Batch OCR Parsing with Tesseract and Python

Calvin Houser

The Process

Tesseract OCR:
https://github.com/tesseract-ocr/tesseract

Hewlett Packard: 1985-98, Google: 2006-18

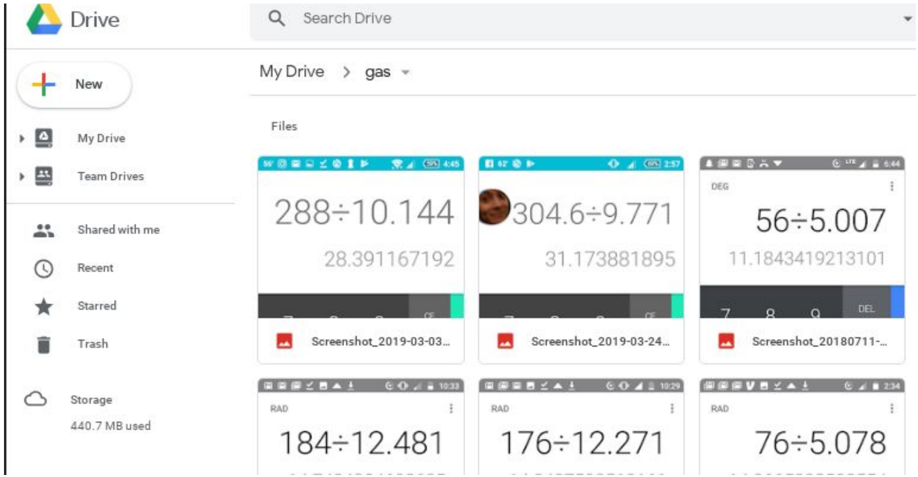
Tesseract 4.0 - LSTM (Recurrent Neural Network architecture, Long Short-Term Memory)

PyTesseract:
https://pypi.org/project/pytesseract/

OpenCV: https://docs.opencv.org/
https://pypi.org/project/opencv-python/

The Data

A set of images, each with useful text in a consistent format.



Reading an image with OpenCV and Tesseract

The images are all stored in subdirectory ‘gas’

OpenCV generates a python object representing the image

pyTesseract generates a string object representing the parsed image text

Image Date

Calculated MPG

```
import cv2
import pytesseract
from os import listdir

img_files = ['./gas/' + name for name
         in listdir(path='./gas')]

im = cv2.imread(img_files[2],
         cv2.IMREAD_COLOR)
text = pytesseract.image_to_string(im,
         config='-l eng --oem 1 --psm
    3')
print(img_files[2],text.splitlines())

OUT:./gas/Screenshot_20180816-102947.pn
g['M@maArvalt @ iD 4 & 10:29', '',
'176412.271', '', '14.3427593513161',
'', ' ', '', 'nn', 'Cr', 'Cr:']
```

Extracting the Relevant Data (Part 1)

```
for image in img_files:
    text = pytesseract.image_to_string(
        cv2.imread(image, cv2.IMREAD_COLOR),
        config='-l eng --oem 1 --psm 3')
    print(image[17:25])
    for line in text.splitlines():
        decimal_idx = line.find('.')
        if decimal_idx == -1: pass
        elif (line[:decimal_idx] +
             line[decimal_idx+1:]).isdigit():
            print(line); break
```

```
OUT:                (continued)
20180803          20190324
15.6588953995761  31.173881895
20180830          20180821
14.5772932853995  20180906
20180816          13.1880385200202
176412.271        20180711
20180823          11.18438419213101
15.4868987094251  20190504
20180815          30.27913380898735
14.7424084608605  20180829
20180929          13.9344262295081
18.6072491682717  20190414
20180909          20180813
13.4028892455858  14.7424084608605
20180914          20190430
13.6540688493525  20190303
```

