**深度學習與電腦視覺期末專題**

**一、專題摘要**

1. 期末專題主題:  浣熊與袋鼠辨識模型
2. 期末專題基本目標:
   * 使用Darknet yolov3訓練一個模型，同時能辨識出浣熊(raccoon)與袋鼠(kangaroo)的類別與位置
   * 使用預訓練模型以及其在影像辨識中的作用，運用模型
   * 把圖片/影片中的袋鼠與浣熊位置找出來

**二、實作方法介紹**

**[使用環境]**

OS: Windows 10

GPU driver: Nvidia GeForce GTX 1070 ti

CUDA: Toolkit 10.2

cuDNN: cuDNN 7.6.5

Visual Studio 2015

Opencv 3.4.2

**[步驟概述]**

1. 環境架設
   1. 安裝GPU相關工具: Nvidia deriver CUDA、cuDNN
   2. 安裝編譯工具VS2015
   3. 安裝 opencv library
   4. 編譯 darknet
2. 資料集處理
   1. 收集資料集
   2. 標註檔轉換為yolo format
   3. 切分train/test set
3. 訓練自己的模型
   1. 撰寫設定檔obj.names與obj.data
   2. 下載pre-train model
   3. 修改model cfg檔案
   4. 開始訓練
4. 測試模型(demo)

**[Step1] 環境架設**

**1.1 安裝GPU相關工具: Nvidia** **driver CUDA、cuDNN**

**1.1.1 Nvidia GPU driver**

先檢查電腦內之驅動版本是否高於「376.51」，低於請下載最新的版本來更新驅動！

      驅動程式下載：<https://www.nvidia.com.tw/Download/index.aspx?lang=tw>

      "控制台/硬體和音效/NVDIA控制面板" 點進去就可以看到驅動版本了！

****

**如果沒有安裝驅動程式**

     請去官網 <http://www.nvidia.com/Download/index.aspx?lang=en-us> 選擇適合自己顯卡的驅動並下載

**1.1.2 CUDA**

      安裝時注意，請選擇自定安裝，因為已經自行安裝較高版本的驅動，所以只需要勾選CUDA即可。

      CUDA下載：<https://developer.nvidia.com/cuda-toolkit-archive>

      我選擇了CUDA10.2版本。

      p.s. 注意Nvidia GPU driver版本, 不同版本driver會有不同cuda版本, 容易出現"CUDA driver version is insufficient for CUDA runtime version"問題

      安裝好Nvidia GPU driver後先用nvidia-smi指令查看driver所使用的cuda版本, 再去下載並安裝相同版本cuda比較保險

      P.s. 安裝位置請別更改，以免有錯

      P.s. 如果之前已有安裝visual studio請在背景把它關掉不然會跟安裝CUDA衝突



**1.1.3 cuDNN**

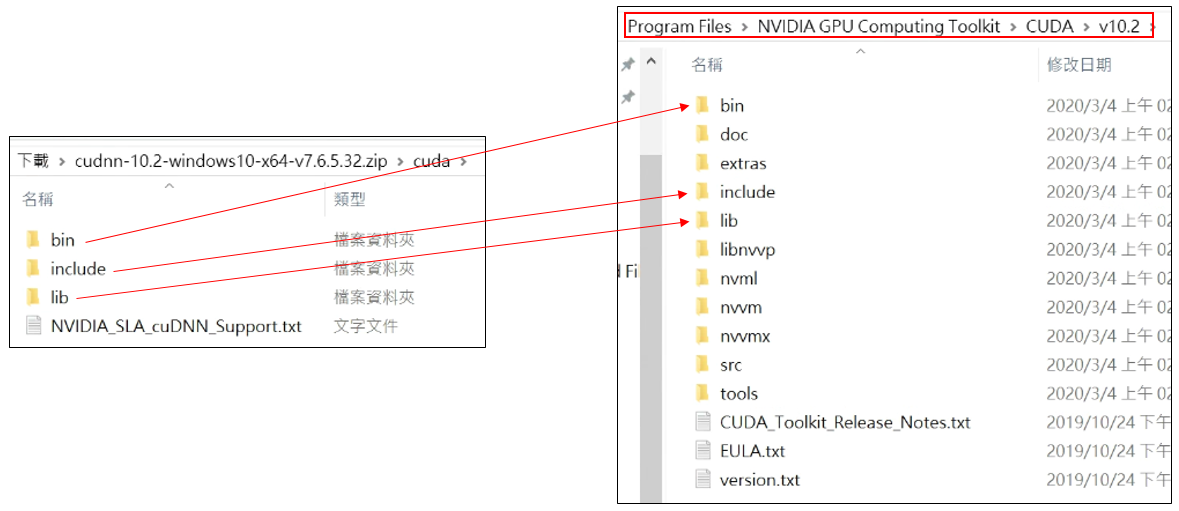
      下載cuDNN: <https://developer.nvidia.com/rdp/cudnn-archive>

