311551180_楊佳誠

Requirements

1. Please go through the referenced literature[2] carefully and complete the "#TODO" parts. For SCCNet implementation, you have to use the predefined arguments to construct each layer. After finishing the code, train 3 models with 4 training schemes and test on **subject 1** **(BCIC_S01_E.mat)**.

Show testing accuracies and plot confusion matrices for your 12 models.

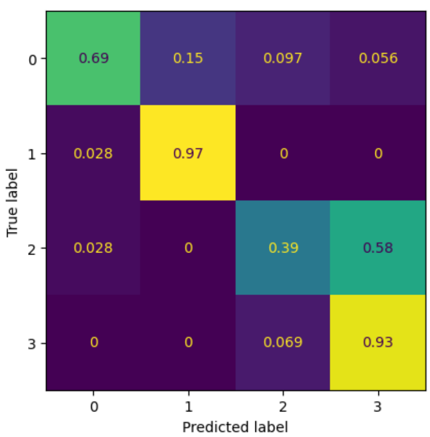
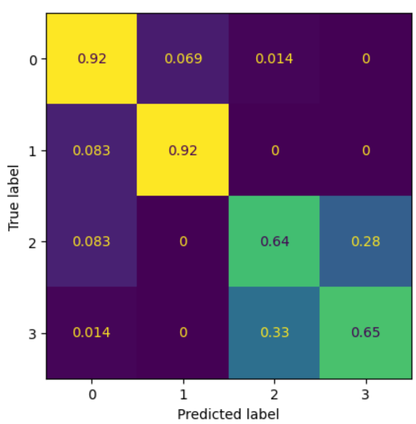
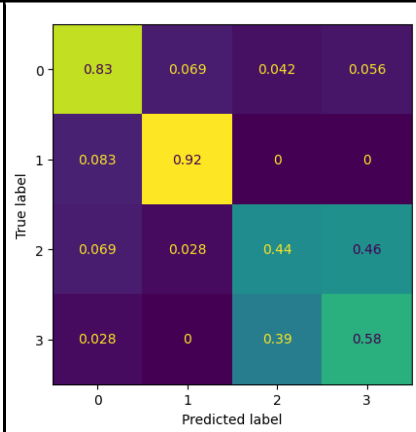
Testing accuracies on Subject 1

SCCNet outperform other models.

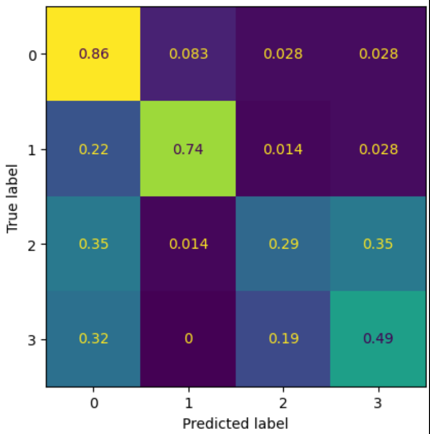
	EEGNet	SCCNet	ShallowConvNet
Ind	0.7465277777777778	0.78125	0.6944444444444444
SI	0.625	0.7013888888888888	0.6076388888888888
SD	0.75	0.7881944444444444	0.7361111111111112
SI+FT	0.6215277777777778	0.8402777777777778	0.6284722222222222

