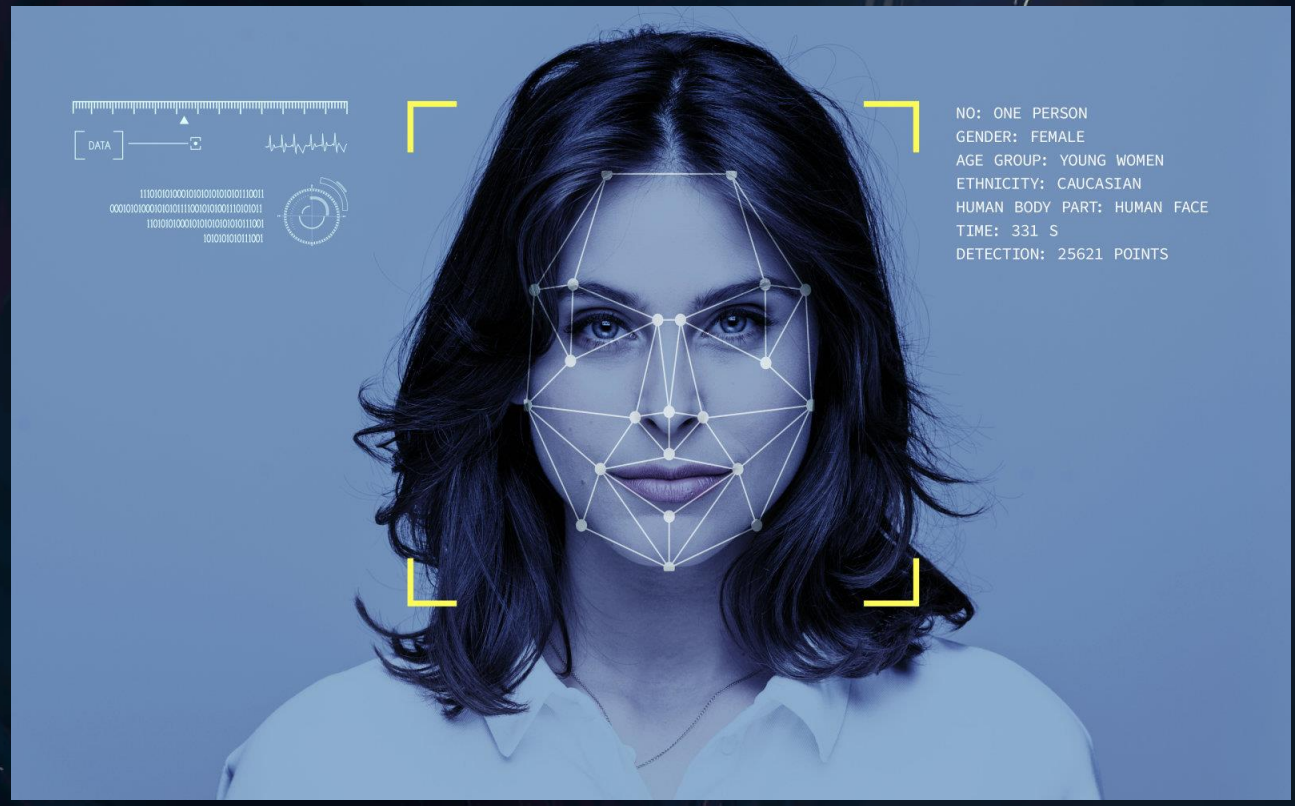
2022
看臉時代

以CNN對人臉進行顏值評分

110030517胡詠晴



Outline
大綱

PART

01

ONE

背景介紹

01 背景介紹 前言

美女的定義

古代美女標準

- 膚若凝脂
- 明眸善睞
- 雲發豐艷
- 蛾眉青黛
- 杏臉桃腮
- 櫻唇貝齒
- 楊柳細腰
- 纖纖素手
- 三寸金蓮
- 軟玉溫香

01 背景介紹 前言

美女的定義

白人標準美女

韓國標準美女

日本標準美女

01 背景介紹

研究動機

在當今“外貌協會”盛行的風氣下，人們對外表的經營和保養越來越重視。這現象也反映在許多電視劇和漫畫情節之中，如“轉學來的女生”其中也有一集針對一女校如何使用即時評估系統，給女學生的顏值進行排名決定其使用校園資源的權益。在看臉時代漫畫中一開始男主也是因為顏值低等問題被霸凌，直到獲得新的高顏值身體人生才得到重生。然而，審美是非常主觀的，因人而異，並沒有一個確切的標準。

研究目的

藉由CNN對個人顏值進行客觀評分,讓顏值能夠被量化，使顏值有更明確的標準，且能提供選美比賽一個參考標準!

