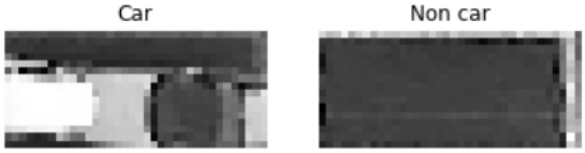
Name: 陳以瑄 StudentID: 109705001

# Task A

## Part 1

- Result:

```
Loading images
The number of training samples loaded: 600
The number of test samples loaded: 600
Show the first and last images of training dataset
```




- Problem I faced and how I solved:

  我原本是用 for foldername in os.listdir(data_path)去讀，但它疑似是隨機的，有時先讀 car 有時先讀 non-car，最後改成直接指定資料夾。

## Part 2

1. Explain the difference between parametric and non-parametric models.

   parametric models 會先假設函數形式，因為它只需要估計一組參數，所以必較容易 fit；non-parametric models 不會對目標函數的形式做太多的假設，雖然需要較多的數據，但可以 fit 任意函數。

2. What is ensemble learning? Please explain the difference between bagging, boosting and stacking.

   Ensemble learning 是將好幾個監督式學習模型，用系統化的方式結合在一起，組合出更強的模型。Bagging 是由多個無關的模型，分別產出各自的輸出值，再用平均或是投票的方式產出最終結果。Boosting 是先有一個模型，再加入另一個可以解決原先預測不好的資料的互補模型，因此它的模型彼此相關。隨著模型一個加一個，最後會產出一個大的模型來做決定。Stacking 也是由多個彼此無關的模型組成，不過它是將這些模型的輸出值統一丟到另一個模型，然後由那個模型產

生最終輸出值。

3. Explain the meaning of the “n_neighbors” parameter in KNeighborsClassifier, “n_estimators” in RandomForestClassifier and AdaBoostClassifier.

   KNeighborsClassifier 的 n_neighbors 指的是要用該資料最接近的多少個鄰居來做決定。RandomForestClassifier 的 n_estimators 是指要結合多少棵決策樹。AdaBoostClassifier 的 n_estimators 是最多可以用多少個 estimator 來估計。

4. Explain the meaning of four numbers in the confusion matrix.

```
Confusion Matrix:
[[287   2]
 [ 13 298]]
```

   以上圖為例，上排表示預測值為真而下排是預測值為假；左邊是實際值為真，右邊則是實際值為假。因此左上角的值 287 為實際值為真且預測值為真(true positive, TP)；右上角的 2 是實際值為假但預測值為真(false positive, FP)；左下角的 13 是實際值為真但預測值為假(true negative, TN)；而右下角的 298 則是實際值跟預測值皆假(false negative, FN)。

5. In addition to “Accuracy”, “Precision” and “Recall” are two common metrics in classification tasks, how to calculate them, and under what circumstances would you use them instead of “Accuracy”.

   Precision 是在所有預測為真的資料中，實際為真的比率是多少。計算方法為 TP/(TP+FP)，上一題的例子 precision = 287/(287+2) = 0.99。Recall 是在所有預測跟實際結果一致的資料中，為真的比率是多少。算法為 TP/(TP+FN)，前一題的例子 recall = 287/(287+298) = 0.49。當資料為 imbalanced 時，會傾向使用 precision 或 recall 而非 accuracy。

Part 3

- Result:

1. Random Forest (n_estimators = 29, random_state = 2)

```
Accuracy: 0.975
Confusion Matrix:
[[287   2]
 [ 13 298]]
```

2. KNN(n_neighbors = 1)

```
Accuracy: 0.8867
Confusion Matrix:
[[297  65]
 [  3 235]]
```

3. Ada Boost (n_estimators = 29)

```
Accuracy: 0.9533
Confusion Matrix:
[[285  13]
 [ 15 287]]
```

Best: Random Forest

- My observation:

1. KNN 跟 Ada Boost 在指定 n_neighbors 或 n_estimators 後，無論何時跑 accuracy 都是一樣的；但是 Random Forest 牽涉到隨機，雖然在同一次連線時固定 random_state 它的 accuracy 會一樣，但是隔一段時間再跑時，accuracy 又會變了。
2. KNN 在預測時，不管 n_neighbors 為多少，它的出錯主要是發生在把 non_car 分成 car，而且 n_neighbors 越大越容易錯。下圖由左至右分別為 n_neighbors = 1, 30, 90。

```
Accuracy: 0.8867
Confusion Matrix:
[[297  65]
 [  3 235]]
```

```
Accuracy: 0.7383
Confusion Matrix:
[[300 157]
 [  0 143]]
```

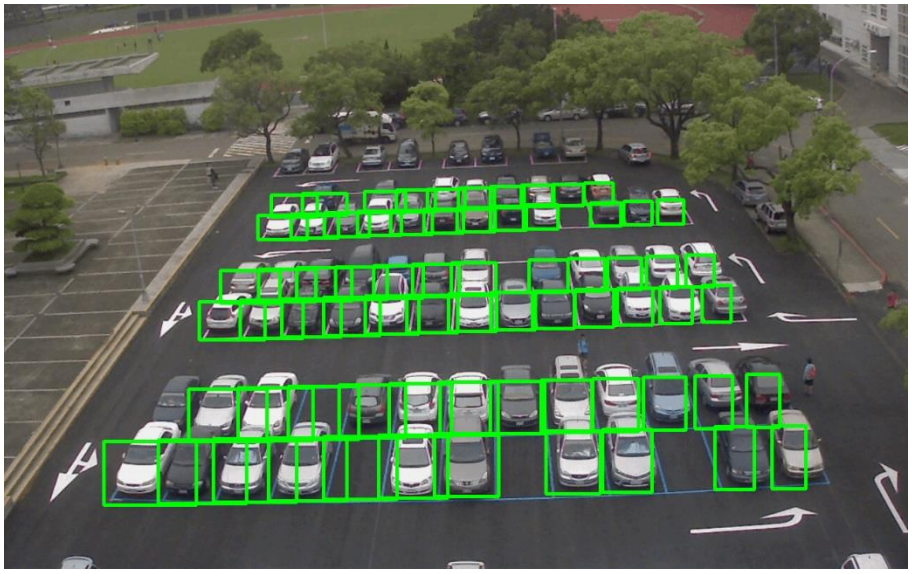
```
Accuracy: 0.645
Confusion Matrix:
[[300 213]
 [  0  87]]
```

3. Ada Boost 從 n_estimators = 1 一直到 30 時，accuracy 進步速度很

快，從一開始的 0.8117 到 0.945，但之後就一直在 0.95 附近。

### Part 4

- Result:



## Task B

### Part 1

- Result:

## Part 2

- Result:
  - Training:

```
Calculate('/content/yolov7/HW1_material/train/','/content/yolov7/runs/detect/train/labels/')
```

```
False Positive Rate: 3/300 (0.010000)
False Negative Rate: 3/300 (0.010000)
Training Accuracy: 594/600 (0.990000)
```

  - Testing:

```
[ ] Calculate('/content/yolov7/HW1_material/test/','/content/yolov7/runs/detect/test/labels/')
```

```
False Positive Rate: 8/300 (0.026667)
False Negative Rate: 5/300 (0.016667)
Training Accuracy: 587/600 (0.978333)
```

- My observation:

  隨著 model 的 epoch 加大，Training 會越來越高。