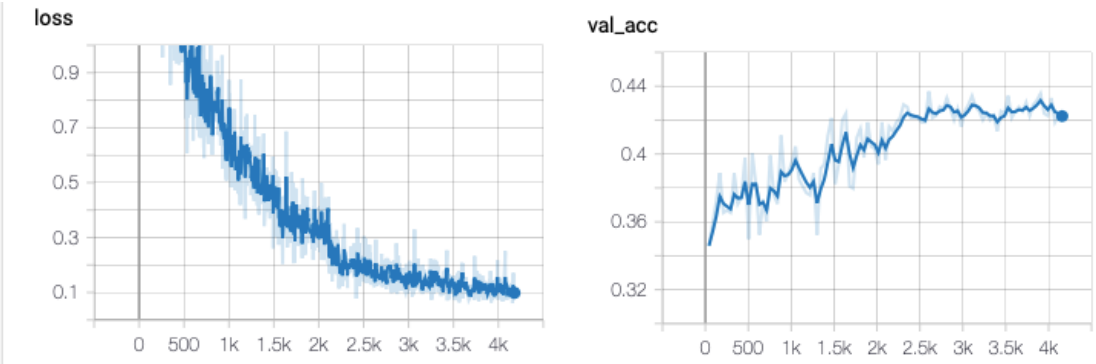
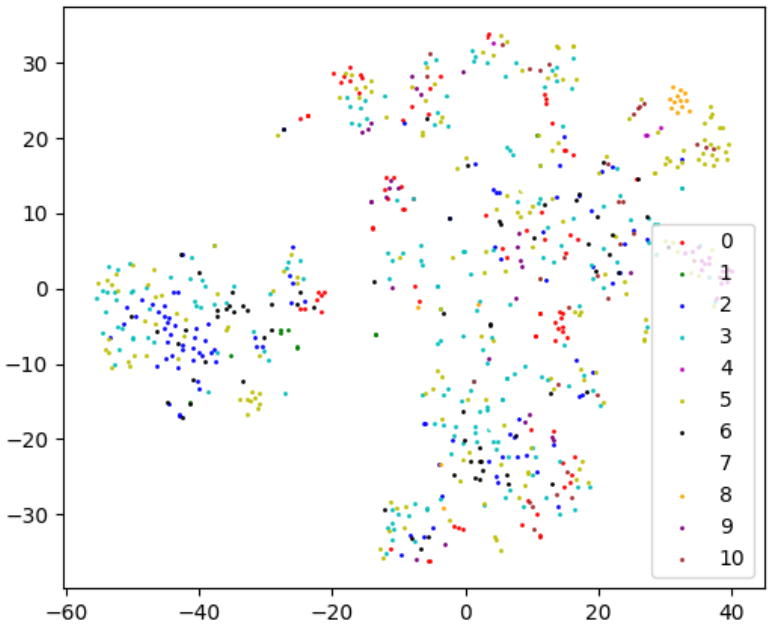
Homework #4

Task 1: Trimmed action recognition w/o RNN (20%)

1. Describe your strategies of extracting CNN-based video features, training the model and other implementation details (which pretrained model) and plot your learning curve. (5%)
   我所使用的 pretrained model 是 ResNet 50。因為助教提供的 reader 是將多個 frame 組在一起，所以我將每一個 frame 個別做 transform 後再合在一起。Transform 的時候有將 image resize 成 224x224。
   因為我沒 fine-tune pretrained model，所以我直接將每一個 video 擷取的 frames 丟進 model，得到相對應的 features(取 mean)，並將這些 features 存成 numpy array，這樣之後在 train 的時候就不用重複抽取 features。