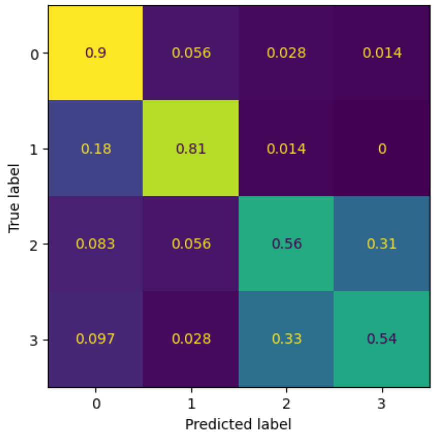
Confusion matrices

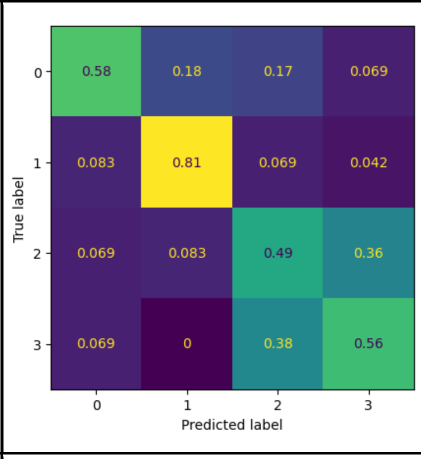
Observed from the confusion matrix, the model has difficulty to classifying between class 3(both feet) and class 4(tongue). Maybe some designs can be added to make the model more capable to classify the differences of two classes.

<table>
<tr><td></td><td>EEGNet</td><td>SCCNet</td><td>ShallowConvNet</td></tr>
<tr><td>Ind</td><td>
</td><td>
</td><td>
</td></tr>
</table>

