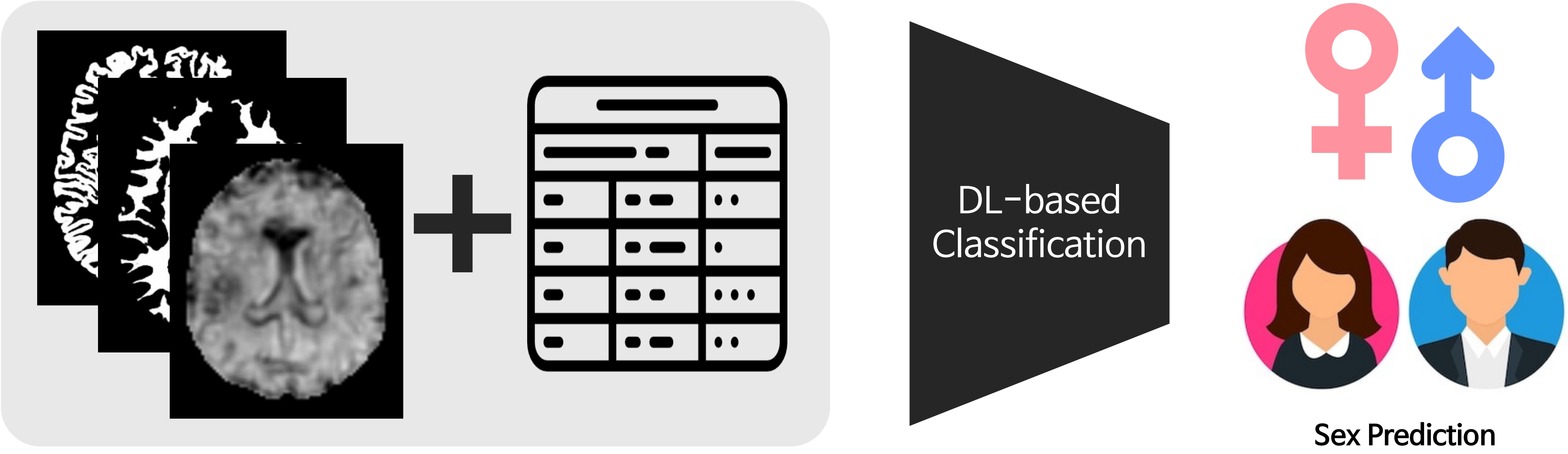
Sex Prediction with Deep Learning

221AIG08 한지원

Introduction

- Goal
  - Designing a deep learning model to predict the sex of each subject, by using MRI scan images and demographic information.



