Assignment 1 find the duck

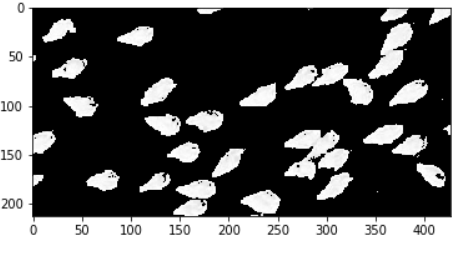
環境：jupyter notebook

程式架構：此次寫了兩支程式duck\_test 與duck\_final，duck\_test只利用一張duck\_pixel的圖片建構兩類之高斯模型，即輸出最終預測結果，duck\_final可自動讀入多張duck\_pixel與non\_duck\_pixel建構高斯模型，再輸出預測結果。

Duck\_test

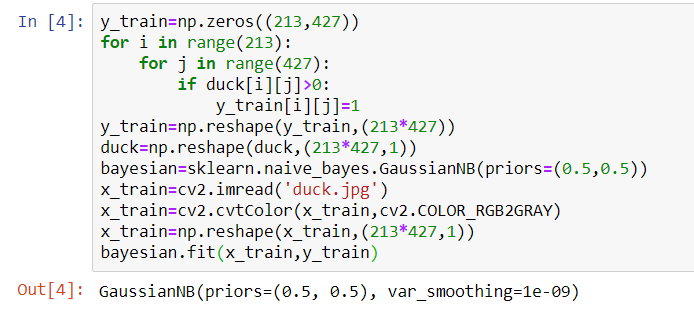


First step:利用小畫家切下full\_duck中有duck的部分，將有duck的影像讀入並轉為灰階，將相素值沒那麼大（較暗沉）的點設為０，結果如下，會留下duck\_body的pixel

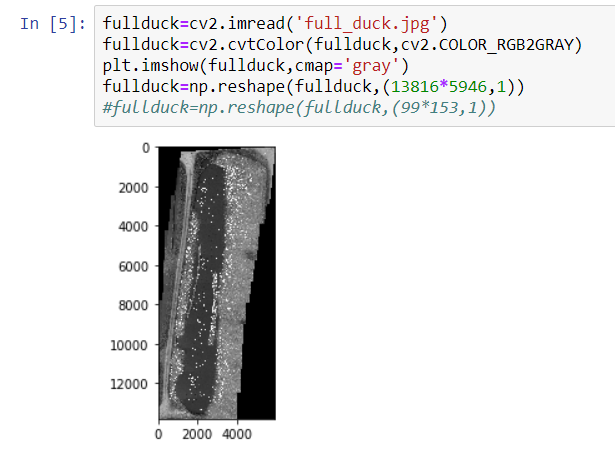


Second step:利用處理完的圖片建立Bayesian classifier需要的label(y\_train)如圖片的pixel大於0　,　y\_train即為1，代表其為duck\_pixel

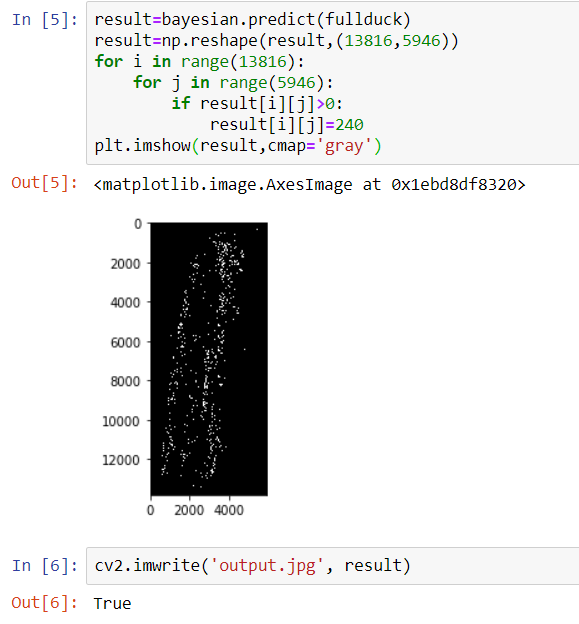
接著讀入原本還未處理過的duck\_pixel(轉灰階)建立x\_train，最後丟入bayesian classifier建立模型



