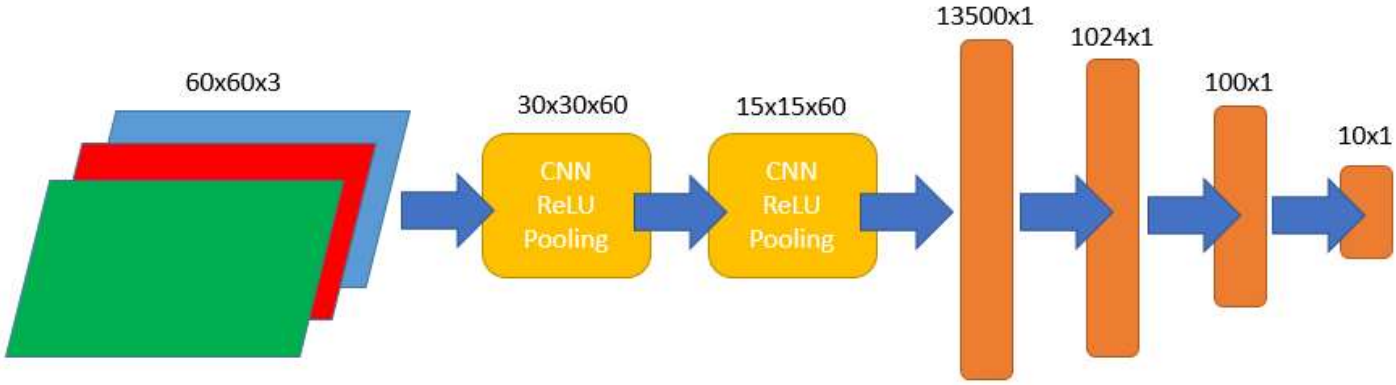
Prob1

此次作業我使用了 putty 連到實驗室裝有 GPU 的伺服器做模型訓練，並且使用自己電腦上的 Jupyter notebook 來打程式。這次總共有 10 種動物類別，每個類別有 1000 張圖片作為 training data，剩餘 400 張圖片作為 testing data。由於每張圖片大小並不相同，我使用了 OpenCV 函式來做圖片處理。首先利用 os.listdir()函式讀出所有路徑下的圖片檔名，接著使用 cv2.imread()讀取圖片並做 cv2.resize()成我所指定的圖片大小，以方便之後丟入神經網路。接著我會示範在不同層數、不同 kernel size、不同 stride 之下所跑出來的 training 與 testing 正確率。

Case 1 配置如下:



詳細規格如下:

圖片 resize=60*60, batch size=50,

Layer1: CNN with stride=1, filter=60, patch size=3*3, from image(60*60*3)→feature map(60*60*60)
→ReLU→Pooling with stride=2→ feature map(30*30*60)

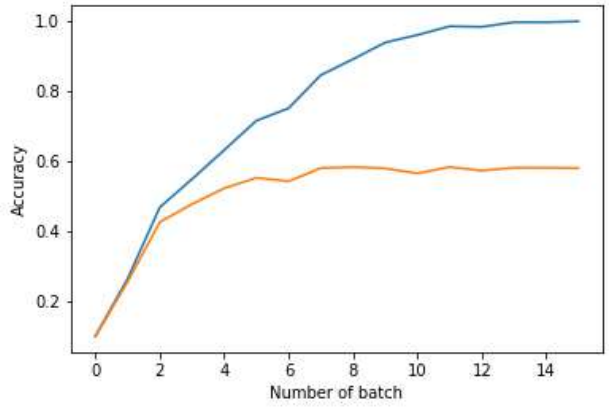
Layer2: CNN with stride=1, filter=60, patch size=3*3, from feature map(30*30*60)
→feature map(30*30*60)→ReLU→Pooling with stride=2
→ feature map(15*15*60)→flat to 15*15*60=13500

Layer3: Fully connected, from 13500→1024

Layer4: Fully connected, from 1024→100

Layer5: Fully connected, from 100→10

下圖為每個 epoch 後，對所有的 training data 與 testing data 做正確率計算(batch→epoch)