DL-based Classification

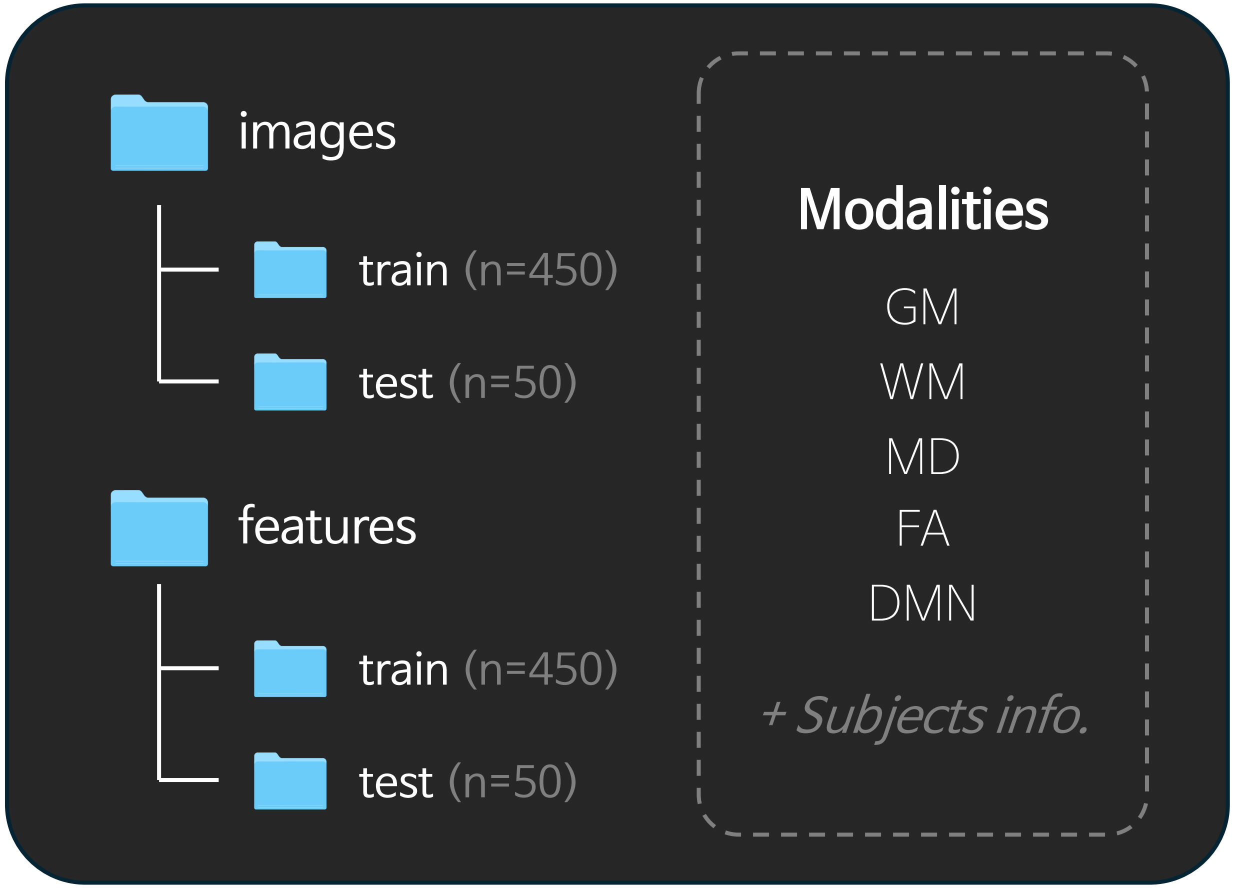
Sex Prediction

Data

- Structure
  - **Images**

    Structural, resting state functional, and diffusion-weighted MRI, demographic information with sex.
  - **Features**

    Extracted features, demographic information with sex.



images

train (n=450)

test (n=50)

features

train (n=450)

test (n=50)

