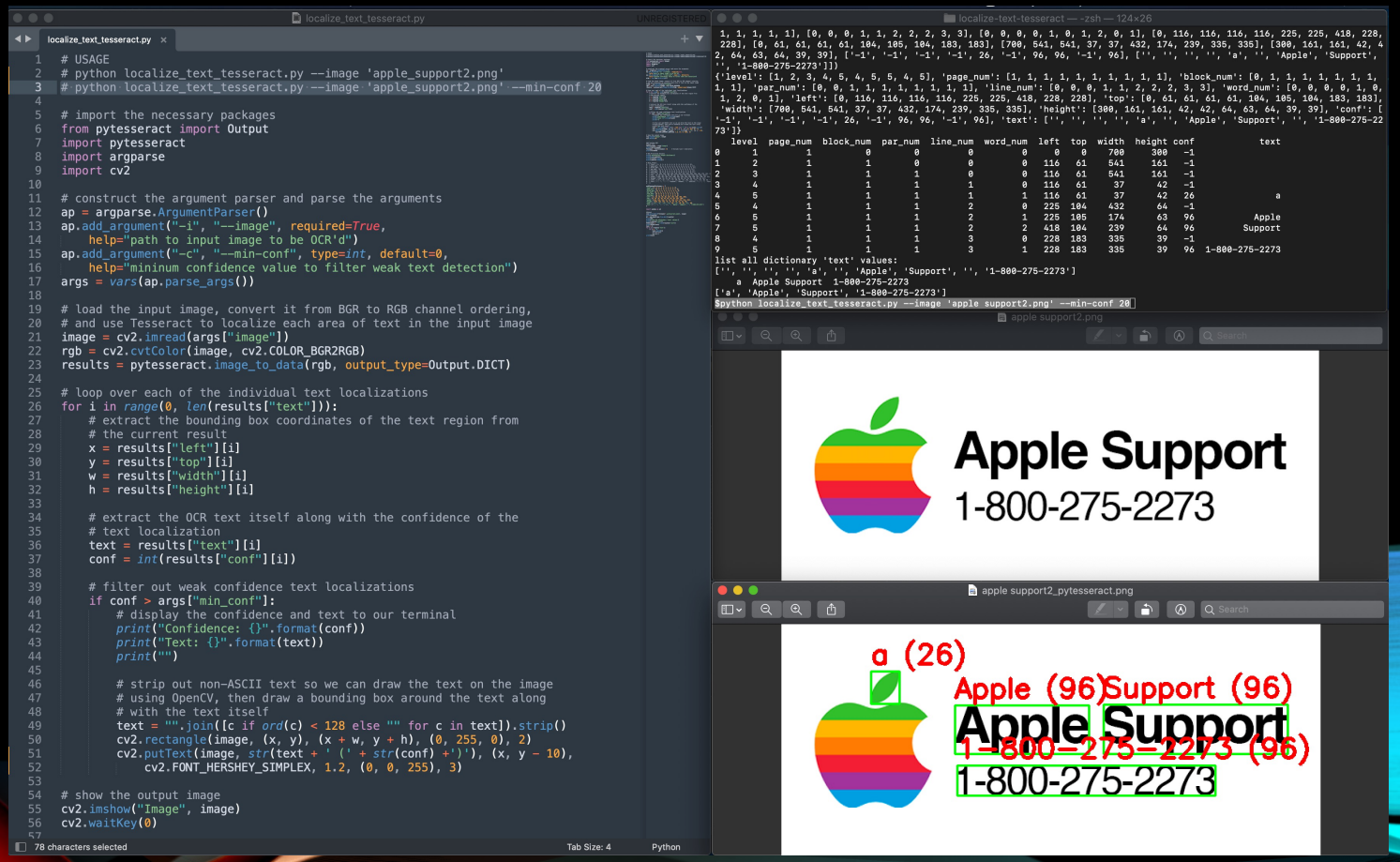
Optical Character Recognition (OCR)

OCR WITH TESSERACT

With Python



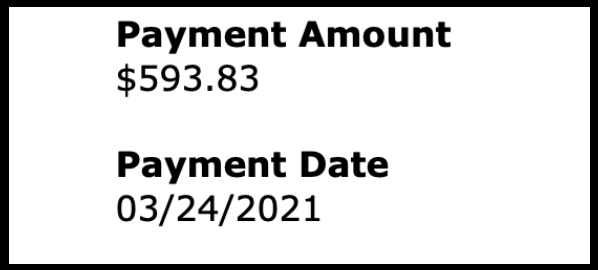
OCR WITH TESSERACT
RESULTS

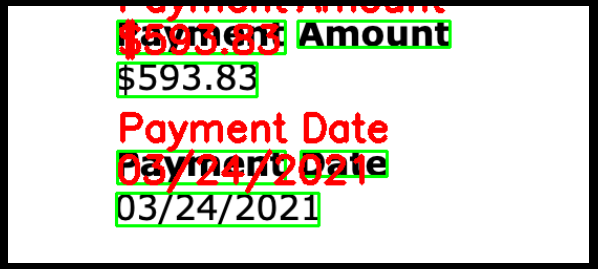
Example 1
image / screenshot

['a', 'Apple', 'Support', '1-800-275-2273']