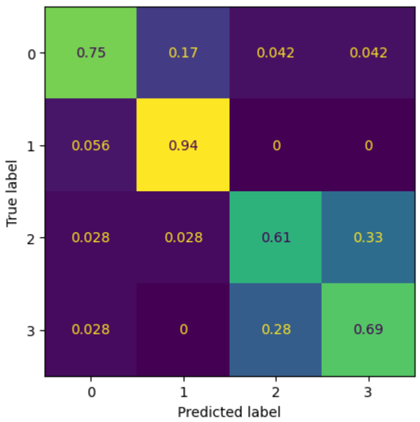
SI

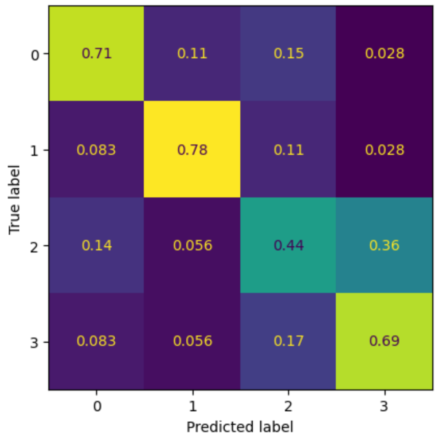


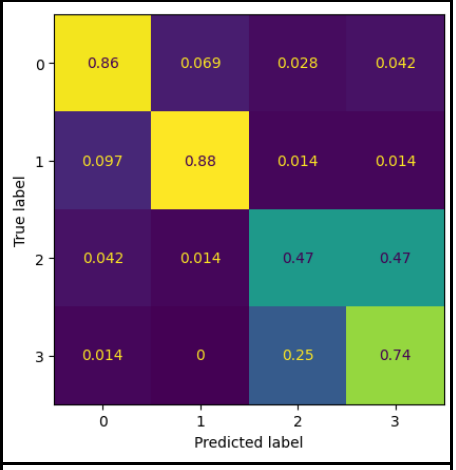




SD

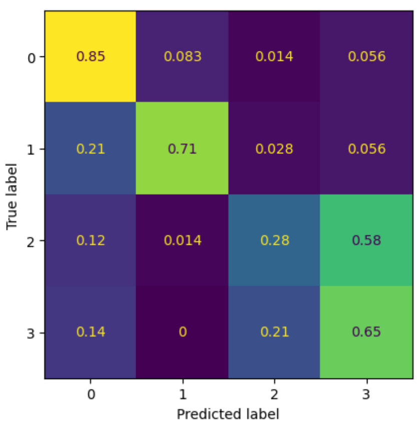


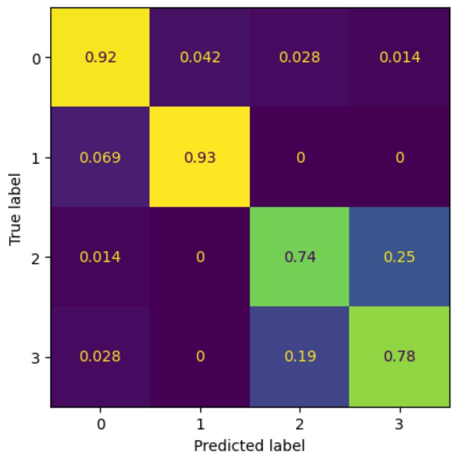


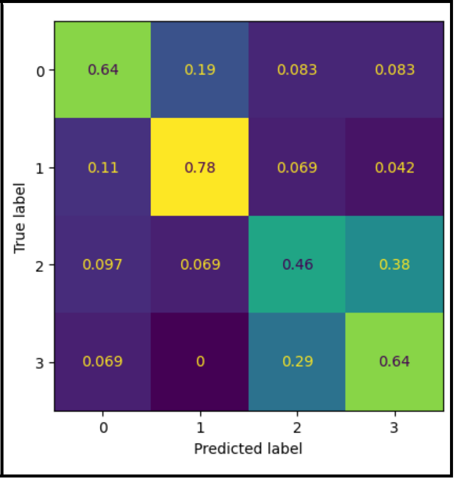


SI+

FT







Kaggle accuracy

	EEGNet	SCCNet (or SCCNet_v2)	ShallowConvNet
SI	0.57777	**0.67361**	0.57777
SD	0.61944	0.65972	0.57361

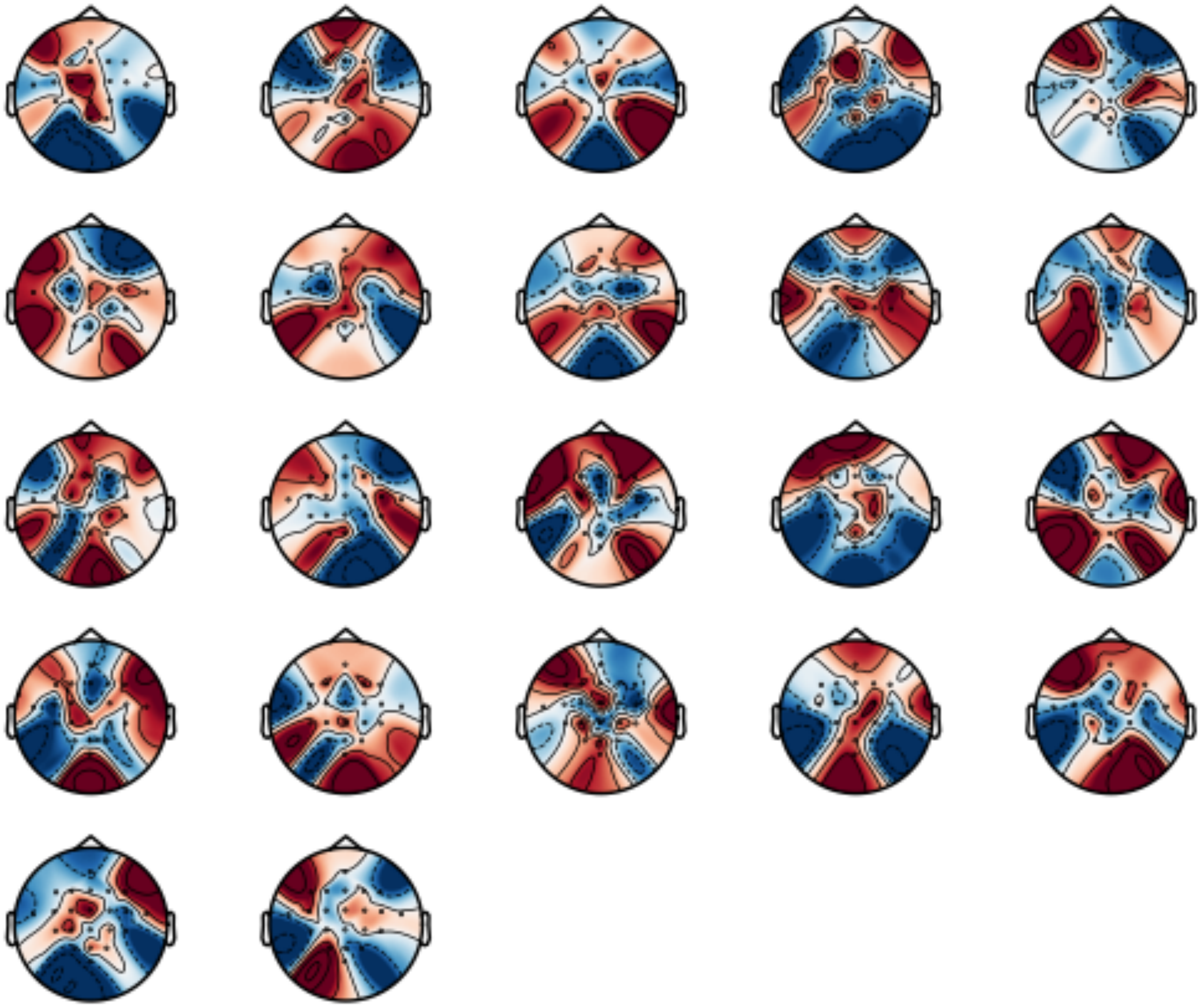
2. Fill in your hyper-parameter settings (Batch size, learning rate, epochs, optimizer, etc).