Modalities

GM

WM

MD

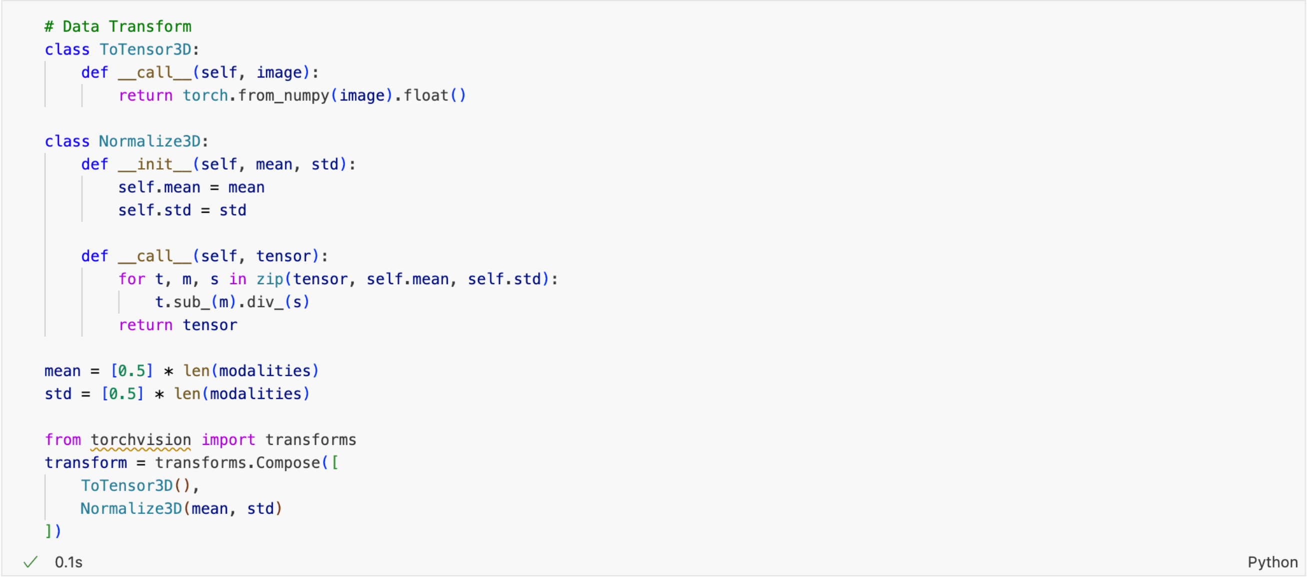
FA

DMN

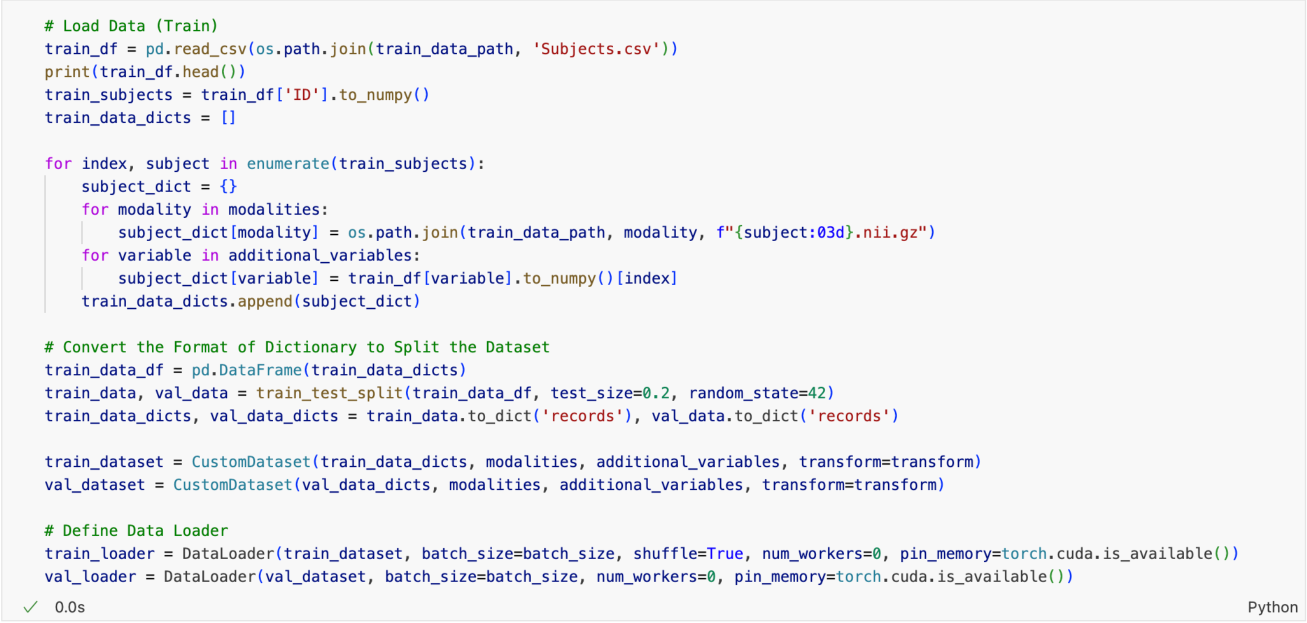
+ Subjects info.

Data

- Pre-processing
  - Merge the images with subjects' demographic information.
  - Split the features into predicator variables and response variable.
  - Define and apply the 3D image transform functions to the image data; Normalize3D, ToTensor3D.



