1. **基礎影像處理**

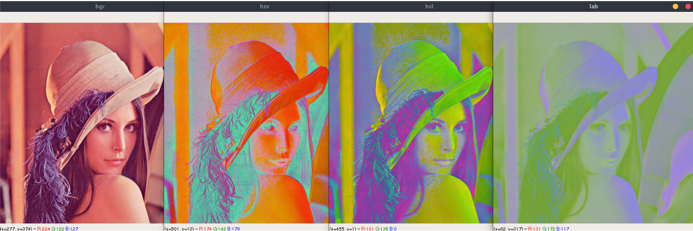
## D1：OpenCV 簡介 + 顯示圖片

## D2：Color presentation 介紹 (RGB, LAB, HSV)

OpenCV 在顯示圖片的時候都還是以 BGR 的方式呈現，但我們可以觀察圖片轉換到其他 color space 後，透過 BGR 呈現會有甚麼差異。

▪參考上一節顯示圖片的程式碼顯示不同 color space 的圖片

▪包含 RGB, HSV, HSL, LAB



## D3：顏色相關的預處理 (改變亮度, 色差)

#### 實作各種不一樣的方式來調整飽和 / 對比 / 明亮

* 改變 color space 來調整飽和度
* 實作直方圖均衡
  + 對 RGB 圖直接處理
  + 改變 color space 改明亮度
* alpha/ beta 調整對比 / 明亮
  + alpha=2.0, beta= 0
  + alpha=1.0, beta=50



## D4：以圖片為例做矩陣操作 (翻轉, 縮放, 平移)

實作本篇提到的三大概念：

* 翻轉：實作上下左右的翻轉
* 縮放：比較鄰近差值與雙立方插值 (或雙線性插值) 的圖片品質
* 平移：建立 Translation Transformation Matrix 來做平移

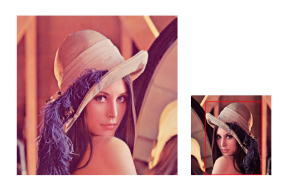


## D5：透過 OpenCV 做圖並顯示 (長方形, 圓形, 直線, 填色)

假設我們希望先對圖片做以下幾點預處理，請印出最後結果

Hint: 注意先後順序，人物原始邊框座標 (60, 40), (420, 510)

* 對明亮度做直方圖均衡處理
* 水平鏡像 + 縮放處理 (0.5 倍)
* 畫出人物矩形邊框
* 並且得到新的矩形邊框座標



## D6：affine transformation 概念與實作

* 練習以旋轉變換 + 平移變換來實現仿射變換
  + 旋轉 45 度 + 縮放 0.5 倍 + 平移 (x+100, y-50)
* 直接建構仿射矩陣 (圖片至少包含 3 個點以上的 pair)

cv2.getAffineTransform (譬如：圖片1的點，圖片2的點，input 應該包含兩張對應圖片的點)

## D7：perspective transformation 概念與實作

【今日無需提交作業，請完成今日範例練習與思考題，按下「我已完成」按鈕即可】

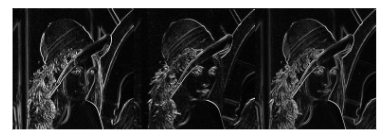
1. 思考一下如果經過透視變換，是否會因為視角轉換的差異導致部份 pixel 遺失造成無法修復的問題？
2. 透視變換是否有甚麼實際上的應用？

若思考題想要跟出題老師討論交流，歡迎您到共學社團上提問。

## D8：Filter 操作 (Sobel edge detect, Gaussian Blur)

實作邊緣檢測 (Sobel Filter)

* 比較 Sobel 如果在 uint8 的情況下做會 overflow 的狀況
* 比較一次與兩次計算偏微分的結果



## D9：SIFT 介紹與實作 (feature extractor)

【今日無需提交作業，請完成今日範例練習與思考題，按下「我已完成」按鈕即可】

思考如果我們把 RGB channel 個別拆開計算 SIFT 是否會得到跟灰階圖的特徵相同？

回顧一下 SIFT 的演算法是否會考慮到顏色的變化？

## D10：SIFT 其他應用 (keypoint matching)

【今日無需提交作業，請完成今日範例練習與思考題，按下「我已完成」按鈕即可】

* 思考 Brute-Force Matching 在應用上是否有甚麼限制
* 特徵點 matching 是否還有其他的應用？panorama?