01 背景介紹 5W1H

When

舉辦選美比賽時，或是想知道自己臉部顏值在哪一個等級時

Who

選美選手、選美評審、想知道自己顏值分數的人

Where

選美比賽現場等

What

選美選手到現場須經由評審評分，評審對美醜的優劣判斷不一

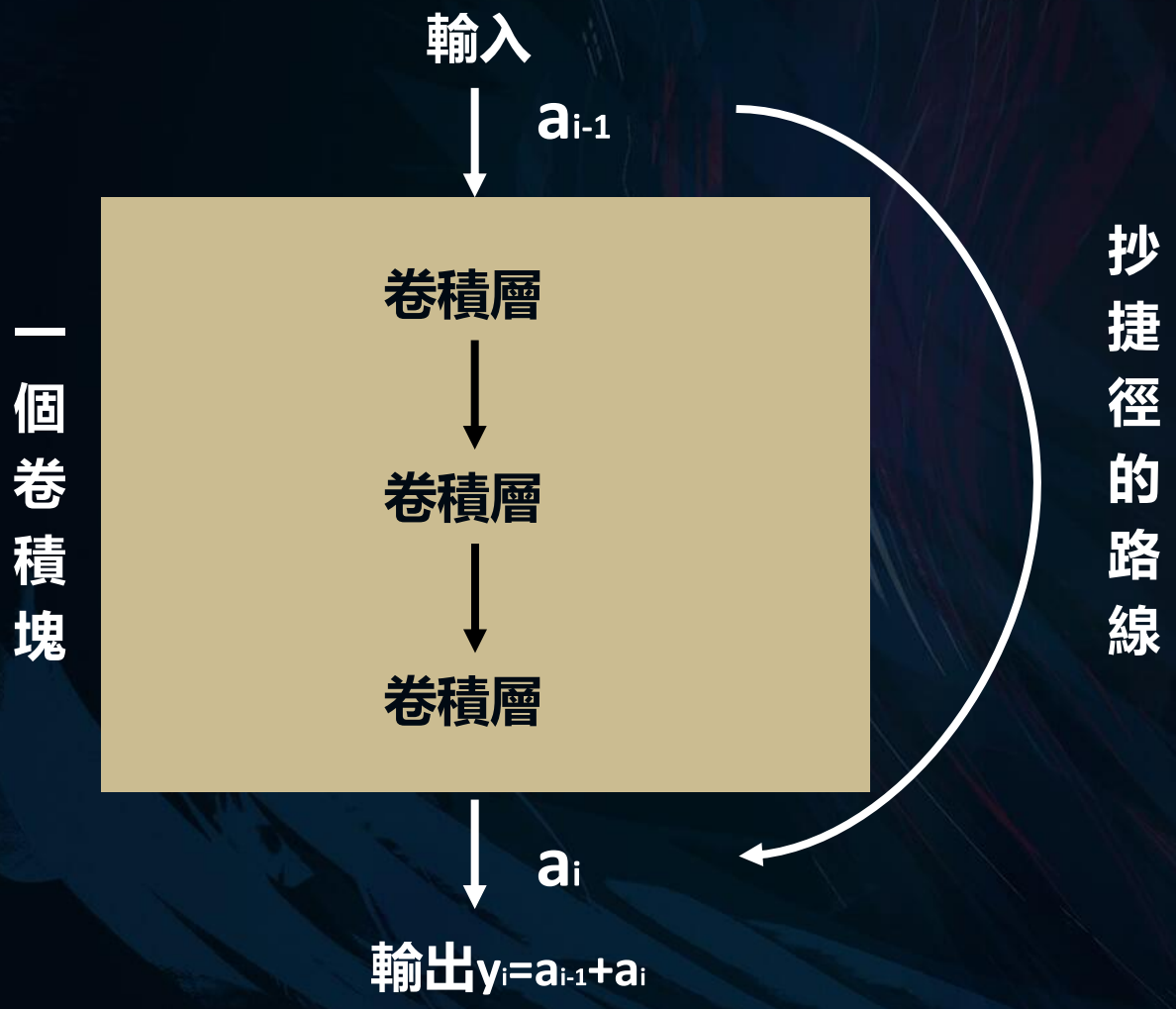
Why

評審評分太過主觀意見，藉由CNN能更客觀描述選美比賽選手的顏值

How

現場拍攝個人正面照片後，以CNN對個人顏值進行客觀評分

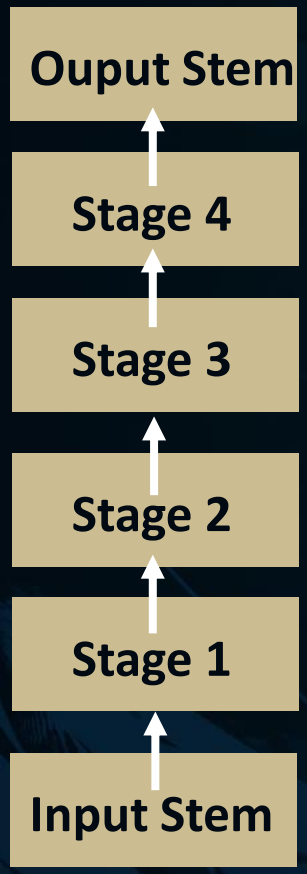
- **殘差神經網路(ResNets)：以卷積塊搭配跳接(Skip connection)設計而成的架構**