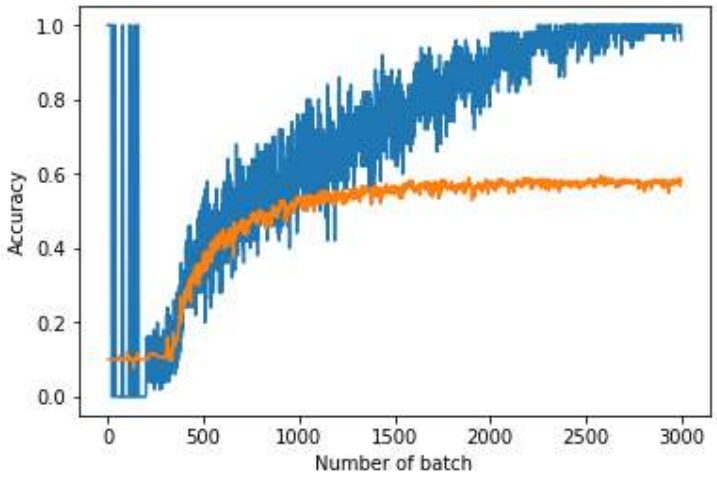


PS. 水平軸應為 Number of epoch

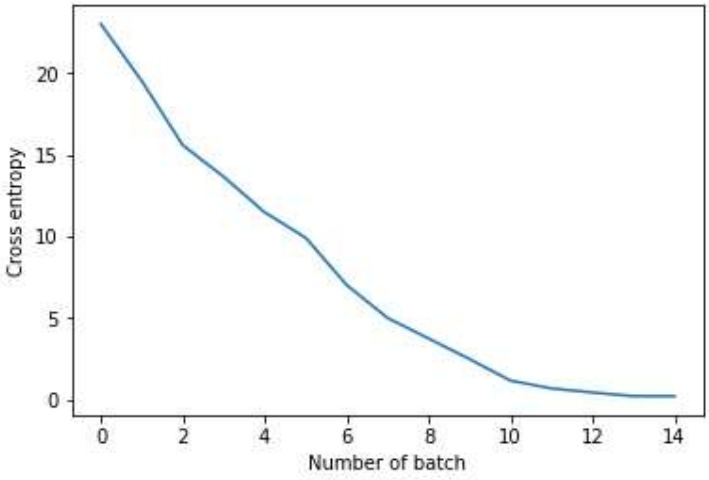
Epoch=15

Final result: training accuracy= 0.9996 , testing accuracy= 0.58025

下圖為每個 batch 後，對每個 batch 的 training data 與全部的 testing data 做正確率計算



下圖為每個 epoch 後，對所有 training data 計算出來的平均 Loss (learning curve)



Final result: Loss= 0.20935150146484374

Case 2 配置如下:

如同 case1 的架構，僅修改了 Layer2 中 CNN 的 kernel size 改成了 4x4

詳細規格如下:

圖片 resize=60*60, batch size=50,

Layer1: CNN with stride=1, filter=60, patch size=3*3, from image(60*60*3)→feature map(60*60*60)
→ReLU→Pooling with stride=2→ feature map(30*30*60)

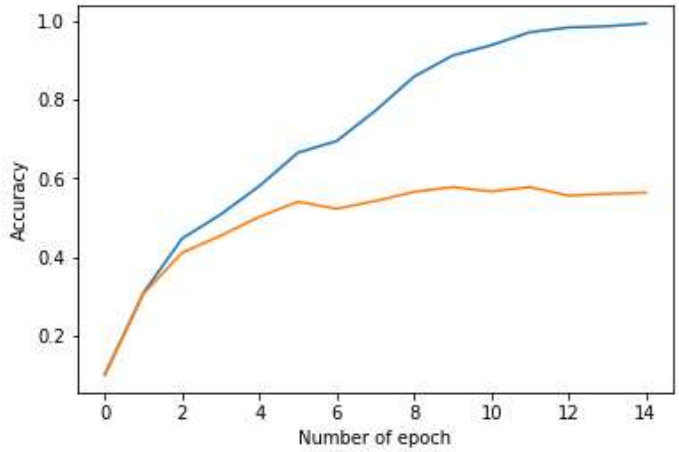
Layer2: CNN with stride=1, filter=60, patch size=4*4, from feature map(30*30*60)
→feature map(30*30*60)→ReLU→Pooling with stride=2
→ feature map(15*15*60)→flat to 15*15*60=13500

Layer3: Fully connected, from 13500→1024

Layer4: Fully connected, from 1024→100

Layer5: Fully connected, from 100→10

下圖為每個 epoch 後，對所有的 training data 與 testing data 做正確率計算(batch→epoch)