若思考題想要跟出題老師討論交流，歡迎您到共學社團上提問。



reference: [**http://www.cs.cmu.edu/afs/andrew/scs/cs/15-463/f05/pub/www/projects/fproj/mrom/**](http://www.cs.cmu.edu/afs/andrew/scs/cs/15-463/f05/pub/www/projects/fproj/mrom/)

**2. 電腦視覺深度學習基礎**

## D11：CNN分類器架構：卷積層

作業內容：

運用Keras搭建簡單的Dense Layer與 Convolution2D Layer，使用相同Neurons數量，計算總參數量相差多少。

## D12：CNN分類器架構：步長、填充

作業內容：

運用Keras搭建簡單的Convolution2D Layer，調整Strides與Padding參數計算輸出feature map大小。

## D13：CNN分類器架構：池化層、全連接層

運用Pooling 與Flatten觀看輸出Tensor的尺寸變化。

## D14：CNN分類器架構：Batch Normalization

運用Keras搭建簡單的Convolution2D Layer＋BN＋Activation Fucntion層

## D15：訓練一個CNN分類器：Cifar10為例

運用Keras 搭建一個CNN 分類器，Training\_Data使用Cifar10，嘗試在驗證集資料上達到準確度80%

## D16：如何使用Data Augmentation

應用Imgaug做Image Augmentation

## D17：AlexNet

今日無需提交作業

完成本日推薦閱讀後請至下方點擊完成作業即可。

1. FLOPS 與 GOPS 計算

[**https://blog.csdn.net/weixin\_36474809/article/details/85675504**](https://blog.csdn.net/weixin_36474809/article/details/85675504)

2. RBF

[**https://www.ycc.idv.tw/ml-course-techniques\_7.html**](https://www.ycc.idv.tw/ml-course-techniques_7.html)

## D18：VGG16 and 19

搭建一個Vgg 16的模型

## D19：InceptionV1-V3

## D20：ResNetV1-V2、InceptionV4、Inception-ResNet

* 搭建Inception-ResNet中的 Inception Block A-C
* 嘗試導入Inception Block到Vgg16的結構中

## D21：Transfer learning

使用Keras做Transfer Learning(以Xception為backbone)

## D22：Breaking Captchas with a CNN

利用CNN做驗證碼識別

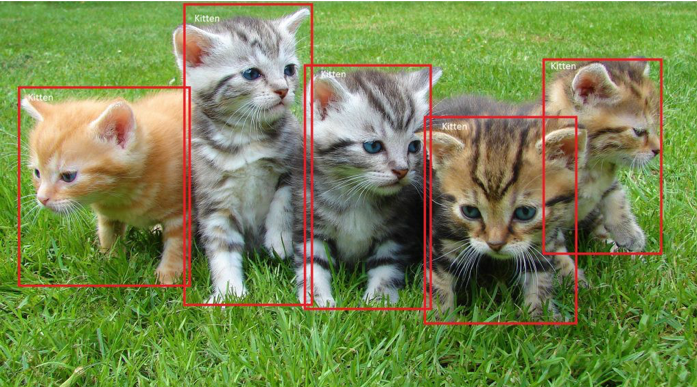
**3. CNN 應用案例學習**

## D23：Object detection原理

### Sample Code & 作業內容

【今日無需提交作業，請完成今日參考資料的閱讀與思考題，按下「我已完成」按鈕即可】

* 請列舉出 Two Stage 與 One Stage 的經典演算法各三個，並比較其運算速度與精度 (MAP)。
* 有空的學員們也可以先預習Object Detection的演進，下一章會較深入的帶大家認識。



參考資料：

* Object Detection: [**https://www.youtube.com/watch?v=5e5pjeojznk**](https://www.youtube.com/watch?v=5e5pjeojznk)
* Selective Search: [**https://blog.gtwang.org/programming/selective-search-for-object-detection/**](https://blog.gtwang.org/programming/selective-search-for-object-detection/)

## D24：Object detection基本介紹、演進

【今日無需提交作業，請閱讀完今日簡報內容，完成今日知識重點，按下「我已完成」按鈕即可】

* 了解 Two Stage 為何通常擁有較高的精度。
* 了解 One Stage 的 Default Anchor Box 概念。
* 了解 SSD 相較於 YoloV1 的優勢。
* 了解 Focal Loss 設計的核心理念，如何處理易分類樣品與類別不均