$$y_i=a_{i-1}+a_i$$

抄捷徑從前一層傳來的　當層算出來的

- **ResNet50**



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

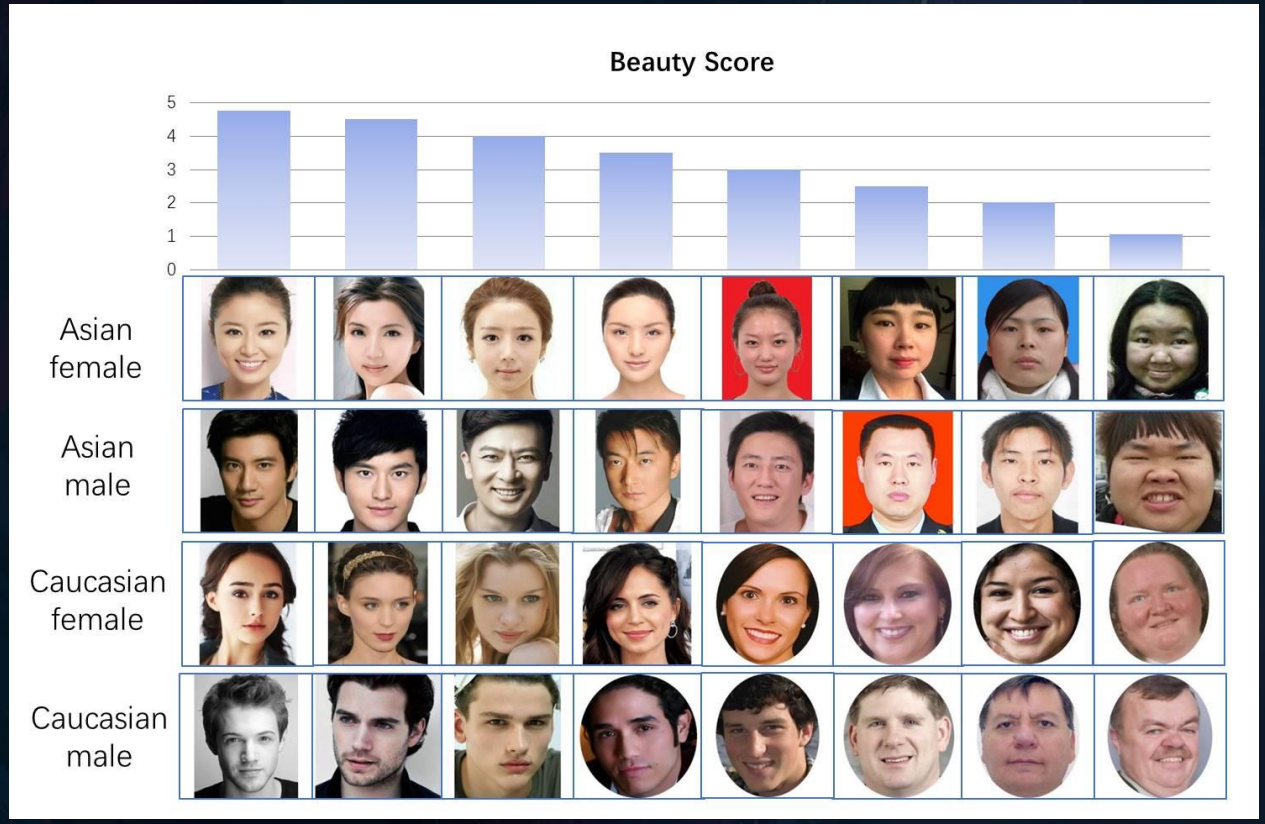
- **Flask**

本次研究以Flask為基礎搭建網頁，Flask是一個使用 Python 撰寫的輕量級 Web 應用程式框架，由於其輕量特性，也稱為micro-framework（微框架）。

01 背景介紹 資料集描述

SCUT-FBP5500

- 共計5500 張人臉正面照片
  1. 2000名亞洲女性（AF）
  2. 2000名亞洲男性（AM）
  3. 750名高加索女性（CF）
  4. 750名高加索男性（CM）
- 標簽
  1. 人臉標記Facial Landmark
  2. 顏值得分（1~5）



- 所有照片皆由 60 名志願者用 [1, 5] 範圍內的美感評分進行標記，每張人臉標記有86個特徵點

PART

02

TWO

資料前處理

02 資料前處理


```
[ ] ratings = pd.read_excel('C:/Users/user/1jupyter notebook example/facial_beauty_prediction-master/All_R
    ratings.head()
```

	Rater	Filename	Rating	original Rating
0	1	CF1.jpg	3	NaN
1	1	CF10.jpg	3	NaN
2	1	CF100.jpg	1	NaN
3	1	CF101.jpg	2	NaN
4	1	CF102.jpg	3	NaN

```
[ ] filenames = ratings.groupby('Filename').size().index.tolist()

    labels = []

    for filename in filenames:
        df = ratings[ratings['Filename'] == filename]
        count = Counter(df['Rating']).most_common(1)[0][0]
        score = round(df['Rating'].mean(), 2)
        labels.append({'Filename': filename, 'most_common': count, 'score': score})

    labels_df = pd.DataFrame(labels)
    labels_df.head()
```

