

MODUL PYTHON

PENGOLAH CITRA MAIN

Penerapan Optical Character Recongntion

Berikut contoh penerapan optical character recongntion pada plat, dapat dilihat pada contoh program dibawah dimana gambar plat dengan menggunakan OCR dapat diekstrak menjadi suatu teks.

```
Optical Character.ipynb
File Edit View Insert Runtime Tools Help All changes saved
RAM
Disk

Files
{ }
sample_data
Plat B.jpeg

+ Code + Text
import cv2
from matplotlib import pyplot as plt
import numpy as np
import imutils
import easyocr

[ ] img = cv2.imread('/content/Plat B.jpeg')
#UBAH GAMBAR WARNA KE GRAYSCALE
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
#TAMPILKAN MENGGUNAKAN LIBRARY MATPLOTLIB
plt.imshow(cv2.cvtColor(gray, cv2.COLOR_BGR2RGB))

[ ] <matplotlib.image.AxesImage at 0x7f535bac6e30>

[ ]
0
50
100
150
200
250
300
0 100 200 300 400

#SOMETHING GAMBAR DENGAN BILATERAL FILTER
bfilter = cv2.bilateralFilter(gray, 11, 17, 17)
#EDGE DETECTION DENGAN METODE CANNY
edge = cv2.Canny(bfilter, 30, 100)
plt.imshow(cv2.cvtColor(edge, cv2.COLOR_BGR2RGB))

<matplotlib.image.AxesImage at 0x7f535b5a5a0>

[ ]
0
50
100
150
200
250
300
0 100 200 300 400
```

```

[ ] """
FIND CONTOURS MENDETEKSI PERUBAHAN WARNA GAMBAR DAN MENANDAINYA SEBAGAI KONTUR.
GAMBAR ATAU ANGKA YANG TERTULIS PADA CITRA AKAN DIDETEKSI SEBAGAI KONTUR
"""
keypoints = cv2.findContours(edge.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
contours = itertools.chain(*keypoints)
contours = sorted(contours, key=cv2.contourArea, reverse=True)[:10]

```

```

[7] """
APPROXPOLYDP MEHUNGKINAN PERKIRAAN POLIGON (GAMBAR BANYAK SUDUT)
JADI JIKA GAMBAR BERISI POLIGON, FUNSI INI AKAN CUKUP AKURATMENDETEKSIINYA.
"""
location = None
for contour in contours:
    approx = cv2.approxPolyDP(contour, 10, True)
    if len(approx)==4:
        location = approx
        break

```

```

[8] location

array([[ 39,  61]],
       [[484,  57]],
       [[479, 165]],
       [[ 55, 170]]], dtype=int32)

```

```

[9] """
DRAWCONTOURS BERFUNKSI UNTUK MELAPISI KONTUR PADA GAMBAR RGB ASLI
"""
mask = np.zeros(gray.shape, np.uint8)
new_image = cv2.drawContours(mask, [location], 0, 255, -1)
new_image = cv2.bitwise_and(img, img, mask=mask)

```

```

[10] plt.imshow(cv2.cvtColor(new_image, cv2.COLOR_BGR2RGB))

```



```

[11] #CIPPING GAMBAR
(x,y) = np.where(mask==255)
(x1, y1) = (np.min(x), np.min(y))
(x2, y2) = (np.max(x), np.max(y))
cropped_image = gray[x1:x2+1, y1:y2+1]

```

```

[12] plt.imshow(cv2.cvtColor(cropped_image, cv2.COLOR_BGR2RGB))

```



```

[13] #PENERAPAN LIBRARY EASYOCR
reader = easyocr.Reader(['en'])
result = reader.readtext(cropped_image)
result

WARNING:easyocr.easyocr: CUDA not available - defaulting to CPU. Note: This module is much faster with a GPU.
WARNING:easyocr.easyocr: Downloading detection model, please wait. This may take several minutes depending upon your network
Progress: [100%] 100.0% CompleteWARNING:easyocr.easyocr: Downloading recognition model, please wait. This may take several minutes depending upon your network
Progress: [100%] 100.0% Complete[[['13', [259, 3], [259, 89], [13, 89]],
(['285, 1], [437, 1], [437, 91], [285, 91]], 'EEK', 0.9970115756101692),
(['176, 84], [274, 84], [274, 110], [176, 110]],
'10 - 16',
0.3899546607559775]]