   All hyper-parameter settings are same and are shown below.

- Batch size:16
- Learning rate:1e-4
- Epochs:750, while if “sift”, Epochs = 850
- Optimizer:Adam
- Scheduler: StepLR(opt_fn, step_size=250, gamma=0.5)
- Others: **Using Repeated K-Fold, K=5, repeat 3 times**

3. Obtain spatial kernel weights from the first convolutional layer of your SCCNet model trained with Ind scheme on subject 1, visualize the weights as topographic maps using the MNE package.

   In this figure, we can visualize what the models learn from our input data.



Discussion

1. Pros and cons of the 3 CNN models and 4 training schemes

Models

a. EEGNet

- The usage of depthwise convolution can capturing spatial information:
  EEG signals are inherently spatial in nature, and depthwise convolution has been shown to

capture spatial information better than traditional convolution. By applying separate filters to each input channel, depthwise convolution captures the unique spatial relationships between the input channels.

- Importantly, when used in EEG-specific applications, this operation provides a direct way to learn spatial filters for each temporal filter, thus enabling the efficient extraction of frequency-specific spatial filters (see the middle column of Figure 1). (Adapted from：EEGNet: A Compact Convolutional Neural Network for EEG-based Brain-Computer Interfaces)
- Less parameters are used compare to normal convnet and the training time is shortened.
- The lack of consideration of spatial-temporal filtering.

b. SCCNet

- The SCCNet has demonstrated superiority in motor imagery EEG classification, outperforming other existing convolutional neural networks. The results of my experiment also support this finding.
- The innovative idea of changing the order of different kind of filter.

c. ShallowConvNet

- Even if its architecture is shallower than Deep ConvNet, Shallow ConvNet performs comparably to Deep ConvNet.
- The use of a larger kernel size(compare to Deep ConvNet) in the temporal convolution of the shallow ConvNet allows a larger range of transformations, which can improve classification accuracy.
- The use of a larger kernel size in the temporal convolution may also result in a loss of some temporal resolution, which could potentially affect classification accuracy.
- The lack of consideration of spatial-temporal filtering.

Training schemes

a. Ind scheme can be easily trained, but the resulting model may not generalize well to different subject, which could result in a significantly decrease in performance.
b. SI scheme exclude the subject's data, which means that the model may not be able to learn the specific information from that excluded subject. As a result, the performance of SI may be worse than the SI+FT scheme, because of not using the subject's train data.
c. In my opinion, the SD scheme contains the most data, which may result in better performance compared to other subjects. However, for a specific suject, the performance may not outperforn SI+FT.