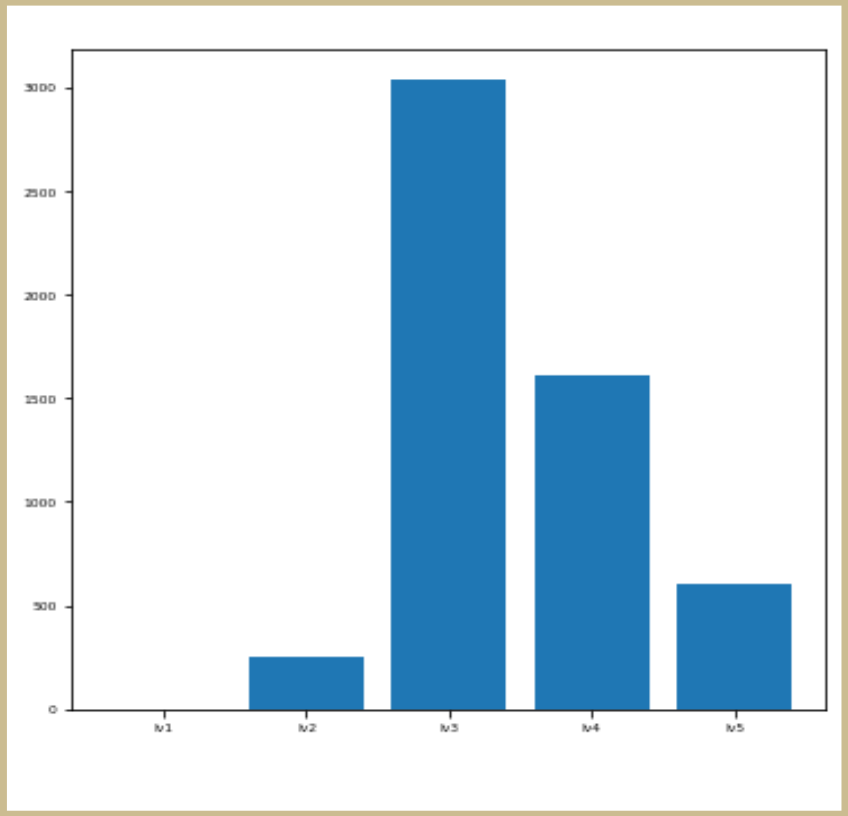
	Filename	most_common	score
0	AF1.jpg	3	2.33
1	AF10.jpg	4	3.43
2	AF100.jpg	3	2.90
3	AF1000.jpg	4	3.97
4	AF1001.jpg	4	3.73

02 資料前處理

➢ 得分分布情況

```
[ ]  scores = sorted(labels_df.score.tolist())
```

```
lv1 = [x for x in scores if x<=1]
lv2 = [x for x in scores if x>1 and x<=2]
lv3 = [x for x in scores if x>2 and x<=3]
lv4 = [x for x in scores if x>3 and x<=4]
lv5 = [x for x in scores if x>4 and x<=5]

plt.bar(['lv1','lv2','lv3','lv4','lv5'],
        [len(x) for x in [lv1, lv2, lv3, lv4, lv5]])
```



分成五個level，以長條圖繪出目前所有的照片之得分分布

超過一半的評分都在 3~4分的等級，兩側極端值人數都極為稀少。

02 資料前處理

將所有照片轉成 numpy.ndarray 格式

```
[ ]  img_width,  img_height,  channels  =  350,  350,  3
     sample_dir  =  'D:/SCUT-FBP5500_v2/Images'
     nb_samples  =  len(os.listdir(sample_dir))
     input_shape  =  (img_width,  img_height,  channels)

     x_total  =  np.empty((nb_samples,  img_width,  img_height,  channels),  dtype=np.float32)
     y_total  =  np.empty((nb_samples,  1),  dtype=np.float32)

     for  i,  fn  in  enumerate(os.listdir(sample_dir)):
             img  =  load_img('%s/%s'  %  (sample_dir,  fn))
             x  =  img_to_array(img).reshape(img_height,  img_width,  channels)
             x  =  x.astype('float32')  /  255.
             y  =  labels_df[labels_df.Filename  ==  fn].score.values
             y  =  y.astype('float32')
             x_total[i]  =  x
             y_total[i]  =  y
```

1 ←

2 →

將數據拆分成訓練集2560個，驗證集640個，測試集800個

```
seed = 42
x_train_all, x_test, y_train_all, y_test = train_test_split(x_total, y_total, test_size=0.2, random_state=seed)
x_train, x_val, y_train, y_val = train_test_split(x_train_all, y_train_all, test_size=0.2, random_state=seed)
```

```
np.save('x_train.npy', x_train)
np.save('y_train.npy', y_train)
np.save('x_val.npy', x_val)
np.save('y_val.npy', y_val)
np.save('x_test.npy', x_test)
np.save('y_test.npy', y_test)
```

```
for item in [x_train, y_train, x_val, y_val, x_test, y_test]:
    print(item.shape)
```

```
(2560, 350, 350, 3)
(2560, 1)
(640, 350, 350, 3)
(640, 1)
(800, 350, 350, 3)
(800, 1)
```

PART

03

THRE
E

模型建構

03 模型建構

```
[ ]  img_width,  img_height,  channels  =  350,  350,  3
     input_shape  =  (img_width,  img_height,  channels)
```

```
[ ]  resnet  =  ResNet50(include_top=False,  pooling='avg',  input_shape=input_shape)
     model  =  Sequential()
     model.add(resnet)
     model.add(Dense(1))
     model.layers[0].trainable  =  False
     model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Model)             (None, 2048)              23587712
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 2049
=================================================================
Total params: 23,589,761
Trainable params: 2,049
Non-trainable params: 23,587,712
_________________________________________________________________
```

