## Submitted By: Enan Imdadul Haque

Student ID: S237802019

Submission Date: 2024/04/22

## 1- part (a): 200 words text

#### (a) Problem-Solving with Image Processing

Problem: Imagine dealing with a digital image containing text that's somewhat unclear due to factors like uneven lighting, background noise, or minor imperfections. The challenge is to accurately extract the text for further processing, such as Optical Character Recognition (OCR).

**Process to Achieve the Goal:**

**Load the Image:** Start by using OpenCV's cv2.imread() function to read the image.

Grayscale Conversion (Optional): If the image has color channels and you only need the text, convert it to grayscale to simplify further steps.

**Thresholding:** Create a binary image where pixels above a certain threshold become white (255, foreground), and those below become black (0, background). This step helps distinguish text from the background.

**Noise Reduction:** Techniques like blurring can help smooth out noise, but avoid blurring the text itself.

Morphological Operations: These are vital for refining the binary image, ensuring accurate text extraction.

#### (b) Morphological Operations: A Crucial Application

**Scenario: Picture an image with text where:**

Foreground (text) objects have varying background illumination, causing some parts of the text to blend with the background.

Background noise exists, such as small objects or texture.

Crucial Role of Morphological Operations (in this specific code):

**The code showcases two morphological operations:**

**Morphological Operation (using closing):**

kernel\_1 = np.ones((5, 5), np.uint8): Defines a 5x5 square kernel.

Closing\_operation = cv2.morphologyEx(image, cv2.MORPH\_CLOSE, kernel\_1): Performs closing, eroding (shrinking objects) followed by dilation (growing objects) using the kernel.

This process enhances the foreground text by eliminating background illumination variations, making darker text areas stand out.

**Dilation:**

kernel\_2 = np.ones((9, 9), np.uint8): Defines a larger 9x9 square kernel.

dilate\_image = cv2.morphologyEx(image, cv2.MORPH\_DILATE, kernel\_2): Performs dilation, thickening thin lines in the text.