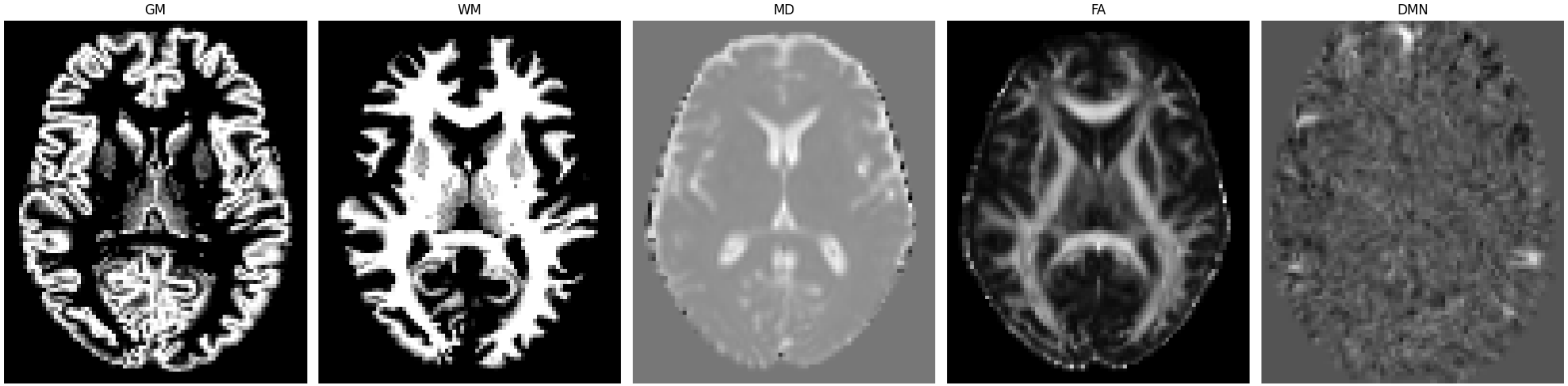
```python
# Data Transform
class ToTensor3D:
    def __call__(self, image):
        return torch.from_numpy(image).float()

class Normalize3D:
    def __init__(self, mean, std):
        self.mean = mean
        self.std = std

    def __call__(self, tensor):
        for t, m, s in zip(tensor, self.mean, self.std):
            t.sub_(m).div_(s)
        return tensor

mean = [0.5] * len(modalities)
std = [0.5] * len(modalities)

from torchvision import transforms
transform = transforms.Compose([
    ToTensor3D(),
    Normalize3D(mean, std)
])
```



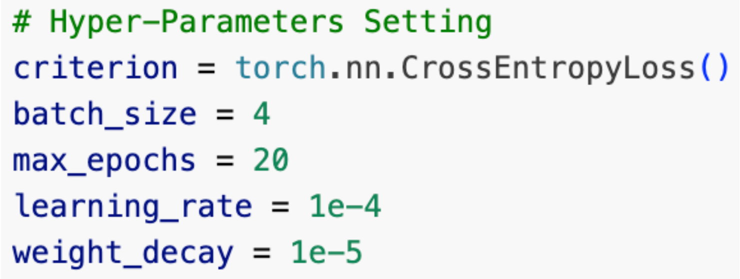
```python
# Load Data (Train)
train_df = pd.read_csv(os.path.join(train_data_path, 'Subjects.csv'))
print(train_df.head())
train_subjects = train_df['ID'].to_numpy()
train_data_dicts = []

for index, subject in enumerate(train_subjects):
    subject_dict = {}
    for modality in modalities:
        subject_dict[modality] = os.path.join(train_data_path, modality, f"{subject:03d}.nii.gz")
    for variable in additional_variables:
        subject_dict[variable] = train_df[variable].to_numpy()[index]
    train_data_dicts.append(subject_dict)

# Convert the Format of Dictionary to Split the Dataset
train_data_df = pd.DataFrame(train_data_dicts)
train_data, val_data = train_test_split(train_data_df, test_size=0.2, random_state=42)
train_data_dicts, val_data_dicts = train_data.to_dict('records'), val_data.to_dict('records')

train_dataset = CustomDataset(train_data_dicts, modalities, additional_variables, transform=transform)
val_dataset = CustomDataset(val_data_dicts, modalities, additional_variables, transform=transform)

# Define Data Loader
train_loader = DataLoader(train_dataset, batch_size=batch_size, shuffle=True, num_workers=0, pin_memory=torch.cuda.is_available())
val_loader = DataLoader(val_dataset, batch_size=batch_size, num_workers=0, pin_memory=torch.cuda.is_available())
```

