APPENDIX

CODING GRAYSCALE

```
for i in range(h):
    for j in range(w):
        p = float(img[i][j][0])*0.2989 + float(img[i][j][1])*0.5870 + float(img[i][j][2])*0.1140
newImage[i][j] = p
```

CODING PREPROCESSING

```
def calculate_template_space(template_side_length):
    return int(template_side_length/2)


def dilation(img, template_side_length):
    template_space = calculate_template_space(template_side_length)


for x in range(template_space, new_image.shape[1] - template_space):
    for y in range(template_space, new_image.shape[0] - template_space):
        for c in range(0, template_side_length):
            for d in range(0, template_side_length):
                if sub > maximum:
                    if sub > 0:
                        maximum = sub
if sub < minimum:
                    if sub > 0:
                        minimum = sub
def optimize(image):
    new_image = erosion(image, 3)
    new_image = dilation(new_image, 3)
return new_image
```

1.06% PLAGIARISM APPROXIMATELY

Report #13990855

INTRODUCTION Background Sometimes processing text data or numbers in images, it makes us difficult to process the data. Ocr is software that converts text in image format or image files into text format that can be read and edited by computer applications. This application is able to extract text in an image. Actually there are lots of applications like this on the internet. But for those of us who care about privacy, sometimes there is our data that we don't want to publish on the internet, therefore using offline applications will certainly be safer. This program will later be more of a text converter in the image into a txt file, not only detecting text. The data will also be easier to process if the final result is in the form of txt. And also not just calling the open source library tesseract because I've tried it and there are some images whose backgrounds make it undetectable or illegible. therefore using the ocr algorithm, before choosing the OCR algorithm, the image must be processed first so

b