      在這裡因為我安裝的是CUDA Toolkit 10.2 所以選對應的進行下載。

      P.s. 請根據你的CUDA版本選擇對應版本下載！



     解壓縮後，將三個資料夾中的檔案，個別複製到CUDA的相對應資料夾中即可。



     接下來將cuda目錄路徑加入系統環境變

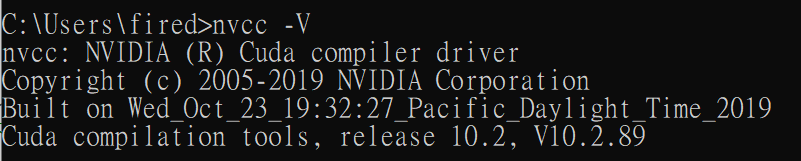
     進入控制台/系統及安全性/進階系統設定/進階/環境變數

     尋找「系統變數System」中的「Path」，新增以下變數：

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.2\bin

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.2\lib\x64

     最後，檢查是否安裝成功  
     進入CMD並輸入「nvcc -V」檢查是否安裝成功



**1.2 安裝編譯工具VS2015**

     visual studio下載：<https://visualstudio.microsoft.com/zh-hant/downloads/>

     選擇想要的安裝路徑，選擇自定安裝，只勾選安裝必要的「Visual C++」，其他功能若有需要請自行安裝

**1.3 安裝 Opencv library**

 opencv下載: <https://opencv.org/releases/>

     下載完成後將壓縮檔解壓縮到自己想放的路徑即可。

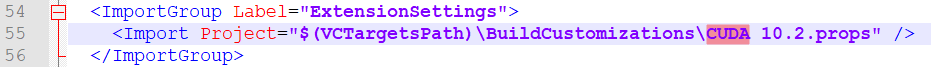
**1.4 編譯darknet**

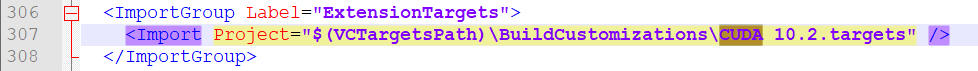
  下載darknet源碼: <https://github.com/AlexeyAB/darknet>

     下載完後，解壓縮放在你想放的目錄即可。

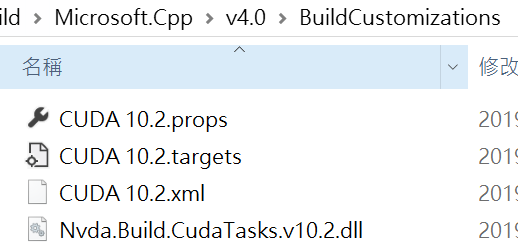
     接著修改build/darknet/darknet.vcxproj 文件 (可用notepad++文字編輯器開啟)

     將CUDA版本號改為自己用的版本 (文件中有兩處需要修改，請善用ctrl+F)





    確認C:\Program Files (x86)\MSBuild\Microsoft.Cpp\v4.0\V140\BuildCustomizations目錄中是否有以下文件

