCR accuracy. Discuss any problems with the Otsu global thresholding algorithm.

Code (in Python 3)**:**

```
def otsu_algo(image):
    total_pixel_count = image.flatten().shape[0]
    total_var = image.flatten().var()
    hist = [np.sum(image == i) for i in range(0, 256)]
    hist_prob = [_ / total_pixel_count for _ in hist]
    intra_class_vars, inter_class_vars = [],[]

    for i in range(0, 256):
        lower_prob, higher_prob = sum(hist_prob[:i]), sum(hist_prob[i:])

        lower_mean = 1/lower_prob * sum([j * hist_prob[j] for j in range(0, i)]) if lower_prob > 0 else 0
        higher_mean = 1/higher_prob * sum([j * hist_prob[j] for j in range(i, 256)]) if higher_prob > 0 else 0

        lower_var = 1/lower_prob * sum([(j - lower_mean)**2 * hist_prob[j] for j in range(0, i)]) if lower_prob > 0 else 0
        higher_var = 1/higher_prob * sum([(j - higher_mean)**2 * hist_prob[j] for j in range(i, 256)]) if higher_prob > 0 else 0

        intra_class_var = lower_prob * lower_var + higher_prob * higher_var
        inter_class_var = lower_prob * higher_prob * (lower_mean - higher_mean)**2
        assert round(inter_class_var + intra_class_var, 7) == round(total_var, 7)
        intra_class_vars.append(intra_class_var)
        inter_class_vars.append(inter_class_var)

    return(intra_class_vars, inter_class_vars)
```

```
def apply_thresh_on_image(image, thresh):
    new_image = image.copy()
    new_image[image > thresh] = 255
    new_image[image <= thresh] = 0
    return new_image

def process_image(image):
    intra_class_vars, inter_class_vars = otsu_algo(image)
    assert intra_class_vars.index(min(intra_class_vars)) == inter_class_vars.index(max(inter_class_vars))
    thresh = intra_class_vars.index(min(intra_class_vars)) - 1
    processed_image = apply_thresh_on_image(image, thresh)
    return processed_image
```

Implementation & Explanation:

Note: For brevity but without loss of generality, we will use Sample 1 here for illustration. Nonetheless, we will cover both Samples 1 and 2 in the results section.

We start by constructing a histogram for the input image, that tabulates the number of pixels belonging to each grey level in the range [0, 255], such as the one shown below:

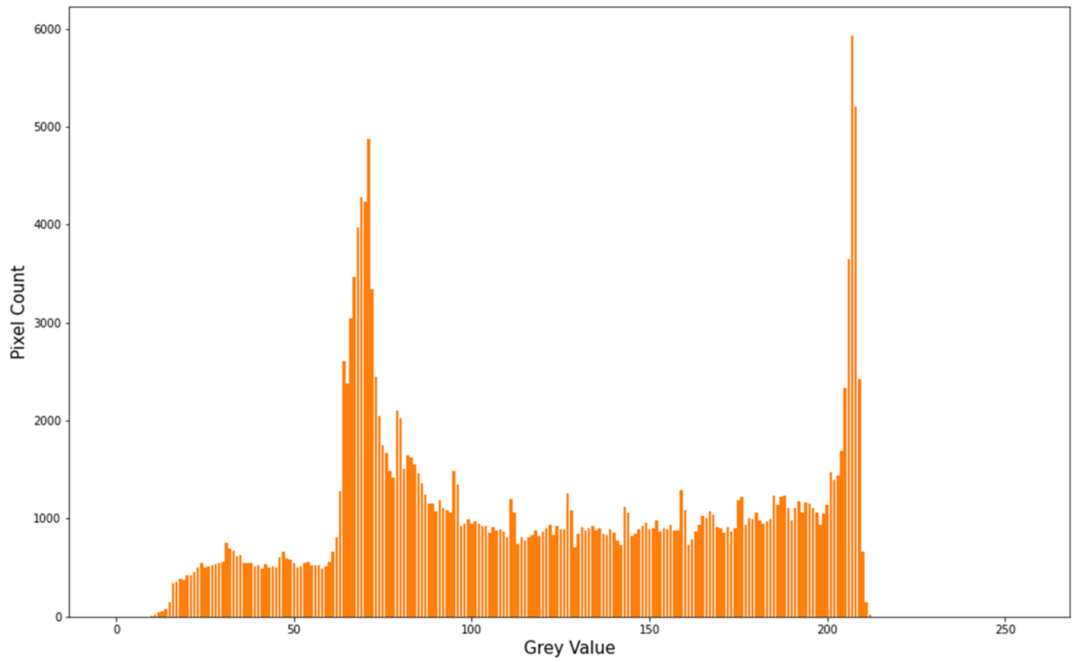


Figure 1 Gray Value Histogram (Sample 1)

For efficiency, we will also pre-compute a relative frequency histogram as such:

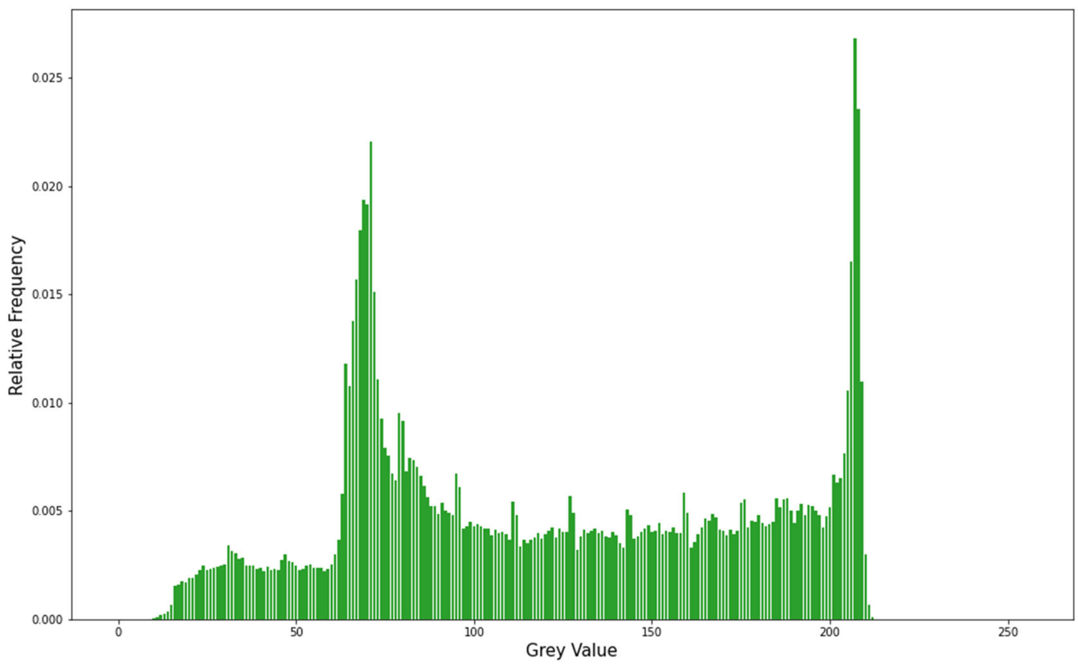


Figure 2 Gray Value Relative Frequency Histogram (Sample 1)

