



機器學習應用實戰 - 第G組報告

人臉口罩辨識

- 結合科技與時下疫情 -

成員名單: 吳浩瑋 | 葉駿成 | 梁芷芸

動機

109年12月1日起，政府公告出入八大類場所強制戴口罩。在公共場合未配戴口罩，除了引起周圍人的反感之外，更會被處以罰鍰。

選擇「人臉口罩辨識」的主題，**第一**、因應目前疫情所需，必須準確分辨哪些人是有戴口罩，哪些是沒戴口罩的；當然，是在不使用肉眼判斷的前提下。**第二**、即使在疫情過後，許多人會持續配戴口罩。許多科技產品有提供人臉解鎖的功能(Apple FaceID)，如何透過少量特徵判斷身份成了重要課題。

本報告將著重於對第一點的探討，第二點將建立在第一點的基礎上，日後自行延伸發揮應用。



資料來源

≡ kaggle

- Home
- Competitions
- Datasets**
- Code
- Discussions
- Courses
- More

Data Explorer

23.28 GB

- with_mask
- without_mask

Summary

- 20.0k files

包含檔案

View Active Events

Search

資料集名稱

Face Mask Lite Dataset

Dataset contains People with mask and without mask

prasoon kottarathil • updated a year ago (Version 1)

Data Tasks Code (17) Discussion (2) Activity Metadata

檔案大小

Download (23 GB)

New Notebook

Usability 8.8

License CC BY-SA 4.0

Tags arts and entertainment, education, covid19, computer vision

可用性

Description

If your approaching this Dataset, please UPVOTE this Dataset to show your appreciation



資料簡介

1. 兩個資料夾(有戴口罩、沒戴口罩), 各 1000 張照片
2. 解析度 1024x1024 像素
3. 口罩是用 P 圖上去的, 每個口罩都相同, 角度不同而已

訓練後發現大問題



請見後面解釋

1.32 GB



1.21 GB



訓練流程

一、尋找合適資料

二、資料前處理：

a. SVD 降維 b. resize

三、建立模型

四、測試模型

五、模型績效評估

六、實際運用



程式碼

```
1
2 import numpy as np
3 import cv2
4 import os
5 import glob
6 img_dir = "/Volumes/NO NAME/without_mask" # Enter Directory of all images
7 data_path = os.path.join(img_dir, '*g')
8 files = glob.glob(data_path)
9 data = []
10 for f1 in files:
11     img = cv2.imread(f1)
12     data.append(img)
13
14 def svd_compression(img, k):
15     res_image = np.zeros_like(img)
16     for i in range(img.shape[2]):
17         U, Sigma, VT = np.linalg.svd(img[:, :, i])
18         res_image[:, :, i] = U[:, :k].dot(np.diag(Sigma[:k])).dot(VT[:, :k, :])
19
20     return res_image
21
22
23 for i in range(len(data)):
24     res1 = svd_compression(data[i], k=100)
25     cv2.imwrite(f"/Volumes/NO NAME/without_mask_compressed/{i}.jpg", res1)
26
```

使用 SVD 壓縮影像

Glob.glob() 從資料夾中讀入檔案

1. 函數功能：匹配所有的符合條件的文件，並將以 list 的形式傳回。

自訂義函數

2. res_image 建立新圖樣板，svd 函數拆解陣列，設定保留的奇異值 k。壓縮過後的影像為傳回值。

執行時間

前 1000 張 未戴口罩 (k=100)

[Done] exited with code=0 in 3458.051 seconds

3. 執行時間 58 分鐘

後 1000張 有戴口罩 (k=100)

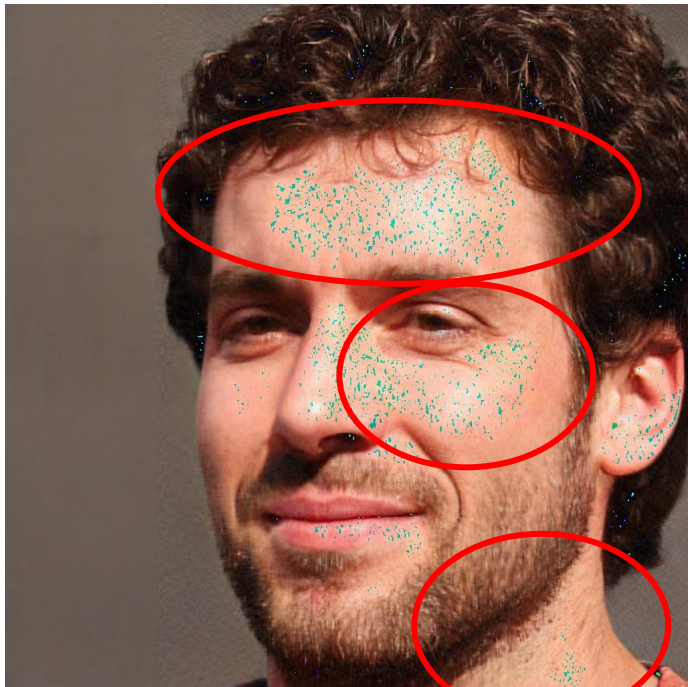
[Done] exited with code=0 in 2343.704 seconds