## D25：Region Proposal、IOU概念

運用給定的BBOX計算兩者之間的IOU

## D26：RPN架構介紹

學習如何架構一個RPN網路

## D27：Bounding Box Regression原理

如何設計一個BBOX Regressor

## D28：Non-Maximum Suppression (NMS)原理

運用Non-Maximum Suppression於BBOX預測

## D29 - D31：程式導讀、實作

今日作業：徹底了解程式中重要步驟的原理

請點擊下方檢視範例參考Data資料夾，作業請提交提交執行完的Build SSD\_VGG.ipynb檔案。

請注意：資料夾「Object Detection 程式導讀」需全部一起下載。

檔案更動：資料夾「Object Detection 程式導讀」>「data」內的 coco.py 有更新，若已經下載資料夾的學員，可以單獨至範例下載 coco.py 檔案，若從未下載資料夾，則只需要下載整個資料夾即可，不需重複下載範例的coco.py 檔案。更新時間 2020/01/10 11:03 AM

新增下載檔案：更新時間 2020/02/15 18:25 AM

因部分資料被移除，除了Data資料以外，再麻煩額外下載以下三個檔案：

1.demo weight(ssd300\_mAP\_77.43\_v2.pth'):

[**https://s3.amazonaws.com/amdegroot-models/ssd300\_mAP\_77.43\_v2.pth**](https://s3.amazonaws.com/amdegroot-models/ssd300_mAP_77.43_v2.pth)

2.載入資料型態

標註：

[**http://images.cocodataset.org/annotations/annotations\_trainval2014.zip**](http://images.cocodataset.org/annotations/annotations_trainval2014.zip)

影像：

[**http://images.cocodataset.org/zips/train2014.zip**](http://images.cocodataset.org/zips/train2014.zip)

## D32：YOLO 簡介及算法理解

相信今天的簡報還有程式碼範例的信息量已經足夠大了，可以多花時間消化簡報和程式碼範例，作業就問一個非常簡單的問題來確保大家有理解到重點，請問在 YOLOv1 的設計下，一張圖片最多能檢測出多少個 bbox 呢？

## D33：YOLO 細節理解 - 網路輸出的後處理

在NMS流程中，IoU重疊率參數(nms\_threshold )調高，試著思考一下輸出的預測框會有甚麼變化?

## D34：YOLO 細節理解 - 損失函數

思考題:

仔細觀察，bbox 寬高計算損失方式和bbox中心計算損失方式有哪邊不一樣嗎? 為什麼要有不同的設計?

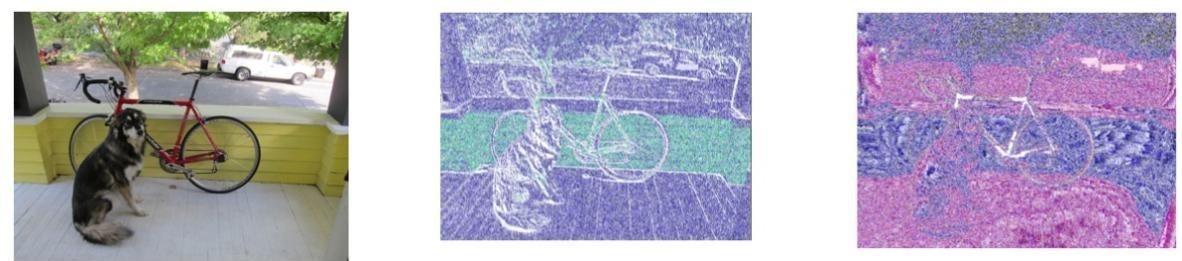
## D35：YOLO 細節理解 - 損失函數程式碼解讀

【今日無需提交作業，請完成今日範例練習，按下「我已完成」按鈕即可】

徹底了解程式中重要步驟的原理，如果第一次看不太懂沒關係，這需要時間慢慢理解，先清楚大架構，yolo系列當上完時，可以再回來看，說不定到時候就會更清楚了。

## D36：YOLO 細節理解 - 網路架構

1. 打開 Day36.YOLO 細節理解-網絡架構\_HW.ipynb
2. 動作做做看，運用 Keras，影像通過  1\*1 和 3\*3 的卷積後，
3. 影像會有什麼變化?
4. 多幾層又有甚麼不同的效果?