Next, the Otsu algorithm will iterate through all possible threshold values in the range [0, 255], and continuously calculate the intra-class variance and the inter-class variance using the following equations:

$$Intra - class\ variance: \sigma_w^2(t) = \frac{1}{\omega_H} * \sigma_H^2(t) * \frac{1}{\omega_L} * \sigma_L^2(t)$$

$$Inter - class\ variance: \sigma_b^2(t) = \omega_H(t) * \omega_L(t) * [\mu_H(t) - \mu_L(t)]^2$$

$$Higher\ group\ probability: \omega_H(t) = \sum_{r=t+1}^{255} p(r)$$

$$Lower\ group\ probability: \omega_L(t) = \sum_{r=0}^{t} p(r)$$

$$Higher\ group\ variance: \sigma_L^2(t) = \frac{1}{\omega_H(t)} * \sum_{r=t+1}^{255} [(r - \mu_H(t))^2 * p(r)]$$

$$Lower\ group\ variance: \sigma_L^2(t) = \frac{1}{\omega_L(t)} * \sum_{r=0}^{t} [(r - \mu_L(t))^2 * p(r)]$$

$$Higher\ group\ mean: \mu_H(t) = \frac{1}{\omega_H(t)} * \sum_{r=t+1}^{255} [r * p(r)]$$

$$Lower\ group\ mean: \mu_L(t) = \frac{1}{\omega_L(t)} * \sum_{r=0}^{t} [r * p(r)]$$

And *t* is threshold, and *p(r)* is relative frequency of grey value *r* (as computed earlier)

The procedures mentioned above have been implemented in *otsu_algo()*.

After collecting the variance metrics, we can plot them out in the following chart:

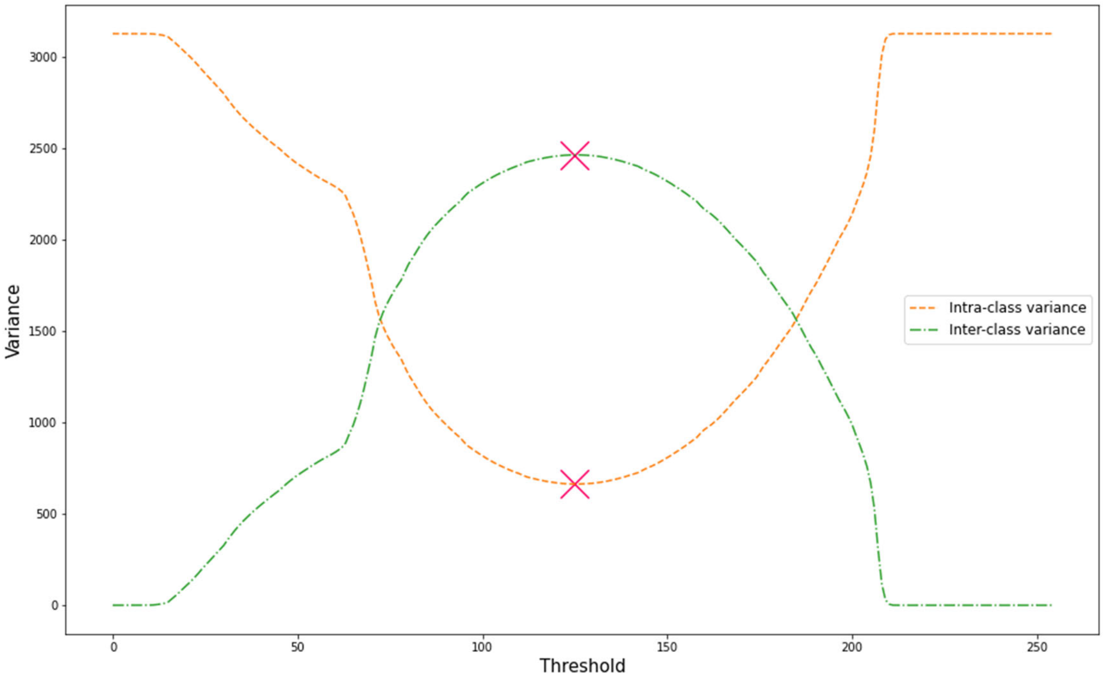


Figure 3 Intra-class and Inter-class Variances (Sample 1)

Note that while intra-class variance and inter-class variance are calculated independently, they always sum to the total variance, which is 3126.648 in this case.

The Otsu algorithm will then select the optimal threshold value (marked out above in red), where the inter-class variance is maximized, and the intra-class variance is minimized. In the case of Sample 1, the optimal threshold value is 125. For Sample 2, we can repeat the process and obtain an optimal threshold value of 141.

The selection of optimal threshold value is implemented in line 2 - 3 of *process_image()*.

To process the image, we will apply binarization in the following manner:

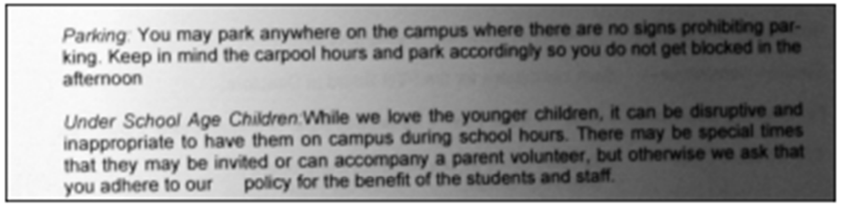
$$r_{new} = \begin{cases} 255 & if\ r_{original} \geq t \\ 0 & if\ r_{original} < t \end{cases}$$

This image binarization procedure is implemented in *apply_thresh_on_image()*.

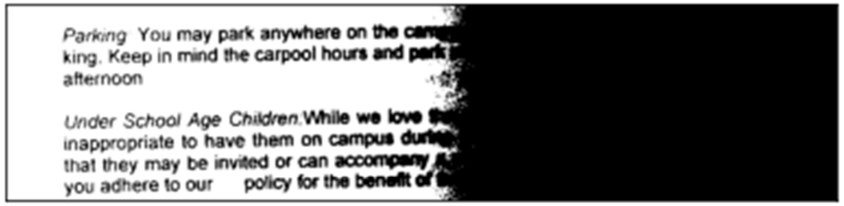
Results & Analysis:

After applying the Otsu algorithm and image binarization, we obtained the following:

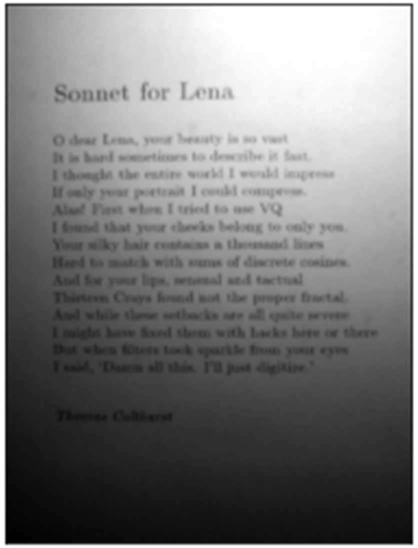
Sample 1 (Before)



Sample 1 (After)



Sample 2 (Before)



Sample 2 (After)

As evident from above, the Otsu algorithm has worked out poorly for both sample images. Instead of separating the foreground text from the background, it has increased contrast in a way that made lighter regions much lighter, while darker regions are completely black.

With at least half of the image completely black, we would not expect OCR to perform so well, and that was indeed the case. In fact, by applying the Otsu algorithm, we have adversely affected the OCR results.

