BCI HW3 Report

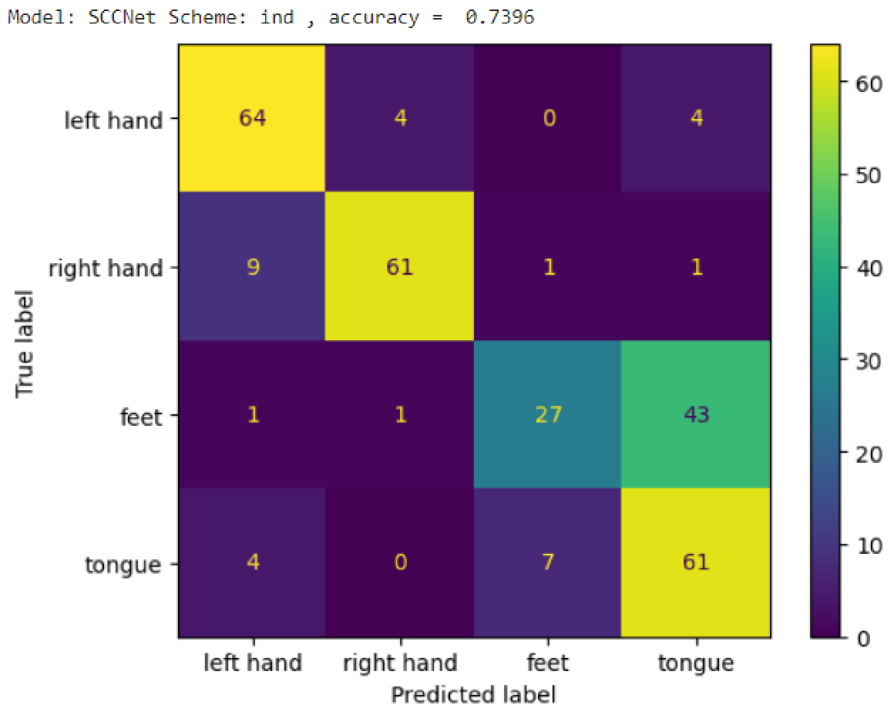
0816077 陳柏安

Hyper Parameter Setting: Default for all models

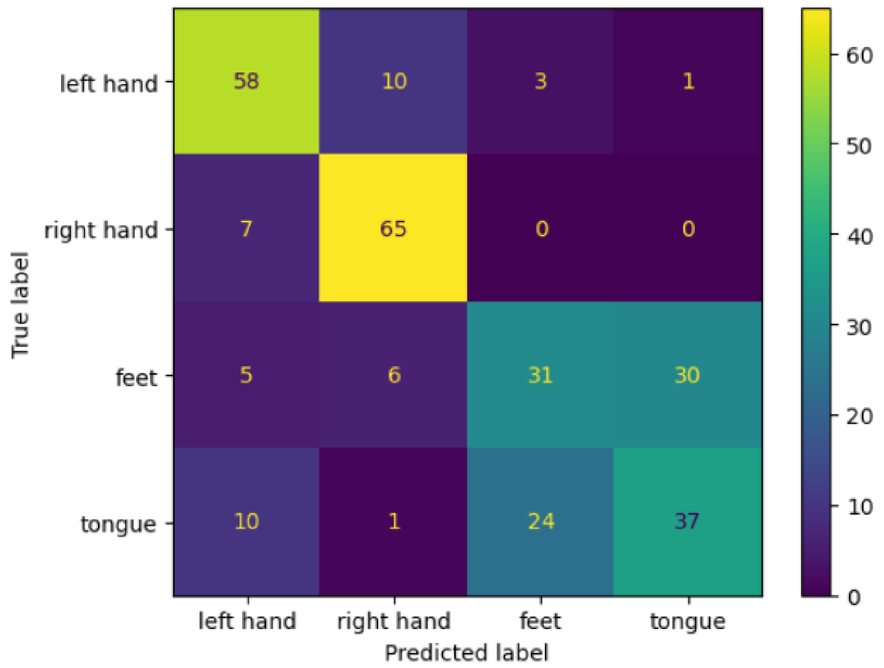
```
epochs = 200
batch_size = 16
lr = 1e-3
```

```
opt_fn = torch.optim.Adam(model.parameters(), lr=lr)
```