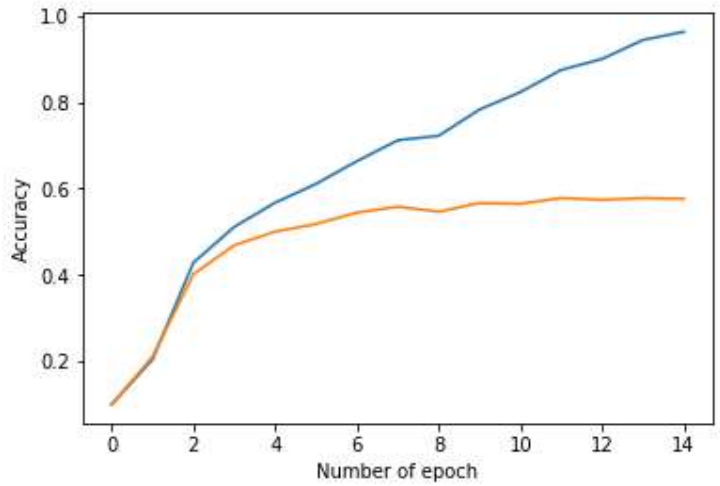
Final result: training accuracy= 0.993 , testing accuracy= 0.56375
收斂速度降低，在 epoch15 次後的正確率也降低

Case 3 配置如下:
如同 case1 的架構，僅修改了 Layer2 中 CNN 的 kernel size 改成了 2x2

詳細規格如下:

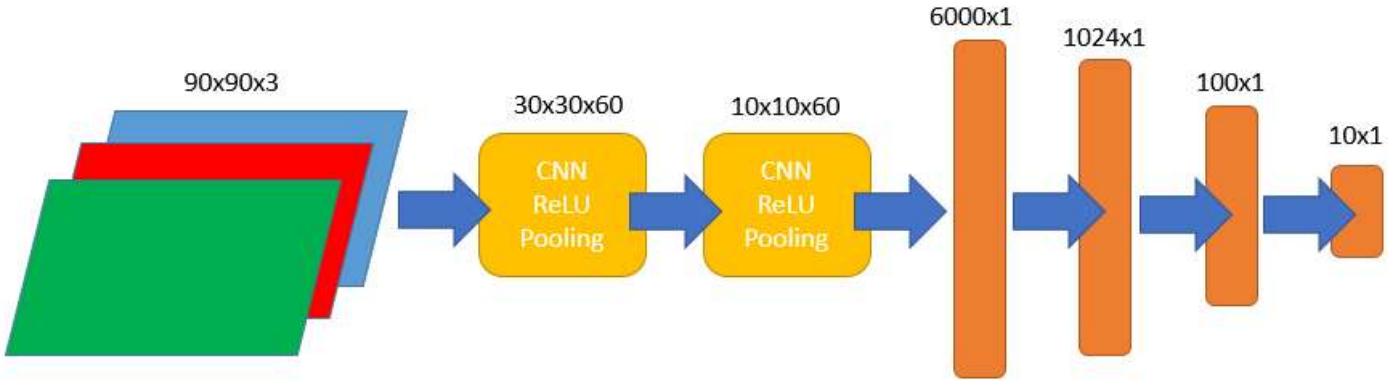
圖片 resize=60*60, batch size=50,
Layer1: CNN with stride=1, filter=60, patch size=3*3, from image(60*60*3)→feature map(60*60*60)
→ReLU→Pooling with stride=2→ feature map(30*30*60)
Layer2: CNN with stride=1, filter=60, patch size=2*2, from feature map(30*30*60)
→feature map(30*30*60)→ReLU→Pooling with stride=2
→ feature map(15*15*60)→flat to 15*15*60=13500
Layer3: Fully connected, from 13500→1024
Layer4: Fully connected, from 1024→100
Layer5: Fully connected, from 100→10

下圖為每個 epoch 後，對所有的 training data 與 testing data 做正確率計算(batch→epoch)



Final result: training accuracy= 0.9636 , testing accuracy= 0.57625

Case 4 配置如下:

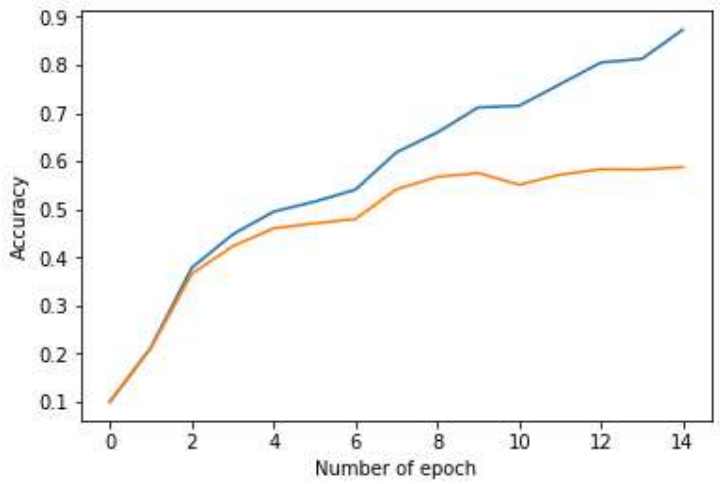


我改成將圖片 resize 成 90x90 的 pixel，並且改變了 pooling 的 stride size 為 3x3

詳細規格如下:

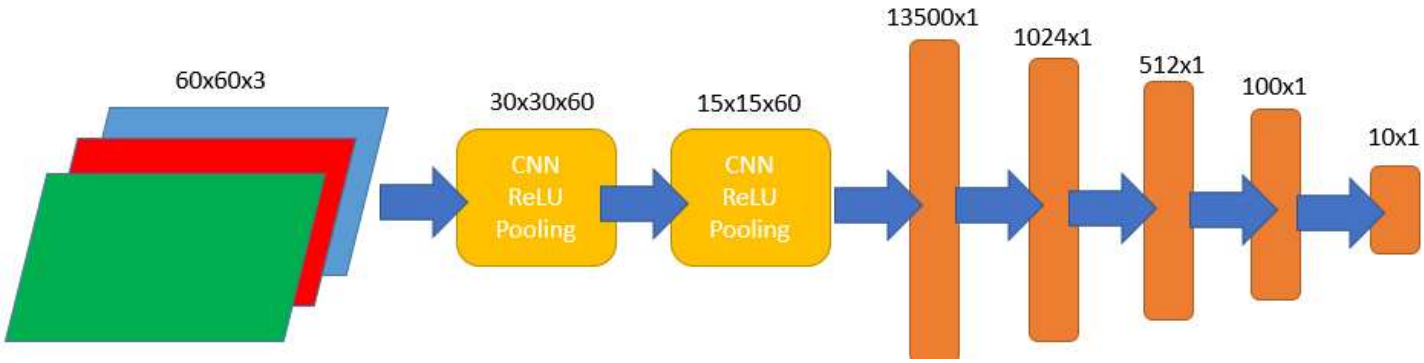
圖片 resize=90*90, batch size=50,
Layer1: CNN with stride=1, filter=60, patch size=3*3, from image(90*90*3)→feature map(90*90*60)
→ReLU→Pooling with stride=3→ feature map(30*30*60)
Layer2: CNN with stride=1, filter=60, patch size=3*3, from feature map(30*30*60)
→feature map(30*30*60)→ReLU→Pooling with stride=3
→ feature map(10*10*60)→flat to 10*10*60=6000
Layer3: Fully connected, from 6000→1024
Layer4: Fully connected, from 1024→100
Layer5: Fully connected, from 100→10

下圖為每個 epoch 後，對所有的 training data 與 testing data 做正確率計算(batch→epoch)



Final result: training accuracy= 0.8733 , testing accuracy= 0.5875
效果並不理想，可能是 feature map 的數量在 CNN 層時快速減少

Case 5 配置如下:



我在 Fully connected 層在加了一層 512 個節點的 hidden layer

詳細規格如下:

圖片 resize=60*60, batch size=50,

Layer1: CNN with stride=1, filter=60, patch size=3*3, from image(60*60*3)→feature map(60*60*60)
→ReLU→Pooling with stride=2→ feature map(30*30*60)

Layer2: CNN with stride=1, filter=60, patch size=3*3, from feature map(30*30*60)
→feature map(30*30*60)→ReLU→Pooling with stride=2
→ feature map(15*15*60)→flat to 15*15*60=13500

