|  |
| --- |
| import numpy as np  from PIL import Image |
| import pytesseract |
| import urllib |
| import cv2 |
| import json |
|  |
|  |
| def ExtractNumbers(List): |
| for index,Row in List.iterrows(): |
| # Get the image from the URL |
| Response = urllib.request.urlopen(Row[0]) |
| Img = np.array(Image.open(Response)) |
|  |
| # Storing images to train |
| Images.append(Img) |
|  |
| # Points of rectangle to crop the full image |
| xt = Row[1][0]['x']\*Img.shape[1] |
| yt = Row[1][0]['y']\*Img.shape[0] |
| xb = Row[1][1]['x']\*Img.shape[1] |
| yb = Row[1][1]['y']\*Img.shape[0] |
|  |
| # Storing image in fullImage |
| fullImage = Image.fromarray(Img) |
|  |
| # Cut the number plate from the fullImage |
| plateImage = fullImage.crop((xt,yt,xb,yb)) |
|  |
| # Storing cropped image |
| Plates.append(np.array(plateImage)) |
| plateImage.save('image.png') |
| img=cv2.imread('image.png') |
|  |
| # As the cropped images size is too small in size |
| # So that we need to increase the size to get proper result |
| img=cv2.resize(img,(int(img.shape[1]\*4),int(img.shape[0]\*4))) |
|  |
| # Pytesseract return characters from the given image |
| Num=pytesseract.image\_to\_string(img) |
|  |
| # It print characters given in the number plate at |
| print(Num) |
|  |
| # Storing Numbers extracted from the image, so we can use it later to improve the program efficiency. |
| Numbers.append(Num) |
|  |
|  |
| # It is requires later to train the program |
| Images =[] #declaring Images to store all the images of queries |
| Plates =[] #declaring Plates to store the cropped part of Image given in the JSON input file |
| Numbers=[] #declaring Numbers to store the results |
|  |
| #main method |
| if \_\_name\_\_== "\_\_main\_\_": |
|  |
| # List is used to read JSON file input |
| # Note:- Use the path where json file is available. In my case it is in the same directory |
| List = pd.read\_json('Indian\_Number\_plates.json', lines=True) |
| pd.set\_option('display.max\_colwidth', -1) |
|  |
| # Removing unwanted extras |
| del List['extras'] |
|  |
| # Organizing all the data in a List |
| List['points'] = List.apply(lambda Row: Row['annotation'][0]['points'], axis=1) |
|  |
| # Removing annotation |
| del List['annotation'] |
|  |
| # Calling method ExtractNumbers to get the characters from the given number plate. |
| ExtractNumbers(List) |