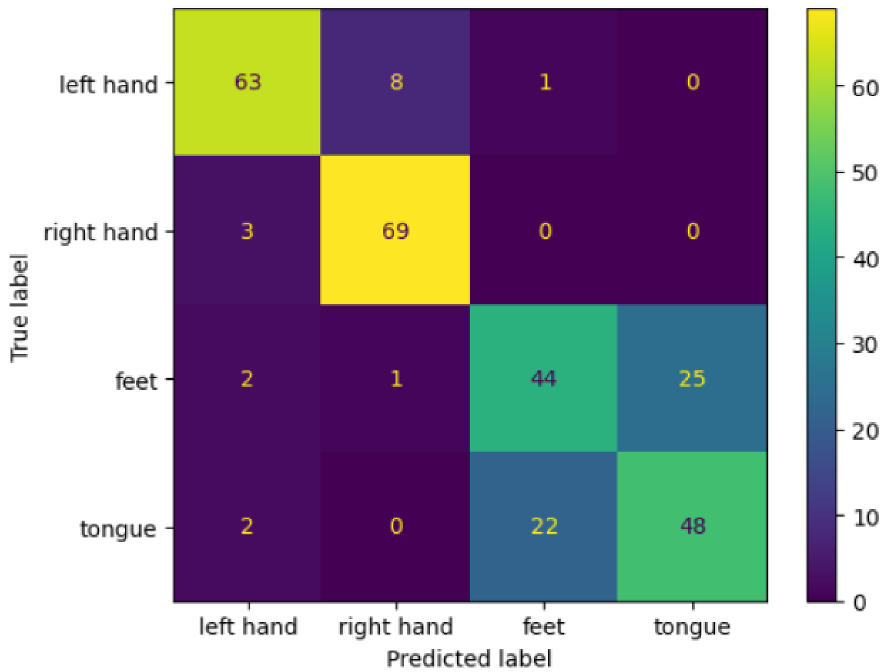
Show testing accuracies and plot confusion matrices for your 12 models.



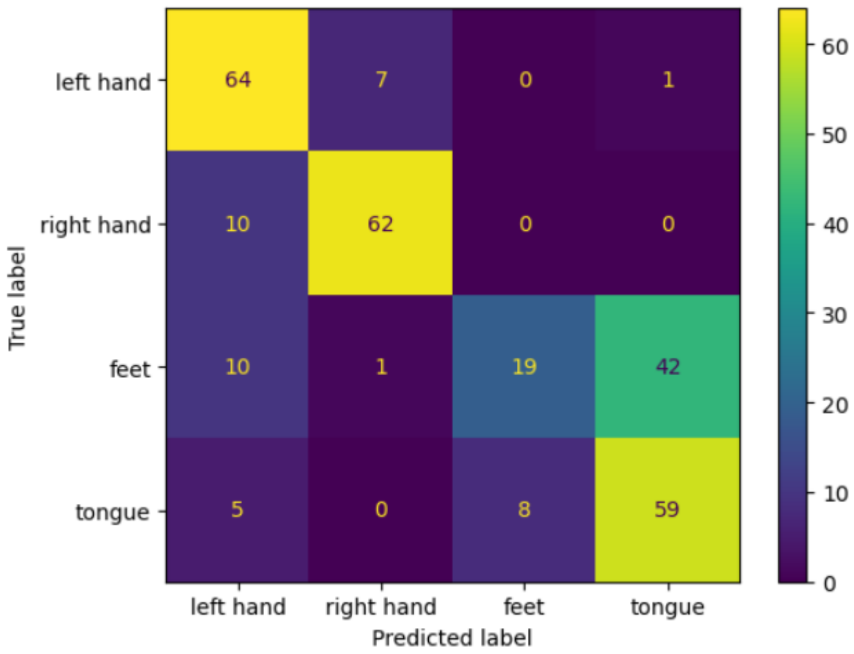
Model: SCCNet Scheme: si , accuracy = 0.6632



