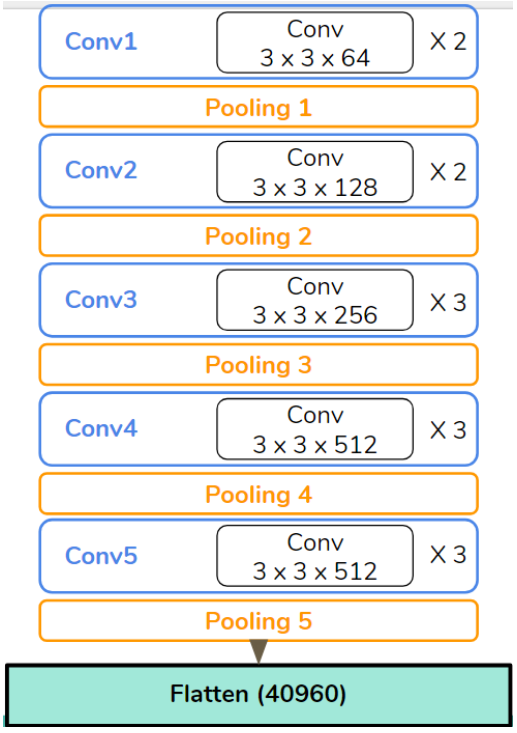
Name: 張鈞閔　Dep.:電機博二　Student ID:D05921027

[Problem1]

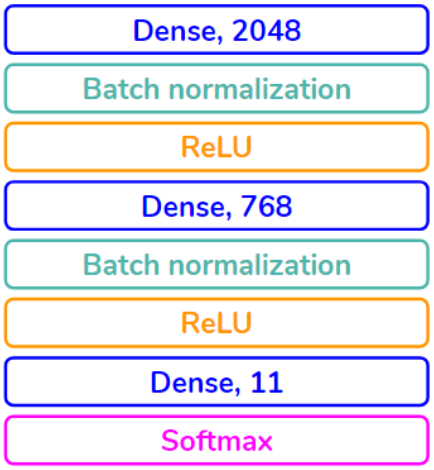
1. (5%) Describe your strategies of extracting CNN-based video features, training the model and other implementation details.

A. I leveraged the *readShortVideo* function in *reader.py* provided by TAs to read videos using downsample=12 and without resizing. Thus, the size of a frame is 240 by 320 pixels. Extract frame-level features by the VGG-16 model pre-trained on ImageNet.



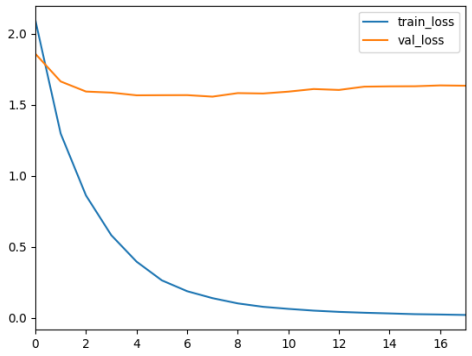
B. For a video, I took the **average** of its frame-level features as the video-level features.

C. My DNN architecture was designed as follow. Training details: batch size (32), learning rate (3e-5), early stop patience (5), epoch (30), Adam optimizer (beta1=0.5)

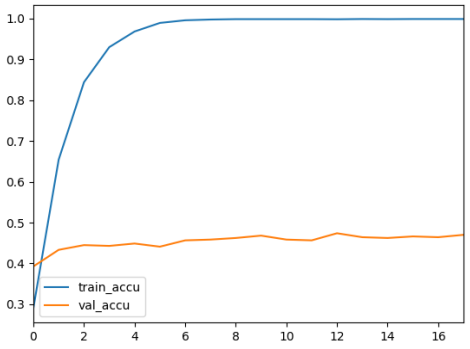


2. (15%) Report your video recognition performance using CNN-based video features and plot the learning curve of your model.

CNN-based video features can achieve accuracy of **0.478** over the validation set. The learning curves were shown below.