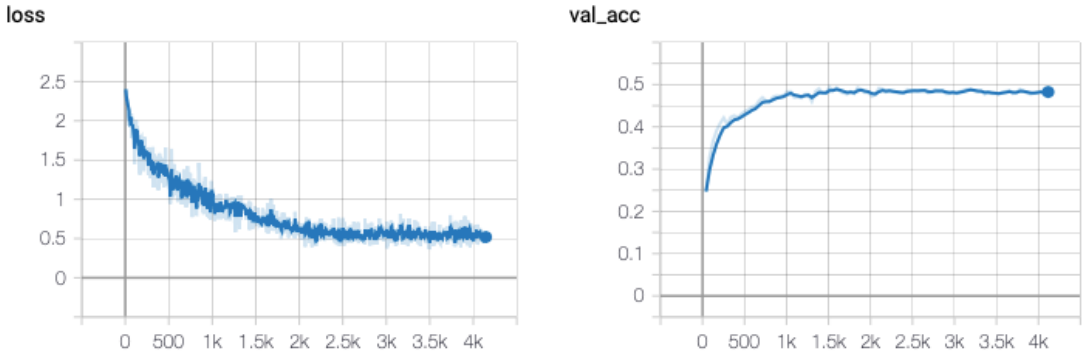
2. Report your video recognition performance (valid) using CNN-based video features and make your code reproduce this result. (5%)
   我的 model 在 validation set 上的 acc 是：0.436
3. Visualize CNN-based video features to 2D space (with t-SNE) in your report. You need to color them with respect to different action labels. (10%)



Task 2: Trimmed action recognition w/ RNN (40%)

1. Describe your RNN models and implementation details for action recognition and plot the learning curve of your model (The loss curve of training set is needed, others are optional). (5%)

   我參考了[1]的 bidirectional LSTM model 架構。Sequence padding 的部分是參考助教提供的[2]。與 task 1 一樣，先將 features 存起來(不過不用取 mean)，避免重複。

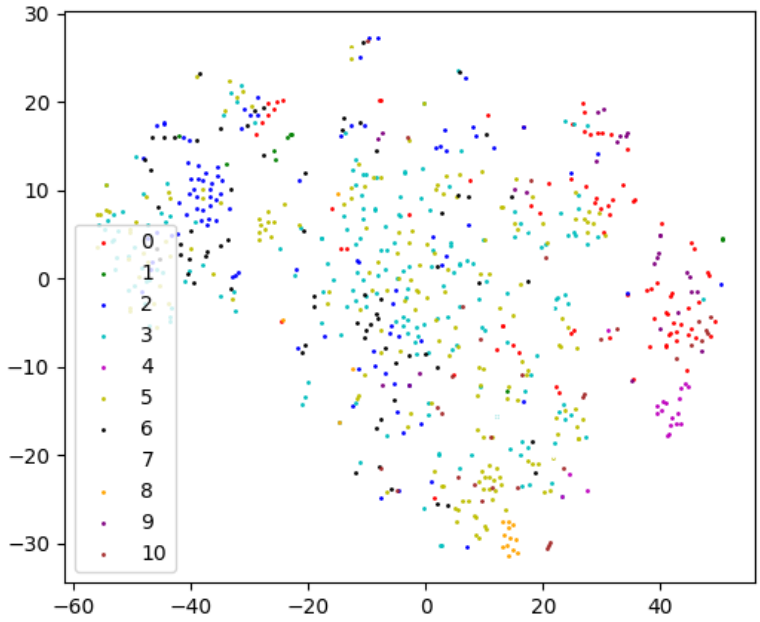


2. Your model should pass the baseline (valid: 0.45 / test: 0.43) validation set (10%) / test set (15%, only TAs have the test set).

   我的 model 在 validation set 上的 acc 是：0.494

3. Visualize RNN-based video features to 2D space (with t-SNE) in your report. You need to color them with respect to different action labels. Do you see any improvement? (10%).

   老實說，單看 task1 與 task2 的 t-SNE，我看不太出有什麼明顯的區別。雖然我畫這兩張圖的時候都是用 validation data，且 task2 在 validation set 的 accuracy 比 task1 高了 5%，感覺應該會比較好，但是畫出來卻沒有。

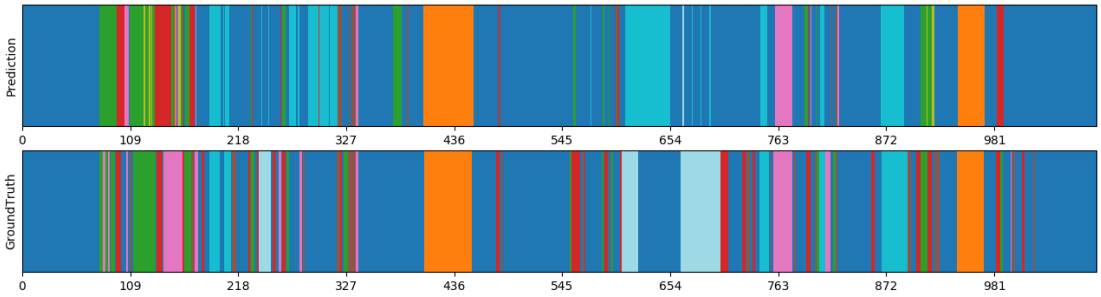


Task 3: Seq-to-Seq prediction in full-length videos (40%)