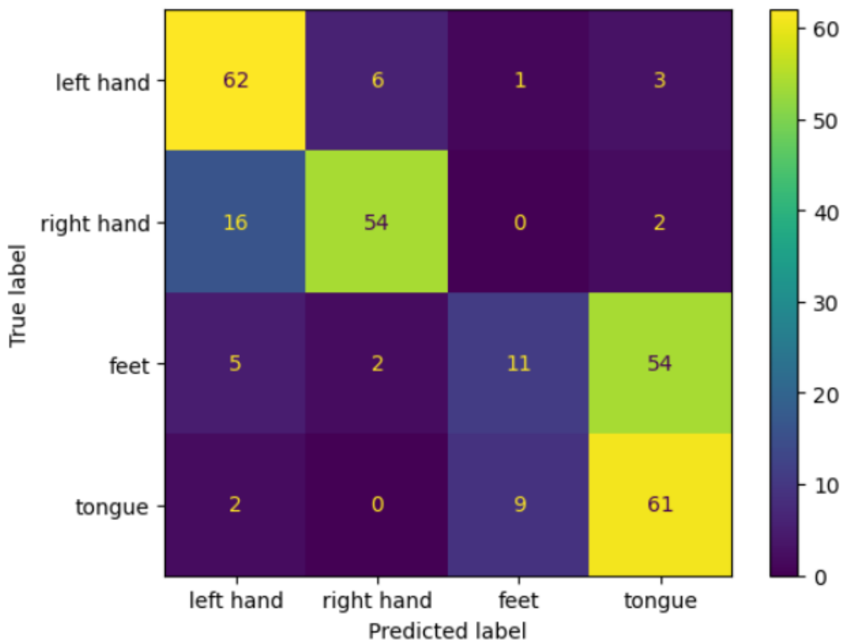
Model: SCCNet Scheme: sd , accuracy = 0.7778



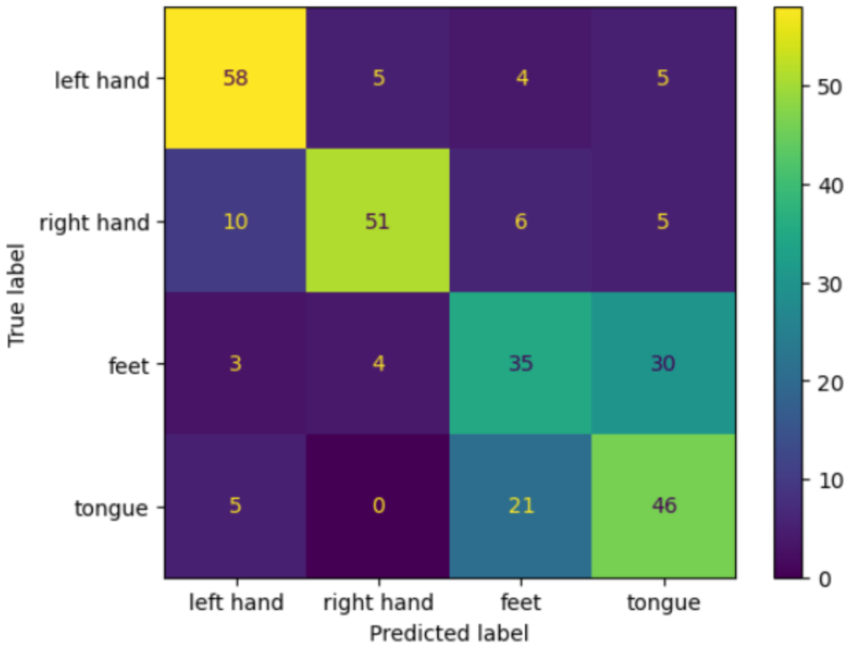
Model: SCCNet Scheme: sift , accuracy = 0.7083



