

รายงาน
เรื่อง Face Mask Detector

เสนอ

ผศ.ดร.พิกุลแก้ว ตังติสานนท์

จัดทำโดย

นายกำธร แซ่เจ็ง รหัสนักศึกษา 60010062

นายกิตติชัย จักรารักษ์ รหัสนักศึกษา 60010063

รายงานนี้เป็นส่วนหนึ่งของวิชาปัญญาประดิษฐ์ รหัสวิชา 01236151

คณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2563

คำนำ

รายงานนี้เป็นส่วนหนึ่งของวิชา ปัญญาประดิษฐ์ รหัสวิชา 01236151 โดยจัดทำขึ้นเพื่อเป็นการนำความรู้ที่ได้รับในวิชานี้มาใช้ โดยได้นำมาประยุกต์ใช้กับการสร้างโปรแกรมที่สามารถบอกได้ว่าบุคคลที่กล้องตรวจจับได้สวมใส่หน้ากากอนามัยหรือไม่ และเพื่อสามารถนำไปประยุกต์ใช้ต่อ ๆ ไป

โดยทางผู้จัดทำหวังว่ารายงานนี้ จะเป็นประโยชน์แก่ผู้อ่าน เพื่อช่วยให้สามารถเข้าใจตัวชิ้นงาน ผลงานที่ทำได้มากขึ้น หากผิดพลาดประการใด ทางผู้จัดทำขออภัยมา ณ ที่นี้

กำธร

แซ่เจ็ง

กิตติชัย

จักราภิรักษ์

ผู้จัดทำ

Face mask detector

1. ที่มาและความสำคัญ

ที่มา

งานที่ทำคือนำ machine learning มาใช้ในการตรวจจับว่าบุคคลใส่หน้ากากอนามัยหรือไม่ ทำเรื่องนี้เพราะว่า ในปัจจุบันมีเรื่องของความสำคัญในการใส่หน้ากาก บางแห่งมีการตรวจการใส่หน้ากาก ก่อนเข้าสถานที่นั้น ๆ ด้วย อันเนื่องมาจากเหตุการณ์ COVID-19 ในปัจจุบัน และมี dataset เป็นจำนวนเพียงพอในการสร้าง model เพื่อนำมาใช้

วัตถุประสงค์

จัดทำขึ้นเพื่อนำไปใช้ตรวจจับได้ว่าบุคคลที่กำลังตรวจจับได้นั้นมีสวมใส่หน้ากากอนามัยหรือไม่

2. ส่วนที่เขียน ai เอง

โดยเริ่มจากที่ว่าตัว Face mask detector แบ่งโมเดลได้ 2 ส่วนคือ 1. ส่วนที่ใช้ในการตรวจจับใบหน้าคน 2. ส่วนที่ใช้ในการทำนายว่าบุคคลในภาพนั้นใส่หน้ากากหรือไม่ โดยในส่วนของโมเดลที่ 1 ได้ทำการนำโมเดลที่มีอยู่แล้วมาใช้ และทำการสร้างโมเดลในส่วนที่ 2 คือส่วนที่นำภาพมาทำนายว่าบุคคลในภาพใส่หน้ากากหรือไม่ โดยที่ใช้ Logistic Regression ในการทำ (ไฟล์ชื่อ predict_mask.ipynb)

โมเดลที่ 1 ใช้ในการจับตรวจใบหน้าคนโดยได้

ที่มา https://github.com/gopinath-balu/computer_vision/tree/master/CAFFE_DNN

การนำมาใช้

```
In [9]: # Load face detection and face mask model

with open ('outfile', 'rb') as fp:
    myw = pickle.load(fp)
|
faceNet = cv2.dnn.readNet(os.path.join(r'deploy.prototxt.txt'),
                           os.path.join('res10_300x300_ssd_iter_140000.caffemodel'))
```

โดยได้นำมาเก็บไว้ในตัวแปรที่ชื่อ faceNet

โมเดลที่ 2 ใช้ในการทำนายว่าบุคคลในภาพที่ได้สวมใส่หน้ากากหรือไม่ โดยการใช้ Logistic Regression