## D37：YOLO 細節理解 - 網路架構程式碼解讀

【今日無需提交作業，請完成今日範例練習，按下「我已完成」按鈕即可】

今日練習：徹底了解程式中重要步驟的原理

## D38：YOLO 演進

【今日無需提交作業，請完成今日推薦閱讀，按下「我已完成」按鈕即可】

有興趣的同學可以嘗試以你最能夠表達的方式 (文字/程式碼) 描述 YOLOv2 預期的 output layer 長什麼樣子，確保理解正確。

#### 推薦閱讀資料：

1. [**YOLO 的發展**](https://zhuanlan.zhihu.com/p/41438057)

2. YOLOv2

* [**YOLOv2--論文學習筆記（算法詳解）**](https://www.twblogs.net/a/5bafd96b2b7177781a0f6394)

3. YOLOv3

[**【目標檢測簡史】進擊的YOLOv3，目標檢測網絡的巔峰之作**](https://zhuanlan.zhihu.com/p/35394369)

## D39：使用 YOLOv3 偵測圖片及影片中的物件

該版本的 yolov3 實現邏輯主要寫在 `[**yolo.py**](http://yolo.py/)` 中 `YOLO` 這個 class 的 `detect\_image` ，其回傳已畫上檢測到的 bboxes 和物件類別的圖片。

請嘗試閱讀及盡量理解 `detect\_image` 的程式碼片段 (當然，如果能完全理解這個 repo 的所有程式碼是最好的！)

請修改/模仿 `detect\_image` 的寫法，使其回傳 bboxes 的信息、信心度及 bboxes 對應的類別

## D40：更快的檢測模型 - tiny YOLOv3

請詳細閱讀 [**https://github.com/qqwweee/keras-yolo3**](https://github.com/qqwweee/keras-yolo3) 的 readme ，按照 readme 的說明嘗試修改程式碼範例，學會使用 tiny YOLOv3 來做檢測，並且計算其平均 fps 以及觀察其檢測結果。



## D41：訓練 YOLOv3

如何使用已經訓練好的模型？

依照 [**https://github.com/qqwweee/keras-yolo3**](https://github.com/qqwweee/keras-yolo3) 的程式碼，請敘述，訓練模型時，資料集的格式是什麼？具體一點的說，要提供什麼格式的文件來描述資料集的圖片以及 bboxes 的信息呢？

**4. 電腦視覺深度學習實戰**

## D42：人臉關鍵點-資料結構簡介

請詳細閱讀程式碼，理解如何利用 pandas 及 numpy 讀取 kaggle 人臉關鍵點資料集 csv 文件並轉換為後續使用的關鍵點以及圖像資料結構。

理解後請嘗試將關鍵點畫在圖像上，確保圖像及資料有正確讀取，如下圖所示。

Hint: 可以用 **[matplotlib](https://matplotlib.org/3.1.1/gallery/shapes_and_collections/scatter.html" \t "_blank)** 的 [**scatter**](https://matplotlib.org/3.1.1/gallery/shapes_and_collections/scatter.html) 或者 **[opencv 的 cv2.circle](https://docs.opencv.org/master/dc/da5/tutorial_py_drawing_functions.html" \t "_blank)**