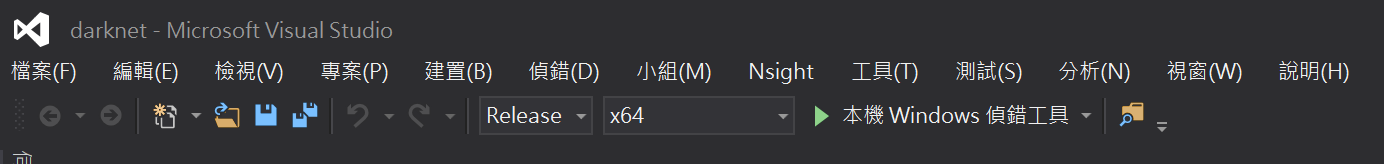


   如果沒有，請把CUDA目錄中的以上4個文件複製到 C:\Program Files (x86)\MSBuild\Microsoft.Cpp\v4.0\V140\BuildCustomizations

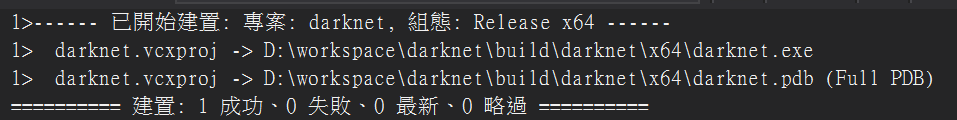
   例如: 我的CUDA路徑是C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.2\extras\visual\_studio\_integration\MSBuildExtensions

   接下來使用VS2015開啟darknet.sln，並將opencv導入project中 (可參考這篇: <https://farmercd.blogspot.com/2017/02/opencv-visual-studio.html>)

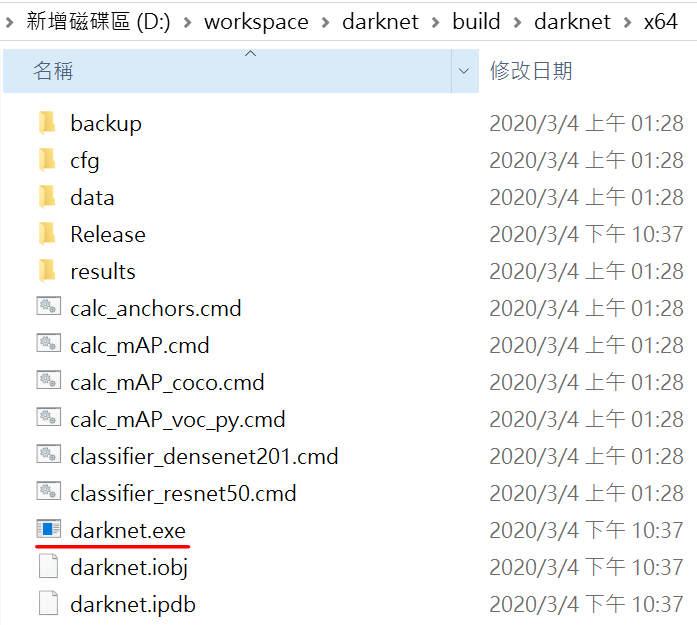
   將編譯組態設定成 release x64，即可按下建置，編譯darknet



   建置完成會出現成功訊息



   在build/darknet/x64目錄中就會看到darknet.exe可執行檔



**[Step2]** **資料集處理**

**2.1 收集資料集**

   浣熊資料集: <https://github.com/experiencor/raccoon_dataset>

   袋鼠資料集: <https://github.com/experiencor/kangaroo>

**2.2 標註檔轉換為yolo format**

   製作資料集目錄:

[dataset root dir]

   |- images: 放所有圖片

   |- labels: 放voc xml格式的標註文件

   |- Yolos: 放yolo format的標註文件與所有圖片

   |- cfg: 放obj.data, obj.names, yolov3.cfg, train.txt, test.txt 文件

       |-weights: 存放模型訓練的weights檔案

   將收集來的所有浣熊與袋鼠圖片放到 images 目錄中，voc xml格式標註文件放到 labels 目錄中

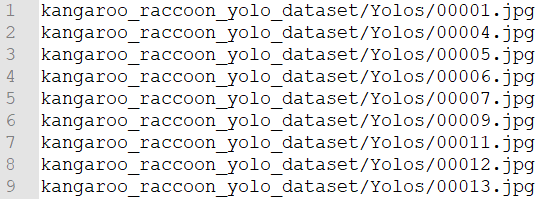
   使用**convert\_vocxml\_to\_yolofrmat.py**程式將 voc xml 標註文件轉換成yolo format的標註文件，並存放在Yolos目錄

