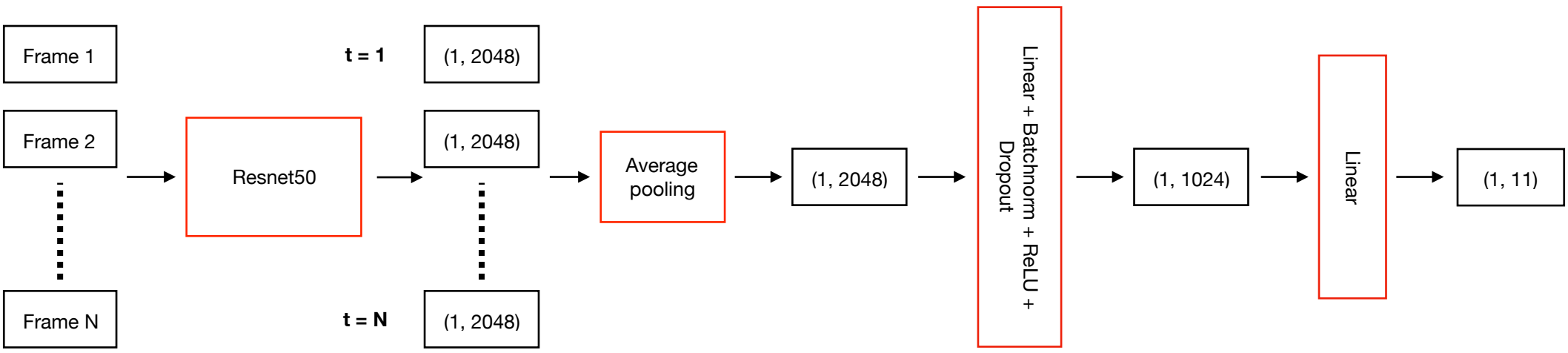
Name：陳紘豪 Dep.：資工碩一 Student ID：R07922134

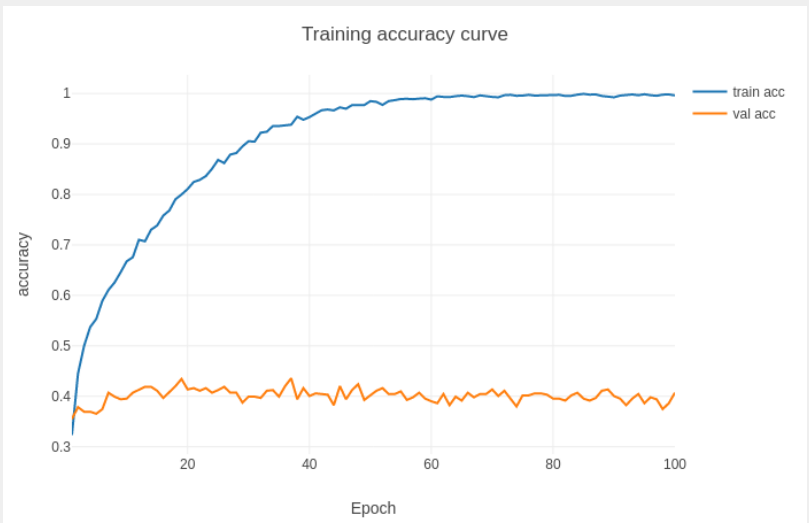
**Problem 1**: Data preprocessing (20%)

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%)

**Model architecture**




Strategies & details for CNN-based video features
- Pre-trained model：Resnet50
- Training 時只 train Resnet50 後面的 fully connected network
- Training preprocessing：
  - 會把每張 frame normalize 到 Imagenet 的 mean, std，其中 mean = [0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]
  - 對每部具有 N 個 frames 的影片，過 Resnet50 後得到 (N, 2048) 的特徵向量，採用 average pooling 的方式得到 (1, 2048) 的特徵向量來代表該部影片
  - 會先將每部影片的特徵向量在前處理時先存起來，training 時直接 load 存起來的特徵向量來 train 後面用來分類的 fully connected network，此做法可以不用每次都過 Resnet50，大大減少了 training 所花的時間
- Adam optimizer lr = 1e-4, weight decay = 1e-5
- Batch size = 64
- Epoch = 100