## D43：人臉關鍵點-檢測網路架構

嘗試使用 keras 來定義一個直接預測 15 個人臉關鍵點坐標的檢測網路，以及適合這個網路的 loss function

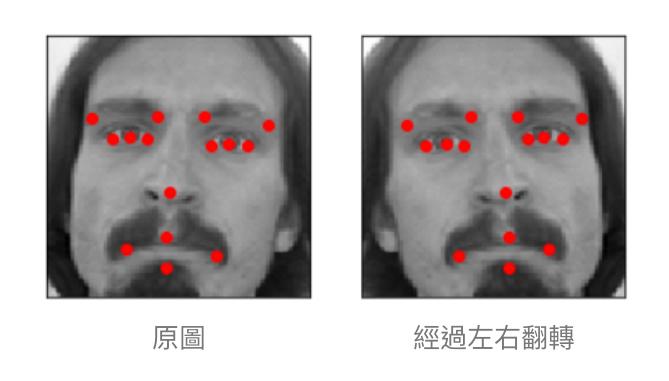
Hint: 參考前面的電腦視覺深度學習基礎

## D44：訓練人臉關鍵點檢測網路

**Sample Code & 作業內容**

請嘗試對原訓練集的圖像做左右翻轉來做 augmentation 以降低人臉關鍵點檢測的 loss

Note: 圖像做左右翻轉之後，groundtruth 的關鍵點也要適當地調整哦

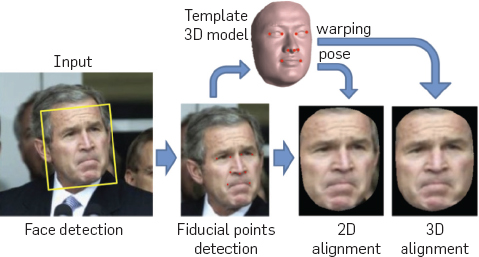


## D45：人臉關鍵點應用

【今日無需提交作業，請完成今日範例練習與推薦閱讀，按下「我已完成」按鈕即可】

人臉關鍵點檢測的知識點內容到今天結束，人臉關鍵點檢測是人臉識別其中一個模塊，本課程知識點並未包含完整的人臉識別流程，今天的作業只需您閱讀以下推薦的人臉識別相關文章，理解人臉關鍵點檢測是如何被應用在人臉識別的任務中。

* [**Deep Face Recognition: A Survey**](https://arxiv.org/pdf/1804.06655.pdf)
  + [**人臉識別：Deep Face Recognition論文閱讀**](https://yongyuan.name/blog/deep-face-recognition-note.html)
  + [**【論文學習】人臉識別 —— Deep Face Recognition: A Survey. 新人必看入門總結**](https://blog.csdn.net/DL_wly/article/details/93902260)



## D46：Mobilenet

嘗試用 keras 的 DepthwiseConv2D 等 layers 實做 Separable Convolution.

* depthwise's filter shape 爲 (3,3), padding = same
* pointwise's filters size 爲 128
* 不需要給 alpha, depth multiplier 參數

## D47：Ｍobilenetv2

嘗試用 keras 的 DepthwiseConv2D 等 layers 實做 Inverted Residual Block.

* depthwise's filter shape 爲 (3,3), padding = same
* 不需要給 alpha, depth multiplier 參數
* expansion 因子爲 6

## D48：Tensorflow Object Detection API

請點擊下方檢視範例，

請參考 Day048\_tensorflow\_object\_detection\_api\_**installation**.ipynb 步驟進行工具安裝

請參考 Day048\_tensorflow\_object\_detection\_api\_**training**.ipynb 步驟執行訓練模型範例

**5. 期末專題**

## D49 - D50：期末專題

### 期末專題完賽流程

Step 1. 學員選擇期末專題主題之一進行

Step 2. 將期末專題程式碼上傳至 github 後，至活動官網完成提交

Step 3. 請至 [**深度學習與電腦視覺共學社團**](https://www.cupoy.com/club/ai_tw/0000016E62FABB7A000000016375706F795F72656C656173654B5741535354434C5542/content/post) 建立期末專案文章分享

Step 4. 請參考 [**深度學習與電腦視覺期末專題\_Cupoy (格式參考)**](https://www.cupoy.com/clubnews/ai_tw/0000016E62FABB7A000000016375706F795F72656C656173654B5741535354434C5542/0000017041EC02FF000000066375706F795F72656C656173654B5741535354434C55424E455753) 文章作為專案文章建立基礎

Step 5. 由其他學員與出題教練進行按讚，主辦單位進行票數統計

Step 6. 票數最高的 10% 學員將獲得期末挑戰紀念品

# 深度學習與電腦視覺期末專題\_Cupoy (格式參考)

2020/02/14

### 一、專題摘要 ****(解釋實作與說明需要解決的問題，限300~500字。)****

1. 期末專題主題
2. 期末專題基本目標

### 二、實作方法介紹 ****(介紹使用的程式碼、模組，並附上實作過程與結果的截圖，需圖文並茂。)****

1. 使用的程式碼介紹
2. 使用的模組介紹

### 三、成果展示 (介紹成果的特點為何，並撰寫心得。)

### 四、結論 ****(總結本次專題的問題與結果)****

### 五、期末專題作者資訊 (請附上作者資訊)

1. 個人Github連結
2. 個人在百日馬拉松顯示名稱