使用Keras內建的ResNet50架構進行訓練

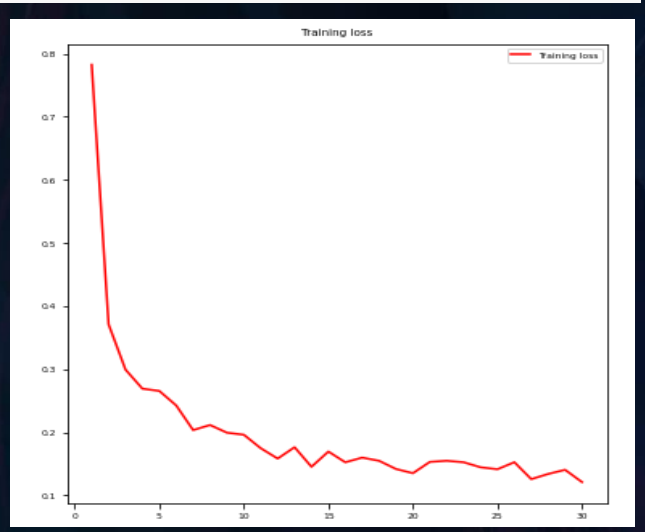
去掉最後的 softmax 層，再加一層 dense 全連接層

將模型的其餘參數設為不可訓練

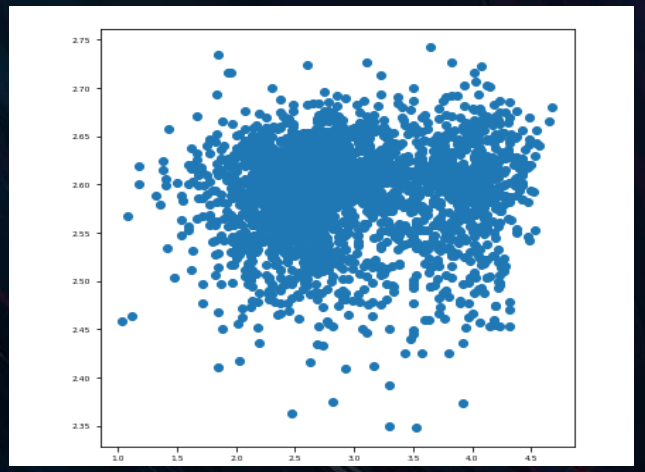
03 模型建構

```
[ ]  model.compile(loss='mse',  optimizer='adam')
     history  =  model.fit(batch_size=32,  x=x_train,  y=y_train,  epochs=30)
```

```
[ ]  plt.rcParams['figure.figsize']  =  (6,6)

     loss  =  history.history['loss']
     epochs  =  range(1,  len(loss)  +  1)

     plt.figure()
     plt.title('Training  loss')
     plt.plot(epochs,  loss,  'red',  label='Training  loss')
     plt.legend()

     plt.show()
```




```
[ ]  plt.scatter(y_train,  model.predict(x_train))
```



以MSE(均方誤差)作為我們的損失函數並訓練30個epoch

將損失函數圖畫出後可以看到其loss下降到大概0.12

以散布圖查看目前數據分布狀況，數據糾結而沒有趨勢

PART

04

FOUR

參數調整

04 參數調整

	OLD	NEW
learning rate	keras內建	min0.00001
batch size	32	8
ResNet50參數	不可訓練	可訓練
Epochs	30	10

```
[ ]  filepath="{epoch:02d}-{val_loss:.2f}.h5"
     checkpoint = ModelCheckpoint(filepath, monitor='val_loss', verbose=1, save_best_only=True, mode='min')
     reduce_learning_rate = ReduceLROnPlateau(monitor='loss',
                                              factor=0.1,
                                              patience=2,
                                              cooldown=2,
                                              min_lr=0.00001,
                                              verbose=1)

     callback_list = [checkpoint, reduce_learning_rate]
```

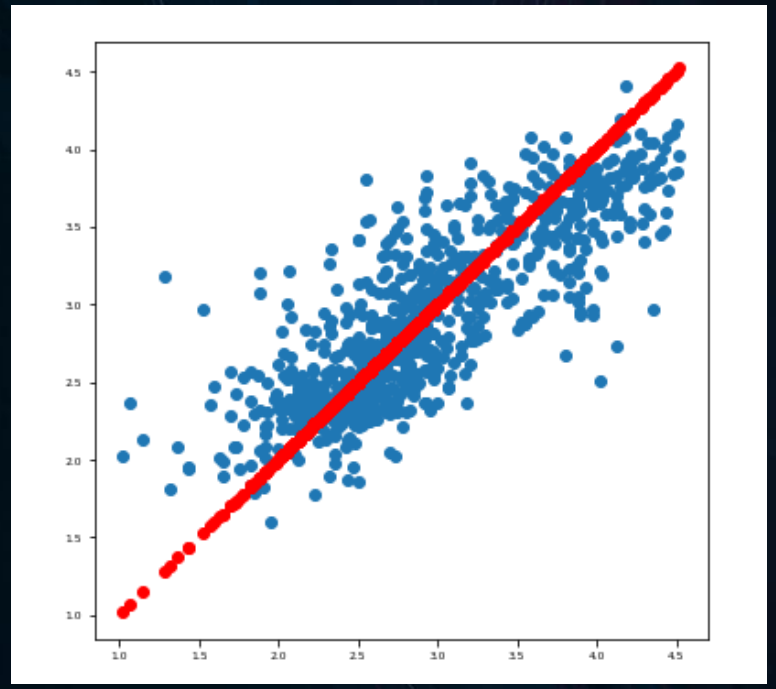
```
[ ]  model.layers[0].trainable = True
     model.compile(loss='mse', optimizer='adam')
     history = model.fit(x=x_train,
                         y=y_train,
                         batch_size=8,
                         epochs=10,
                         validation_data=(x_val, y_val),
                         callbacks=callback_list)
```