Model

- CNN-based Classification Model
  - [**Input**: Image & Subject Info] → [**Output**: Male/Female Possibilities]
    - Image Size: (#modalities, 79, 95, 79)
    - Subject Info: ID, Age

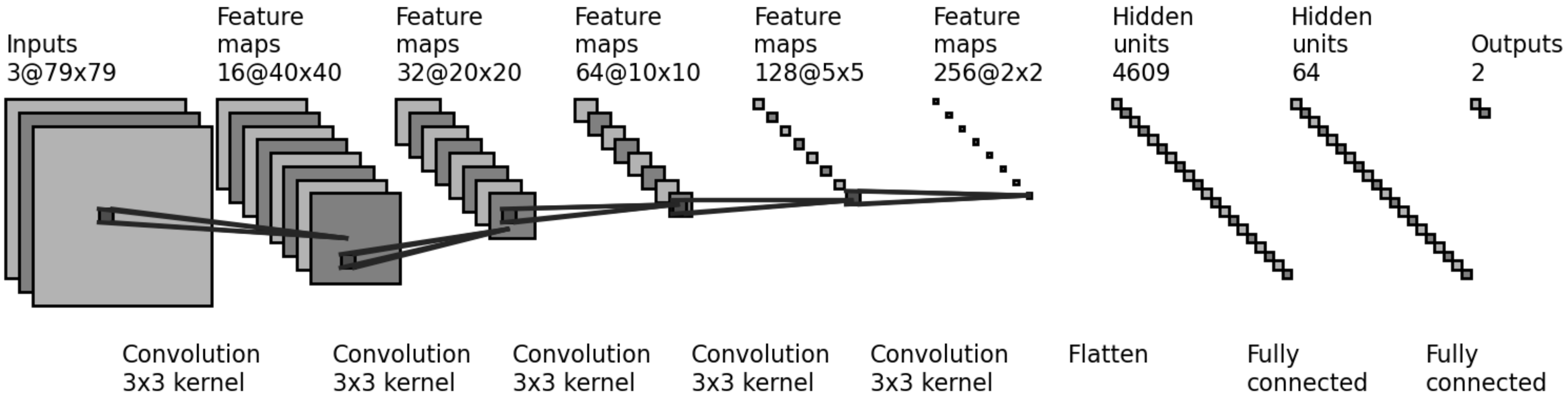


Model



```
# Hyper-Parameters Setting
criterion = torch.nn.CrossEntropyLoss()
batch_size = 4
max_epochs = 20
learning_rate = 1e-4
weight_decay = 1e-5
```