โดยมี dataset คือรูปภาพของหน้าที่ใส่หน้ากาก และไม่ใส่หน้ากาก โดยได้นำมาเข้าโปรแกรมแปลงชื่อไฟล์ และปรับขนาดรูปให้มีขนาดเท่ากันทุกรูป โดยกำหนดไว้ที่ 54*96 pixels นามสกุล .jpg

ที่มา dataset <https://github.com/chandrikadeb7/Face-Mask-Detection/tree/master/dataset>

ในการสร้างโมเดลมีขั้นตอนดังนี้

2.1.Import Library

```
In [14]: import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import imageio
import glob
```

โดยมี numpy เพื่อใช้ประมวลผลทางด้านคณิตศาสตร์ต่าง ๆ

Matplotlib เพื่อนำมาพล็อตกราฟแสดงผลของค่า Error ที่เกิดขึ้น

imageio เพื่ออ่านไฟล์รูปภาพออกมาเป็นข้อมูล

glob เพื่ออ่านชื่อและพาร์ทไฟล์รูป

2.2. สร้างสมการของ Logistic Regression

```
In [15]: def LoR_create_Xb(X):  
    N = X.shape[0]  
    ones = np.ones([N, 1])  
    Xb = np.hstack([ones, X])  
    return Xb
```

```
In [16]: def LoR_find_W_local_mul_class(X, Y, epoch = 1000, lr = 0.05): #<<<<<<<<<  
    Xb = LoR_create_Xb(X)  
    N = Xb.shape[0]  
    D_1 = Xb.shape[1]  
    K = Y.shape[1]  
    W = np.random.randn(D_1, K)/np.sqrt(D_1)  
    error_list = []  
    for i in range(epoch):  
        Yhat = LoR_find_Yhat_mul_class(X, W)  
        error = (-Y*np.log(Yhat)).sum()  
        error_list.append(error)  
        S = np.dot(Xb.T, Y-Yhat)  
        W = W + (lr/N)*S  
    return W, error_list
```

```
In [17]: def LoR_find_Yhat_mul_class(X, W):  
    Xb = LoR_create_Xb(X)  
    Z = np.dot(Xb, W)  
    Yhat = np.exp(Z)/np.exp(Z).sum(axis=1, keepdims = True)  
    return Yhat
```

```
In [18]: def find_error_mul_class(Y, Yhat):  
    N = Y.shape[0]  
    Y_argmax = np.argmax(Y, axis=1)  
    Yhat_argmax = np.argmax(Yhat, axis=1)  
    error = 100*(Y_argmax != Yhat_argmax).sum()/N  
    return error
```

```
In [19]: def create_onehot_target(label):  
    K = len(np.unique(label))  
    N = label.shape[0]  
    onehot = np.zeros([N, K])  
    for i in range(N):  
        onehot[i, label[i, 0]] = 1  
    return onehot
```

2.3. อ่านข้อมูลรูป และนำมา สร้าง train test เพื่อนำไปสร้างโมเดลต่อไป

```
In [20]: #glob.glob("train/*.jpg")
```

```
In [39]: image_train = np.empty((0,15552), int) #54*96*3=15552
label_train = np.empty((0,1), int)
```

```
num_train = 2326 #2326
i = 0
for image_path in glob.glob("train/*.jpg"):
    image = imageio.imread(image_path)
    image = np.array(image)
    image = image.reshape(1, -1)
    image_train = np.vstack((image_train, image))

    label = int(image_path[:-1][4])
    label_train = np.vstack((label_train, label))

    i += 1
    if i >= num_train:
        break

print(image_train.shape)
print(label_train.shape)

(2326, 15552)
(2326, 1)
```

```
In [22]: X_Train = image_train/255.0
Y_Train = create_onehot_target(label_train)
```

```
In [40]: image_test = np.empty((0,15552), int) #54*96*3=15552
label_test = np.empty((0,1), int)
```

```
num_test = 1392 #1392
i = 0
for image_path in glob.glob("test/*.jpg"):
    image = imageio.imread(image_path)
    image = np.array(image)
    image = image.reshape(1, -1)
    image_test = np.vstack((image_test, image))

    label = int(image_path[:-1][4])
    label_test = np.vstack((label_test, label))

    i += 1
    if i >= num_test:
        break

print(image_test.shape)
print(label_test.shape)

(1392, 15552)
(1392, 1)
```

```
In [24]: X_Test = image_test/255.0
Y_Test = create_onehot_target(label_test)
```