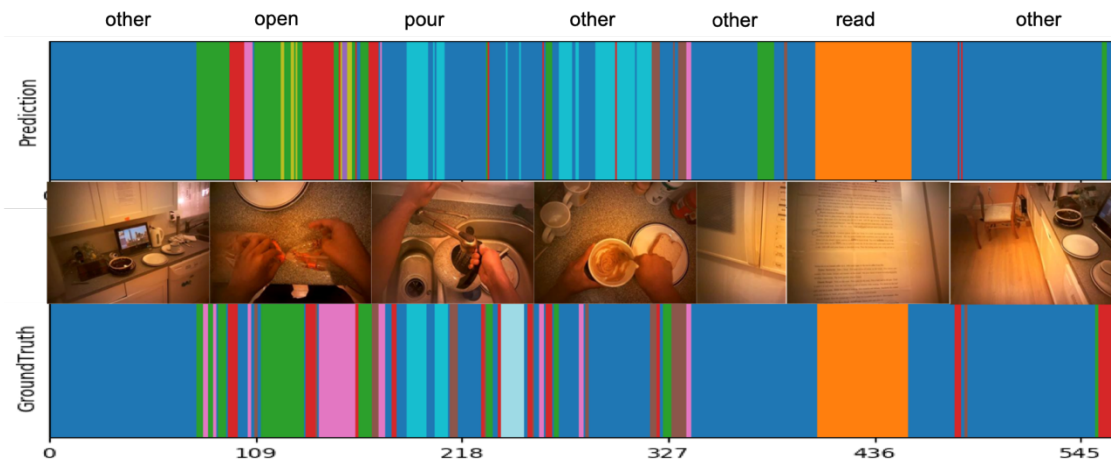
1. Describe any extension of your RNN models, training tricks, and post-processing techniques you used for temporal action segmentation. (5%)
   我 RNN 的 model 架構用跟 task 2 一模一樣，只不過 task2 是將最後一個 hidden cell 的結合(因為是 bi-direction)輸入進 FC layer，而 task3 是將每一個 timestamp 的 output 輸入進 FC layer。且我有將每次輸入進 RNN 的 sequence length 限制在 300 以下。另外，因為 image 是有順序性的，所以我沒有像前面兩題一樣做 shuffle，然後每個 video 是獨立的，不會跟其他 video 放在同一個 batch。
2. Report validation accuracy in your report and make your code reproduce this result. (20%)
   我的 model 在 validation set (不同 category)上的 acc 是：

Category	Accuracy
OP01-R02-TurkeySandwich	0.5355731225296443
OP01-R04-ContinentalBreakfast	0.6028513238289206
OP01-R07-Pizza	0.6195872116552004
OP03-R04-ContinentalBreakfast	0.5466816647919011
OP04-R04-ContinentalBreakfast	0.656221198156682
OP05-R04-ContinentalBreakfast	0.5474683544303798
OP06-R03-BaconAndEggs	0.5686653771760155

   Overall accuracy: 0.5824354646526777
3. Choose one video from the 7 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 500 continuous frames). (15%)
   因為 category 'OP04-R04-ContinentalBreakfast'在 validation set 上的 accuracy 最好，所以我選擇用它來 visual
   下圖是 prediction 與 ground truth 在每個時間上預測結果的比較



這個 category 總共有 1085 張 images，但是因為這樣太小，不容易觀察，所以我只取前面 545 張 images 的 predict v.s. ground truth 來做 visualize:



Reference:

[1] https://github.com/yunjey/pytorch-tutorial/blob/master/tutorials/02-intermediate/bidirectional_recurrent_neural_network/main.py

[2] https://zhuanlan.zhihu.com/p/34418001