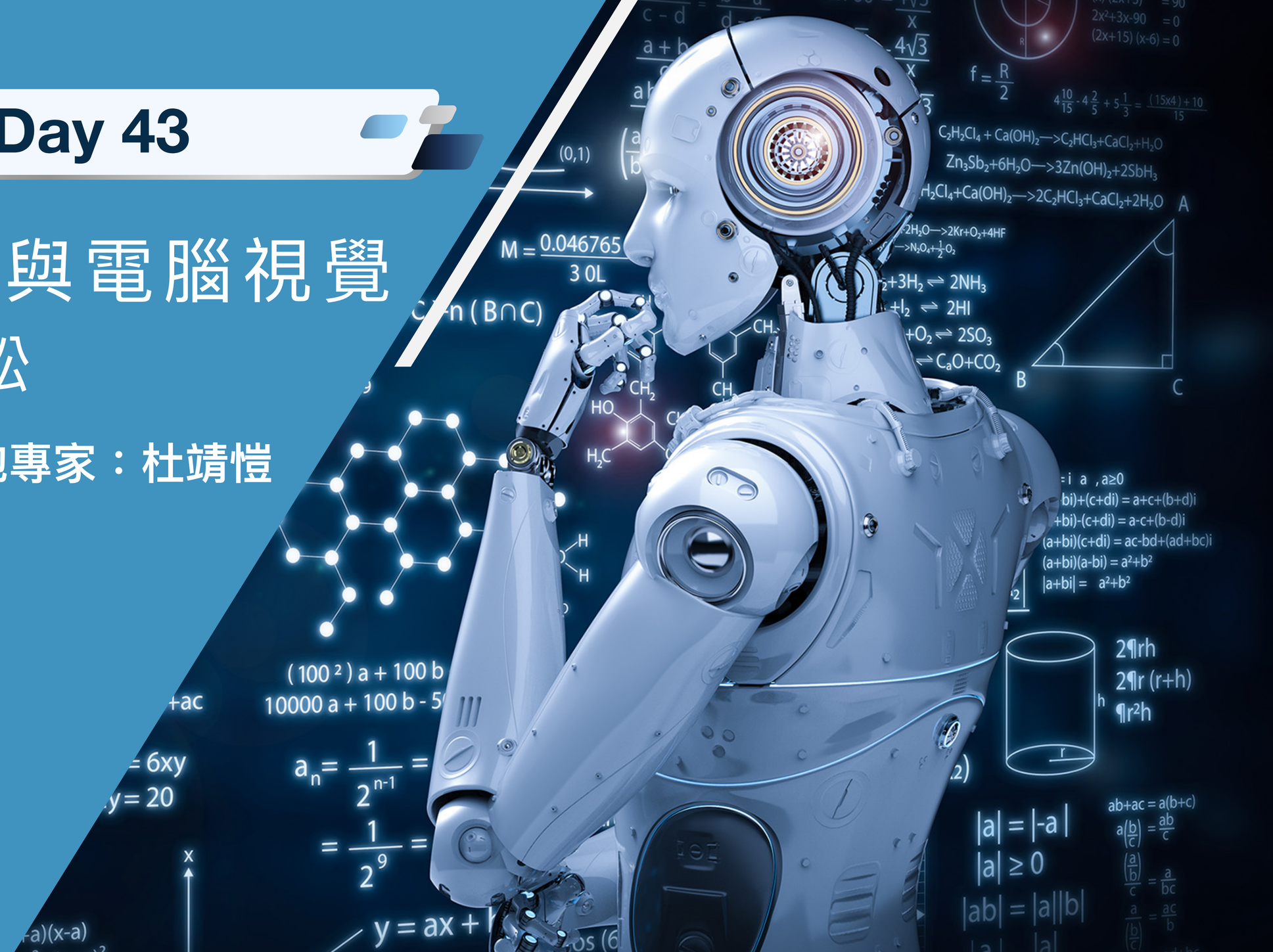
Day 43

深度學習與電腦視覺

學習馬拉松

cupoy 陪跑專家：杜靖愷



CUPOY

人臉關鍵點檢測網路架構

重要知識點

CUPOY

目標
知識點

學習用 keras 定義人臉關鍵點檢測的網路架構

獲得
知識點

完成今日課程後你應該可以了解

- 如何定義人臉關鍵點檢測相關的網路

定義人臉關鍵點檢測網路

CUPOY

完成前一天對人臉關鍵點資料的探索後，今天的課程主要是實作練習，希望學員能夠更深入的體會，在理解資料的 「長相」之後（尤其是輸入及輸出的部分），如何能夠定義所需要的網路結構。