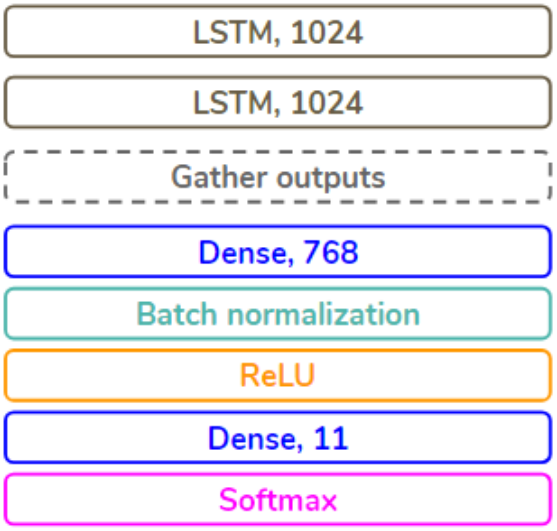
x-axis: number of epoch
y-axis: cross-entropy loss (**train**, **val**)



x-axis: number of epoch
y-axis: accuracy (**train**, **val**)

[Problem2]

1. (5%) Describe your RNN models and implementation details for action recognition.
   A. My RNN architecture was designed as follow. I stacked two *tf.contrib.rnn.BasicLSTMCell* layers by *tf.contrib.rnn.MultiRNNCell*, and used *tf.nn.dynamic_rnn* to tackle the dynamic context of videos.



   B. The "Gather outputs" step implied that if a video length is *k*, I will extract the RNN's output at the *k*-th timestamp as its RNN features. By doing so, we can

avoid the information diluted by the padding zeros. Each video extract its own outputs at the corresponding timestamp.

C. The maximum video length was set at 25. If a video's length less than 25, I pad zero vectors until its time length reaches 25; If a video's length greater than 25, I uniformly pick 25 frames out.

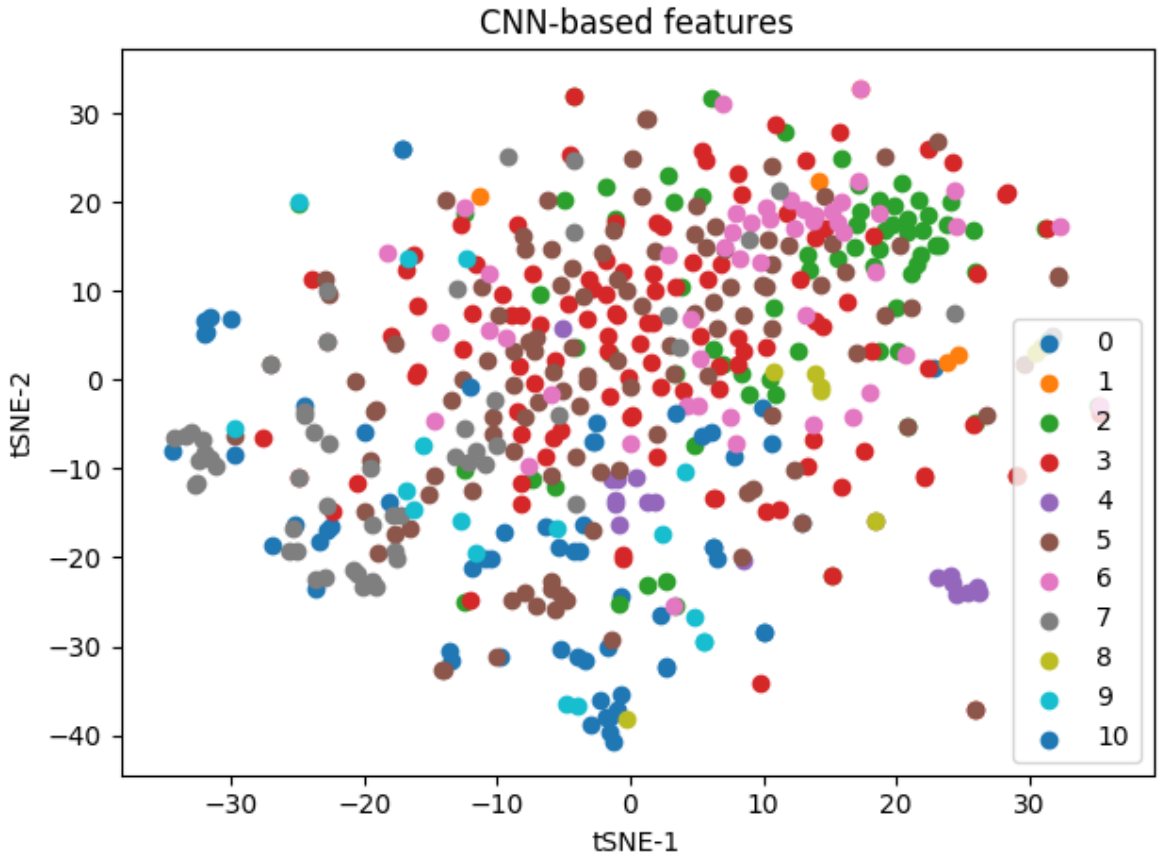
2. (15%) Visualize CNN-based video features and RNN-based video features to 2D space (with tSNE). You need to generate two separate graphs and color them with respect to different action labels. Do you see any improvement for action recognition? Please explain your observation.