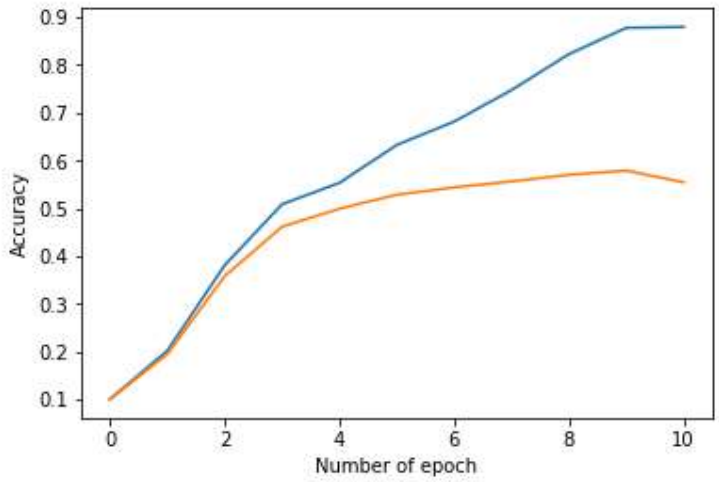
Layer3: Fully connected, from 13500→1024

Layer4: Fully connected, from 1024→512

Layer5: Fully connected, from 512→100

Layer6: Fully connected, from 100→10

下圖為每個 epoch 後，對所有的 training data 與 testing data 做正確率計算(batch→epoch)



Final result: training accuracy= 0.8799 , testing accuracy= 0.5545

探討不同動物種類的分類正確率

以下的測試我使用 case1 的架構來做訓練，記錄在不同 epoch 時，每種動物正確分類的機率

詳細規格如下:

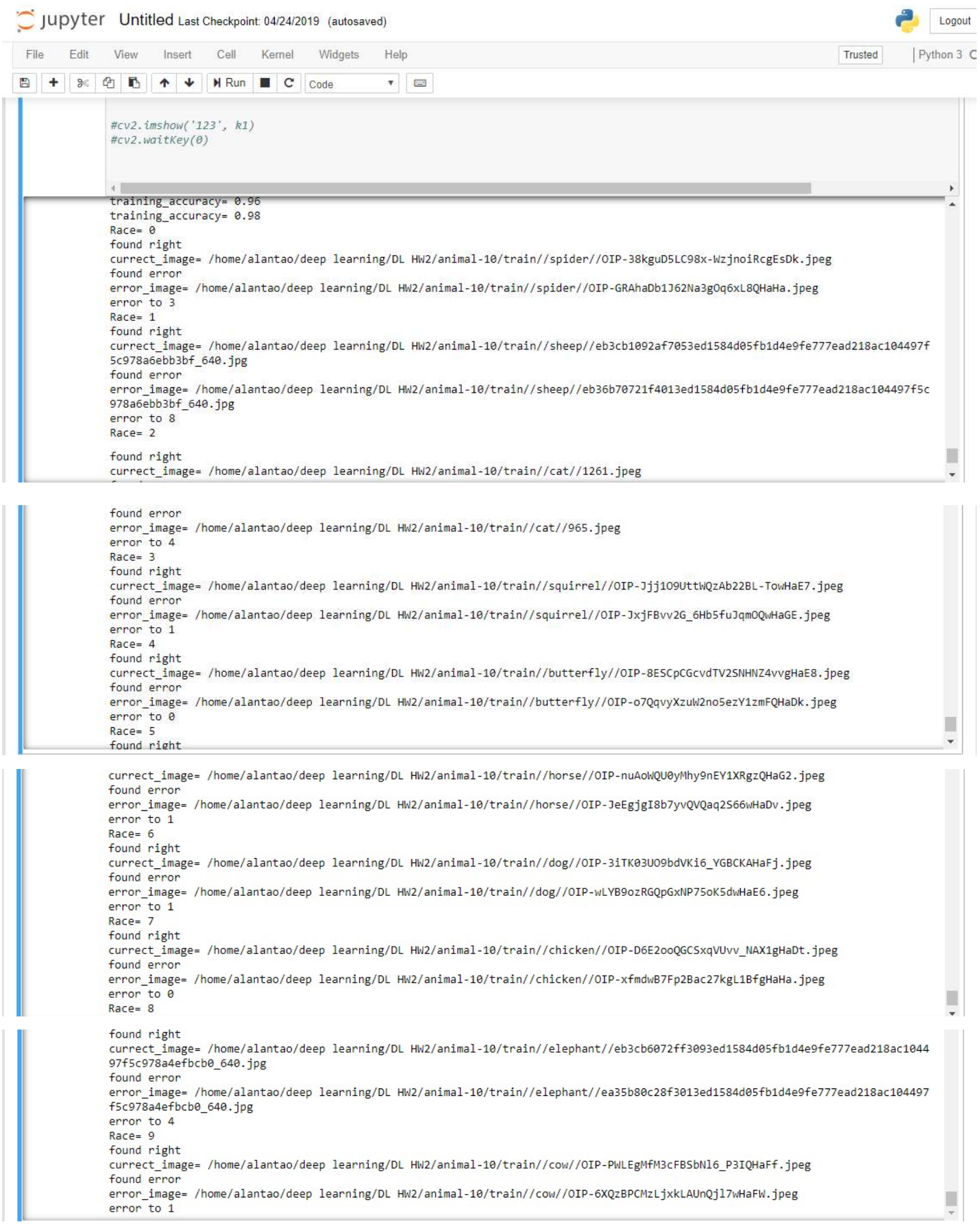
圖片 resize=60*60, batch size=50,
Layer1: CNN with stride=1, filter=60, patch size=3*3, from image(60*60*3)→feature map(60*60*60)
→ReLU→Pooling with stride=2→ feature map(30*30*60)
Layer2: CNN with stride=1, filter=60, patch size=3*3, from feature map(30*30*60)
→feature map(30*30*60)→ReLU→Pooling with stride=2
→ feature map(15*15*60)→flat to 15*15*60=13500
Layer3: Fully connected, from 13500→1024
Layer4: Fully connected, from 1024→100
Layer5: Fully connected, from 100→10