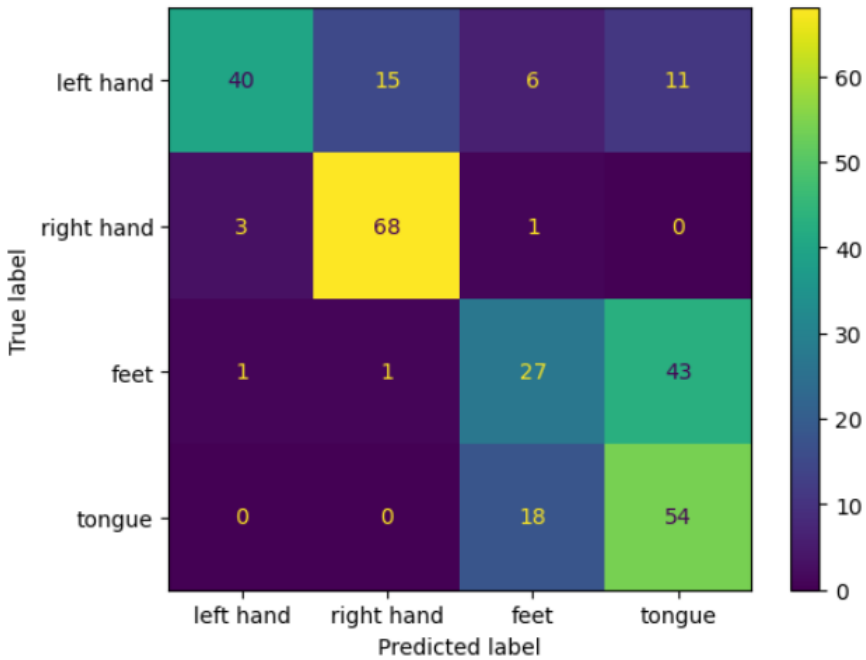
Model: EEGNet Scheme: ind , accuracy = 0.6528



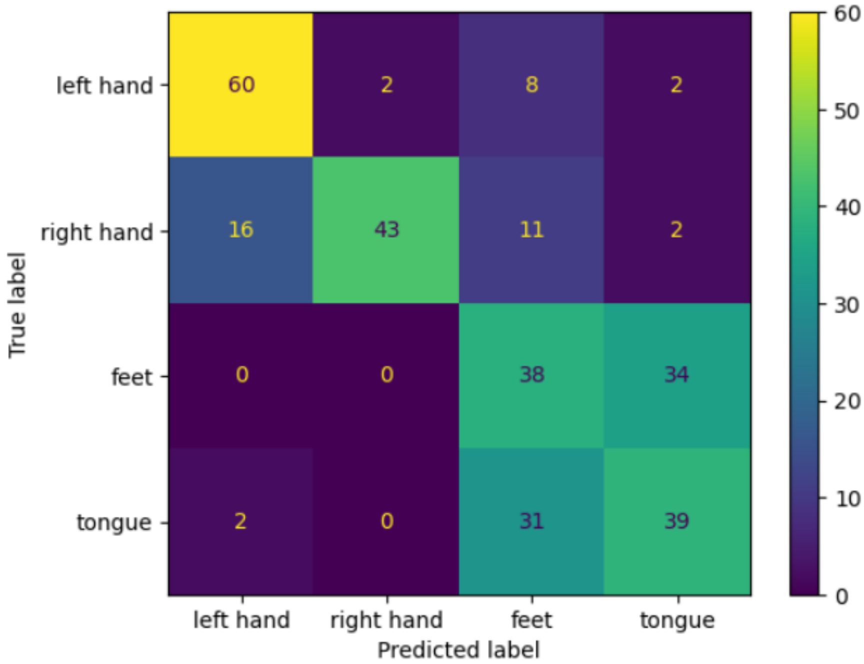
Model: EEGNet Scheme: si , accuracy = 0.6597