#### (c) Why Morphological Operations are Indispensable

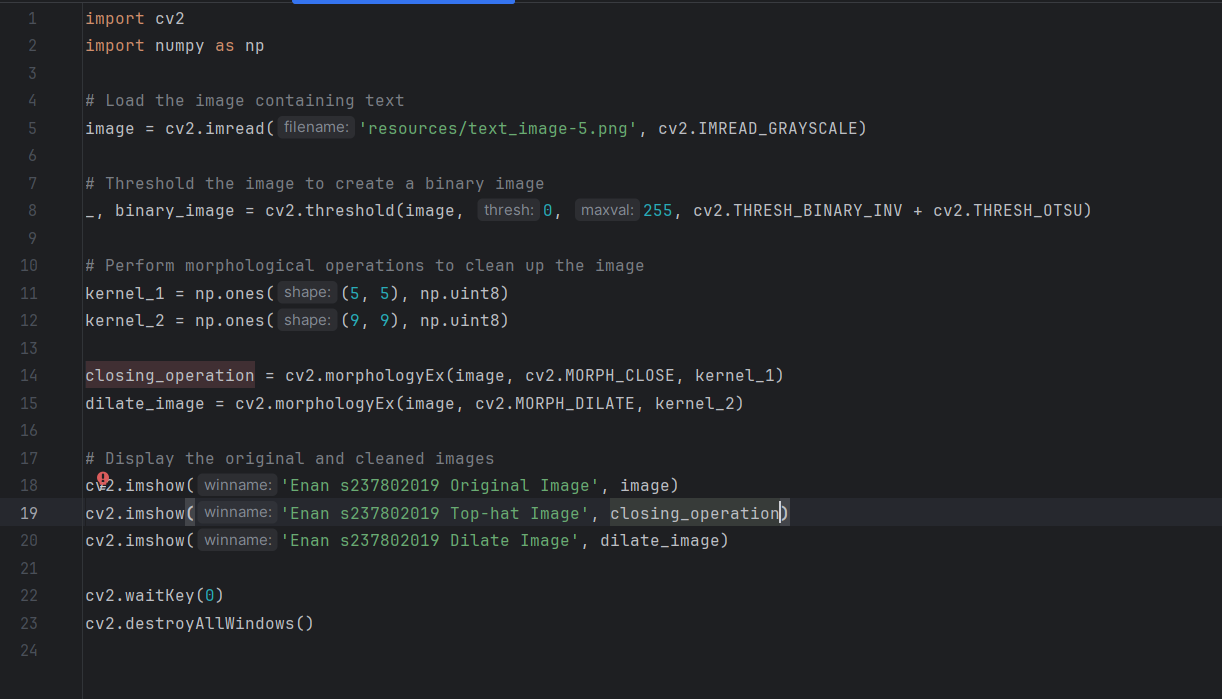
**Without morphological operations:**

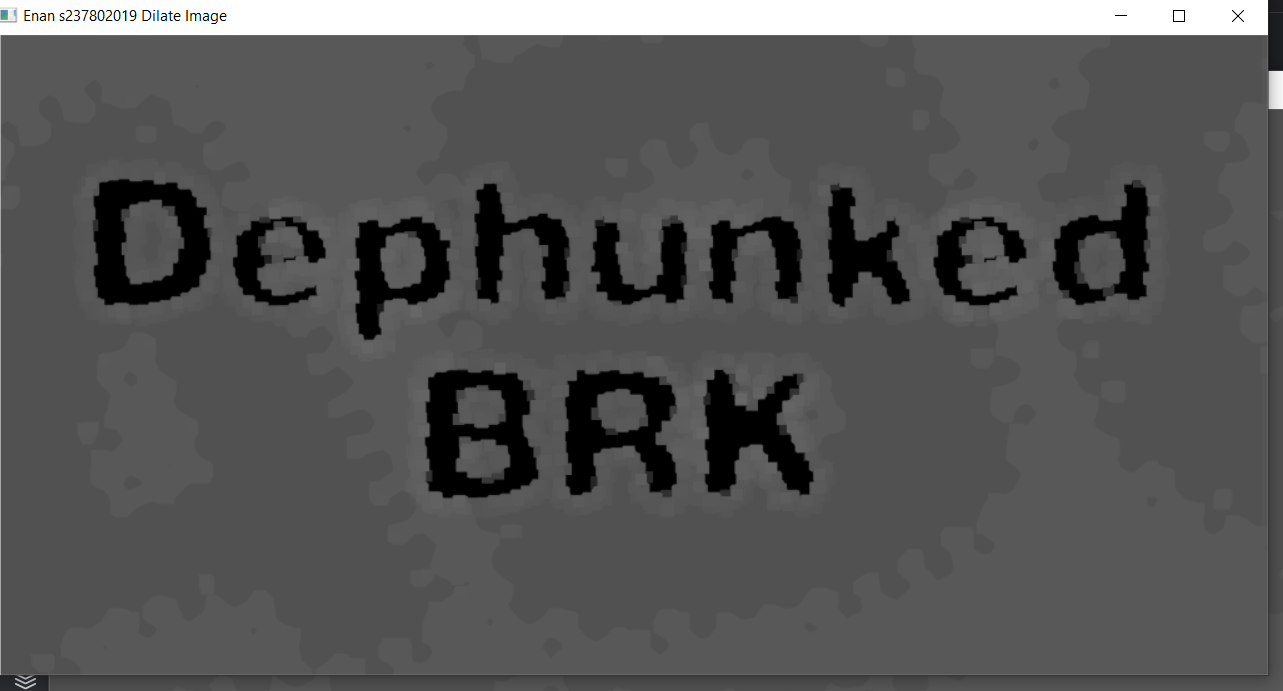
Uneven background illumination may cause text loss during thresholding.

Background noise could persist, potentially interfering with later stages like OCR.

The top-hat transform helps counter uneven illumination by enhancing foreground text, while dilation thickens thin lines. These operations improve the binary image's quality, facilitating accurate text extraction.

## 2 - part (b): a screenshot (jpg) displaying the OpenCV program and the result in the image.





## part (c): a screenshot (jpg) displaying the OpenCV program and result the image of before the morphological process showing the result in an output image with the title as in (2)