設置了 checkpoint callback，將驗證集的 loss 設為監控項

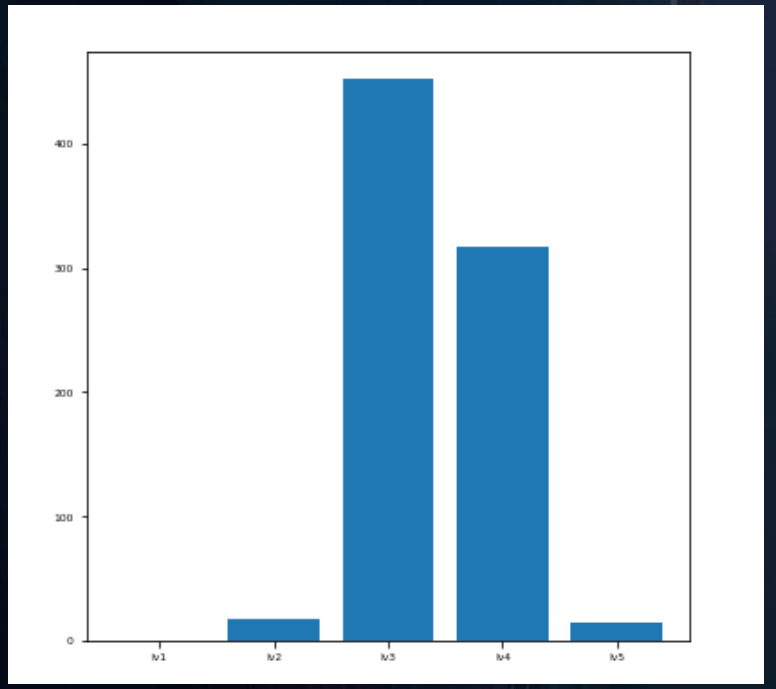
val loss下降則保存其該值最低的模型

val loss沒有下降則判定為沒有改善，而不儲存結果。

04 參數調整



測試集之回歸分析，
其結果與目標值接近



測試集之長條圖其原始數據大
致相同都是中間高極端值較少

04 參數調整 模型結果

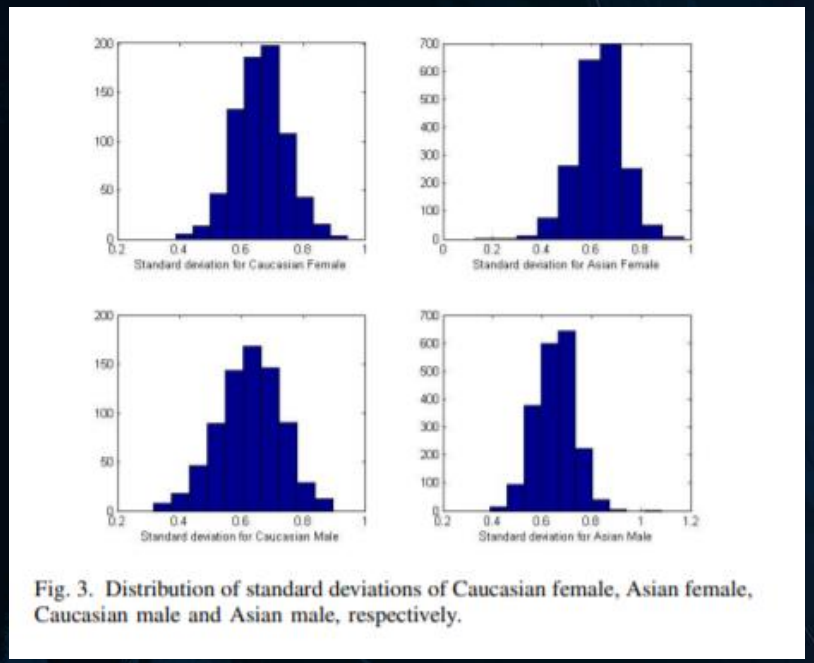


Fig. 3. Distribution of standard deviations of Caucasian female, Asian female, Caucasian male and Asian male, respectively.

