Please use this report template, and upload it in the PDF format. Reports in other format will result in ZERO point. Reports written in either Chinese or English is acceptable. The length of your report should NOT exceed 8 pages.

Name: 黃宇謙　Dep.:資工碩一　Student ID:R06922072

[Problem1]

1. (5%) Describe your strategies of extracting CNN-based video features, training the model and other implementation details.

InceptionV3:

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         (None, 240, 320, 3)       0
_________________________________________________________________
inception_v3 (Model)         (None, 6, 8, 2048)        21802784
_________________________________________________________________
global_average_pooling2d_1 ( (None, 2048)              0
=================================================================
Total params: 21,802,784
Trainable params: 21,768,352
Non-trainable params: 34,432
_________________________________________________________________
```

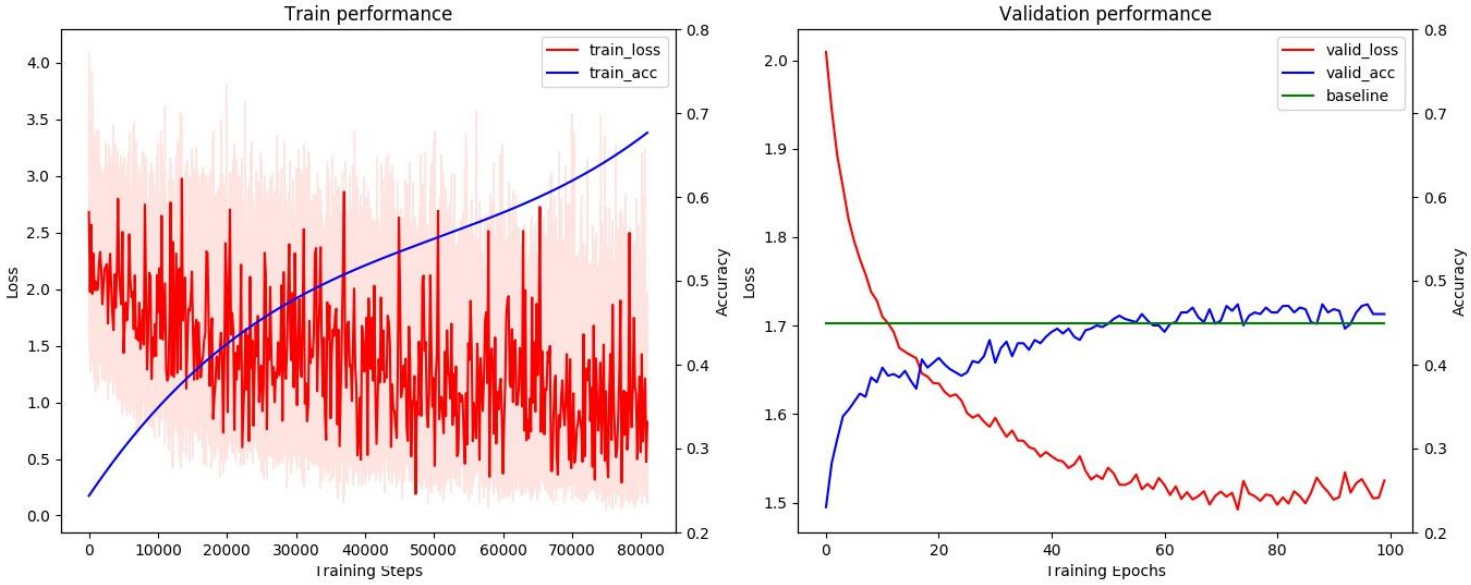
Fully connected layers:

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_3 (InputLayer)         (None, 2048)              0
_________________________________________________________________
fc1 (Dense)                  (None, 512)               1049088
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
fc2 (Dense)                  (None, 512)               262656
_________________________________________________________________
dropout_2 (Dropout)          (None, 512)               0
_________________________________________________________________
predictions (Dense)          (None, 11)                5643
=================================================================
Total params: 1,317,387
Trainable params: 1,317,387
Non-trainable params: 0
_________________________________________________________________
```

在 Problem1 中，我使用 InceptionV3 的 pre-trained model 來提取影像中的 features，在 InceptionV3 的 CNN 架構後面，接上 GlobalAveraePooling，將每個 frame 的輸出大小壓成 2048 維，而在取 feature 策略方面，我使用平均方法，將同一個影片的每一個 frame 通過 InceptionV3 的輸出(N, 2048)，做平均的運算(1, 2048)，最後再將其丟入 fully connected layers 中做 classify。

Fully connected layers 的架構如上圖，總共有三層，前兩層都是 512 個 Kernel，最後接上分類數(11)個 Kernel，並運用 Softmax 下去做預測。

2. (15%) Report your video recognition performance using CNN-based video features and plot the learning curve of your model.