Sample 1: Before Otsu vs After Otsu

Parking: You may park anywhere on the ce king. Keep in mind the carpool hours and park afternoon Under School Age Children:While we love inappropriate to have them on campus @) that they may be invited or can accompany : you adhere to our _ policy for the benefit of	Parking You may park anywhere on the cf king. Keep in mind the carpool hours and peri, afternoon Under School Age Children:While we love inappropriate to have them on campus @ i that they may be invited or can accompany J you adhere to our —_policy for the benefit of

Sample 2: Before Otsu vs After Otsu

Sonnet for Lena	Sonnet for ler

As a global thresholding technique, one pre-requisite for the Otsu algorithm is that “objects of different classes occupy a separable range in the gray-level histogram”[1], which is clearly not the case here. The images have been taken under non-uniform illumination, which resulted in uneven shading. In fact, if we looked more closely, we could see that in the original Sample 2, the text in the top right corner, i.e. “vast” has an gray level of about 180 (> threshold of 141, so it became white), which is higher than that of the “Thomas” in the bottom left corner, which has an gray level of about 50 (< threshold of 141, so it became black). Additionally, while the foreground texts can be discernibly darker than the background in some brightly-lit regions, this difference would get significantly diminished in other, darker regions.

With no clear separation between the foreground text and background in both of our 2 sample images, any attempt to find a global threshold and subsequently using for image binarization is akin to an “one-size-fit-all” approach that will not work.

[1] Don, H. A noise attribute thresholding method for document image binarization. *IJDAR* **4,** 131–138 (2001). https://doi.org/10.1007/s100320100062

Part 2

Design your own algorithms to address the problem of Otsu global thresholding algorithm, and evaluate OCR accuracy for the binary images as produced by your algorithms. You may explore different approaches such as adaptive thresholding, image enhancement, etc., and the target is to achieve the best OCR accuracy.

Analysis

From Part 1, we observed how the Otsu algorithm still performs poorly despite a bimodal image histogram. This is because the underlying assumption of the Otsu algorithm is that the foreground text and the background will fall under two non-overlapping ranges of grey levels.

However, this is clearly not the case here. In Sample 2 for instance, the lower half of the image is dark, and joins the foreground text near the same mode in the histogram. When the Otsu algorithm sets all pixels with grey levels below the global threshold to 0, the entire lower region, along with most of the foreground text, turns black.

To address this problem, we will introduce the following 3 different possible image enhancement methods, and follow up with a discussion of results and limitations:

1. Adaptive Gaussian Thresholding + Gaussian Blur
2. Background Layer Removal
3. A Simplified Approach to Background Layer Removal

Note: For brevity but without loss of generality, we will use Sample 2 here for illustration. Nonetheless, we will cover both Samples 1 and 2 in the results section.

A quick summary of our methods' accuracies is as follows:

Method	Sample 1	Sample 2
Adaptive Gaussian Thresholding + Gaussian Blur	99.61%	92.51%
Background Layer Removal	99.81%	99.35%
A Simplified Approach to Background Layer Removal	99.81%	99.18%

Method 1: Adaptive Gaussian Thresholding + Gaussian Blur

Code (in Python 3)**:**

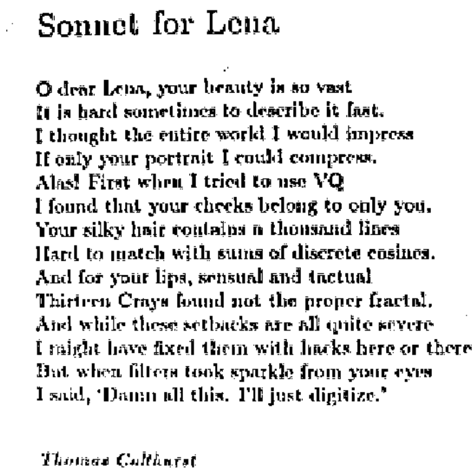
```
def method_1(image):
    # Step 1: Gaussian Adaptive Threshold
    step_1_result = cv2.adaptiveThreshold(image,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv
2.THRESH_BINARY,19,5)
    # Step 2: Gaussian Blur
    step_2_result = cv2.GaussianBlur(step_1_result,(3,3),3)

    return step_2_result
```

Implementation & Explanation:

As can be observed in the original images, the text is always darker than the immediate background. This suggests that we can partition the image into different parts and apply thresholding locally. Adaptive thresholding is the method where the threshold value is calculated for smaller, individual regions.

In the code above, we first take the Gaussian-weighted sums of 19 x 19 windows, before subtracting a constant of 5. After applying this step, we can obtain Figure 4.

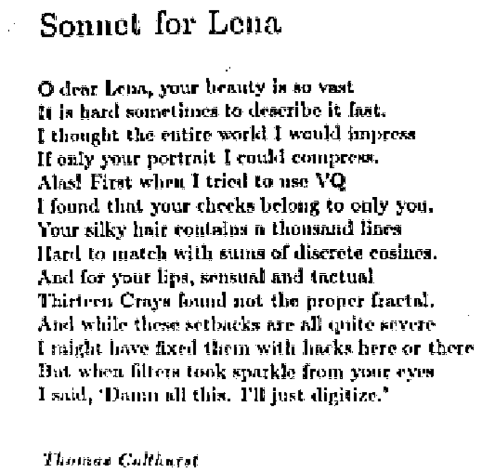
Next, we apply a Gaussian blur to remove the high-frequency components that are ‘cutting’ some part of the character. The result can be seen in Figure 5.



Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, ‘Damn all this. I'll just digitize.’

Thomas Colthurst



Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, ‘Damn all this. I'll just digitize.’

