```
In [1]: import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt
```

1. Using Closing

```
In [2]: img = cv.imread("txtimg.png")
        img1 = cv.cvtColor(img,cv.COLOR BGR2RGB)
        #converting to grayscale
        gray = cv.cvtColor(img,cv.COLOR BGR2GRAY)
        #applying gaussian blur
        blur = cv.GaussianBlur(gray, (3,3),0)
        #thresholding to convert into binary
        thresholded = cv.adaptiveThreshold(blur, 255, cv. ADAPTIVE THRESH GAUSSIAN C, cv. THRESH BINA
        #dilate image to connect text contours
        kernel = cv.getStructuringElement(cv.MORPH RECT, (35,21))
        dilated = cv.morphologyEx(thresholded,cv.MORPH CLOSE,kernel)
        #get external contours
        contours = cv.findContours(dilated,cv.RETR EXTERNAL,cv.CHAIN APPROX SIMPLE)
        contours = contours[0] if len(contours) == 2 else contours[1]
        #draw contours
        result = imq1.copy()
        for points in contours:
           pad = 10
            x, y, w, h = cv.boundingRect(points)
            cv.rectangle(result, (x, y), (x+w, y+h), (0, 0, 255), 2)
        plt.imshow(result)
        plt.axis('off')
```

Out[2]: (-0.5, 699.5, 439.5, -0.5)

It was the best of times, it was the worst of times, it was the age of wisdom, it was the age age of foolishness...

```
In [3]: img = cv.imread("txtimg1.png")
    img1 = cv.cvtColor(img,cv.COLOR_BGR2RGB)
    #converting to grayscale
    gray = cv.cvtColor(img,cv.COLOR_BGR2GRAY)
    #applying gaussian blur
    blur = cv.GaussianBlur(gray,(3,3),0)
    #thresholding to convert into binary
    thresholded = cv.adaptiveThreshold(blur,255,cv.ADAPTIVE_THRESH_GAUSSIAN_C,cv.THRESH_BINA
    #dilate image to connect text contours
    kernel = cv.getStructuringElement(cv.MORPH_RECT,(6,5))
    dilated = cv.morphologyEx(thresholded,cv.MORPH_CLOSE,kernel)
    #get external contours
    contours = cv.findContours(dilated,cv.RETR_EXTERNAL,cv.CHAIN_APPROX_SIMPLE)
    contours = contours[0] if len(contours)==2 else contours[1]
    #draw contours
```

```
result = img1.copy()
for points in contours:
    pad = 10
    x,y,w,h = cv.boundingRect(points)
    cv.rectangle(result,(x,y),(x+w,y+h),(0,0,255),2)
plt.imshow(result)
plt.axis('off')
```

Out[3]: (-0.5, 297.5, 168.5, -0.5)

```
A basic Paragraph

State of the control of the cont
```

```
In [4]: img = cv.imread("txtimg2.png")
        img1 = cv.cvtColor(img,cv.COLOR BGR2RGB)
        #converting to grayscale
        gray = cv.cvtColor(img,cv.COLOR BGR2GRAY)
        #applying gaussian blur
        blur = cv.GaussianBlur(gray, (3,3), 0)
        #thresholding to convert into binary
        thresholded = cv.adaptiveThreshold(blur, 255, cv. ADAPTIVE THRESH GAUSSIAN C, cv. THRESH BINA
        #dilate image to connect text contours
        kernel = cv.getStructuringElement(cv.MORPH RECT, (15,2))
        dilated = cv.morphologyEx(thresholded,cv.MORPH CLOSE,kernel)
        #get external contours
        contours = cv.findContours(dilated,cv.RETR EXTERNAL,cv.CHAIN APPROX SIMPLE)
        contours = contours[0] if len(contours) == 2 else contours[1]
        #draw contours
        result = img1.copy()
        for points in contours:
            pad = 10
            x, y, w, h = cv.boundingRect(points)
            cv.rectangle(result, (x, y), (x+w, y+h), (0, 0, 255), 2)
        plt.imshow(result)
        plt.axis('off')
```

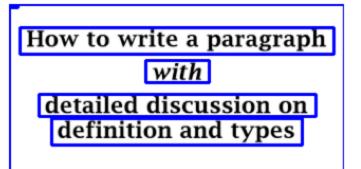
Out[4]: (-0.5, 182.5, 274.5, -0.5)

```
CORTO IS THE TIME OF THE EIGHT STATES OF THE E
```

2. Using Dilation

```
In [5]: img = cv.imread("txtimg3.jpeg")
        img1 = cv.cvtColor(img,cv.COLOR BGR2RGB)
        #converting to grayscale
        gray = cv.cvtColor(img,cv.COLOR BGR2GRAY)
        #applying gaussian blur
        blur = cv.GaussianBlur(gray, (3,3),0)
        #thresholding to convert into binary
        thresholded = cv.adaptiveThreshold(blur, 255, cv. ADAPTIVE THRESH GAUSSIAN C, cv. THRESH BINA
        #dilate image to connect text contours
        kernel = cv.getStructuringElement(cv.MORPH RECT, (15,5))
        # dilated = cv.morphologyEx(thresholded,cv.MORPH CLOSE,kernel)
        dilated = cv.dilate(thresholded, kernel, iterations=1)
        #get external contours
        contours = cv.findContours(dilated,cv.RETR EXTERNAL,cv.CHAIN APPROX SIMPLE)
        contours = contours[0] if len(contours) == 2 else contours[1]
        #draw contours
        result = imq1.copy()
        for points in contours:
            pad = 10
            x,y,w,h = cv.boundingRect(points)
            cv.rectangle(result, (x, y), (x+w, y+h), (0, 0, 255), 2)
        plt.imshow(result)
        plt.axis('off')
```