華南理工大學之數據平均標準差在0.7左右

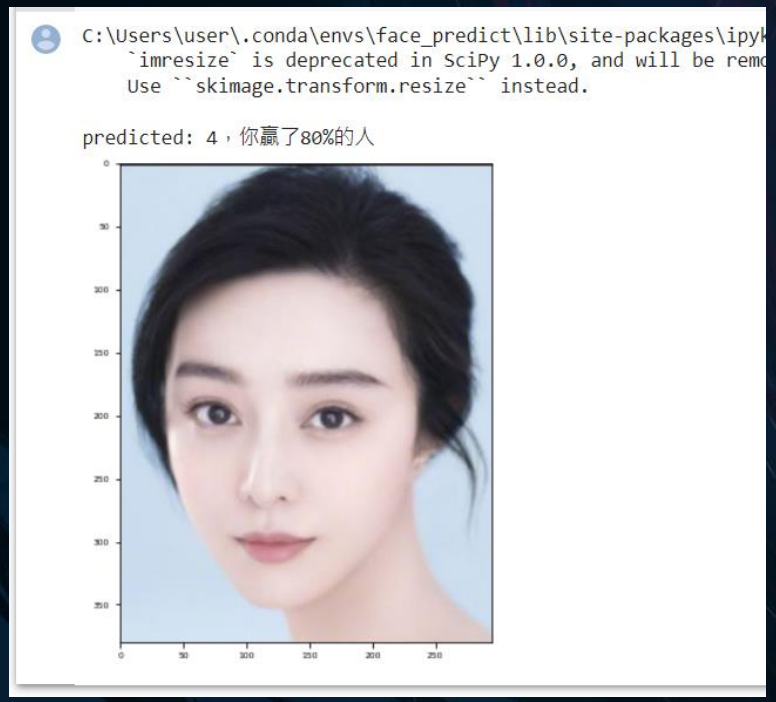
實際預測結果偏差也在0.7以內

將分數分為五個lv區間，並隨機印出測試之結果

04 參數調整 模型結果

- **實測網路隨機下載照片並進行預測**

以范冰冰為例其結果為4分



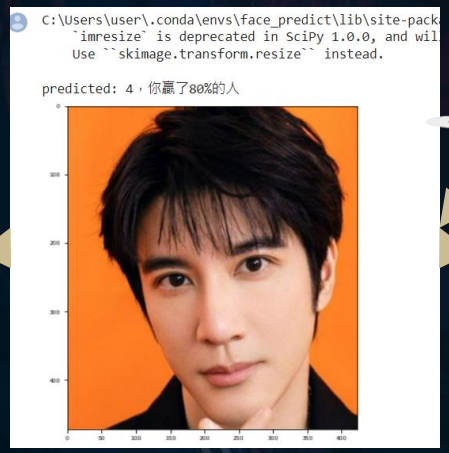
不同張范冰冰照片其結果同為4分

同一人不同張照片其結果分數一致

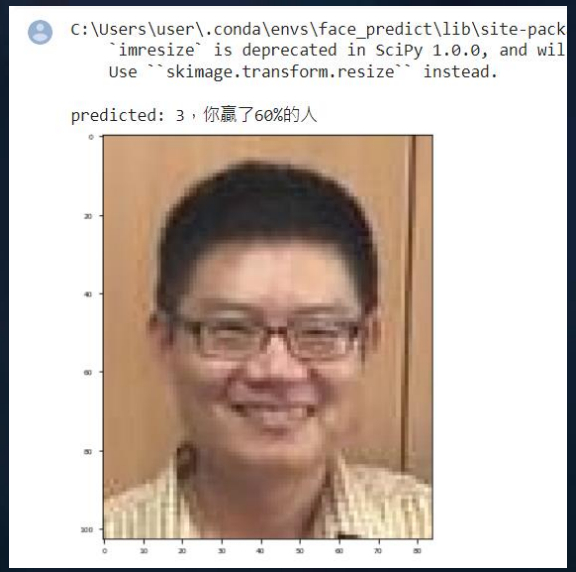
04 參數調整 模型結果

李靜蕾(3分)

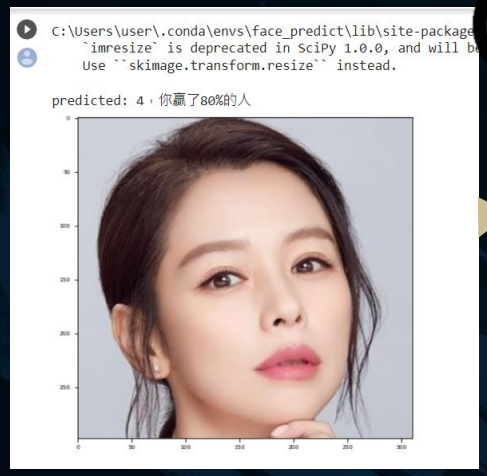
王力宏(4分)



老師(3分)



徐若瑄(4分)



BY2 YUMI(4分)