Example 2 .PNG
pdf / screenshot of web receipt

['Payment', 'Amount', '$593.83', 'Payment', 'Date', '03/24/2021']





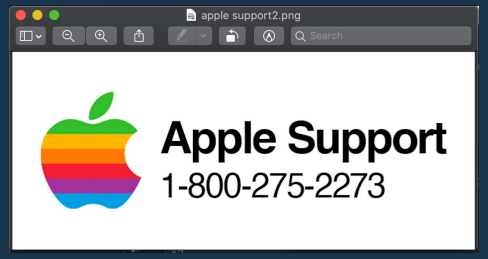
Example 3 .JPG
Large digital photo

[' ', ' ', ' ', '®', 'iy', 'See', 'Ss', '~—sCOTRAIL', '&', 'ae', 'ue', 'OMe', 'ae', 'val', 'i(s', ' ', ' ', ' ', ' ', ' ', ' ', 'Elevation', 'Change', '=', '37', 'feet', ' ', 'ie', '|', ' ', ' ', ' ', ' ', ' ', 'me:', '~', 'Lish', 'Yet]', 'WA', 'Ae', 'Boardwalk', ')', 'ase', 'a', 'aces', 'Typical', 'ee', 'Surface', 'Vy',

OCR WITH TESSERACT
OVERVIEW

Inputs

- Image Filename



- Confidence Threshold [-1 to 100]

Python Script

- Read / Process input image
- Draw box around image text
- Display result text and confidence

```
localize_text_tesseract.py
# USAGE
# python localize_text_tesseract.py --image 'apple_support2.png'
# python localize_text_tesseract.py --image 'apple_support2.png' --min-conf 20

# import the necessary packages
from pytesseract import Output
import pytesseract
import argparse
import cv2

# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--image", required=True,
    help="path to input image to be OCR'd")
ap.add_argument("-c", "--min-conf", type=int, default=0,
    help="mininum confidence value to filter weak text detection")
args = vars(ap.parse_args())

# load the input image, convert it from BGR to RGB channel ordering,
# and use Tesseract to localize each area of text in the input image
image = cv2.imread(args["image"])
rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
results = pytesseract.image_to_data(rgb, output_type=Output.DICT)

# loop over each of the individual text localizations
for i in range(0, len(results["text"])):
    # extract the bounding box coordinates of the text region from
    # the current result
    x = results["left"][i]
    y = results["top"][i]
    w = results["width"][i]
    h = results["height"][i]

    # extract the OCR text itself along with the confidence of the
    # text localization
    text = results["text"][i]
    conf = int(results["conf"][i])

    # filter out weak confidence text localizations
    if conf > args["min_conf"]:
        # display the confidence and text to our terminal
        print("Confidence: {}".format(conf))
        print("Text: {}".format(text))
        print("")

        # strip out non-ASCII text so we can draw the text on the image
        # using OpenCV, then draw a bounding box around the text along
        # with the text itself
        text = "".join([c if ord(c) < 128 else "" for c in text]).strip()
        cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)
        cv2.putText(image, str(text + ' (' + str(conf) +')'), (x, y - 10),
            cv2.FONT_HERSHEY_SIMPLEX, 1.2, (0, 0, 255), 3)

# show the output image
cv2.imshow("Image", image)
cv2.waitKey(0)
```

Outputs

- Dictionary of OCR Results

```
{
'level': [1, 2, 3, 4, 5, 4, 5, 5, 4, 5],
'page_num': [1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
'block_num': [0, 1, 1, 1, 1, 1, 1, 1, 1, 1],
'par_num': [0, 0, 1, 1, 1, 1, 1, 1, 1, 1],
'line_num': [0, 0, 0, 1, 1, 2, 2, 2, 3, 3],
'word_num': [0, 0, 0, 0, 1, 0, 1, 2, 0, 1],
'left': [0, 88, 88, 88, 88, 197, 197, 390, 200, 200],
'top': [0, 39, 39, 39, 39, 82, 83, 82, 161, 161],
'width': [644, 541, 541, 37, 37, 432, 174, 239, 335, 335],
'height': [256, 161, 161, 42, 42, 64, 63, 64, 39, 39],
'conf': ['-1', '-1', '-1', '-1', 26, '-1', 96, 96, '-1', 96],
'text': ['', '', '', '', 'a', '', 'Apple', 'Support', '', '1-800-275-2273']
}
```

- OCR Text

['a', 'Apple', 'Support', '1-800-275-2273']

- Output Image(s)


```
# python localize_text_tesseract.py --image 'apple support2.png' --min-conf 20
```