

画像選択の機械学習データ作成プログラム (Data_FigSelection.py) 使用説明書

目次

1. 機能
2. 動作環境
3. 概要
 - 3.1 構造
 - 3.2 プログラムリストの概要
 - 3.3 入出力
 - 3.3.1 入力
 - 3.3.2 出力
 - 3.3.3 データの出力形式
4. プログラムリスト

1. 機能

このプログラム(Data_FigSelection.py)は、予め行われた画像判別結果を元に、機械学習用データを生成する。生成されたデータは機械学習(Learning_FigSelection.py)で使用する

2. 動作環境

1) このプログラムはWindows10, Ubuntu18-04での動作が確認されている

2) このプログラムを実行するには、以下のモジュールがあらかじめ準備された環境が必要である。また、記載されている以外のVersionでの動作確認は行っていないので注意が必要である

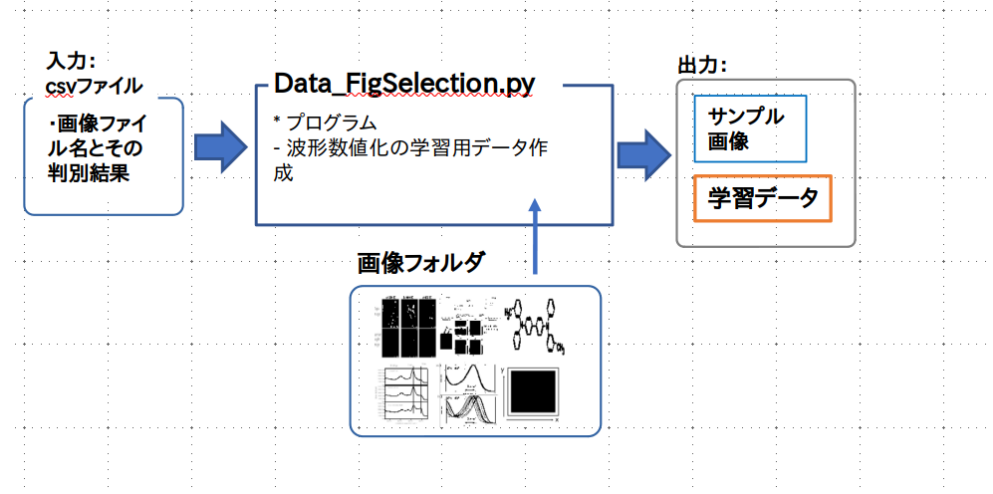
- Python 3.6
- Tensorflow-gpu 1.4.0
- Keras 2.2.4
- sklearn 0.19.1, 0.21.2
- matplotlib 2.2.2

3) このプログラムには、画像ファイルを収納したフォルダと、この各画像ファイルに対して合否判別を行った判別結果をまとめたcsvファイルが必要である

3. 構造

3.1 概要

このプログラムは、画像ファイルとその合否判別結果 からなるcsvファイルを元に、画像選別機械学習(Learning_FigSelection.py)に対応した学習データを生成する



3.2 プログラムリストの概要

プログラムリストの概要を示す。全リストは、5. プログラムリストに示す

Learning_FigSelection.py

1-8 行: 開始コメント 10-17 行: モジュールのインポート
19-30 行: 開始準備
32-48 行: CSVファイル読み込み
50-93 行: データ作成
95-130 行: データチェック
132-150 行: データの保存
152 行: END

3.3 入出力

以下、Jupyter Notebook上で実行された例を示して説明する

3.3.1 入力およびプログラムの実行

このプログラムを実行するには、画像リストとその合否判別結果からなるcsvファイルが予め作られていることが必要である

svcファイルの構造

svcファイルの構造は、N行(データ数N-1の場合)、2列で、第1行目にコメント、2行以降がデータで、第1列に画像ファイル名、第2列にその合否判定が記述される

合否判定: Pass = 1, NG = 0もしくは空白, 保留 = 0.5

```

1 name, OK1/NG0
2 acs-0. png,
3 acs-1. png, 0.5
4 acs-2. png,
5 acs-3. png, 1
6 acs-4. png,
7 acs-5. png,
8 acs-6. png, 1
9 acs-7. png,
10 acs-8. png,
11 acs-9. png, 1
12 acs-10. png, 1
13 acs-11. png,
14 acs-12. png,
15 acs-13. png,
16 acs-14. png,
17 acs-15. png,
18 acs-16. png,
19 acs-17. png,

```

プログラムの実行

プログラム実行は、下記のように、36行目にcsvファイルのディレクトリ名を、37行目にcsvファイルのファイル名を入力して保存し、<2. 動作環境>を満たした環境下でPython上で実行する

```

35 # Jading data read
36 dir_n = 'D:\TDM_DATA\TDM-PNG_k2_n4207/'
37 d_n = 'OK-NG_k2'.