2. Your observations regarding the models (structures, size .etc)

- Trainable parameter： ShallowConvNet>SCCNet>EEGNet
- Model Depth：EEGNet> SCCNet>ShallowConvNet

3. Difficulties you encountered in this homework.

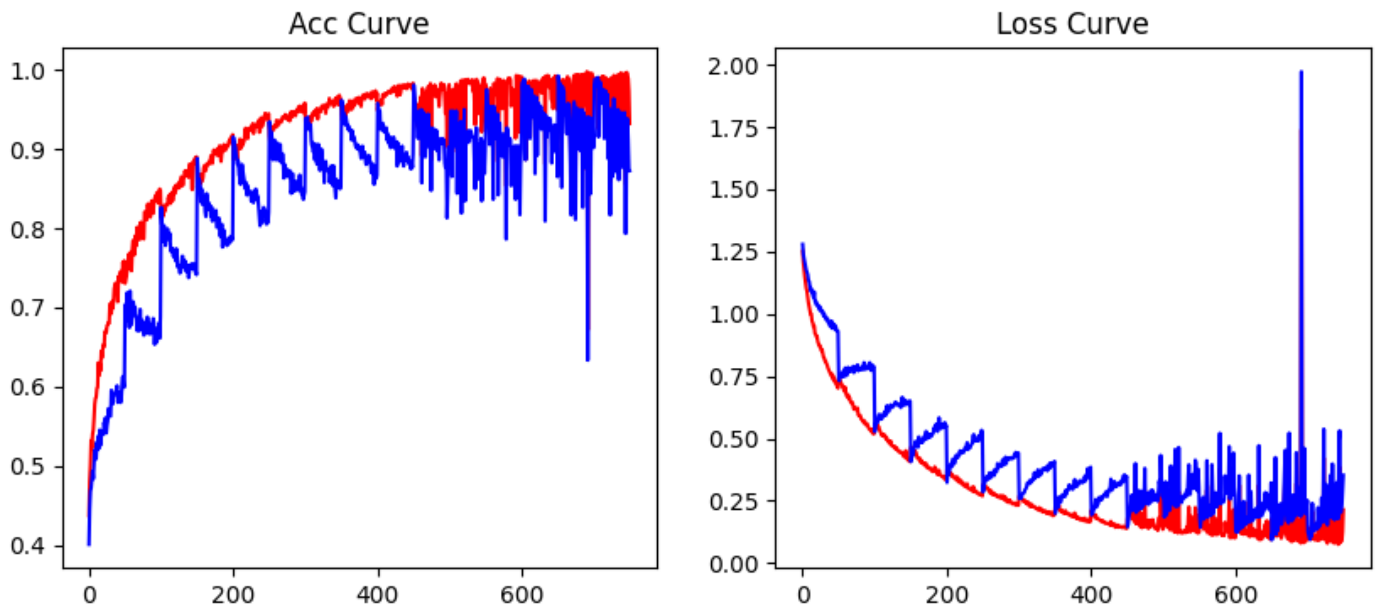
- There are many commonly used classes in mne python, such as in plot_topomap, a info object should be create and montage need to be set.
- It takes lots of effort to merge 22 topographic maps into one image. It turns out to be the "show" parameter of plot_topomap should be set to False.
- It is challenging to implement a model by reading paper.