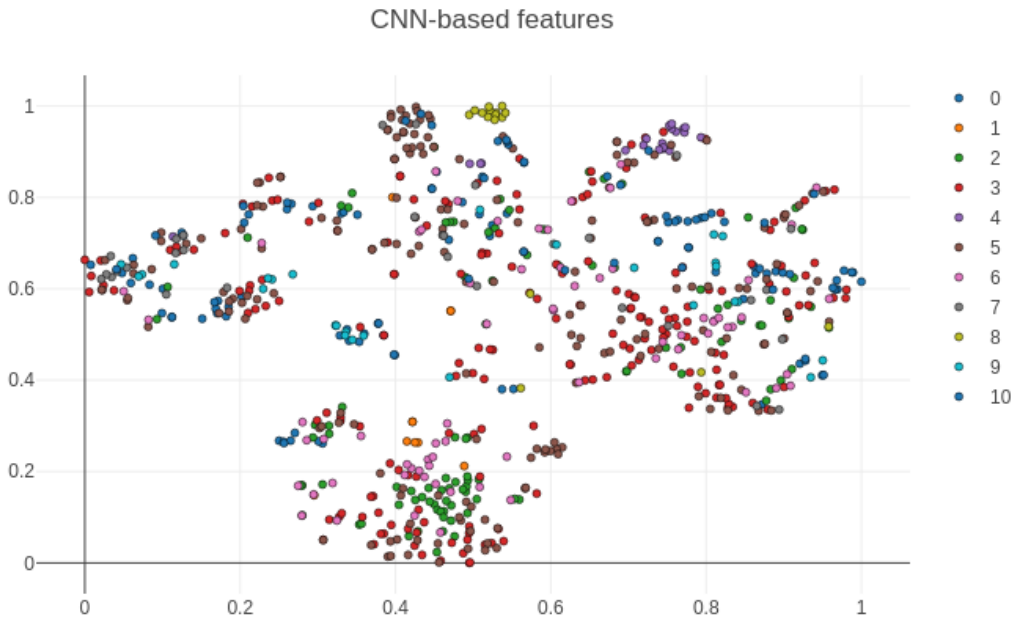
| Loss | Accuracy |
| --- | --- |
|  |  |



2. Report your video recognition performance (valid) using CNN-based video features. (5%)

Validation accuracy：0.435630689206762 = 43.56%

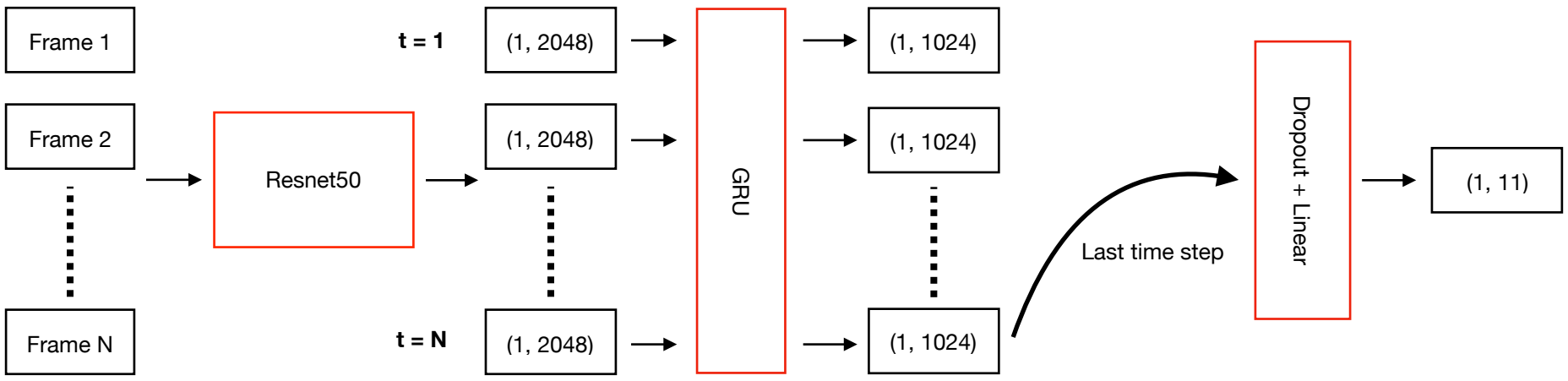
3. Visualize CNN-based video features to 2D space (with tSNE) in your report. You need to color them with respect to different action labels. (10%)



