import cv2  
import numpy as np  
import pytesseract  
import os  
  
# img = cv2.imread('hand.png')  
# Adding custom options  
custom\_config = r'--oem 3 --psm 1 -c preserve\_interword\_spaces=1'  
UPLOAD\_FOLDER = 'static/uploads/'  
  
  
# ocr\_.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER  
# ans=pytesseract.image\_to\_string(img, config=custom\_config)  
  
# print(ans[:-1])  
  
# noise removal  
def remove\_noise(image):  
 return cv2.medianBlur(image, 7)  
  
  
# opening - erosion followed by dilation  
def opening(image):  
 kernel = np.ones((7, 7), np.uint8)  
 return cv2.morphologyEx(image, cv2.MORPH\_OPEN, kernel)  
  
  
# closing - dilation followed by erosion  
def closing(image):  
 kernel = np.ones((5, 5), np.uint8)  
 return cv2.morphologyEx(image, cv2.MORPH\_CLOSE, kernel)  
  
  
def ocr\_driver(fname):  
 image = cv2.imread(os.path.join(UPLOAD\_FOLDER, fname))  
 ocr1 = pytesseract.image\_to\_string(image, config=custom\_config)  
 image\_med = remove\_noise(image)  
 # cv2.imshow('img',image\_med)  
 # cv2.waitKey(0)  
 ocr2 = pytesseract.image\_to\_string(image\_med, config=custom\_config)  
 image\_op = opening(image)  
 # cv2.imshow('img',image\_op)  
 # cv2.waitKey(0)  
 ocr3 = pytesseract.image\_to\_string(image\_op, config=custom\_config)  
 image\_cl = closing(image)  
 # cv2.imshow('img',image\_cl)  
 # cv2.waitKey(0)  
 ocr4 = pytesseract.image\_to\_string(image\_cl, config=custom\_config)  
 return ocr1[:-1], ocr2[:-1], ocr3[:-1]  
  
  
def ocr\_fun(fname):  
 image = cv2.imread(os.path.join(UPLOAD\_FOLDER, fname))  
 ocr1 = pytesseract.image\_to\_string(image, config=custom\_config)  
 # print(ocr1)  
 return ocr1  
  
  
# ans=ocr\_driver(img)  
# for \_ in ans:  
# print(\_)  
  
cv2.destroyAllWindows()