ARTIFICIAL INTELLEGNECE

Lab 02

Abstract

Feature Extraction, bounding boxes, centroid, and transition

Syed Muhammad Akbar

Msyed.bscs17seecs

Preparations

```
!pwd
!pip install -q Pillow
from google.colab.patches import cv2_imshow
```

```
/content
```

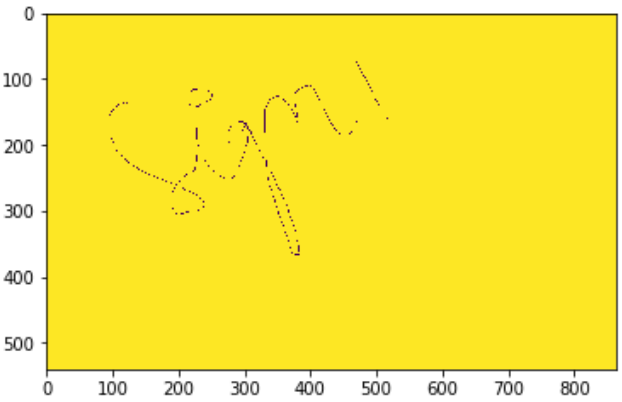
```
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt

# Load a color image in grayscale
img = cv.imread('sign image.PNG', 0)


plt.imshow(img)
```

```
plt.imshow(img)
```

```
<matplotlib.image.AxesImage at 0x7f13a46ff2e8>
```



TASK 01

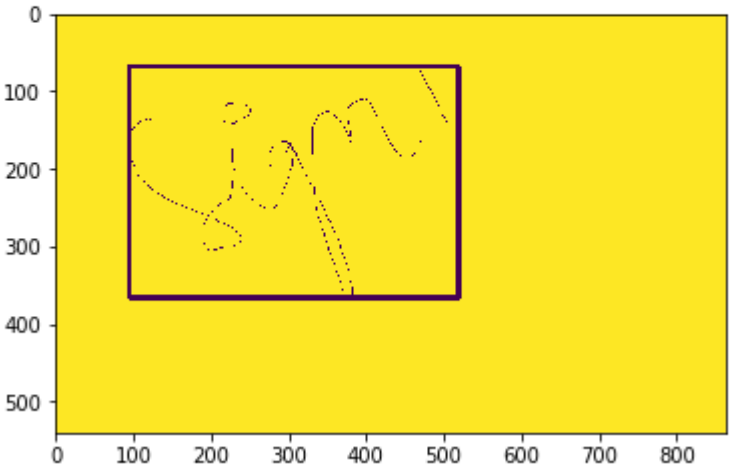
```
def box(img):
  height, width = img.shape
  print(width, height)
  left = width
  right = 0
  top = height
  bottom = 0
  for x in range(width):
    for y in range(height):
      #print(x, y)
```

```
        color = img[y, x]
        if color == 0:
          if x > right:
            right = x
          if x < left:
            left = x
          if y > bottom:
            bottom = y
          if y < top:
            top = y
    print(top, bottom, left, right)
    img = cv.rectangle(img, (left, top), (right, bottom), (0, 0, 0), 5)
    plt.imshow(img)
    return [img, left, top, right, bottom]

boxes= box(img)
#print(boxes)
```

```
boxes= box(img)
#print(boxes)
```

```
864 540
69 366 95 519
```

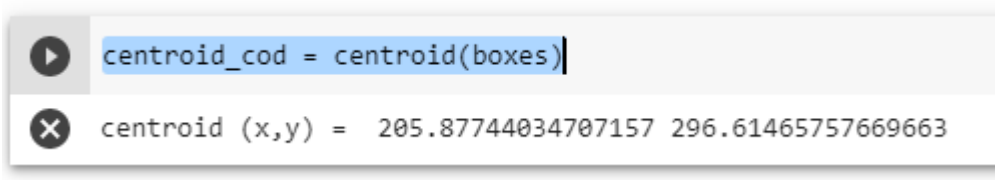


TASK 02

```
def centroid(boxes):
  height, width = boxes[0].shape
  cx = boxes[1]
  cy = boxes[2]
  n = 0
  img = boxes[0]
  for  x  in  range(boxes[1], boxes[3]):
```

```
    for y  in  range(boxes[2], boxes[4]):
      if img[y,x] == 0:
        cx  =  cx  +  x
        cy  =  cy  +  y
        n  =  n  +  1
  cx  =  cx  /  n
  cy  =  cy  /  n
  print('centroid (x,y) = ', cy, cx)
  return (cy, cx)
```

```
centroid_cod = centroid(boxes)
```

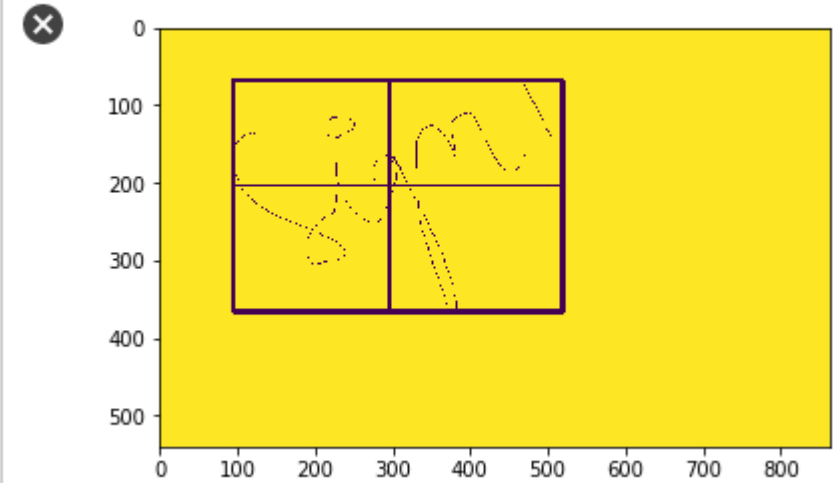


TASK 03

```
def div_four(boxes, cent):
  y, x = cent
  y = int(y)
  x = int(x)
  img = boxes[0]
  left = boxes[1]
  top = boxes[2]
  right = boxes[3]
  bottom = boxes[4]
  img = cv.rectangle(img, (left, top), (x, y), (0, 0, 0), 2)
  img = cv.rectangle(img, (boxes[1], y), (x, boxes[4]), (0, 0, 0), 2)
  img = cv.rectangle(img,  (x, y), (boxes[3], boxes[4]), (0, 0, 0), 2)
  img = cv.rectangle(img, (x, boxes[2]), (boxes[3], y), (0, 0, 0), 2)
  im1 = img[:y, :x]
  im2 = img[:y, x:]
  im3 = img[y:, :x]
  im4 = img[y:, x:]
  # cv2_imshow(im1)
  # cv2_imshow(im2)
  # cv2_imshow(im3)
  # cv2_imshow(im4)
  plt.imshow(img)
  return [im1, im2, im3, im4]
```

```
segments = div_four(boxes, centroid_cod)
```

```
segments = div_four(boxes, centroid_cod)
```



TASK 04

```
def white_transition(img):
  height, width = img.shape
  prev = img[0, 0]
  n = 0
  for x in range(1, width):
    for y in range(1, height):
      curr = img[y, x]
      if curr == 255 and prev == 0:
        n = n + 1
      prev = curr
  print(n)
```

```
for segment in segments:
  white_transition(segment)
```

```
for segment in segments:
  white_transition(segment)
```

```
537
672
680
545
```