**Problem 2**: Trimmed action recognition (20%)

1. Describe your RNN models and implementation details for action recognition and plot the learning curve of your model. (5%)

**Model architecture**




Strategies & details for RNN-based video features

- Pre-trained model：Resnet50
- Training 時只 train Resnet50 後面的 GRU 和 fully connected network
- Training preprocessing：
  - 會把每張 frame normalize 到 Imagenet 的 mean, std，其中 mean = [0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]
  - 有做 data augmentation，對影片的 frame 做水平翻轉
  - 會先將每部影片的特徵向量在前處理時先存起來，training 時直接 load 存起來的特徵向量來 train 後面的 GRU 和 fully connected network，此做法可以不用每次都過 Resnet50，大大減少了 training 所花的時間
- Adam optimizer lr = 1e-4, weight decay = 1e-5
- Batch size = 64
- Epoch = 100
- 有試過把每部影片過 GRU 後的 outputs 給做 average pooling 丟給 fully connected 做分類，但效果沒有只取最後一個 time step 來的好

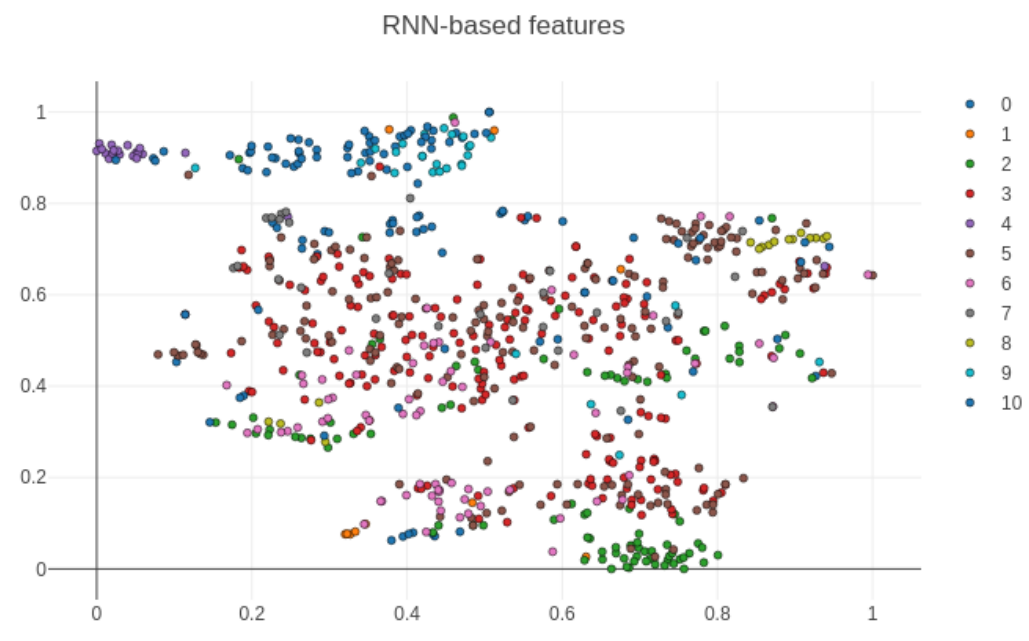
| Loss | Accuracy |
| --- | --- |
|  |  |



2. Your model should pass the baseline (valid: 0.45 / test: 0.43) (5%) validation set (10%) / test set (15%, only TAs have the test set)

Validation accuracy：0.49934980494148246 = 49.93%

3. Visualize RNN-based video features to 2D space (with tSNE) in your report. You need to color them with respect to different action labels. Do you see any improvement for action recognition compared to CNN-based video features? Why? Please explain your observation (10%)