執行時間 39 分鐘

壓縮後影像

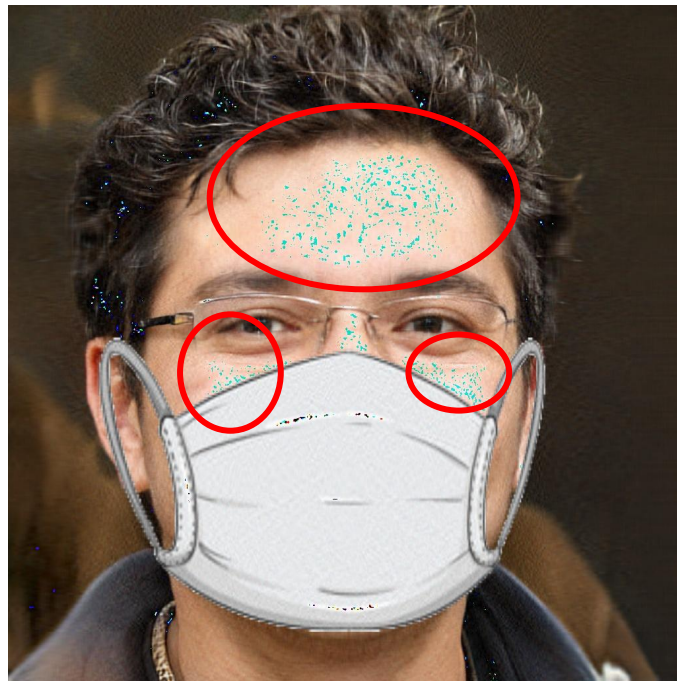
未戴口罩

資料夾大小: 1.32gb -> 244 mb



有戴口罩

資料夾大小: 1.21gb -> 264 mb



建立模型

```
1 import cv2
2 import glob
3 import numpy as np
4 import matplotlib.pyplot as plt
5 import os
6
7 ROWS=64
8 COLS=64
9 CHANNELS=3
10
11 FILE_DIR = "training_folder/"
12 files = glob.glob(FILE_DIR)
13
14
15 data=[] # 存放大小更改後的照片
16 data_labels=[] # 存放 照片的標籤 0 dog 1 cat
17 classes={0:"Person is wearing mask.",1:"Person is not wearing mask."}
18 #此功能使用opencv讀入照片，更改照片解析度並回傳
19 def read_image(image):
20     img=cv2.imread(image, cv2.IMREAD_COLOR)
21     return cv2.resize(img,(ROWS,COLS),
22         interpolation=cv2.INTER_CUBIC)
23
24 # 此功能將照片特徵轉變為一維
25 def prep_data(img):
26     X = img.reshape((1,CHANNELS*ROWS*COLS))
27     return X
```

讀入檔案

Resize 圖片

```
54 def predict_folder(folder_path):
55     TEST_DIR=os.path.join(folder_path,'*g')
56     files_test=glob.glob(TEST_DIR)
57     for i in range(len(files_test)):
58         img = read_image(files_test[i])
59         show_image_prediction(img,i,clf)
60
61 predict_folder("testing_folder")
```

```
29 for fl in files:
30     img = read_image(fl)
31     if "with_mask" in fl:
32         data_labels.append(0)
33     else:
34         data_labels.append(1)
35     data.append(prepare_data(img).flatten())
36
37
```

```
38 from sklearn.linear_model import LogisticRegression
```

```
39
40 clf = LogisticRegression()
41 clf.fit(data,data_labels)
```