```

```
✓ [14] 0s """
DARI KONVERSI OCR TERDAPAT 3 LIST (DALAM BAHASA PEMROGRAMAN LAIN DISEBUT DENGAN ARRAY),
UNTUK MENGELIMINASI BILANGAN YANG TIDAK PENTING DAN HANYA MENGAMBIL VALUE HASIL OCR,
TULISKAN KODE DIBAWAH.

PENULISAN KODE DIBAWAH DISESUAIKAN DENGAN HASIL OCR,
APABILA 2 LIST, HANYA TULISKAN SAMPAI LIST KEDUA,
APABILA TIGA LIST, HANYA TULISKAN SAMPAI LIST KETIGA,
"""

text1 = str.upper(result[0][1])
text2 = str.upper(result[1][1])
text3 = str.upper(result[2][1])

✓ [15] 0s #TAMPILKAN ISI VARIABEL TEXT
text1, text2, text3

('B 313', 'EEK', '10 . 16')
```

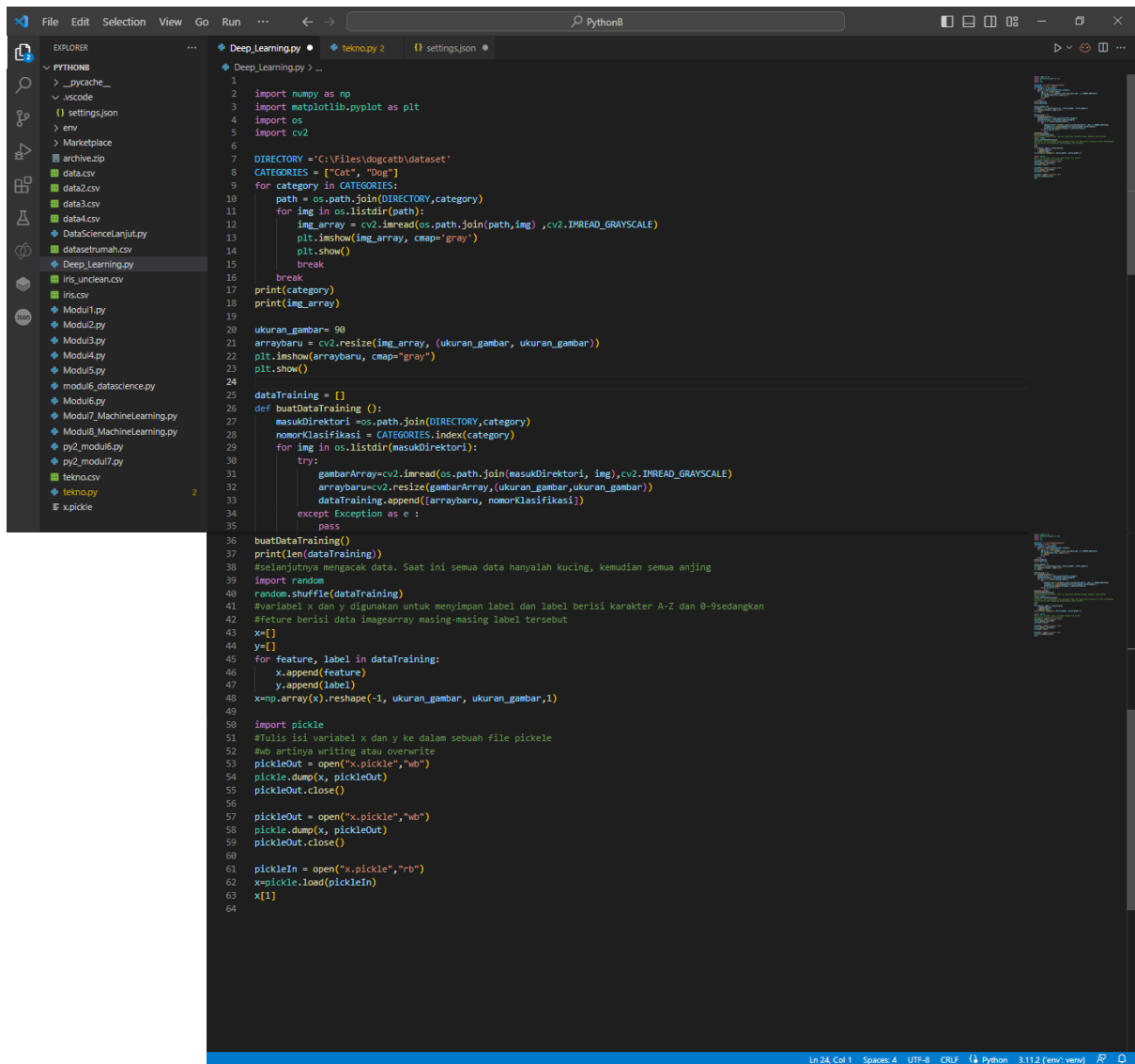
✓ 0s completed at 8:32 AM



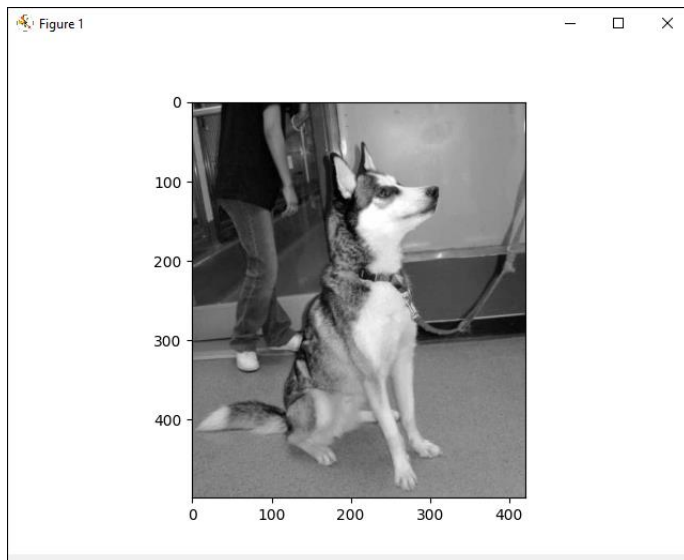
MODUL PYTHON

PENGOLAH CITRA MAIN

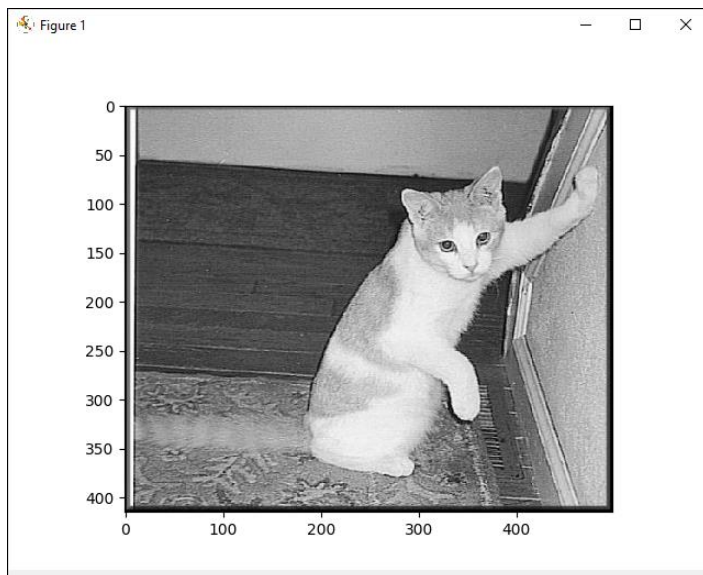
Cat and Dog



```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import os
4 import cv2
5
6 DIRECTORY = 'C:\Files\dogcat\b\dataset'
7 CATEGORIES = ["Cat", "Dog"]
8 for category in CATEGORIES:
9     path = os.path.join(DIRECTORY, category)
10    for img in os.listdir(path):
11        img_array = cv2.imread(os.path.join(path, img), cv2.IMREAD_GRAYSCALE)
12        plt.imshow(img_array, cmap='gray')
13        plt.show()
14        break
15    print(category)
16    print(img_array)
17
18 ukuran_gambar = 90
19 arraybaru = cv2.resize(img_array, (ukuran_gambar, ukuran_gambar))
20 plt.imshow(arraybaru, cmap='gray')
21 plt.show()
22
23 dataTraining = []