โดยจากขนาดรูปคือ 54*96 และเป็นรูปสี จึงต้องทำ matrix ตัวรับขนาด 15552 จาก 54*96*3 เกิดไว้ที่ตัวแปร image_train, image_test ของ train, test ตามลำดับ

2.4. สร้างโมเดล

```
In [25]: W, error_list = LoR_find_W_local_mul_class(X_Train, Y_Train)
```

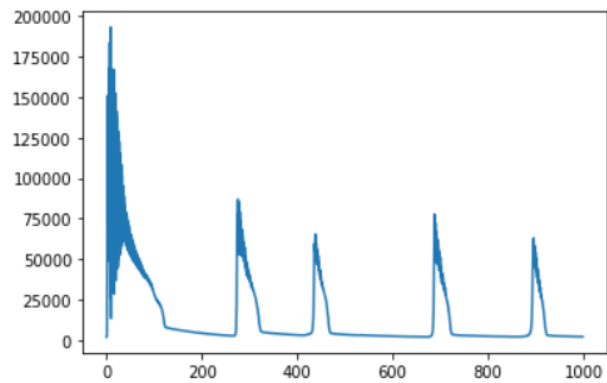
```
In [26]: W.shape
```

```
Out[26]: (15553, 2)
```

ค่า Error ที่ได้

```
In [27]: plt.plot(error_list)
```

```
Out[27]: [<matplotlib.lines.Line2D at 0x1eb278db220>]
```



```
In [29]: Yhat_Train = LoR_find_Yhat_mul_class(X_Train, W)
```

```
In [30]: error_Train = find_error_mul_class(Y_Train, Yhat_Train)
```

```
In [31]: error_Train
```

```
Out[31]: 5.803955288048152
```

2.5. นำโมเดลมาทดสอบ และค่า Error ที่ได้

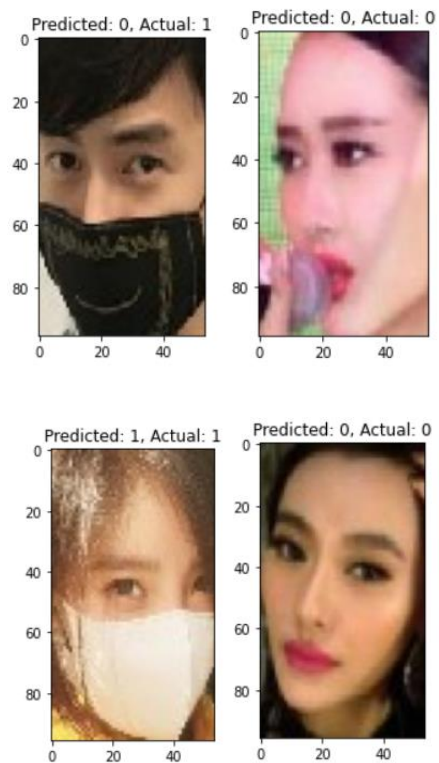
```
In [35]: Yhat_Test = LoR_find_Yhat_mul_class(X_Test, W)
```

```
In [36]: error_Test = find_error_mul_class(Y_Test, Yhat_Test)
```

```
In [37]: error_Test
```

```
Out[37]: 8.405172413793103
```

ตัวอย่างจริงบางส่วน



3. วิธี export model วิธีนำมาใช้

Export โดยการนำ W ที่ได้จากการสร้างโมเดลมาเก็บไว้ในไฟล์ ชื่อ outfile เพื่อไปใช้ในการทำนายต่อไป

```
In [379]: import pickle
```

```
In [380]: with open('outfile', 'wb') as fp:  
          pickle.dump(W, fp)
```