邏輯斯迴歸

```
42
43 def show_image_prediction(X,i,model):
44     image = X.reshape(1,-1)
45     image_class=classes[model.predict(image)[0]]
46     image=image.reshape((ROWS,COLS,CHANNELS))
47     plt.figure(figsize=(4,2))
48     plt.xticks([])
49     plt.yticks([])
50     plt.imshow(image[...,:-1]) # RGB-> BGR
51     plt.title(f"Test {i+1}: {image_class}")
52     plt.show()
```

顯示影像的函數

傳入資料夾，
顯示預測結果

進行到這邊，發現資料集有嚴重問題：

1. 因為口罩是用 p 圖的關係，訓練出來的資料，在測試同樣的資料集時，準確率 是百分之百（過擬合）。
2. 但是只要用自己上傳的照片，幾乎每張都會判斷為『未戴口罩』，因為沒有白色的，清楚的口罩。
3. 因此，決定重新找資料集。

資料來源

≡ kaggle

🏠 Home

🏆 Competitions

📁 Datasets

<> Code

🗨️ Discussions

🎓 Courses

✓ More

Data Explorer

226.32 MB

📁 Dataset

- ▶ 📁 mask_worn_incorrect
- ▶ 📁 with_mask
- ▶ 📁 without_mask

Summary

▶ 📁 8982 files

包含檔案

📅 View Active Events

🔍 Search

資料集名稱

📁 Dataset

Face Mask Detection

Building a face mask classifier



vijay kumar • updated 3 months ago (Version 1)

Data

Tasks (1)

Code (7)

Discussion

Activity

Metadata

檔案大小

Download (226 MB)

New Notebook

Your Dataset download has started.
Show your appreciation with an upvote

40



📁 Usability 8.8

📄 License CC0: Public Domain

🏷️ Tags health, image data, covid19, public safety, cnn

可用性

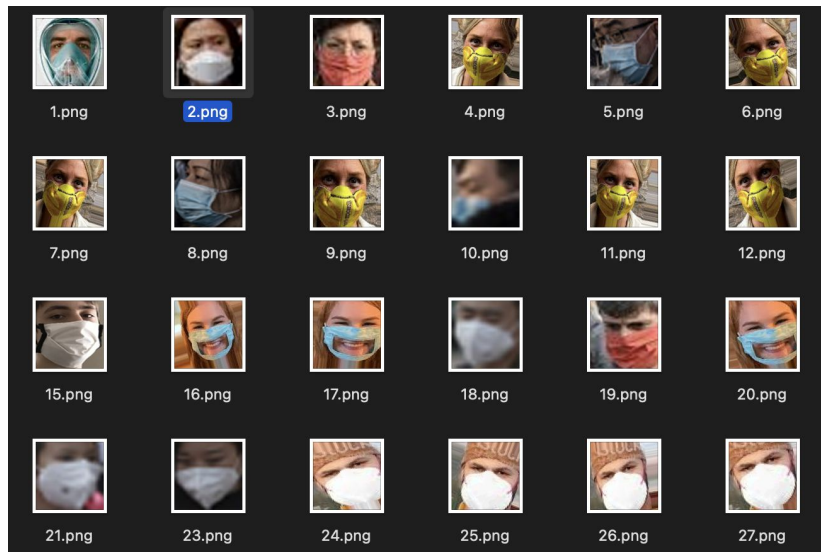
Context

Having seen multiple datasets related to face mask detection on Kaggle, one dataset which stood out contained 3 classes (with mask, without a mask, and wearing mask incorrectly), unfortunately, the dataset was highly imbalanced and uncleaned. So to improve this dataset, images had to be augmented in such a way that each class has an equal distribution of images and removing noisy images which could be considered as outliers. Thus this dataset that I've created is a combination of an existing dataset that has been cleaned and equally distributed across each class.