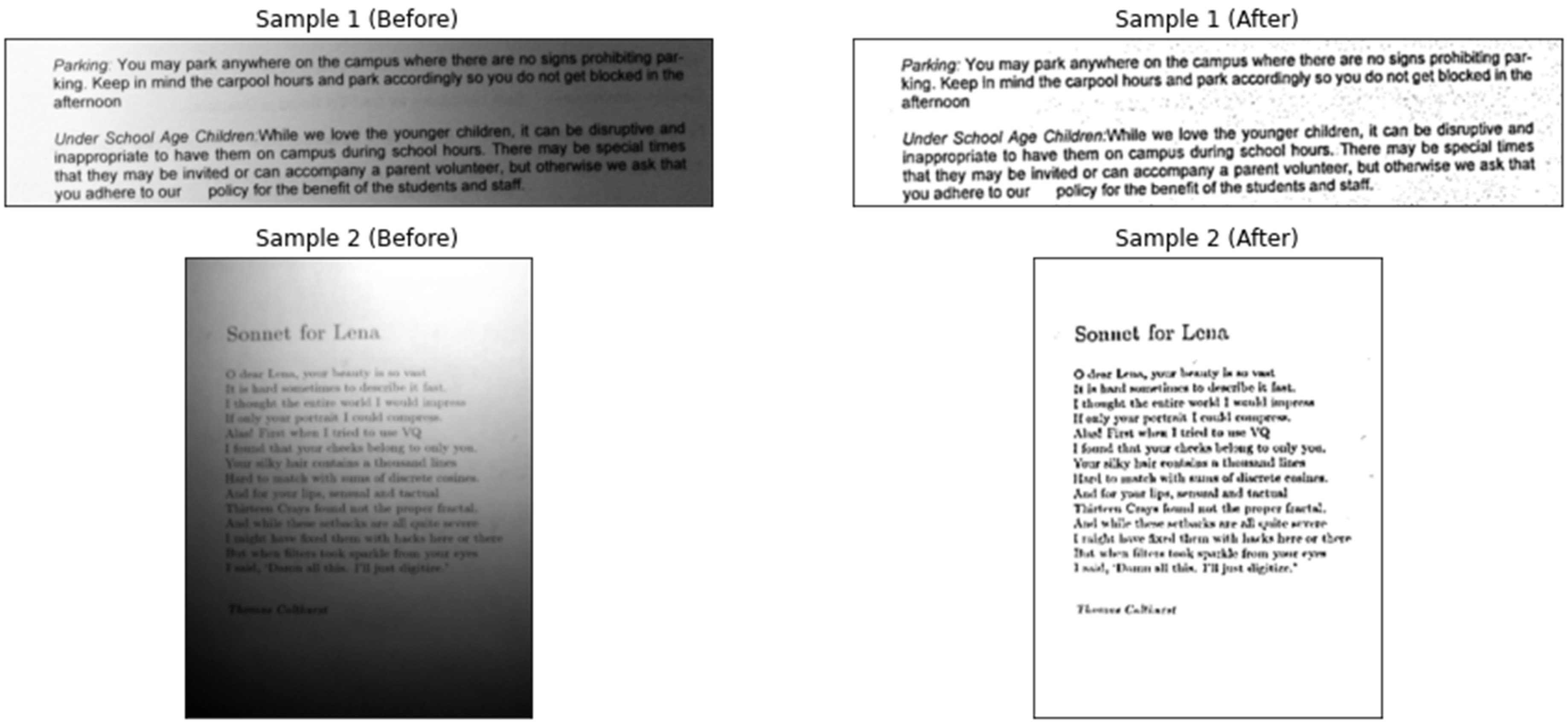
Thomas Colthurst

Figure 4 After Adaptive Gaussian Thresh (Sample 2)

Figure 5 After Gaussian Blur (Sample 2)

Results & Analysis:

After applying Method 1, we obtained the following:



While our method performs well for Sample 1, there are still many mistakes in Sample 2. This is because while Adaptive Gaussian Thresholding reduces non-uniform lightning, it also removes details from the characters as different parts of a character can have different intensities.

Sample 1: Our Method 1 vs Ground Truth (2 mistakes in 516 characters = 99.61% accuracy)

Parking: You may park anywhere on the campus where there are no signs prohibiting par-king. Keep in mind the carpool hours and park accordingly so you do not get blocked in the afternoon - : . Under School Age Children:While we love the younger children, it can be disruptive and inappropriate to have them on campus during school hours. There may be special times that they may be invited or can accompany a parent volunteer, but otherwise we ask that you adhere to our policy for the benefit of the students and staff.	Parking: You may park anywhere on the campus where there are no signs prohibiting par-king. Keep in mind the carpool hours and park accordingly so you do not get blocked in the afternoon Under School Age Children:While we love the younger children, it can be disruptive and inappropriate to have them on campus during school hours. There may be special times that they may be invited or can accompany a parent volunteer, but otherwise we ask that you adhere to our policy for the benefit of the students and staff.

Sample 2: Our Method 1 vs Ground Truth (46 mistakes in 614 characters = 92.51% accuracy)

Sonnet for Lena O dear Lena, your beauty in so vast te in harct sometinaes to deseribe it fant. Tthought the entice work! 1 would ianpress If only your portrait [coukl comprens, Alas! First when T trict! to use VQ 1 found that your cheeks belong to only you. Your silky haic coulalns a thousand lines Hard to match with suing of discrete cosines. And for your lips, sensual and tactual Thirteen, Crays found not the proper fractal, Aird while these setbacks are all qnite severe LU raight have fixed them with hacks bere or there Bat when filters took sparkle from your exes Tsaid, ‘Damn all this, I'l just digitize.” thomas Cultingst	Sonnet for Lena O dear Lena, your beauty is so vast It is hard sometimes to describe it fast. I thought the entire world I would impress If only your portrait I could compress. Alas! First when I tried to use VQ I found that your cheeks belong to only you. Your silky hair contains a thousand lines 27 Hard to match with sums of discrete cosines. And for your lips, sensual and tactual Thirteen Crays found not the proper fractal. And while these setbacks are all quite severe I might have fixed them with hacks here or there But when filters took sparkle from your eyes I said, ‘Damn all this. I’ll just digitize.” Thomas Colthurst

Method 2: Background Layer Removal

Code (in Python 3)**:**

```
def contrast_stretching(img):
    max_x = img.max()
    min_x = img.min()
    contrast_image = 255.0*(img - min_x)/(max_x-min_x)
    return contrast_image.astype(np.uint8)

def method_2(image):
    mask = image.copy()
    mask = cv2.Canny(mask, 30, 50)
    kernel = np.ones((3,3),np.uint8)
    mask = cv2.dilate(mask, kernel ,iterations = 1)
    mask = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kernel)

    background = image.copy()
    background = cv2.inpaint(background, mask, 3, cv2.INPAINT_TELEA)
    background = cv2.inpaint(background, mask, 100, cv2.INPAINT_TELEA)

    new_background = cv2.GaussianBlur(background, (3,3), 2)
    new_image = image / new_background
    norm_new_image = contrast_stretching(new_image)

    return norm_new_image
```

Implementation & Explanation:

Learning from our earlier experience, we noted that despite the uneven background shading, the human eye can still identify foreground text because it is always going to be darker than the immediate background. Mathematically, there will always be a sudden drop in grey level at the border of every character, but this drop will be less observable in the background, because its shading is changing much more gradually.

To tap on this special property, our overall strategy will be to:

1. Use edge detection and inpainting to reconstruct the background layer
2. Separate the background effect from the overall layer
3. Perform contrast stretching to enhance resulting image before OCR

To generate the inpainting mask, we have designed the following workflow:

1. Canny edge detection
   30 and 50 are the best thresholds to use here, because while they are low enough to capture most if not all of the character's borders, they are not too low, since we also do not want to misclassify the change in background shading as edges. After applying this step on Sample 2, we can obtain Figure 6.

2. Morphological dilation followed by closing
   As can be observed, the characters in Figure 6 are all ‘hollow’. Hence, the purpose of this step is to ‘fill’ these characters up, so as to mask all the foreground pixels as completely as possible, thereby reducing noise after inpainting. After this step, we can obtain Figure 7.