Out[5]: (-0.5, 316.5, 158.5, -0.5)



```
In [6]: img = cv.imread("txtimg2.png")
        img1 = cv.cvtColor(img,cv.COLOR BGR2RGB)
        #converting to grayscale
        gray = cv.cvtColor(img,cv.COLOR BGR2GRAY)
        #applying gaussian blur
        blur = cv.GaussianBlur(gray, (3,3),0)
        #thresholding to convert into binary
        thresholded = cv.adaptiveThreshold(blur, 255, cv.ADAPTIVE THRESH GAUSSIAN C, cv.THRESH BINA
        #dilate image to connect text contours
        kernel = cv.getStructuringElement(cv.MORPH RECT, (15,2))
        # dilated = cv.morphologyEx(thresholded,cv.MORPH CLOSE,kernel)
        dilated = cv.dilate(thresholded, kernel, iterations=1)
        #get external contours
        contours = cv.findContours(dilated,cv.RETR EXTERNAL,cv.CHAIN APPROX SIMPLE)
        contours = contours[0] if len(contours) == 2 else contours[1]
        #draw contours
        result = img1.copy()
        for points in contours:
            pad = 10
            x,y,w,h = cv.boundingRect(points)
            cv.rectangle(result, (x, y), (x+w, y+h), (0, 0, 255), 2)
        plt.imshow(result)
        plt.axis('off')
```

```
Out[6]: (-0.5, 182.5, 274.5, -0.5)
```

```
PARAGRAPH UN EARTH

Earth is the third of the eight planets in the solar system from the Sun, Earth is the only planet on which lite has come into existence. Earth and its resources make life possible on the earth.

The earth revolves around the sun in a special cacle collect the orbital, Earth is the home not only of man but also of millions of other living beings, the only place in the universe where life exists.

**EXAL MOVE**

Inters://studyparagraphs.co/
```

```
In [7]: img = cv.imread("txtimg.png")
        img1 = cv.cvtColor(img,cv.COLOR BGR2RGB)
        #converting to grayscale
        gray = cv.cvtColor(img,cv.COLOR BGR2GRAY)
        #applying gaussian blur
        blur = cv.GaussianBlur(gray, (3,3),0)
        #thresholding to convert into binary
        thresholded = cv.adaptiveThreshold(blur, 255, cv. ADAPTIVE THRESH GAUSSIAN C, cv. THRESH BINA
        #dilate image to connect text contours
        kernel = cv.getStructuringElement(cv.MORPH RECT, (35,10))
        # dilated = cv.morphologyEx(thresholded,cv.MORPH CLOSE,kernel)
        dilated = cv.dilate(thresholded, kernel)
        #get external contours
        contours = cv.findContours(dilated,cv.RETR EXTERNAL,cv.CHAIN APPROX SIMPLE)
        contours = contours[0] if len(contours) == 2 else contours[1]
        #draw contours
        result = imq1.copy()
        for points in contours:
            pad = 10
            x, y, w, h = cv.boundingRect(points)
            cv.rectangle(result, (x, y), (x+w, y+h), (0, 0, 255), 2)
        plt.imshow(result)
        plt.axis('off')
```

Out[7]: (-0.5, 699.5, 439.5, -0.5)

```
It was the best of times, it was the worst of times, it was the age of wisdom, it was the age age of foolishness...
```

```
img = cv.imread("txtimg1.png")
img1 = cv.cvtColor(img,cv.COLOR_BGR2RGB)
#converting to grayscale
gray = cv.cvtColor(img,cv.COLOR_BGR2GRAY)
#applying gaussian blur
blur = cv.GaussianBlur(gray,(3,3),0)
#thresholding to convert into binary
thresholded = cv.adaptiveThreshold(blur,255,cv.ADAPTIVE_THRESH_GAUSSIAN_C,cv.THRESH_BINA)
```

```
#dilate image to connect text contours
kernel = cv.getStructuringElement(cv.MORPH_RECT,(5,5))
# dilated = cv.morphologyEx(thresholded,cv.MORPH_CLOSE,kernel)
dilated = cv.dilate(thresholded,kernel)
#get external contours
contours = cv.findContours(dilated,cv.RETR_EXTERNAL,cv.CHAIN_APPROX_SIMPLE)
contours = contours[0] if len(contours)==2 else contours[1]
#draw contours
result = img1.copy()
for points in contours:
    pad = 10
    x,y,w,h = cv.boundingRect(points)
    cv.rectangle(result,(x,y),(x+w,y+h),(0,0,255),2)
plt.imshow(result)
plt.axis('off')
```

Out[8]: (-0.5, 297.5, 168.5, -0.5)