讀取資料集的程式碼範例

CUPOY

```python
# 讀取資料集以及做前處理的函數
def load_data(dirname):
    # 讀取 csv 文件
    data = pd.read_csv(dirname)
    # 過濾有缺失值的 row
    data = data.dropna()

    # 將圖片像素值讀取為 numpy array 的形態
    data['Image'] = data['Image'].apply(lambda img: np.fromstring(img, sep=' ')).values

    # 單獨把圖像 array 抽取出來
    imgs = np.vstack(data['Image'].values)/255
    # reshape 為 96 x 96
    imgs = imgs.reshape(data.shape[0], 96, 96)
    # 轉換為 float
    imgs = imgs.astype(np.float32)

    # 提取坐標的部分
    points = data[data.columns[:-1]].values

    # 轉換為 float
    points = points.astype(np.float32)

    # normalize 坐標值到 [-0.5, 0.5]
    points = points/96 - 0.5

    return imgs, points
```

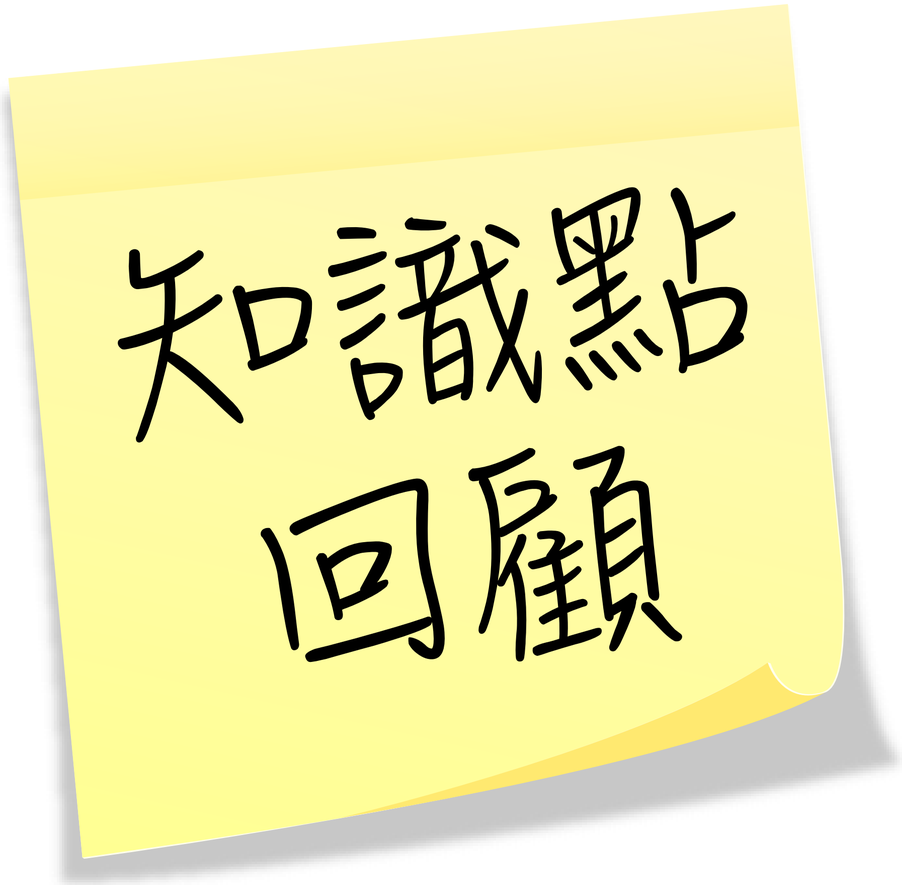
```python
# 讀取資料
imgs_train, points_train = load_data(dirname = 'training.csv')
print("圖像資料:", imgs_train.shape, "\n關鍵點資料:", points_train.shape)
```

```
圖像資料: (2140, 96, 96)
關鍵點資料: (2140, 30)
```