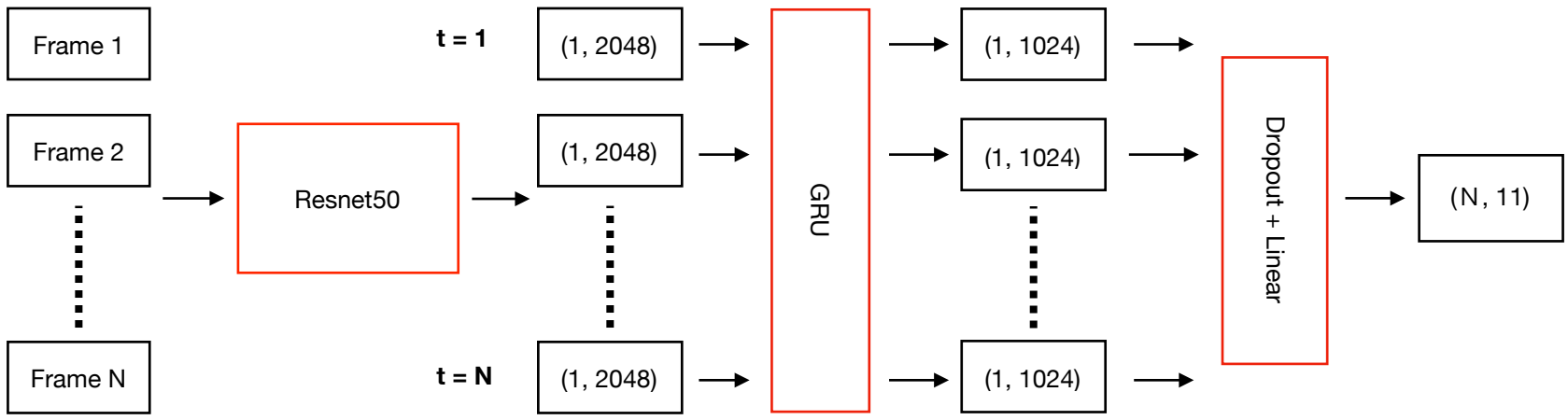

很明顯看得出來使用 RNN 來從影片取 features，有達到讓相同 類別的影片在 feature space 上群聚的效果，若單純只使用 CNN 來取 features，只有少數類別彼此會在 feature space 上距離較近（例如 8: Divide/ Pull Apart），其餘的類別都是彼此在 feature space 上混成一團，看不出兩兩類別之間的分別。

就結果來看 RNN-based features 的 validation accuracy 的確也比 CNN-based features 的 validation accuracy 來得高，也難怪 t-SNE 視覺化後的結果也比較好，但真的要探究原因的話，我推測可能是因為輸入是「影片」的關係，RNN 比較能處理這種具有時間資訊的輸入，比起單純使用 CNN 然後做 average pooling 取特徵，RNN 可以讓 model 自己觀察不同時間的輸入最終學出該影片的特徵，因此具有較好的 accuracy。