```

3.3.2 出力

計算が終了すると、作成されたデータの中から不適合と判断された画像(下図 NG_figure_sample)と適合すると判断された画像(下図 OK_figure_sample)から、それぞれ8個づつがランダムに選ばれ表示される

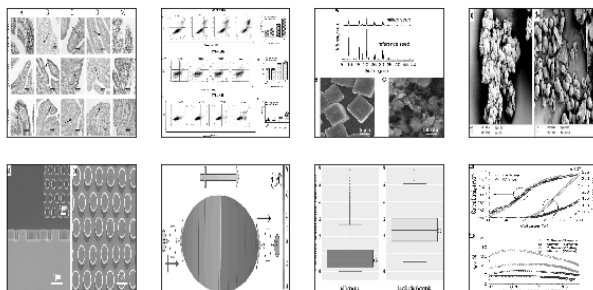
< Data Creating of Machine Learning for FigSelection 200628-1403 >

```

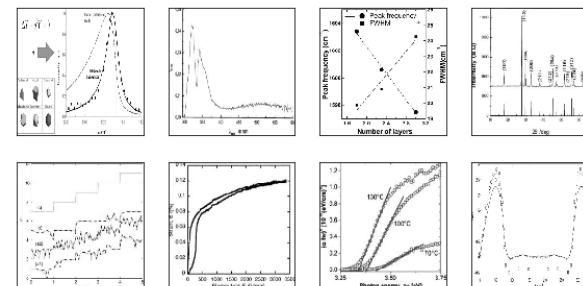
Dimendion = 256
Data directory : /media/tkono/USB_115G/TDM-PNG_n2_n4207/
Judge List : OK-NG_test.csv

```

< NG figure Sample > [132 148 95 106 297 187 130 294]



< OK figure Sample > [9 73 56 59 42 70 36 38]



次に、画像の個数が表示され、データを保存するかどうかの待機待ち-->になる
保存する場合は'y'を入力すると、データは保存され最終段にデータ名が表示され、プログラムは終了する

```

< Results >
Total = 500
OK = 76
NG = 424
Data save ? y or n --> y
Data was saved : ./data/FigSelection_256_500_200628-1403.pik

```

3.3.3 データの出力形式

以下に出力されるデータの形式について説明する

参照:4.1 データ作成本体(Data_FigSelection.py)プログラムリスト,133-151 行:モデルの保存 及び A1_Learning_FigSelection_Manyual、3.3.1 入力

- pickle形式で保存された辞書型のデータ <データ名.pik >
- 'data','out','para'の3つのキーを持ち、下表のような要素で構成されている

key	item
data	画像データ
out	教師データ
para	画像の大きさ, データの数

画像データと教師データの形式

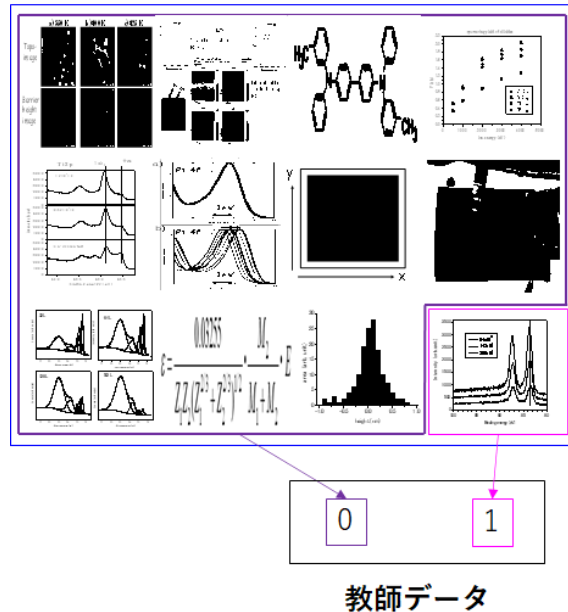
画像データ:

0-255の整数の数値を取り、そのサイズが256x256で1チャンネル(グレイタイプ)の画像配列(256,256,1)

教師データ:

0: 適合しない,1:適合するのどちらかの整数値を取る1個の整数配列(1)

画像データ (256 x 256)



5. プログラムリスト

5.1 Learning_FigSelection.py

1-8 行: 開始コメント
 10-17 行: モジュールのインポート
 19-30 行: 開始準備
 32-48 行: CSVファイル読み込み
 50-93 行: データ作成
 95-130 行: データチェック
 132-150 行: データの保存
 152 行: END

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Oct.16,2019 / Fixed on Jun.30,2020
4
5 < csv-file の判別結果をもとに画像選別の機械学習データを作成 >
6
7 @author: T.KONO
8 """
9
10 import numpy as np
11 import pandas as pd

```