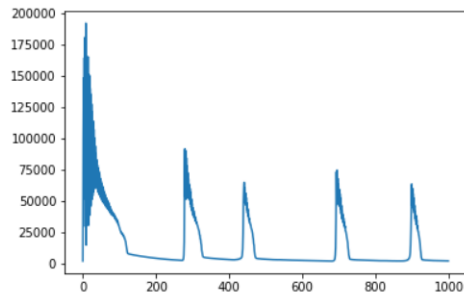
```
In [381]: with open ('outfile', 'rb') as fp:  
          myW = pickle.load(fp)
```

4. ความแม่นยำที่ได้

Train

```
In [374]: plt.plot(error_list)
```

```
Out[374]: [<matplotlib.lines.Line2D at 0x2eb70fe2df0>]
```



```
In [375]: Yhat_Train = LoR_find_Yhat_mul_class(X_Train, myW)
```

```
In [376]: error_Train = find_error_mul_class(Y_Train, Yhat_Train)
```

```
In [377]: error_Train
```

```
Out[377]: 3.353396388650043
```

Test

```
In [382]: Yhat_Test = LoR_find_Yhat_mul_class(X_Test, W)
```

```
In [383]: error_Test = find_error_mul_class(Y_Test, Yhat_Test)
```

```
In [384]: error_Test
```

```
Out[384]: 8.261494252873563
```

5. อื่นๆ

การนำโค้ดมาสร้างเป็นโปรแกรม

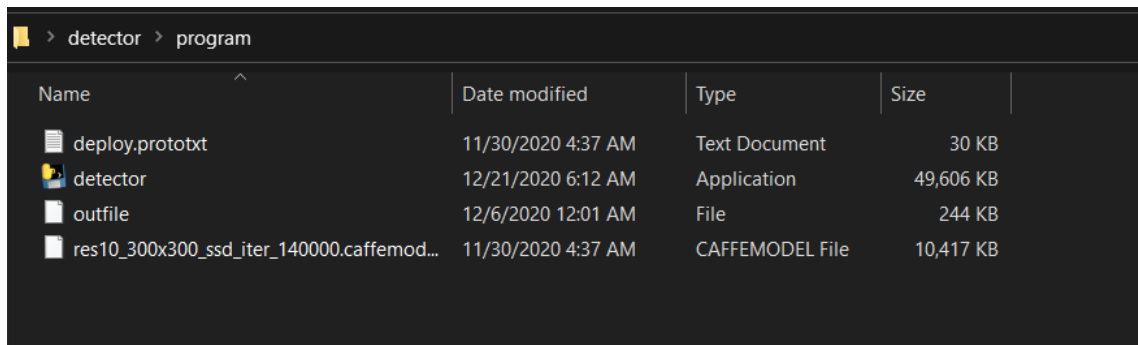
ชิ้นงานที่ได้นั้นได้ทำการใช้ pyinstaller ในการแปลงโค้ดจาก python เป็น exe เพื่อที่จะสามารถนำไปใช้ได้สะดวกต่อไป (อยู่ในโฟลเดอร์ program ไฟล์ชื่อ detector.exe)

ในขั้นแรกที่ทำได้ทำเป็น web application โดยใช้ Django จนสามารถใช้งานบน localhost ได้ แต่ในการนำไปใช้งานจริงพบว่าคำสั่งที่ใช้เปิดกล้องใน python ของ opencv นั้นเป็นการใช้กล้องของฝั่ง server จึงไม่สามารถนำไป deploy เพื่อใช้ได้)

Git: <https://github.com/achilles050/detector.git>

การใช้งานโปรแกรม

โดยโปรแกรมจะอยู่ในโฟลเดอร์ Program ชื่อว่า detector.exe โดยในนั้นจะมีไฟล์อื่นอีก 3 ไฟล์ คือไฟล์ที่เป็นการนำโมเดลมาใช้งานในโปรแกรม



Name	Date modified	Type	Size
deploy.prototxt	11/30/2020 4:37 AM	Text Document	30 KB
detector	12/21/2020 6:12 AM	Application	49,606 KB
outfile	12/6/2020 12:01 AM	File	244 KB
res10_300x300_ssd_iter_140000.caffemod...	11/30/2020 4:37 AM	CAFFEMODEL File	10,417 KB