在這部分，我有用 VGG-19 和 InceptionV3 當作提取 features 的 model，接上同樣架構的 fully connected layers，最後結果顯示使用 InceptionV3 做提取 feature 的 model 有比較好的表現，使用 InceptionV3 的準確率可以達到 47%，而使用 VGG-19 則是 36%。而在取 feature 的策略方面，我嘗試 Average 以及 Concatenate 的方案，在使用 Concatenate 方案下，我發現在 fully connected layers 的訓練及準確值表現的都不太好，因此最後決定使用 average 來做取 feature 的策略。

[Problem2]

1. (5%) Describe your RNN models and implementation details for action recognition.

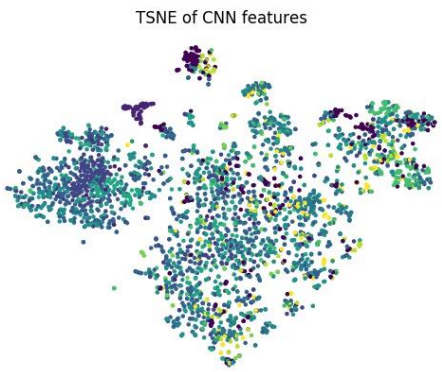
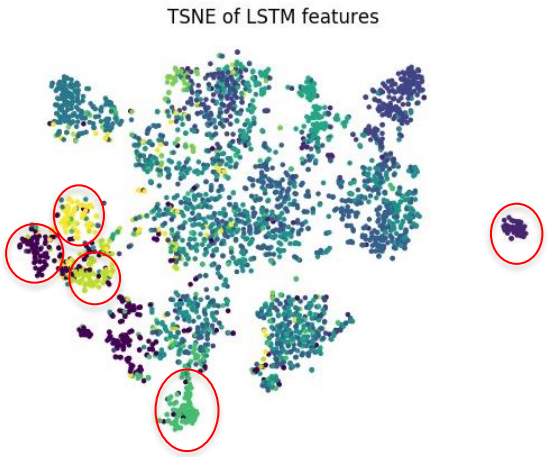
   LSTM:

Layer (type)	Output Shape	Param #
input_3 (InputLayer)	(None, None, 2048)	0
bidirectional_1 (Bidirection	(None, None, 512)	4720640
lstm_2 (LSTM)	(None, 256)	787456
fc1 (Dense)	(None, 128)	32896
dropout_1 (Dropout)	(None, 128)	0
predictions (Dense)	(None, 11)	1419

Total params: 5,542,411
Trainable params: 5,542,411
Non-trainable params: 0

   前半段的 model 我一樣使用到 InceptionV3，而 RNN 的部分則是使用到兩層 LSTM，第一層 LSTM 中，我加上 Bidirectional 的架構，希望 LSTM 能從兩個不同方向的 features 關係取得某一些特性，在 Bidirectional 的 merge 策略中，我使用的是 Concatenate，而第一層 LSTM 的輸出為 Sequence 的形式，並將其丟入下一層的 LSTM，第二層的輸出為 256 維，其輸出會進入 fully connected layers 中做 classify。(action recognition)

2. (15%) Visualize CNN-based video features and RNN-based video features to 2D space (with tSNE). You need to generate two separate graphs and color them with respect to different action labels. Do you see any improvement for action recognition? Please explain your observation.
   在此題中，我使用 Train data 去做 tSNE，選擇 train data 來做 tSNE 主要是因為 data 量比較多。
   可以從下圖觀察到 RNN-based video features 所畫出的影像中，點的分布比較跟 labels 有所關係，除了 other 類別之外，其他相同顏色(同類別)的點基本上都是分布在一起(同區塊)，反觀以 CNN-based video features 所畫出來的影像，則比較沒有這些特性。由此可以推測以 RNN-based 所提取的 features 的特性比較好，這個特性也反應到準確率的表現上，使用 CNN-based 提取 features 的準確率為 47%，而使用 RNN-based 提取 feature 的準確率則是 51%。

CNN	RNN
	

[Problem3]

1. (5%) Describe any extension of your RNN models, training tricks, and post-processing techniques you used for temporal action segmentation.