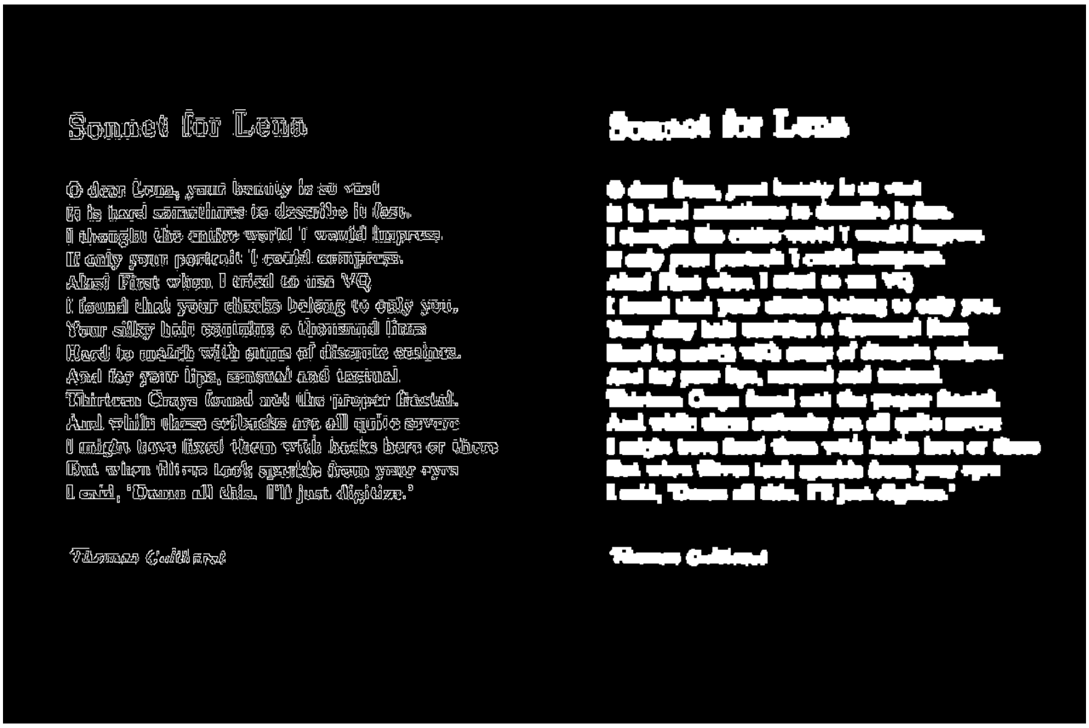


Figure 6 After Canny Edge (Sample 2)   *Figure 7 After Dilation & Closing (Sample 2)*

After generating the inpainting mask, we want to use it to extract the background.

3. Inpainting with Gaussian blur
   This step makes use of the mask generated earlier (see Figure 7). Wherever the mask contains a positive value (i.e. non-black), the algorithm will look for the corresponding location in the original image, and replace the area with neighbouring pixels, as if we are ‘repairing’ these areas.

   By taking out these foreground texts, we are left with the background. Taking it one step further, we then apply a Gaussian Blur to further remove noise. This will leave us with the following Figure 8:

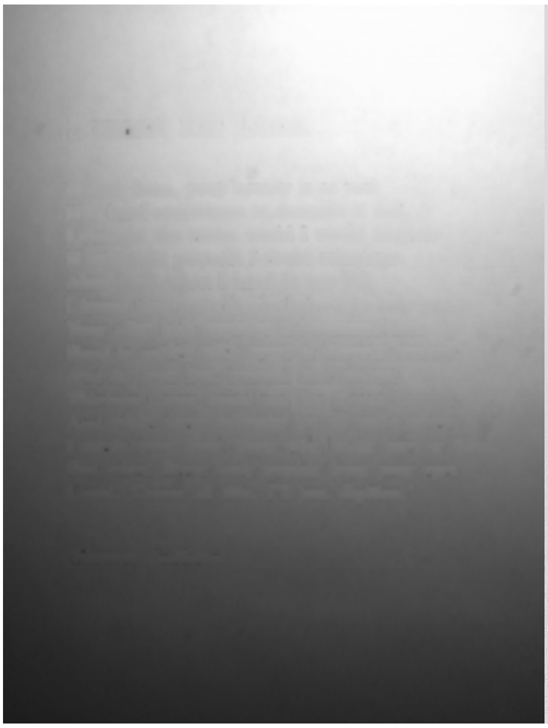
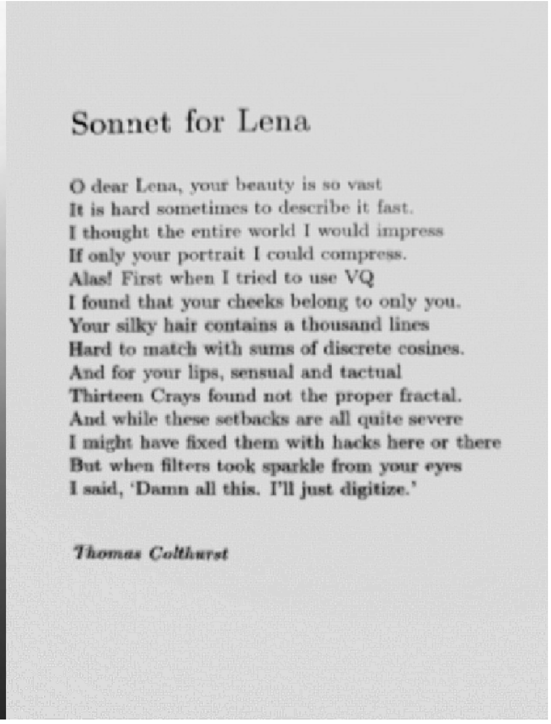


Figure 8 Background (Sample 2)



Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst

Figure 9 Post-processing (Sample 2)

4. Diminish background presence in the original image
   For this step, we have chosen to divide the entire image by the background. While minus might sound like a more intuitive operation, we must understand that background repair is never perfect (especially when we have applied a Gaussian filter earlier), and hence, if we do a minus operation, any resulting pixel with a high grey level, e.g. 255, will cause the image to have a wide range of grey values, e.g. [0, 255], rendering any subsequent contrast stretching efforts ineffective.

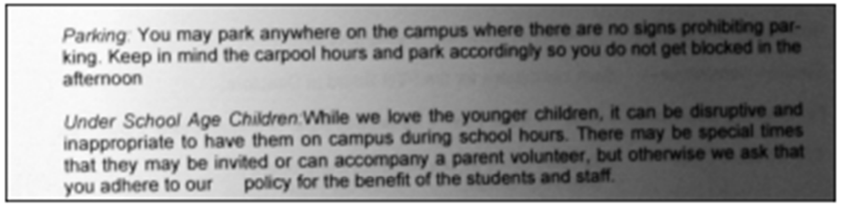
   Instead, by applying the division operation, we can obtain a "image / background ratio" matrix that has the same dimensions as that of the original image. For Sample 2, this will be a 589 x 782 matrix with values ranging from 0.4529 to 1.0828.

5. Contrast stretching
   Finally, we can take the earlier matrix and stretch it to fill the entire range of [0, 255]. This will give us Figure 9 (above).

Results & Analysis:

After applying Method 2, we obtained the following:

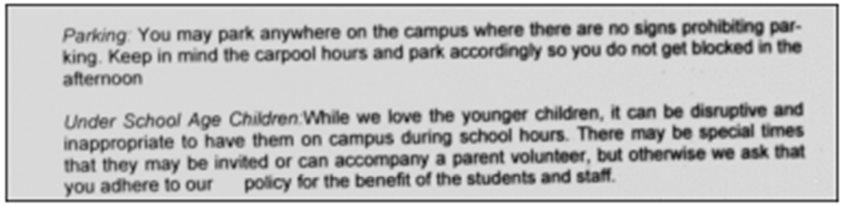
Sample 1 (Before)



Parking: You may park anywhere on the campus where there are no signs prohibiting parking. Keep in mind the carpool hours and park accordingly so you do not get blocked in the afternoon

Under School Age Children: While we love the younger children, it can be disruptive and inappropriate to have them on campus during school hours. There may be special times that they may be invited or can accompany a parent volunteer, but otherwise we ask that you adhere to our policy for the benefit of the students and staff.