Model: EEGNet Scheme: sd , accuracy = 0.6562

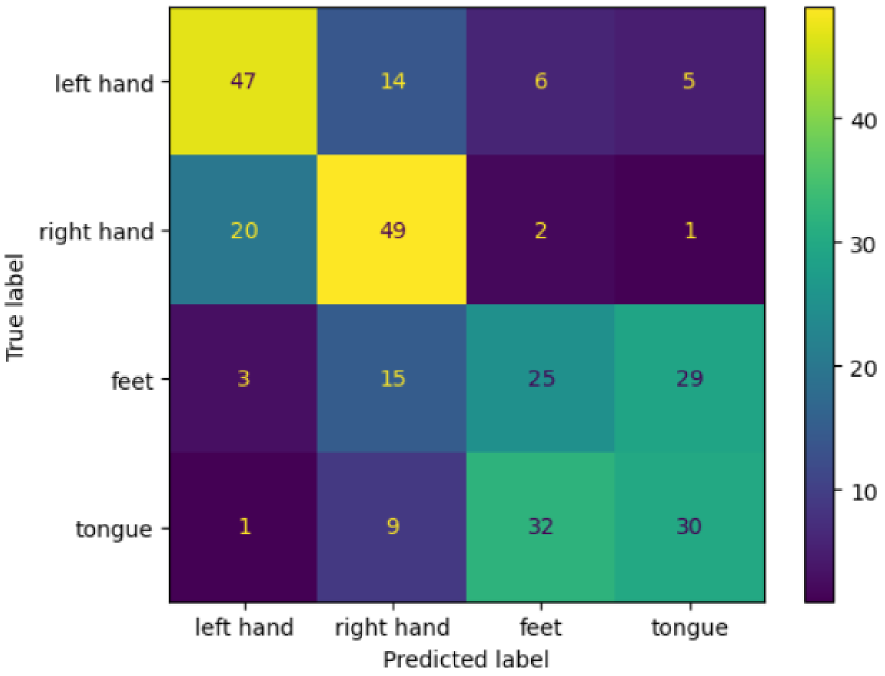


Model: EEGNet Scheme: sift , accuracy = 0.625

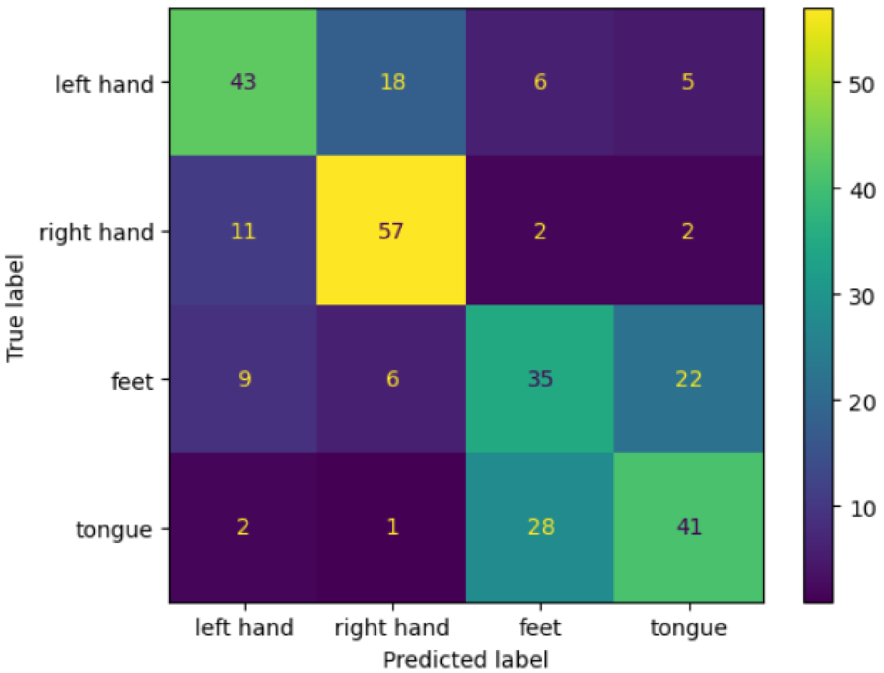


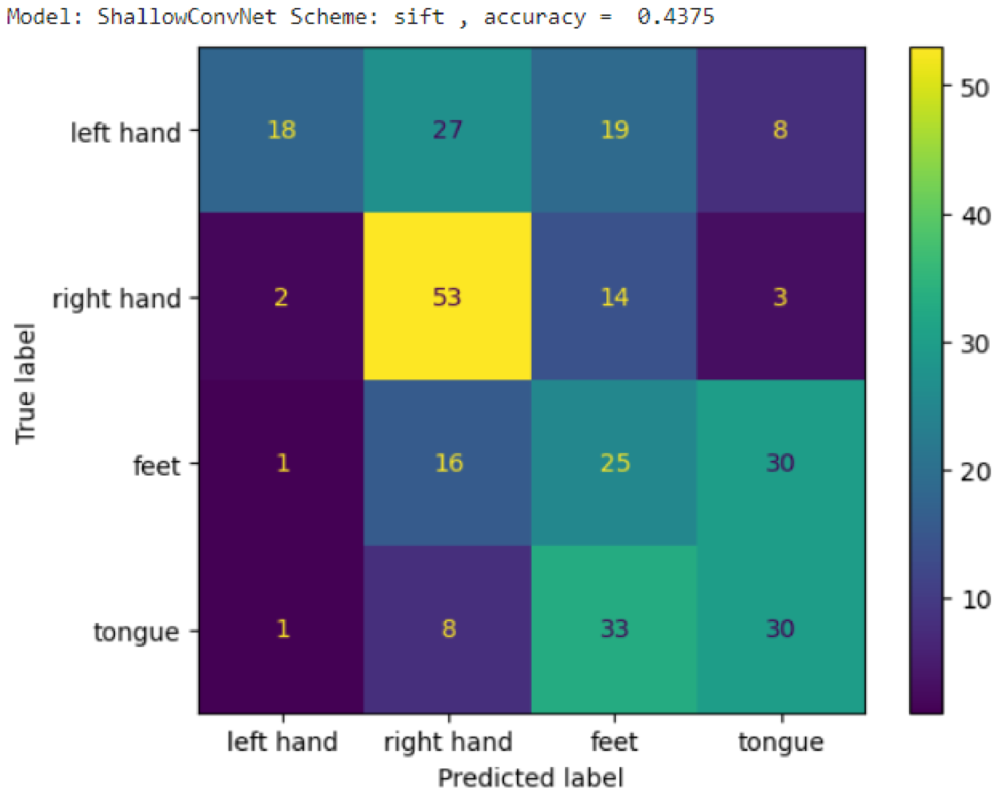
Model: ShallowConvNet Scheme: ind , accuracy = 0.5833

Model: ShallowConvNet Scheme: si , accuracy = 0.5243



Model: ShallowConvNet Scheme: sd , accuracy = 0.6111





Discussion

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

3 CNN models:

SCCNet:

Has a lightened structure to avoid over-fitting to the highly variant EEG data

Has the best performance result among all 3 CNN based models

EEGNet:

Can learn a wide variety of interpretable features over a range of BCI tasks

Not performing better than models built to perform specific tasks

ShallowConvNet:

The size is very big compared to the other two CNN based models

Have many parameters to train, which may lead to over-fitting

4 Training Schemes:

Individual:

The model is trained and tested within a single subject. It is simple and fast but could have bad performance when there are large inter-subject differences since it is only trained on a single subject.

Subject Independent:

The subjects in the dataset are separated into 1 test data and other training data. The model is tested on the second session of the test subject. To me this is the best approach to evaluate the performance of a model since the training and testing dataset have no relationship between, unlike the other schemes that use sessions from the same subject.

Subject Dependent:

It includes the first session of the test subject with all sessions from other subjects as the training data. The model is tested on the second session of the test subject. Can have one more session of data for training, and since the first session of testing subject is included, the validation accuracy may be higher.

Subject Independent + Fine-Tuning