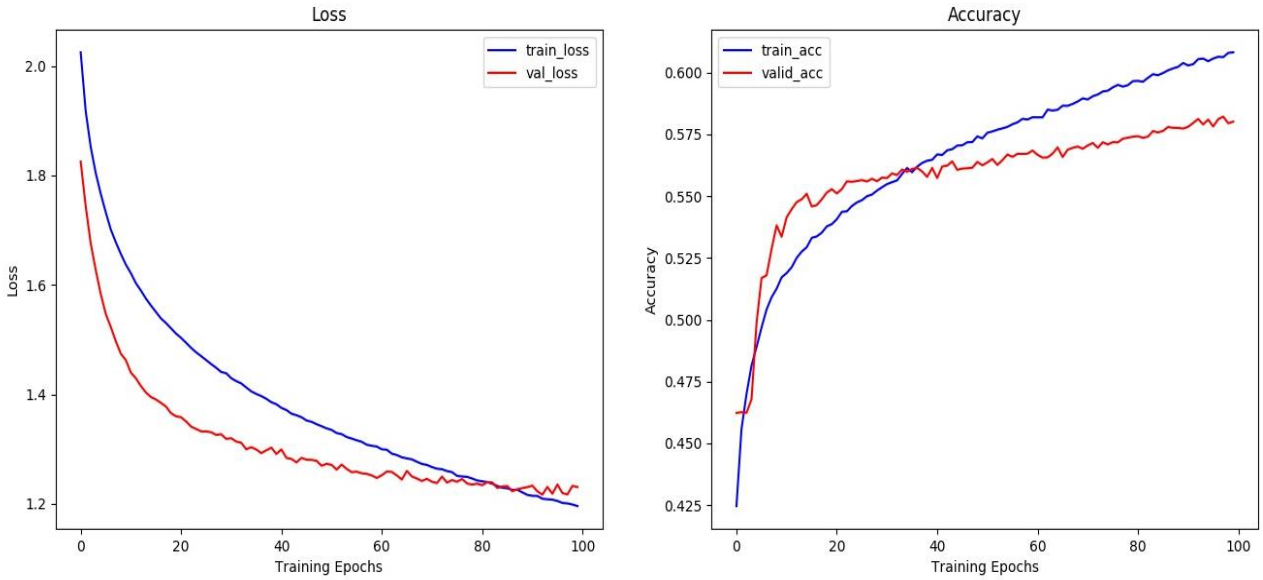
LSTM:

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_3 (InputLayer)         (None, None, 2048)        0
_________________________________________________________________
bidirectional_1 (Bidirection (None, None, 512)         4720640
_________________________________________________________________
lstm_2 (LSTM)                (None, None, 256)         787456
_________________________________________________________________
time_distributed_1 (TimeDist (None, None, 128)         32896
_________________________________________________________________
time_distributed_2 (TimeDist (None, None, 128)         0
_________________________________________________________________
time_distributed_3 (TimeDist (None, None, 11)          1419
=================================================================
Total params: 5,542,411
Trainable params: 5,542,411
Non-trainable params: 0
_________________________________________________________________
```

此題的 RNN 架構跟第二題相同，不同的地方在於 fully connected layers 都加上 TimeDistributed 使其可以針對 LSTM 的每個 output 都做 classify。

在 model 的 input 方面，我使用 10 張影像(5 秒的影片)作為一個 input。前處理方面我則是使用 Sliding window 的形式來取 data，一次位移一張影像。(e.g. 如果有 20 張影像，則 input data 有 20-10+1=11 個)，而它們的 output 則是對應到各自影像的 label。在 predict 方面，我也是用同樣的方式取 input data，由於位移是 1，因此每張影像最多會有 10 個預測結果，接著在從這些結果中，以投票的方式找到最佳的解當成最終的預測結果。

2. (10%) Report validation accuracy and plot the learning curve.



3. (10%) Choose one video from the 5 validation videos to visualize the best prediction result in comparison with the ground-truth scores in your report. Please make your figure clear and explain your visualization results. You need to plot at least 300 continuous frames (2.5 mins).

圖片中上排為 ground truth 對應時間軸的部分，下排為 predict 出來的結果，而在這之中又標出了幾個出現時間比較長的動作(e.g. Take, Open ...)，可以看到 Open, Take, Inspect/Read 都有不錯的 detection 結果。而中間的部分則是將影片中的十一個區段顯示出來。

[BONUS]