Checking for Mistakes

```
im1 = cv2.imread(img_files[9], cv2.IMREAD_COLOR)
text = pytesseract.image_to_string(im1, config='-l eng --oem 1 --psm 3')

im2 = cv2.imread(img_files[14], cv2.IMREAD_COLOR)
text2 = pytesseract.image_to_string(im2, config='-l eng --oem 1 --psm 3')

im3 = cv2.imread(img_files[16], cv2.IMREAD_COLOR)
text3 = pytesseract.image_to_string(im3, config='-l eng --oem 1 --psm 3')

im4 = cv2.imread(img_files[17], cv2.IMREAD_COLOR)
text4 = pytesseract.image_to_string(im4, config='-l eng --oem 1 --psm 3')
print(text.splitlines()); print(text2.splitlines())
print(text3.splitlines()); print(text4.splitlines())
OUT: []
['aS Sg me eed Rae', '', '°', '°', 'e', '', '168.1+6.181', '',
'2/7.19624656204497', '', ' ', '', 'nn', 'nn', 'O = +¢']
['c.f reo — ON KL a', '', ' ', '', 'ene', '1 2 3', '0 = +']
['cOMNY ORE Bal', '', ' ', '', ' ', '', ' ', '', ' ', '', 'EY', '', '288
710.144', '', '28.39116/192', '', ' ', '', 'a 5 6', '1 W )', '0 = +']
```

Extracting the Relevant Data (Part 2)

```
import matplotlib.dates as dates; from datetime import date

data_points = []
for image in img_files:
    text = pytesseract.image_to_string(cv2.imread(image, cv2.IMREAD_COLOR),
            config='-l eng --oem 1 --psm 3')
    year = int(image[17:21])
    month = int(image[21:23])
    day = int(image[23:25])
    img_ts = date(year, month, day)
    for line in text.splitlines():
        line2 = line.replace('/','')
        decimal_idx = line2.find('.')
        if decimal_idx == -1: pass
        if (line2[:decimal_idx] + line2[decimal_idx+1:]).isdigit()
                and float(line2) < 50:
            data_points.append((dates.date2num(img_ts), float(line2)))
```

Viewing the Results

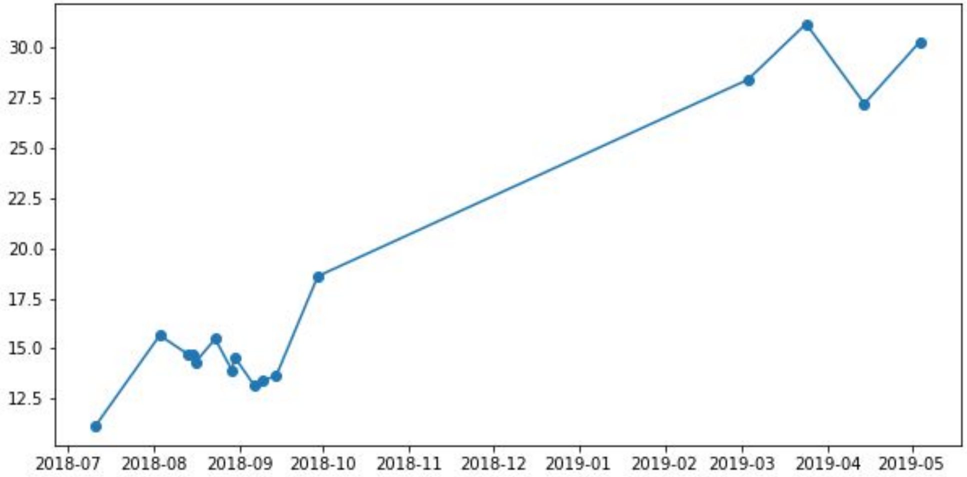
```
sorted_points = sorted(data_points,
         key=lambda tup: tup[0])
print(*sorted_points, sep="\n")
```