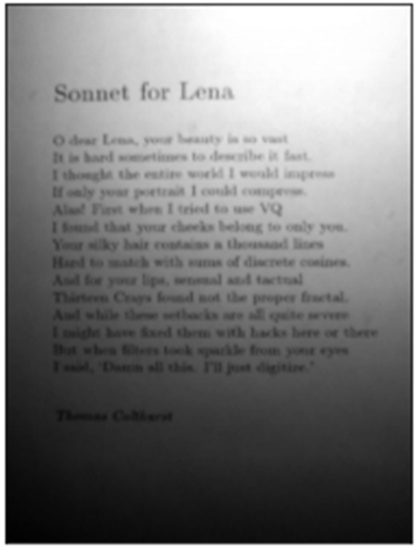
Sample 1 (After)



Parking: You may park anywhere on the campus where there are no signs prohibiting parking. Keep in mind the carpool hours and park accordingly so you do not get blocked in the afternoon

Under School Age Children: While we love the younger children, it can be disruptive and inappropriate to have them on campus during school hours. There may be special times that they may be invited or can accompany a parent volunteer, but otherwise we ask that you adhere to our policy for the benefit of the students and staff.

Sample 2 (Before)

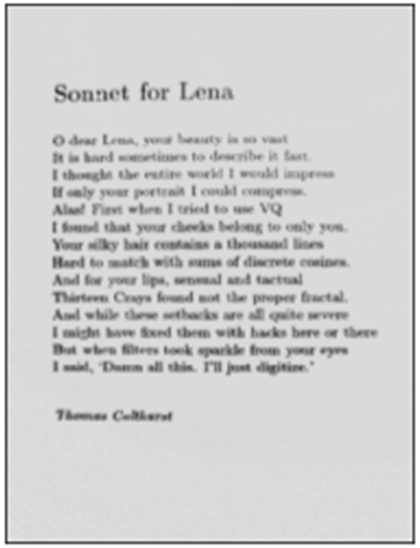


Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst

Sample 2 (After)



Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst

As evident from above, our algorithm offers a significant improvement over the Otsu algorithm. With no loss of text information, and the foreground being clearly distinguishable from the background, running OCR on the resulting image has yielded comforting results.

Sample 1: Our Method 2 vs Ground Truth (1 mistake in 516 characters = 99.81% accuracy)

Parking: You may park anywhere on the campus where there are no signs prohibiting par-king. Keep in mind the carpool hours and park accordingly so you do not get blocked in the afternoon Under School Age Children:While we love the younger children, it can be disruptive and inappropriate to have them on campus during school hours. There may be special times that they may be invited or can accompany a parent volunteer, but otherwise we ask that you adhere to our _ policy for the benefit of the students and staff.	Parking: You may park anywhere on the campus where there are no signs prohibiting par-king. Keep in mind the carpool hours and park accordingly so you do not get blocked in the afternoon Under School Age Children:While we love the younger children, it can be disruptive and inappropriate to have them on campus during school hours. There may be special times that they may be invited or can accompany a parent volunteer, but otherwise we ask that you adhere to our policy for the benefit of the students and staff.

Sample 2: Our Method 2 vs Ground Truth (4 mistakes in 614 characters = 99.35% accuracy)

Sonnet for Lena © dear Lena, your beauty is so vast It is hard sometimes to describe it fast, I thought the entire world I would impress If only your portrait I could compress. Alas! First when I tried to use VQ I found that your cheeks belong to only you. Your silky hair contains a thousand lines Hard to match with sums of discrete cosines. And for your lips, sensual and tactual Thirteen Crays found not the proper fractal. And while these setbacks are all quite severe I might have fixed them with hacks here or there But when filters took sparkle from your eyes I said, 'Damn all this. I'll just digitize." Thomas Colthurst	Sonnet for Lena O dear Lena, your beauty is so vast It is hard sometimes to describe it fast I thought the entire world I would impress If only your portrait I could compress. Alas! First when I tried to use VQ I found that your cheeks belong to only you. Your silky hair contains a thousand lines Hard to match with sums of discrete cosines. And for your lips, sensual and tactual Thirteen Crays found not the proper fractal. And while these setbacks are all quite severe I might have fixed them with hacks here or there But when filters took sparkle from your eyes I said, 'Damn all this. I'll just digitize." Thomas Colthurst

Method 3: A Simplified Approach to Background Layer Removal

Code (in Python 3)**:**

```
def method_3(image):
    background = cv2.GaussianBlur(image, (75, 75), 20)
    new_image = image / background
    norm_new_image = contrast_stretching(new_image)
    return norm_new_image
```

Implementation & Explanation:

The process of background removal above may require us to tune many parameters to maximize the accuracy of OCR eg. size of Canny mask, size of Gaussian Blur, Morphological Transformation and inpaint parameters. In this part, we will introduce an alternative way to reconstruct the background layer with less effort to tune needed to tune the parameters while still capable of achieving decent accuracy. The rationale is that there is a uniform gradient background across the image. We can use blur the whole image to get a result that is close to the actual background.

Our overall strategy will be to:

1. Apply Gaussian Blur
   With a large enough kernel size, we can use apply a Gaussian blur to blur out the text on the image and obtain the background layer. This attempts to achieve to same effect as Steps 1 – 3 of Method 2.

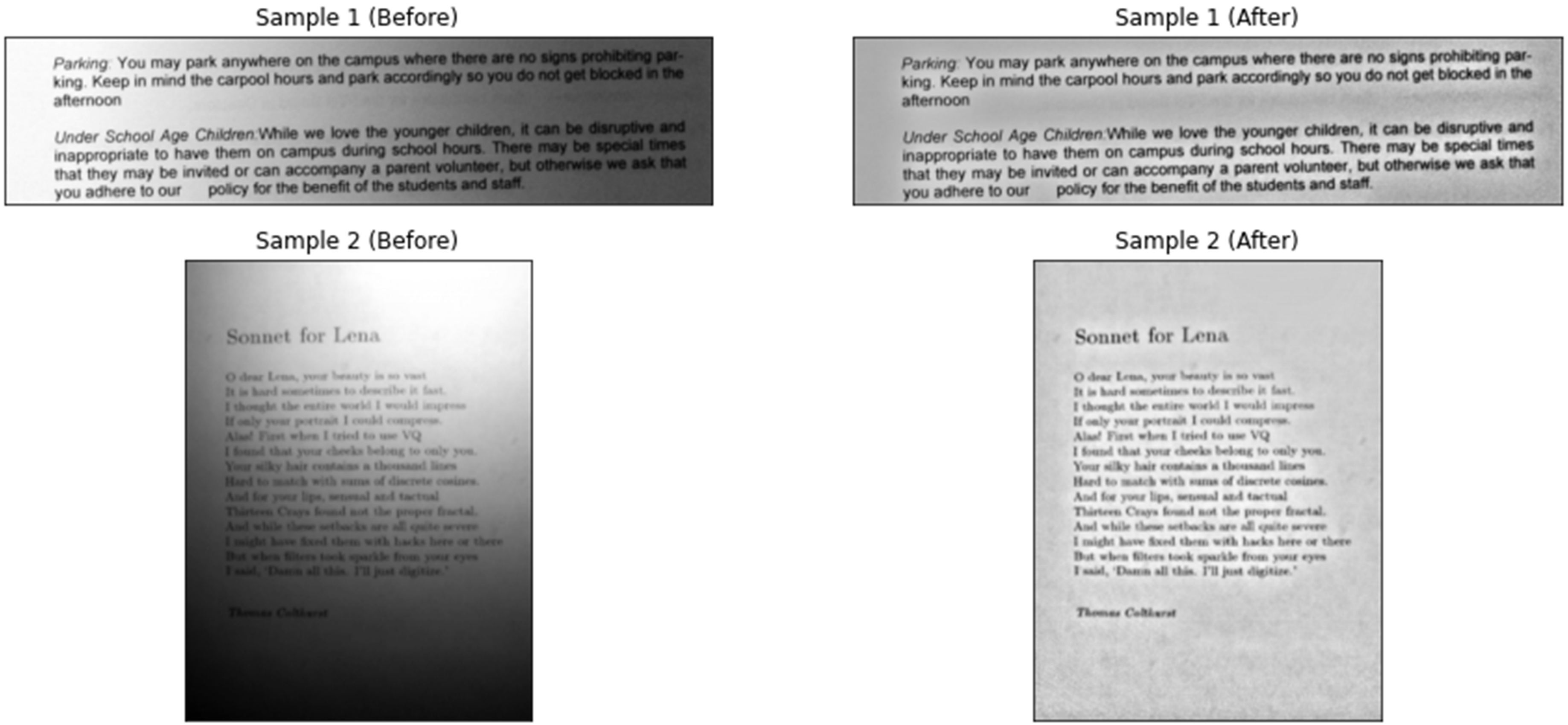
2. Diminish background presence in the original image
   This step is driven by the rationale as Step 4 of Method 2.