The data of the first session of the test subject are used in the fine-tuning process to tune the parameters of the pretrained SI model. Applying fine tuning should make the model perform better on data of target subject. It may take more time and effort to do so.

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

SCCNet:

```
==========================================================================================
Layer (type:depth-idx)                   Output Shape              Param #
==========================================================================================
SCCNet                                   [16, 4]                   --
├─Sequential: 1-1                        [16, 22, 1, 562]          --
│    └─Conv2d: 2-1                       [16, 22, 1, 562]          484
│    └─BatchNorm2d: 2-2                  [16, 22, 1, 562]          44
├─Sequential: 1-2                        [16, 20, 1, 551]          --
│    └─Conv2d: 2-3                       [16, 20, 1, 551]          5,280
│    └─BatchNorm2d: 2-4                  [16, 20, 1, 551]          40
│    └─Dropout: 2-5                      [16, 20, 1, 551]          --
├─AvgPool2d: 1-3                         [16, 1, 20, 41]           --
├─Flatten: 1-4                           [16, 820]                 --
├─Sequential: 1-5                        [16, 4]                   --
│    └─Linear: 2-6                       [16, 4]                   3,284
│    └─Softmax: 2-7                      [16, 4]                   --
==========================================================================================
Total params: 9,132
Trainable params: 9,132
Non-trainable params: 0
Total mult-adds (M): 50.95
==========================================================================================
Input size (MB): 0.79
Forward/backward pass size (MB): 5.99
Params size (MB): 0.04
Estimated Total Size (MB): 6.81
==========================================================================================
```