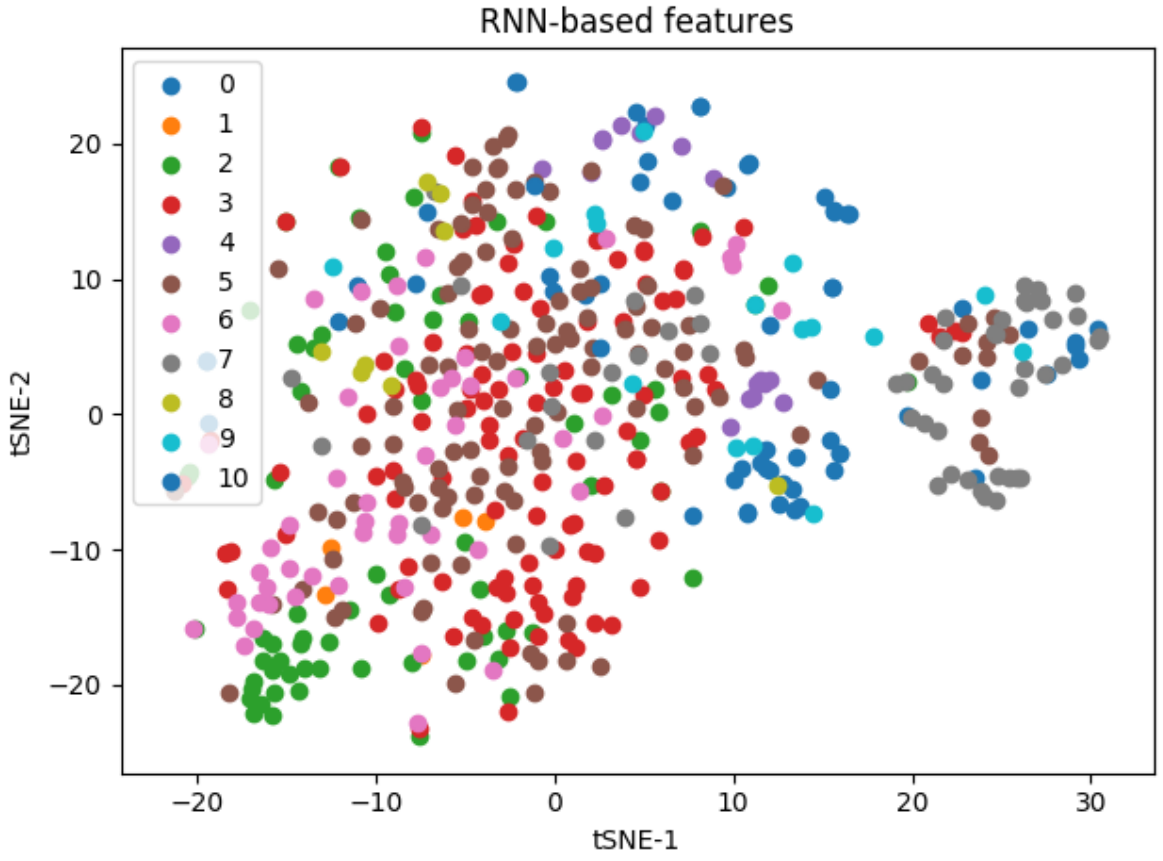
   A. RNN-based model can achieve accuracy up to **0.489**, slightly better than DNN-based model (0.478). RNN slightly improved the results and I made several observations from the t-SNE visualization results to explain it.

   B. Class 3, 5, and 6 were usually mingled together. And, RNN-based features seem more concentrated than CNN-based features.

   C. Class 7, 9, and 10 were more closer to each other in comparison with other classes. Also, for class 7, the t-SNE result using RNN-based features had a conspicuous cluster on the right-hand side but cannot observe this pattern while using CNN-based features.

   D. Class 1 and 8 had less samples and thus more difficult to distinguish the interaction with other classes from the t-SNE, for both features.





[Problem3]

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

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).

[BONUS]