3. Contrast stretching
   This step is driven by the rationale as Step 5 (i.e. the final step) of Method 2.

However, the limitation of this method is that it is only applicable when the background spans across the whole image and there is uniform background.

Results & Analysis:

After applying Method 2, we obtained the following:



The comparison below shows that the results from using Gaussian Blur to obtain the background layer is as good as using Morphological Transformation as the accuracies are almost exactly the same.

Sample 1: Our Method 3 vs Ground Truth (1 mistake in 516 characters = 99.81% accuracy)

Parking: You may park anywhere on the campus where there are no signs prohibiting par-king. Keep in mind the carpool hours and park accordingly so you do not get blocked in the afternoon Under School Age Children:While we love the younger children, it can be disruptive and inappropriate to have them on campus during school hours. There may be special times that they may be invited or can accompany a parent volunteer, but otherwise we ask that you adhere to our _ policy for the benefit of the students and staff.	Parking: You may park anywhere on the campus where there are no signs prohibiting par-king. Keep in mind the carpool hours and park accordingly so you do not get blocked in the afternoon Under School Age Children:While we love the younger children, it can be disruptive and inappropriate to have them on campus during school hours. There may be special times that they may be invited or can accompany a parent volunteer, but otherwise we ask that you adhere to our policy for the benefit of the students and staff.

Sample 2: Our Method 3 vs Ground Truth (5 mistakes in 614 characters = 99.18% accuracy)

Sonnet for Lena © dear Lena, your beauty is so vast It is hard sometimes to describe it fast, I thought the entire world | would impress If only your portrait [could compress. Alas! First when I tried to use VQ I found that your cheeks belong to only you, Your silky hair contains a thousand lines Hard to match with sums of discrete cosines. And for your lips, sensual and tactual Thirteen Crays found not the proper fractal. And while these setbacks are all quite severe I might have fixed them with hacks here or there But when filters took sparkle from your eyes I said, ‘Damn all this. I'll just digitize.” Thomas Colthurst	Sonnet for Lena O dear Lena, your beauty is so vast It is hard sometimes to describe it fast I thought the entire world I would impress If only your portrait I could compress. Alas! First when I tried to use VQ I found that your cheeks belong to only you. Your silky hair contains a thousand lines Hard to match with sums of discrete cosines. And for your lips, sensual and tactual Thirteen Crays found not the proper fractal. And while these setbacks are all quite severe I might have fixed them with hacks here or there But when filters took sparkle from your eyes I said, ‘Damn all this. I’ll just digitize.” Thomas Colthurst

Part 3

Discuss how to improve recognition algorithms for more robust and accurate character recognition while document images suffer from different types of image degradation. This is an open and optional task. There will be bonus points if you have good ideas.

In this section we will be proposing three novel ideas, which we believe will lead to an improved OCR performance:

1. Location-aware correction

Based on our experience working with Tesseract thus far, one interesting observation that we made was that: Tesseract seems to be more accurate when we 'zoom' into the region of interest.

For instance, when we were performing OCR on the entire Sample 2, we were getting "| would" and "| could" instead of "I would" and "I could" (see previous page). However, if we chose to take a snippet of the region of interest (Figure 10), and then perform another round of OCR, we would get the correct results (Figure 11).

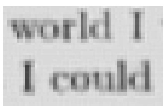


Figure 10 Snippet (Sample 2)

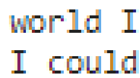


Figure 11 Correct Result (Sample 2)

Given how most character recognition algorithms these days have some form of neural networks built-in (Tesseract, for instance, is a LSTM-based OCR engine), we propose that whenever future algorithms encounter an region of low confidence, it should zoom in and perform a second round of character recognition for that region separately.

2. A search-based pre-processing approach

Our research shows that character recognition algorithms perform best when the input image follows a set of rules, such as having a high resolution (> 300 DPI), large fonts (> 12) and no illumination issues. However, most character recognition algorithms either do not possess the ability to do image pre-processing, or have it shrouded in a black-box, something that the user could not easily relate to.

In our earlier proposed algorithm, we used trial-and-error and leveraged on existing research to establish a set of pre-processing steps. Although our algorithm has performed well on the two sample images, there is no guarantee that it would generalize well on other image types.

Hence, we propose for future character recognition systems to incorporate the various methods into a single search space and look for the optimal sequence using a Depth-First-Search, as illustrated by the following example:

Assuming that there are only two methods to pre-process an image: A and B, we can look for the optimal sequence in the following manner:

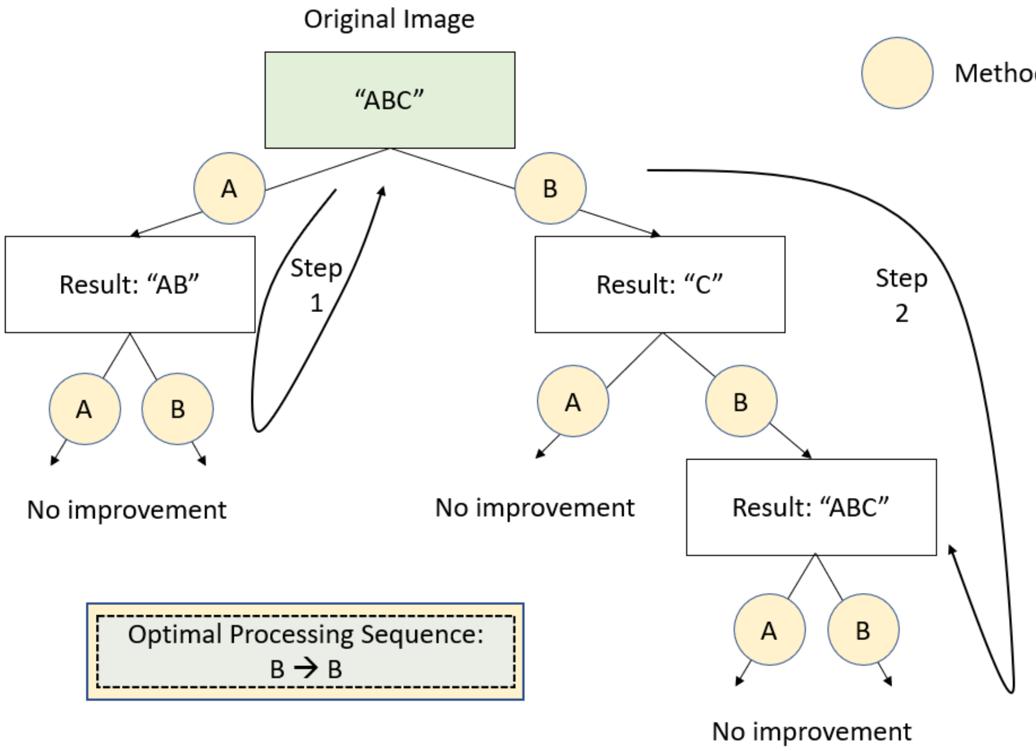


Figure 12 A search-based pre-processing approach

By further expanding on the most promising method at every stage, and back-tracking when there is no further improvement, we can look for the optimal pre-processing sequence in the same way we look for an optimal path.

3. Bounding box adjustment

Degraded documents often contain noisy artefacts such as ink marks, which degrades OCR performance. However, humans are still able to recognize the characters/words in these documents because we can guess a word from the neighbouring words. Hence, by assuming that the document follows the right grammar and syntax, we can incorporate these natural language techniques into the OCR. Furthermore, we can also include the use of spell-checks to verify the correctness of each individual words.

One possible way to introduce the above-mentioned techniques into OCR is by readjusting the bounding box. Figure 13 shows an example of a vandalised document with a line across multiple words. Typically, OCR will consider the whole image as one bounding box. However, in our case, treating the whole image as one bounding box results in no detection.

Hence, we propose for OCR to use the spaces between lines and words to segregate each word into their bounding boxes, such as in Figure 14. Interestingly, just by drawing these boxes, we can already start to get positive OCR detections! In this case, our detection is "| Cultivators ihad| So|amucn |", far from perfect but a huge improvement in the right direction.

Furthermore, if we look at these bounding boxes individually, we may achieve even better results! For instance, we are almost getting perfect detection in Figure 15.

[2] J. Banerjee, A. M. Namboodiri and C. V. Jawahar, "Contextual restoration of severely degraded document images," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, 2009, pp. 517-524, doi: 10.1109/CVPR.2009.5206601.

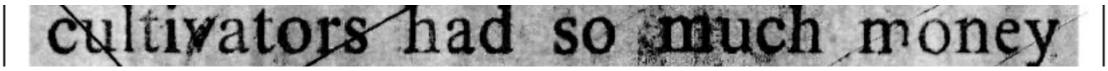


Figure 13 Vandalised document – No detection

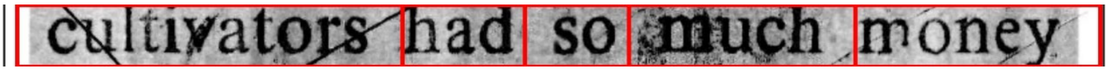


Figure 14 Vandalised document with adjusted bounding boxes – Good detection

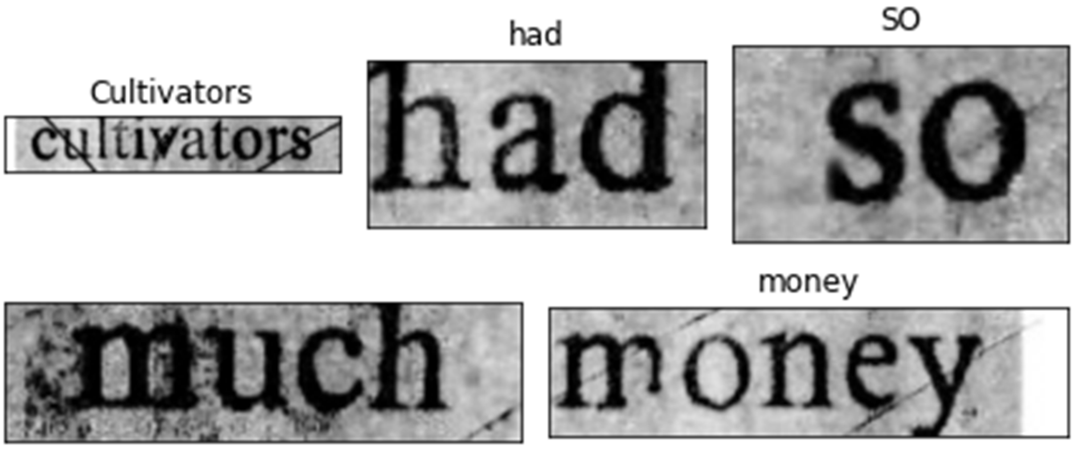


Figure 15 Looking at individual boxes – Almost perfect detection

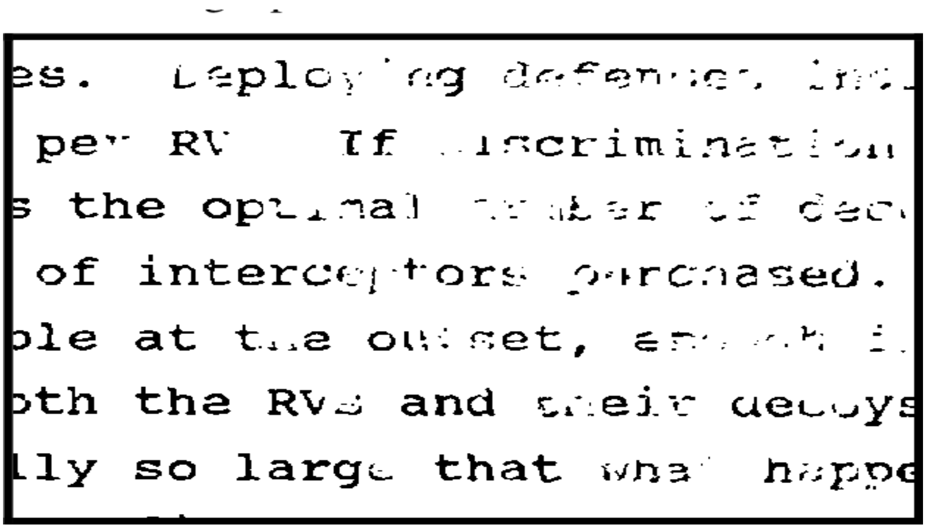


Figure 16 Document with missing information

When processing documents with missing information, such as the one shown in Figure 16, we can also take a similar approach. First, we will use OCR to generate bounding boxes for each word using spaces. With this step, OCR can recognize some words in the image and leave the unknown words blank. Next, we can use existing language models such as N-gram or deep learning model to predict candidates for each unrecognized word. We can iterate through each candidate and reapply OCR to compute the probability that the word in the bounding box is the candidate word. Thus, the final prediction is the word that follows the following equation:

$$\underset{word \in candidates}{\operatorname{argmax}} \quad P(word \mid context) * P(bounding\ box\ contains\ word \mid word)$$