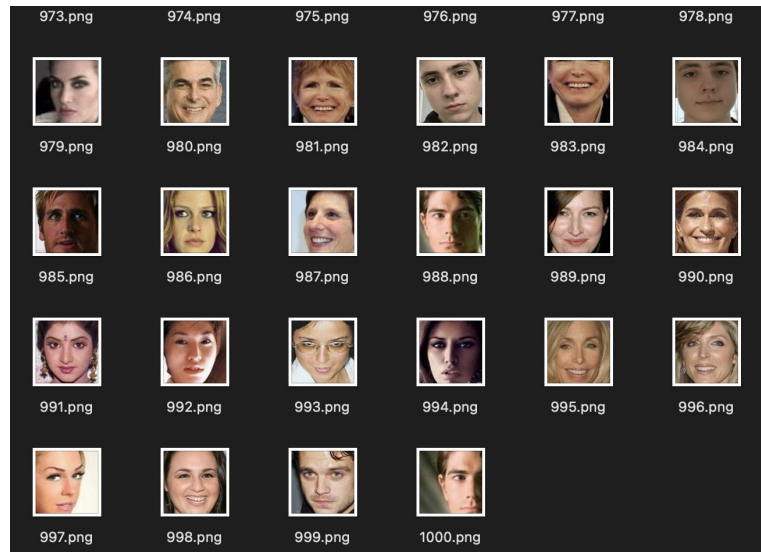
資料簡介

1. 兩個資料夾(有戴口罩、沒戴口罩), 各 1000 張照片
2. 圖片大小 128x128 像素
3. 與前面的資料集比較, 每張的口罩都是不同的

34.1 MB



36.6 MB



測試模型

建立模型步驟，
進行到這邊都與前者相同，
現在資料集沒問題後接續前者執行下去。

```
X_test = []
y_test = []
TEST_DIR = "testing_folder/*/g"
test_files = glob.glob(TEST_DIR)
for fl in test_files:
    img = read_image(fl)
    if "with_mask" in fl:
        y_test.append(0)
    else:
        y_test.append (function) prep_data: (img) -> Ar
X_test.append(prep_data(img).flatten())
```

傳入測試資料夾

```
y_prediction = clf.predict(X_test)
```

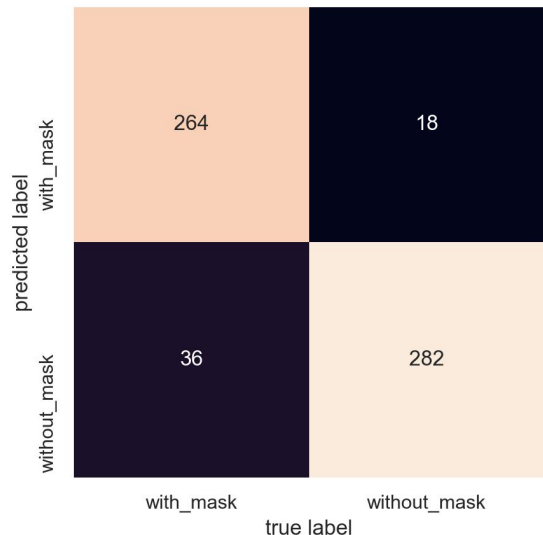
得到 predict 結果

績效評估

```
56 y_prediction = clf.predict(X_test)
57 from sklearn.metrics import classification_report
58 print(classification_report(y_test,y_prediction,target_names=['with_mask','without_mask']))
59
60
61 import matplotlib.pyplot as plt
62 import seaborn as sns; sns.set()
63 from sklearn.metrics import confusion_matrix
64
65 mat = confusion_matrix(y_test,y_prediction)
66
67 sns.heatmap(data=mat.T,
68             square=True,
69             annot=True,
70             fmt='d',
71             cbar=False,
72             xticklabels=['with_mask','without_mask'],
73             yticklabels=['with_mask','without_mask'])
74 plt.xlabel('true label')
75 plt.ylabel('predicted label')
76 plt.show()
```

	precision	recall	f1-score	support
with_mask	0.94	0.88	0.91	300
without_mask	0.89	0.94	0.91	300
accuracy			0.91	600
macro avg	0.91	0.91	0.91	600
weighted avg	0.91	0.91	0.91	600

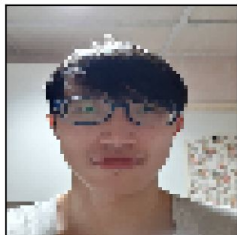
產生混淆矩陣



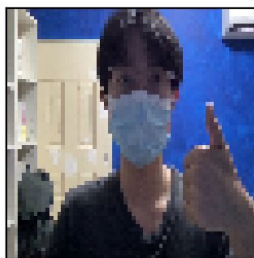
模型評估報表

實際運用

Test 1: Person is not wearing mask.



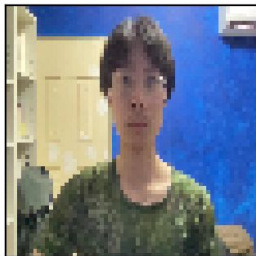
Test 2: Person is wearing mask.



Test 3: Person is wearing mask.



Test 4: Person is not wearing mask



Test 4: Person is wearing mask.



Test 5: Person is not wearing mask.

