Govardhan Seshasayee

Text detection using OpenCV and OCR

Text detection is the process of detecting the text present in the image, followed by surrounding it with a rectangular bounding box. Text detection can be carried out using image-based techniques or machine learning-based techniques. In image-based techniques, an image is segmented into multiple segments, and statistical features of connected components are utilised to form the text. Whereas, Machine learning approaches use support vector machines and convolutional neural networks to classify the components into text and non-text.

The necessary steps that you need to perform are:

1. Image pre-processing
2. Find possible contours that can represent the textual areas.
3. Apply optical character recognition (using python-tesseract, google OCR engine.

Output: Text detection

Stacking of Detected texts:

Input Code:

```
#Importing required libraries
import cv2
import pytesseract #Install Tesseract.exe in the location below.
import numpy as np
from PIL import ImageGrab
import time

#image reading
pytesseract.pytesseract.tesseract_cmd = 'C:\\Program Files\\Tesseract-OCR\\tesseract.exe'
img = cv2.imread('img.png')
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

#image to data and rewriting;
boxes = pytesseract.image_to_data(img)
for a,b in enumerate(boxes.splitlines()):
        #print(b)
        if a!=0:
            b = b.split()
            if len(b)==12:
                x,y,w,h = int(b[6]),int(b[7]),int(b[8]),int(b[9])
                cv2.putText(img,b[11],(x,y-5),cv2.FONT_HERSHEY_SIMPLEX,1,(50,50,255),2)
                cv2.rectangle(img, (x,y), (x+w, y+h), (50, 50, 255), 2)

cv2.imshow('img', img)
cv2.waitKey(0)
```

```
import cv2
import numpy as np

def stackImages(scale,imgArray):
    rows = len(imgArray)
    cols = len(imgArray[0])
    rowsAvailable = isinstance(imgArray[0], list)
    width = imgArray[0][0].shape[1]
    height = imgArray[0][0].shape[0]
    if rowsAvailable:
        for x in range ( 0, rows):
            for y in range(0, cols):
                if imgArray[x][y].shape[:2] == imgArray[0][0].shape [:2]:
                    imgArray[x][y] = cv2.resize(imgArray[x][y], (0, 0), None, scale, scale)
                else:
                    imgArray[x][y] = cv2.resize(imgArray[x][y], (imgArray[0][0].shape[1], imgArray[0][0].shape[0]), None, scale,
                if len(imgArray[x][y].shape) == 2: imgArray[x][y]= cv2.cvtColor( imgArray[x][y], cv2.COLOR_GRAY2BGR)
        imageBlank = np.zeros((height, width, 3), np.uint8)
        hor = [imageBlank]*rows
        hor_con = [imageBlank]*rows
        for x in range(0, rows):
            hor[x] = np.hstack(imgArray[x])
        ver = np.vstack(hor)
    else:
        for x in range(0, rows):
            if imgArray[x].shape[:2] == imgArray[0].shape[:2]:
                imgArray[x] = cv2.resize(imgArray[x], (0, 0), None, scale, scale)
            else:
                imgArray[x] = cv2.resize(imgArray[x], (imgArray[0].shape[1], imgArray[0].shape[0]), None,scale, scale)
            if len(imgArray[x].shape) == 2: imgArray[x] = cv2.cvtColor(imgArray[x], cv2.COLOR_GRAY2BGR)
        hor= np.hstack(imgArray)
        ver = hor
    return ver
```

```python
def getContours(img):
    contours,hierarchy = cv2.findContours(img,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_NONE)
    for cnt in contours:
        area = cv2.contourArea(cnt)
        print(area)
        if area>500:
            cv2.drawContours(imgContour, cnt, -1, (255, 0, 0), 3)
            peri = cv2.arcLength(cnt,True)
            #print(peri)
            approx = cv2.approxPolyDP(cnt,0.02*peri,True)
            print(len(approx))
            objCor = len(approx)
            x, y, w, h = cv2.boundingRect(approx)

            if objCor ==3: objectType ="Tri"
            elif objCor == 4:
                aspRatio = w/float(h)
                if aspRatio >0.98 and aspRatio <1.03: objectType= "Square"
                else:objectType="Rectangle"
            elif objCor>4: objectType= "Circles"
            else:objectType="None"



            cv2.rectangle(imgContour,(x,y),(x+w,y+h),(0,255,0),2)
            cv2.putText(imgContour,objectType,
                        (x+(w//2)-10,y+(h//2)-10),cv2.FONT_HERSHEY_COMPLEX,0.7,
                        (0,0,0),2)
```

```python
path = './img.png'
img = cv2.imread(path)
imgContour = img.copy()

imgGray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
imgBlur = cv2.GaussianBlur(imgGray,(7,7),1)
imgCanny = cv2.Canny(imgBlur,50,50)
getContours(imgCanny)

imgBlank = np.zeros_like(img)
imgStack = stackImages(0.8,([img,imgGray,imgBlur],
                            [imgCanny,imgContour,imgBlank]))

cv2.imshow("Stack", imgStack)
cv2.waitKey(0)
```