**2.3 切分train/test set**

   使用split\_train\_test.py程式切分train與test set清單

   train/test.txt的檔案內容，就是訓練/測試圖片的路徑

   ex: train.txt



**[Step3]** **訓練自己的模型**

**3.1 撰寫設定檔obj.names與obj.data**

   obj.names：此檔內容為label的列表，例如kangaroo與raccoon，YOLO在訓練與預測時皆需要讀取此檔。

kangaroo

raccoon

   obj.data：定義label數目以及各個設定檔及weights目錄的path，YOLO訓練及預測時皆會讀取

classes=2

train=kangaroo\_raccoon\_yolo\_dataset/cfg/train.txt

valid=kangaroo\_raccoon\_yolo\_dataset/cfg/test.txt

namse=kangaroo\_raccoon\_yolo\_dataset/cfg/obj.names

backup=kangaroo\_raccoon\_yolo\_dataset/cfg/weights

**3.2 下載pre-train model**

   pre-train model: <https://pjreddie.com/media/files/darknet53.conv.74>

**3.3 修改model cfg檔案**

   從darknet目錄中將 yolov3.cfg 複製過來

   修改batch size:

batch=24  #是指每批次取幾張圖片進行訓練

subdivisions=8  #是指要將每批次拆成幾組，以避免GPU memory不夠

   修改輸出類別數量:

   將每個yolo層的前一個convolutional層的filters值改成 (偵測類別數量 + 5)\*3，ex: (2+5)\*3=21

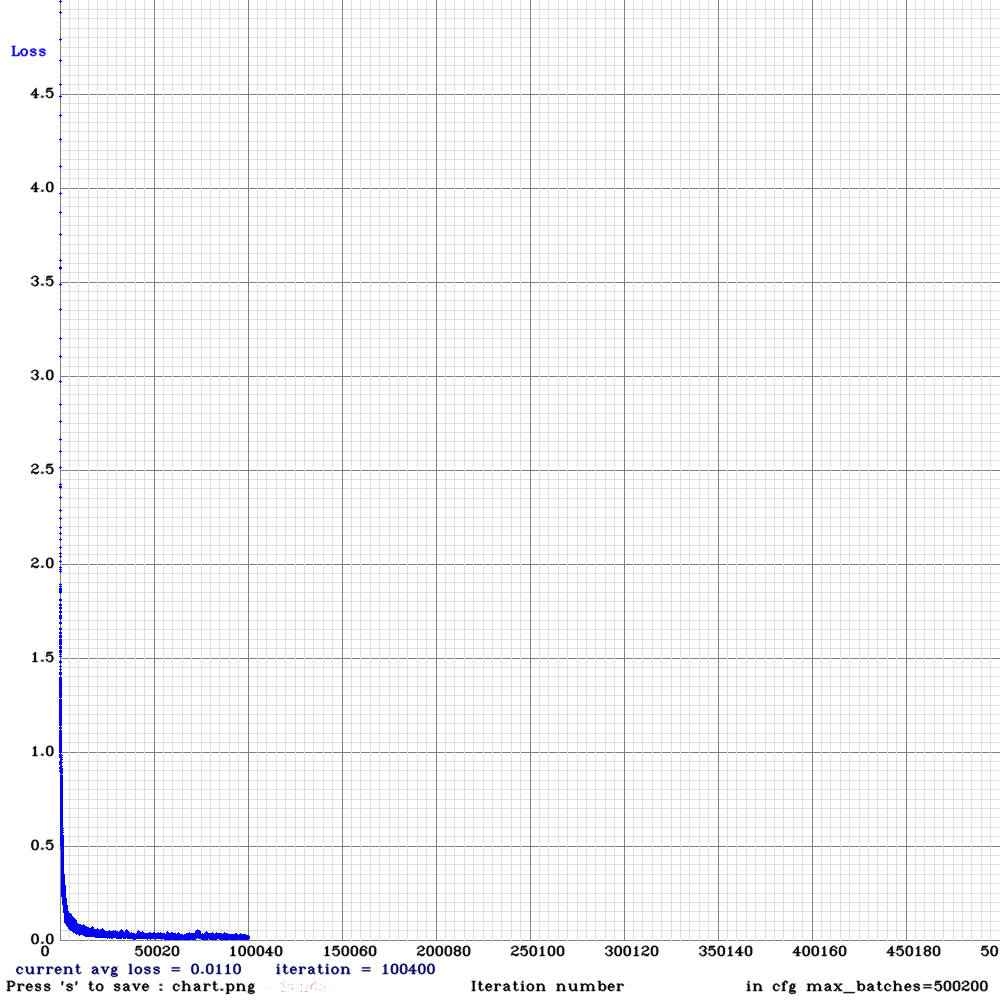
filters=21   #in our case filters=21

**3.4 開始訓練**

   開啟cmd，使用以下指令開始訓練

darknet -i 0 detector train kangaroo\_raccoon\_yolo\_dataset/cfg/obj.data kangaroo\_raccoon\_yolo\_dataset/cfg/yolov3.cfg pretrained/darknet53.conv.74

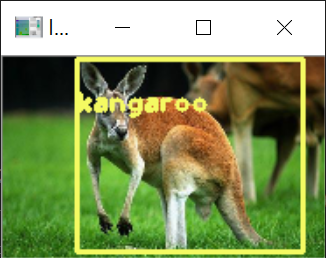
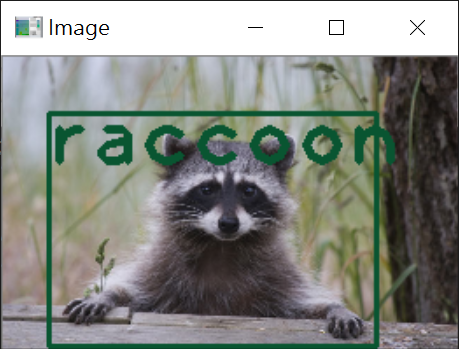
   訓練過程的loss圖:



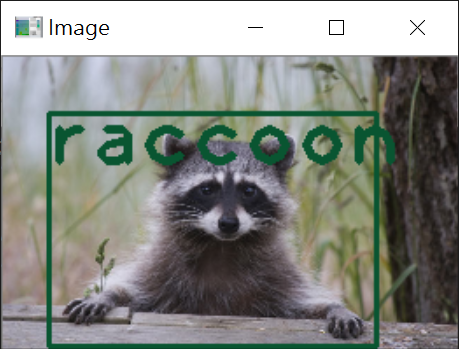
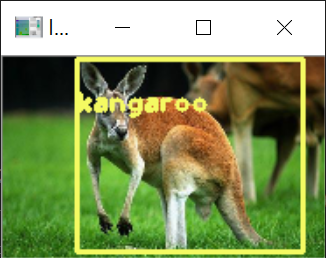
**[Step4]** **測試模型(Demo)**

   使用demo.py以下程式調用訓練好的model

   單張圖片測試結果:



**三、成果展示**

使用浣熊影片測試時 第一次使用darknet，從環境準備如CUDA、cuDNN、VS安裝，資料集處理、訓練、成果展示…等，過程花了不少時間摸索。  
雖然最後跑測試影片的效果差強人意，不過能將這整個流程親自操作一次覺得很有成就感，是很棒的入門經驗。**** ****

**四、結論**

使用浣熊影片測試時效果不太好，可能是因為訓練資料太少且類型較單一，例如:浣熊訓練資料大部分都是浣熊正面照片，而測試影片中的浣熊並非都是正面入鏡，所以才會偵測較果較差，若能找到更多訓練資料加入訓練應該能有更好的表現。

**五、期末專題作者資訊**

1. 個人Github連結:   
   <https://github.com/weishinyi/1st-DL-CVMarathon/tree/master/homework/Day_049-050>
2. 個人在百日馬拉松顯示名稱: 魏欣怡