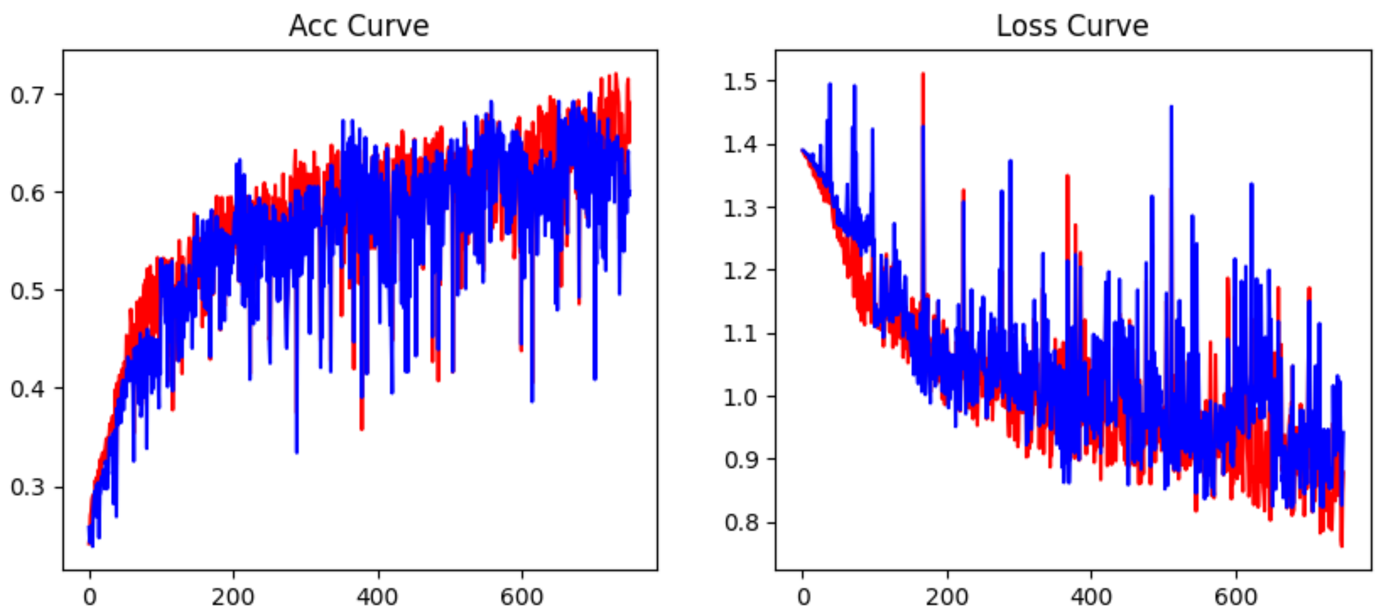
4. For models trained with different subjects, what are the possible reasons for the difference in model performance.

- Maybe some subjects are active, they created more noise to the raw data.

4. Other topics you find worthy to discuss.

- The amplitude of loss curve is smoother in SCCNet than other Nets.

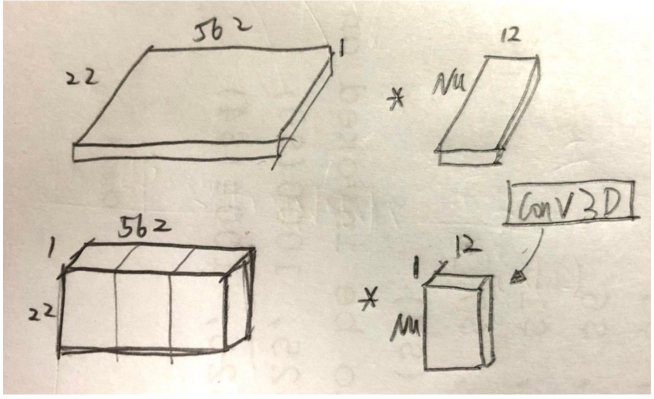




Bonus Challenge

SCCNet_v2

The SSCNet_v1's conv2D with permute is equals to SCCNet_v2's conv3D with some squeeze/unsqueeze operations in order to match the input of nn.Conv3D.



Kaggle's <u>second</u> best performance

Accuracy	model	epoch	Batch size	lr	optimizer	Scheduler	others
0.66527	SCCNet	500	16	1e-4	Adam	x	K_fold * 2

Algorithm

K_fold * N :

1. concat train set and valid set.
2. spilt dataset to 5 groups, choose one as the valid set to train an epoch, until all fold are used as valid set.
3. repeat step 2. (N-1) times.