Third step：讀入full\_duck\_pixel並轉換為可分類的資料維度



利用訓練完的bayesian classifier預測處理完的full\_duck其label,利用for迴圈將值為１的資料轉為較大數值，並轉為二維陣列，儲存輸出結果



Duck\_final

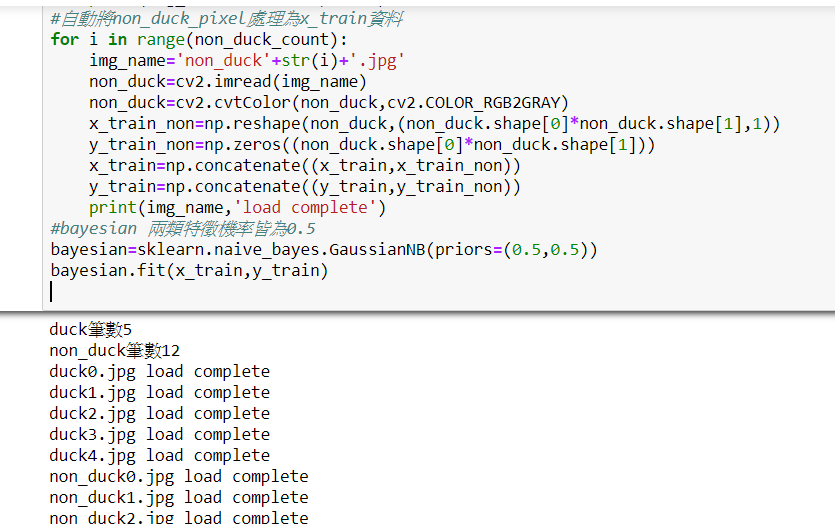
大致上與duck\_test相同，有鑑於本來架構讀取圖片很麻煩需要手打程式碼，因此增加了可讓程式自動讀入training data 的處理，並可印出兩類prior的高斯分布。