分類正確率:

Epoch/animal	butterfly	cat	chicken	cow	dog	elephant	horse	sheep	spider	squirrel
Epoch1	0	0	0	0	0.0375	0	0	0	0	0.9875
Epoch2	0.465	0.295	0.2075	0.2775	0.2625	0.1825	0.2825	0	0.46	0.2125
Epoch3	0.55	0.5075	0.6375	0.3575	0.435	0.1575	0.12	0.25	0.505	0.4925
Epoch4	0.58	0.545	0.3825	0.3125	0.67	0.4975	0.3	0.43	0.72	0.395
Epoch5	0.5525	0.6025	0.5525	0.54	0.585	0.5175	0.3325	0.575	0.6525	0.3775
Epoch6	0.6975	0.4375	0.3875	0.58	0.59	0.665	0.425	0.7025	0.685	0.3475
Epoch7	0.49	0.615	0.1825	0.61	0.7925	0.635	0.3	0.7675	0.625	0.41
Epoch8	0.695	0.515	0.5	0.5275	0.685	0.6525	0.33	0.645	0.7425	0.5075
Epoch9	0.6675	0.4975	0.63	0.645	0.715	0.67	0.4225	0.5925	0.645	0.4675
Epoch10	0.4725	0.5375	0.575	0.5825	0.7575	0.6375	0.5	0.665	0.5925	0.5175
Epoch11	0.6625	0.5475	0.5725	0.4775	0.625	0.6325	0.5275	0.6375	0.7475	0.415
Epoch12	0.515	0.515	0.5525	0.6325	0.79	0.78	0.435	0.58	0.535	0.43
Epoch13	0.58	0.425	0.6125	0.645	0.695	0.68	0.355	0.655	0.7225	0.4275
Epoch14	0.6875	0.5275	0.5875	0.54	0.6725	0.6425	0.4425	0.64	0.6575	0.495
Epoch15	0.705	0.53	0.5775	0.64	0.735	0.6125	0.37	0.6075	0.605	0.4875

探討分類正確與錯誤的動物圖片

我先利用 case1 的架構訓練完，再隨機從每種動物類別當中挑出 1 張分類錯誤的與 1 張分類正確的圖片路徑，錯誤的圖片要找出他被誤判的類別，以下為使用 jupyter notebook 的執行結果:



動物種類	被分類正確的圖片	被分類錯誤的圖片	被錯誤分類成的種類
butterfly			Spider 畢竟只是紋身，判斷錯誤很合理
Cat			butterfly 可能是因為貓呈側躺姿勢，會配誤判呈蝴蝶的翅膀
chicken			spider 這隻公雞的外型很明顯，不知道是否受到背景為圓形的關係影響
Cow			sheep 從側面看確實有可能看成羊，這判斷錯也很合理

Dog			sheep 可能是狗的頭部形狀有點類似羊
Elephant			butterfly 因為圖片當中的大象群都擠在一起，不容易做辨識
Horse			sheep 馬和羊外型有點類似
Sheep			elephant 考慮是把羊角誤認為是象牙了

Spider			squirrel 這張特徵頗為明顯，理當不應該判斷錯才對
Squirrel			sheep 可能是有點駝背的角度看起來有點接近羊的外型