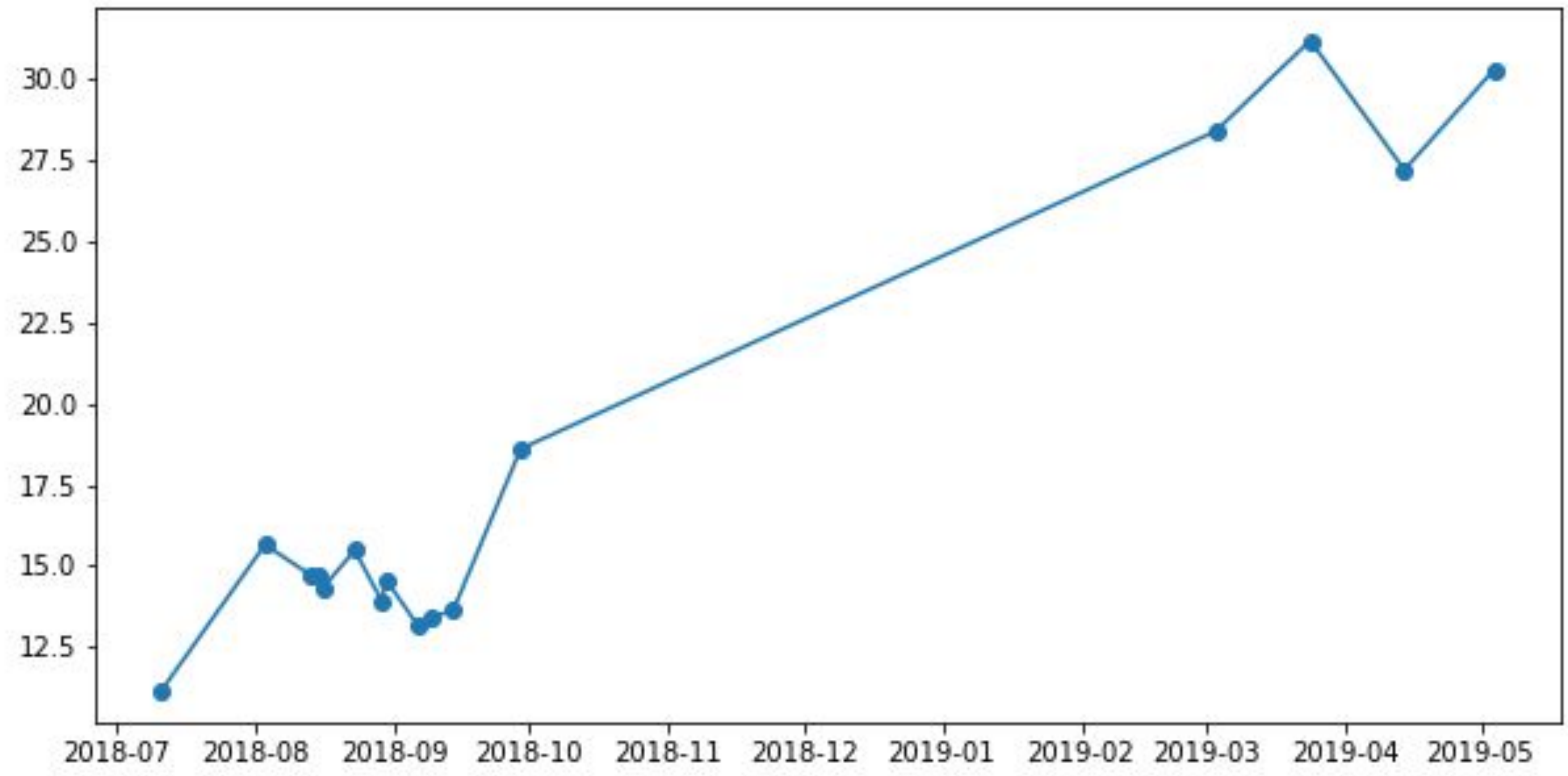
```
OUT:
(736886.0, 11.18438419213101)
(736909.0, 15.6588953995761)
(736919.0, 14.7424084608605)
(736921.0, 14.7424084608605)
(736922.0, 14.3427593513161)
(736929.0, 15.4868987094251)
(736935.0, 13.9344262295081)
(736936.0, 14.5772932853995)
(736943.0, 13.1880385200202)
(736946.0, 13.4028892455858)
(736951.0, 13.6540688493525)
(736966.0, 18.6072491682717)
(737121.0, 28.39116192)
(737142.0, 31.173881895)
(737163.0, 27.19624656204497)
(737183.0, 30.27913380898735)
```

```
import matplotlib.pyplot as plt

#https://stackoverflow.com/questions/18458734/
#python-plot-list-of-tuples
plt.plot_date(*zip(*sorted_points),
         linestyle='solid')

plt.savefig(img_files[0][17:25]+'-'
         +img_files[-1][17:25]+'.png')
```




30.0
27.5
25.0
22.5
20.0
17.5
15.0
12.5
2018-07
2018-08
2018-09
2018-10
2018-11
2018-12
2019-01
2019-02
2019-03
2019-04
2019-05