**Problem 3**: Temporal action segmentation (40%)

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

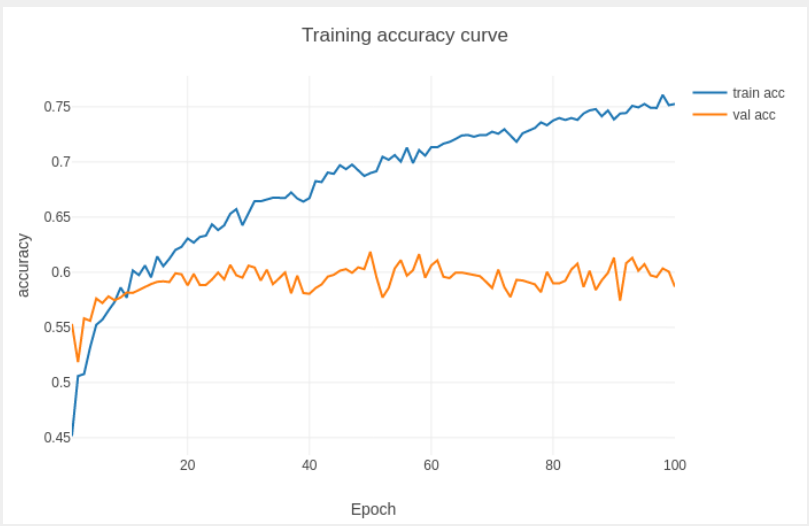
**Model architecture**




Strategies & details

Pre-trained model：Resnet50

- Training 時只 train Resnet50 後面的 GRU 和 fully connected network
- Training preprocessing：
  - 會把每張 frame normalize 到 Imagenet 的 mean, std，其中 mean = [0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]
  - 會先將每部影片的特徵向量在前處理時先存起來，training 時直接 load 存起來的特徵向量來 train 後面的 GRU 和 fully connected network，此做法可以不用每次都過 Resnet50，大大減少了 training 所花的時間
- Adam optimizer lr = 1e-4, weight decay = 1e-5
- Batch size = 3
- Epoch = 100
- 在這題的架構大致上和第二題相同，只是 forward path 有些稍許不同，這邊會把每個 time step 的 output 都丟到後面的 fully connected network 去做分類，以產生每個 frame 的預測
- 對於每部影片，是每次隨機選取 512 張 frame （但會是 frame 彼此之間會保持原有的時間前後關係）進來訓練，所以 GRU model 的 input 會變成（3, 512, 2048），因為是隨機的關係所以使得每個 epoch 讀的 frame 也不一樣，變相的增加了 training data 的數量

| Loss | Accuracy |
| --- | --- |
|  |  |

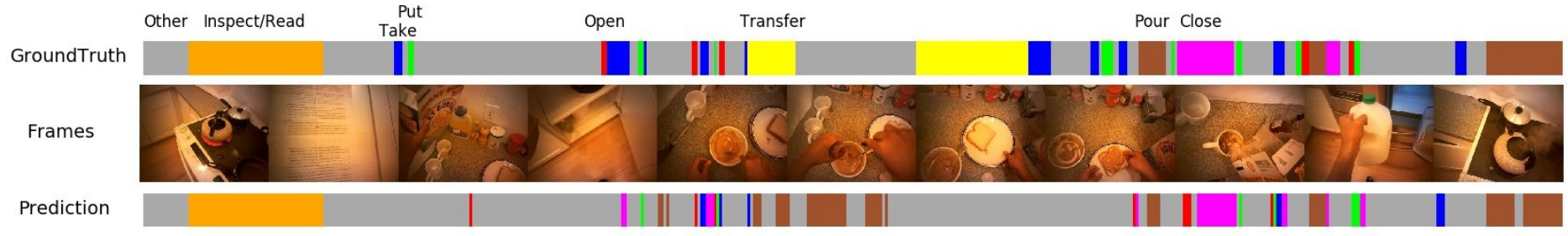


2. Report validation accuracy in your report. (20%)

| | | |
| --- | --- | --- |
| OP01-R02-TurkeySandwich | 0.539525691699605 = 53.95% | 546 / 1012 |
| OP01-R04-ContinentalBreakfast | 0.621181262729124 = 62.12% | 610 / 982 |
| OP01-R07-Pizza | 0.636179684338325 = 63.62% | 1572 / 2471 |
| OP03-R04-ContinentalBreakfast | 0.58830146231721 = 58.83% | 523 / 889 |
| OP04-R04-ContinentalBreakfast | 0.656221198156682 = 65.62% | 712 / 1085 |
| OP05-R04-ContinentalBreakfast | 0.55168776371308 = 55.17% | 523 / 948 |
| OP06-R03-BaconAndEggs | 0.671824629271438 = 67.18% | 1042 / 1551 |
| Total | 0.618482882076527 = **61.85%** | 5528 / 8939 |

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%)

選擇「OP04-R04-ContinentalBreakfast」的 frame 390 到 890 來視覺化，結果如下




Ground-truth 中不同顏色對應的 label 我有把它標在該顏色第一次出現的地方，由此可見預測最準確的是 Inspect/Read 這個動作，可以幾乎 100% 預測正確，再來是 Pour 的正確率也頗高，但是 Transfer 這個動作卻完全預測不出來，我猜可能是 Transfer 這個動作本身就很抽象，不像 Inspect/Read 或 Pour 有個明確的動作特徵。

再來就是 Other 這個類別很常被預測出來，可能是因為本身 Ground-truth 中就有很多的 Other 所以被學的比較多，以及在 training 時 random sample 512 個 frame 太常出現都是一堆 0 的狀況了，所以學到了一堆 Other 的特徵而沒學到其他的動作。