24 def buatDataTraining():
25     masukDirektori = os.path.join(DIRECTORY, category)
26     nomorKlasifikasi = CATEGORIES.index(category)
27     for img in os.listdir(masukDirektori):
28         try:
29             gambarArray = cv2.imread(os.path.join(masukDirektori, img), cv2.IMREAD_GRAYSCALE)
30             arraybaru = cv2.resize(gambarArray, (ukuran_gambar, ukuran_gambar))
31             dataTraining.append([arraybaru, nomorKlasifikasi])
32         except Exception as e:
33             pass
34
35 buatDataTraining()
36 print(len(dataTraining))
37
38 #selanjutnya mengacak data. Saat ini semua data hanyalah kucing, kemudian semua anjing
39 import random
40 random.shuffle(dataTraining)
41 #variabel x dan y digunakan untuk menyimpan label dan label berisi karakter A-Z dan 0-9sedangkan
42 #feature berisi data Imagearray masing-masing label tersebut
43 x=[]
44 y=[]
45 for feature, label in dataTraining:
46     x.append(feature)
47     y.append(label)
48 x=np.array(x).reshape(-1, ukuran_gambar, ukuran_gambar, 1)
49
50 import pickle
51 #Tulis isi variabel x dan y ke dalam sebuah file pickle
52 #wb artinya writing atau overwrite
53 pickleOut = open("x.pickle", "wb")
54 pickle.dump(x, pickleOut)
55 pickleOut.close()
56
57 pickleOut = open("x.pickle", "wb")
58 pickle.dump(x, pickleOut)
59 pickleOut.close()
60
61 pickleIn = open("x.pickle", "rb")
62 x=pickle.load(pickleIn)
63 x[1]
```



Dog



Cat