```

12 import pickle
13 import matplotlib.pyplot as plt
14
15 import cv2
16
17 from datetime import datetime
18
19 # -----
20 #
21 #       Data Creating of Machine Learning for FigSelection
22 #       - from jading data of csv-file
23 #
24 # -----
25
26 # --- Opning -----
27 # date and time
28 now = datetime.today().strftime('%y%m%d-%H%M')
29 # start comment
30 print('\n< Data Creating of Machine Learning for FigSelection', now,'>')
31
32 # --- Data Read -----
33 # dimansion
34 dim = 256
35 # Jading data read
36 dir_n = 'D:\TDM_DATA\TDM-PNG_k2_n4207/'
37 d_n = 'OK-NG_k2'
38 fd_n = dir_n+d_n+'.csv'
39
40 print('\n Dimendion = ',dim,
41       '\n Data directory :',dir_n,
42       '\n Judge List : ',d_n+'.csv')
43
44 # データ読み出し(欠損値は0に置換)
45 df = pd.read_csv(fd_n).fillna(0)
46
47 # 判別結果を行列に
48 ar_d=df[['PN']].values
49
50 # --- Data Create -----
51 ng_out = []
52 ng_ima = []
53 ng_list = []
54 ok_out = []
55 ok_ima = []
56 ok_list = []
57
58 loop = 0
59 for f_n in df['name']:
60     # file name
61     f_name = dir_n+f_n+'.png'
62
63     # File Open
64     im_ = cv2.imread(f_name,0) # gray image
65     sz = im_.size
66
67     # Resise
68     re_ = cv2.resize(im_,(dim,dim),cv2.INTER_LANCZOS4)
69
70     # array
71     arrayimage = np.asarray(re_).astype(np.uint8)
72     arrayimage = arrayimage.reshape([dim,dim,-1])

```

```

73
74 # data append
75 fg = ar_d[loop,0]
76 if 1.1 > fg > 0.4 :
77     '''曖昧な画像を0.5と判定しているのでこれもOKに分類する'''
78     ...
79     ok_ima.append(arrayimage)
80     ok_out.append(1)
81     ok_list.append(loop)
82
83 else :
84     ng_ima.append(arrayimage)
85     ng_out.append(0)
86     ng_list.append(loop)
87
88     loop=loop+1
89
90 ok_ima = np.asarray(ok_ima)
91 ok_out = np.asarray(ok_out)
92 ng_ima = np.asarray(ng_ima)
93 ng_out = np.asarray(ng_out)
94
95 #--- Data Check -----
96 fig=plt.figure(figsize=(10,5))
97 n = 8
98 n_list= np.random.choice(len(ng_list), n)
99 print('\n< NG figure Sample >', n_list)
100 for i in range(n):
101     nn = n_list[i]
102     ax = plt.subplot(2,4, i+1)
103     plt.imshow(np.reshape(ng_ima[nn,:],(dim,- 1)))
104     plt.gray()
105     ax.get_xaxis().set_visible(False)
106     ax.get_yaxis().set_visible(False)
107 plt.show()
108
109 fig=plt.figure(figsize=(10,5))
110 n = 8
111 n_list= np.random.choice(len(ok_list), n)
112 print('\n< OK figure Sample >', n_list)
113 for i in range(n):
114     nn = n_list[i]
115     ax = plt.subplot(2,4, i+1)
116     plt.imshow(np.reshape(ok_ima[nn,:],(dim,- 1)))
117     plt.gray()
118     ax.get_xaxis().set_visible(False)
119     ax.get_yaxis().set_visible(False)
120 plt.show()
121
122 print('< Results >\n','Total = ',len(df),
123       '\n   OK = ',len(ok_out),
124       '\n   NG = ',len(ng_out)
125       )
126
127 # --- Creating Learning data -----
128 # learning data
129 x = np.concatenate([ok_ima, ng_ima], axis=0)
130 # Teaching data
131 y = np.concatenate([ok_out, ng_out], axis=0)
132
133 # --- Data save -----

```

```

134 fg = input('Data save ? y or n --> ')
135 if fg == 'y' :
136
137     test_data = {'in':x,
138                  'out':y,
139                  'para':(dim,str(len(x)))
140                  }
141
142     data_n = './data/FigSerection_'+str(dim)+'_'+str(
143         len(x))+str(now)+'.pik'
144
145     with open(data_n,'wb') as f1:
146         pickle.dump(test_data, f1, protocol=4)
147
148     print(' Data was saved :',data_n)
149
150 else :
151     print(' Data was NOT saved. ')
152
153 # _____ END OF CODE _____

```

End