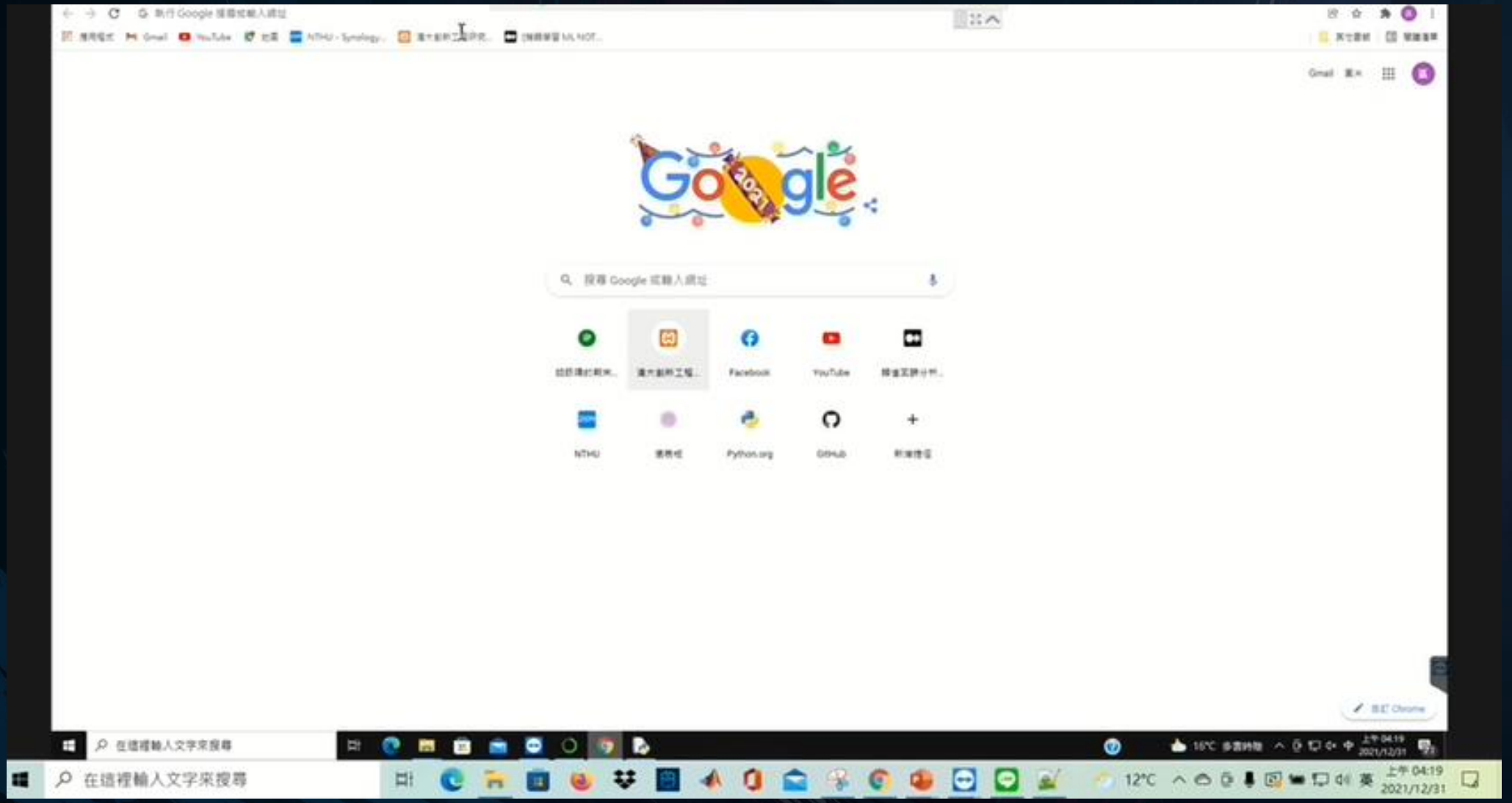
PART

05

FIVE

網頁DEMO

05 網頁DEMO


Google
Facebook
YouTube
NTHU
Python.org
GitHub
在這裡輸入文字來搜尋
12°C
上午 04:19
2021/12/31

PART

06

SIX

結　　論

06 結論

研究結論

該資料集相較以前的數據集有更清晰的分類，更豐富的人臉庫，但該數據集沒有有色人種，可能存在一點倫理上的問題，期盼有更多的資料讓整個模型結果更為精準，本次研究結果針對每一張人臉照都有做出分數的判斷，但因為數據分數本身的量尺分數只有1-5分造成區間落在3-4的比例居多，但還是能做為一個簡單的顏值評分工具。

結論

未來展望

在做模型評估的時候因為時間限制所以參數調整的部分較不完全，未來如果有機會自己製作數據集的時候可以把量尺分數拉寬，這樣的分數可以更加精準，在選美比賽上分數也更為直觀，同時此技術也可以用在基礎整形外科臉部評分中，讓患者知道自己的整形前後顏值各是落在哪個區間！

參考文獻

中國古代的美女標準竟如此嚴苛

https://kknews.cc/zh-tw/culture/p55zjgp.html

衡量美女標準是什麼？1-10分，5分就算美女!

https://www.xuehua.us/a/5ec1fb8d6fcd4773f3aa3983?lang=zh-hk

SCUT-FBP5500: A Diverse Benchmark Dataset for Multi-Paradigm Facial Beauty Prediction

【深度學習】ResNet解讀及程式碼實現

https://www.itread01.com/content/1542368643.html

Residual Leaning: 認識ResNet與他的冠名後繼者ResNeXt、ResNeSt

https://medium.com/%E8%BB%9F%E9%AB%94%E4%B9%8B%E5%BF%83/deep-learning-residual-leaning-%E8%AA%8D%E8%AD%98resnet%E8%88%87%E4%BB%96%E7%9A%84%E5%86%A0%E5%90%8D%E5%BE%8C%E7%B9%BC%E8%80%85resnext-resnest-6bedf9389ce

How Attractive Are You in the Eyes of Deep Neural Network?

https://towardsdatascience.com/how-attractive-are-you-in-the-eyes-of-deep-neural-network-3d71c0755ccc

SCUT-FBP5500 人臉美學預測 論文解讀

https://zhuanlan.zhihu.com/p/346096100

Keras Callback 的使用

https://ithelp.ithome.com.tw/m/articles/10234641

基於深度學習的顏值打分器

https://www.zhihu.com/column/p/36138077