#importing libraries

import numpy as np

import pandas as pd

from matplotlib import pyplot as plt

import cv2

import imutils

import pytesseract

#reading image

img = cv2.imread('i1.jpg')

#performing grayscaling

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

plt.imshow(cv2.cvtColor(gray, cv2.COLOR\_BGR2RGB))

bfilter = cv2.bilateralFilter(gray, 11, 17, 17) #Noise reduction

edged = cv2.Canny(bfilter, 30, 200) #Edge detection

plt.imshow(cv2.cvtColor(edged, cv2.COLOR\_BGR2RGB))

#finding contours

keypts = cv2.findContours(edged.copy(), cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)

conts = imutils.grab\_contours(keypts)

conts = sorted(conts, key=cv2.contourArea, reverse=True)[:10]

location = None

for contour in conts:

approx = cv2.approxPolyDP(contour, 10, True)

if len(approx) == 4:

location = approx

break

location

#masking

mask = np.zeros(gray.shape, np.uint8)

new\_img = cv2.drawContours(mask, [location], 0,255, -1)

new\_img = cv2.bitwise\_and(img, img, mask=mask)

plt.imshow(cv2.cvtColor(new\_img, cv2.COLOR\_BGR2RGB))

(x,y) = np.where(mask==255)

(x1, y1) = (np.min(x), np.min(y))

(x2, y2) = (np.max(x), np.max(y))

cropped\_img = gray[x1:x2+1, y1:y2+1]

plt.imshow(cv2.cvtColor(cropped\_img, cv2.COLOR\_BGR2RGB))

pytesseract.pytesseract.tesseract\_cmd = r'C:\Program Files (x86)\Tesseract-OCR\tesseract'

# Run tesseract OCR on image

text = pytesseract.image\_to\_string(cropped\_img, lang= 'eng')

text