今天的範例基本上是把前一天讀取資料的程式碼寫成一個函數，在定義出檢測網路以前，我們要先意識到，這個檢測網路和我們所擁有的訓練資料的**「形狀」**有很大的關係，以這個 kaggle 資料集為例，

input shape 是 96 x 96

output shape 是一個 30 維的向量



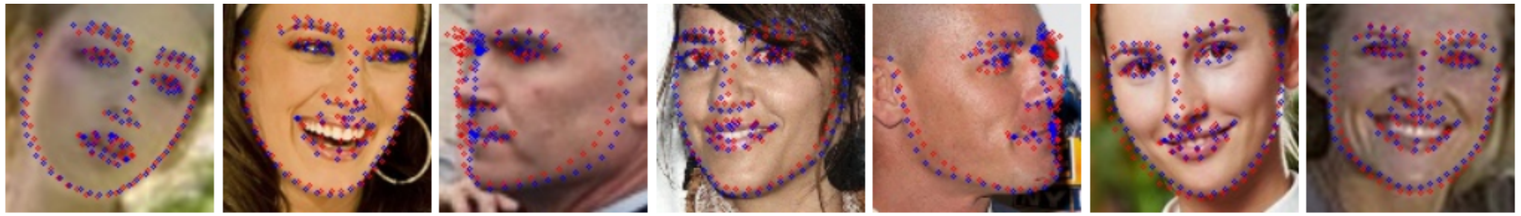
- 定義一個神經網路，有兩個最基本的模塊，一個是輸入輸出的設計，另外就是網路骨幹
- 網路骨幹我們通常會直接使用擁有強大計算能力的研究單位發表的網路結構，比如 Google 的 Inception 系列
- 在應用的角度，需要對輸入輸出有更高的敏感度，然後以有限的計算力對別人發表的骨幹網路結構適度地基於應用需求更改及 finetune

參考資料

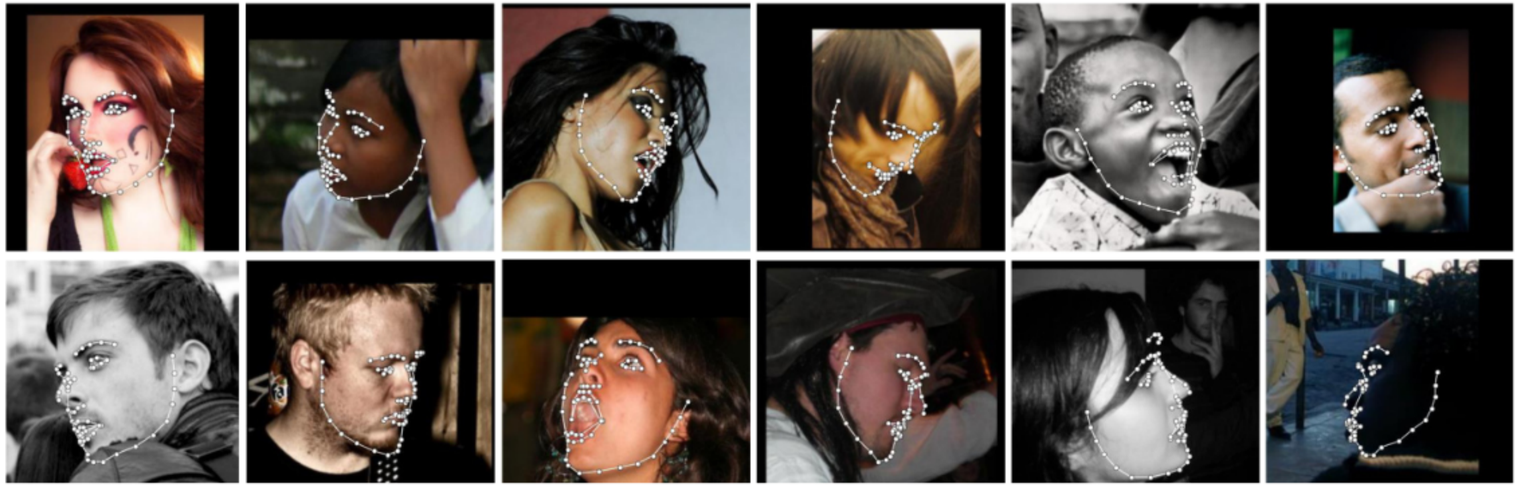
CUPOY

這裡介紹兩個很棒的人臉關鍵點研究，有興趣的可以去詳細的看：

https://github.com/guoqiangqi/PFLD



https://github.com/1adrianb/face-alignment



解題時間 Let’s Crack It

CUPOY

請跳出 PDF 至官網 Sample Code & 作業開始解題