Comparing the size of the models, SCCNet is the smallest among all (6.81MB). Although the total parameter number is not the smallest, it has the least total mult-add count among the 3 models, which could lower the computation time thus lower the overall training time.

```
==========================================================================================
Layer (type:depth-idx)                   Output Shape              Param #
==========================================================================================
EEGNet                                   [16, 4]                   --
├─Sequential: 1-1                        [16, 8, 22, 499]          --
│    └─Conv2d: 2-1                       [16, 8, 22, 499]          512
│    └─BatchNorm2d: 2-2                  [16, 8, 22, 499]          16
├─Sequential: 1-2                        [16, 16, 1, 124]          --
│    └─Conv2d: 2-3                       [16, 16, 1, 499]          352
│    └─BatchNorm2d: 2-4                  [16, 16, 1, 499]          32
│    └─ELU: 2-5                          [16, 16, 1, 499]          --
│    └─AvgPool2d: 2-6                    [16, 16, 1, 124]          --
│    └─Dropout: 2-7                      [16, 16, 1, 124]          --
├─Sequential: 1-3                        [16, 16, 1, 15]           --
│    └─Conv2d: 2-8                       [16, 16, 1, 125]          256
│    └─Conv2d: 2-9                       [16, 16, 1, 125]          256
│    └─BatchNorm2d: 2-10                 [16, 16, 1, 125]          32
│    └─ELU: 2-11                         [16, 16, 1, 125]          --
│    └─AvgPool2d: 2-12                   [16, 16, 1, 15]           --
│    └─Dropout: 2-13                     [16, 16, 1, 15]           --
├─Linear: 1-4                            [16, 4]                   964
==========================================================================================
Total params: 2,420
Trainable params: 2,420
Non-trainable params: 0
Total mult-adds (M): 93.78
==========================================================================================
Input size (MB): 0.79
Forward/backward pass size (MB): 25.30
Params size (MB): 0.01
Estimated Total Size (MB): 26.10
==========================================================================================
```

EEGNet is also considered a small model among the CNN based models.
It uses a total of 4 convolution layers, which is quite deep comparing with the other two models, which only use 2.

```
==========================================================================================
Layer (type:depth-idx)                   Output Shape              Param #
==========================================================================================
ShallowConvNet                           [16, 4]                   --
├─Conv2d: 1-1                            [16, 40, 22, 551]         480
├─Conv2d: 1-2                            [16, 40, 1, 551]          35,200
├─BatchNorm2d: 1-3                       [16, 40, 1, 551]          80
├─AvgPool2d: 1-4                         [16, 40, 1, 74]           --
├─Dropout: 1-5                           [16, 40, 1, 74]           --
├─Linear: 1-6                            [16, 4]                   11,844
==========================================================================================
Total params: 47,604
Trainable params: 47,604
Non-trainable params: 0
Total mult-adds (M): 403.61
==========================================================================================
Input size (MB): 0.79
Forward/backward pass size (MB): 67.71
Params size (MB): 0.19
Estimated Total Size (MB): 68.69
==========================================================================================
```

The Shallow Convolution Network is shallow, of course, but with a lot more parameters to learn when training.

3. Difficulties you encountered in this homework.

I had a lot of trouble working on this homework since I started. For the first part, we are asked to complete the SCCNet structure using PyTorch. I had done some work using PyTorch before, but they were just some tiny CNN models for image classification. And this is my first time using EEG signal as the input of a neural network so it took me a lot of time to figure out how the model actually work, and then it took me even more time to construct the model. Until now I am still not sure if the structure is correct.

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

5. Other topics you find worthy to discuss.