1. 首先給定總共有幾筆有duck\_pixel與non\_duck\_pixel

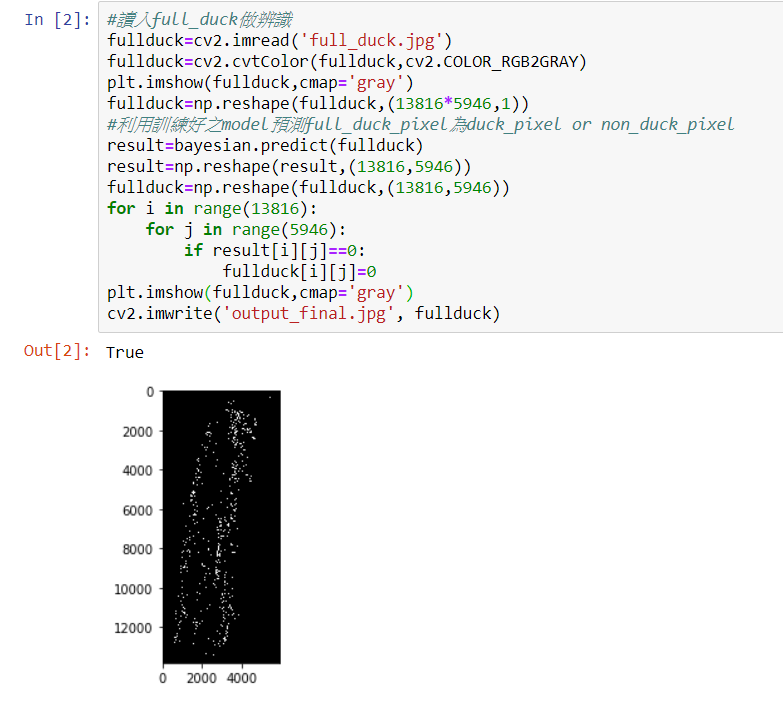


1. duck\_pixel的處理，將圖片讀入後整理為x\_train的資料維度丟入x\_train

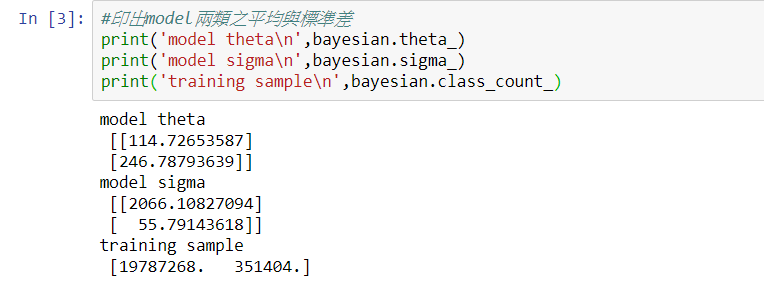
另外建立一變數y\_train\_duck儲存此筆duck\_pixel的label，最後丟入y\_train，即進行完此次duck\_pixel的處理



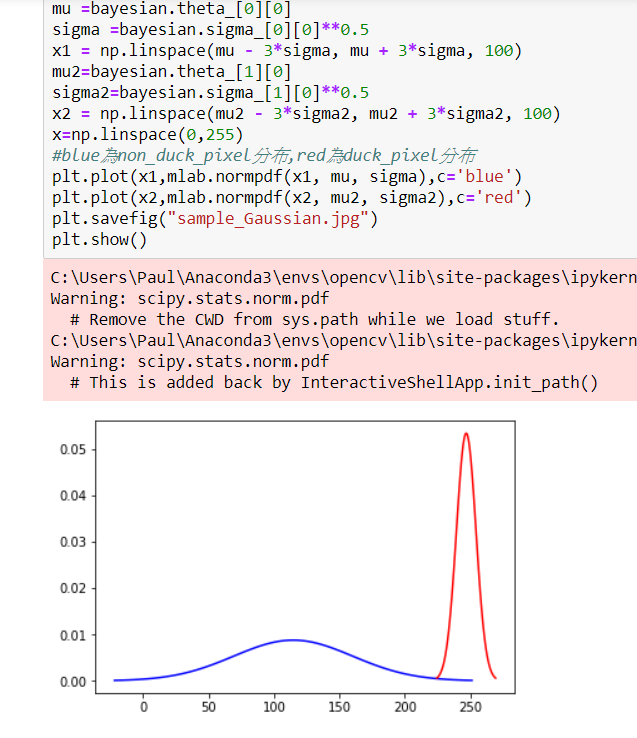
1. non\_duck\_pixel的處理，與duck\_pixel相同，但不需做轉換y\_train的處理，只需將y\_train設為維度與輸入圖片相同的0陣列即可，接著將處理完的兩類樣本丟入bayesian classifier裡做訓練



1. 和duck\_test類似，讀入full\_duck做預測產生圖片，利用預測出來的result處理full\_duck，如果result為0，對應的full\_duck像素也會被設為0



1. 最後印出兩類平均與標準差，繪製高斯曲線圖



Result

這次的作業總使用第一支程式duck\_test產生最一開始的output : sample\_only\_duck\_pixel\_output作為比較

第二支產

Sample1:

model theta

[[106.93681475]

[246.78793639]]

model sigma

[[1814.28031059]

[ 55.79143618]]

training sample

(non\_duck0~non\_duck7) (duck0~duck4)

[11272884. 351404.]

Sample2:

model theta

[[102.42188502]

[246.57023706]]

model sigma

[[2208.95641757]

[ 54.94298873]]

training sample

(non\_duck0~non\_duck7) (duck0~duck4+duck\_paint+duck\_black)

[11809284. 419636.]

Sample3:

model theta

[[114.72653587]

[246.78793639]]

model sigma

[[2066.10827094]

[ 55.79143618]]

training sample

(non\_duck0~non\_duck11) (duck0~duck4)

[19787268. 351404.]