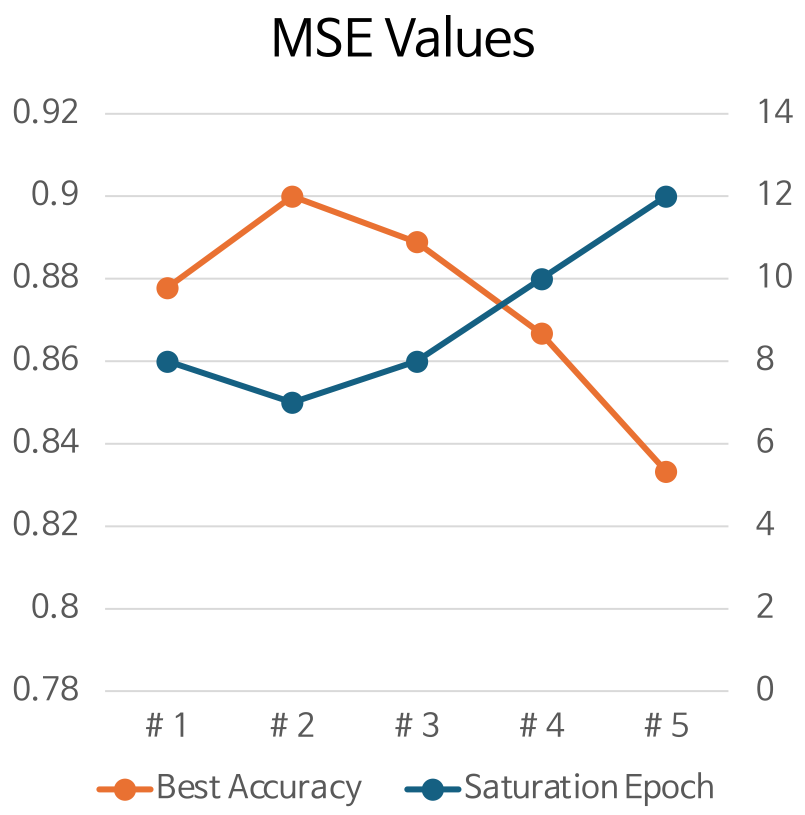
- CNN-based Classification Model
  - Setting Hyper-Parameters
    - Criterion: Cross Entropy Loss
    - Batch size, Epochs, Learning Rate, Weight Decay, Dropout



Result

- Model Comparison: Accuracy & Computing Time

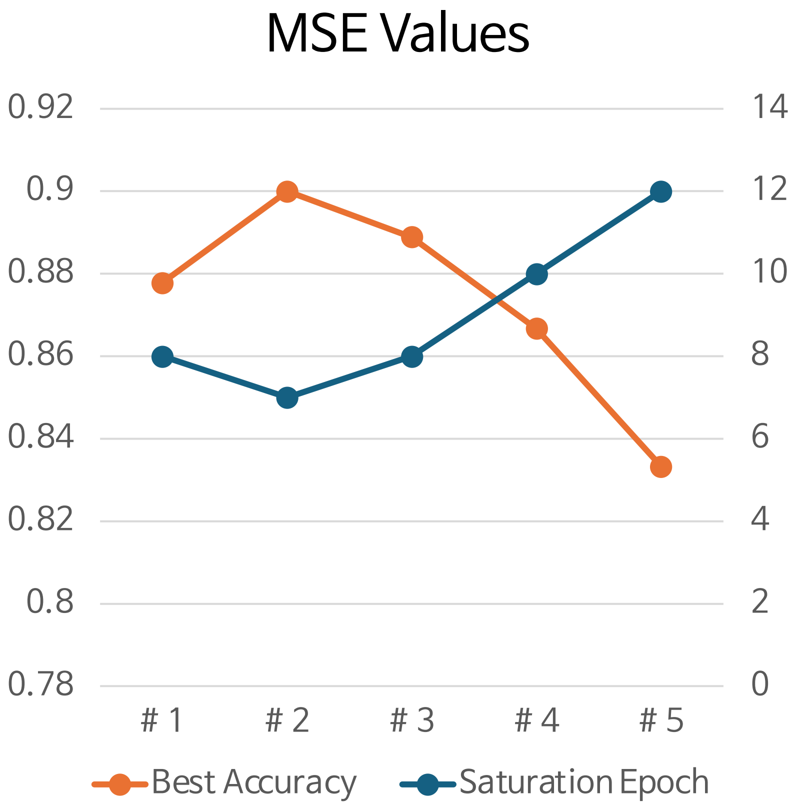
Modalities	Best Accuracy	Time Consumption	Saturation Epoch
GM	0.878	26 min	8
WM+GM	0.900	40 min	7
MD+WM+GM	0.889	47 min	8
FA+MD+WM+GM	0.867	34 min	10
DMN+FA+MD+WM+GM	0.833	43 min	12



Result

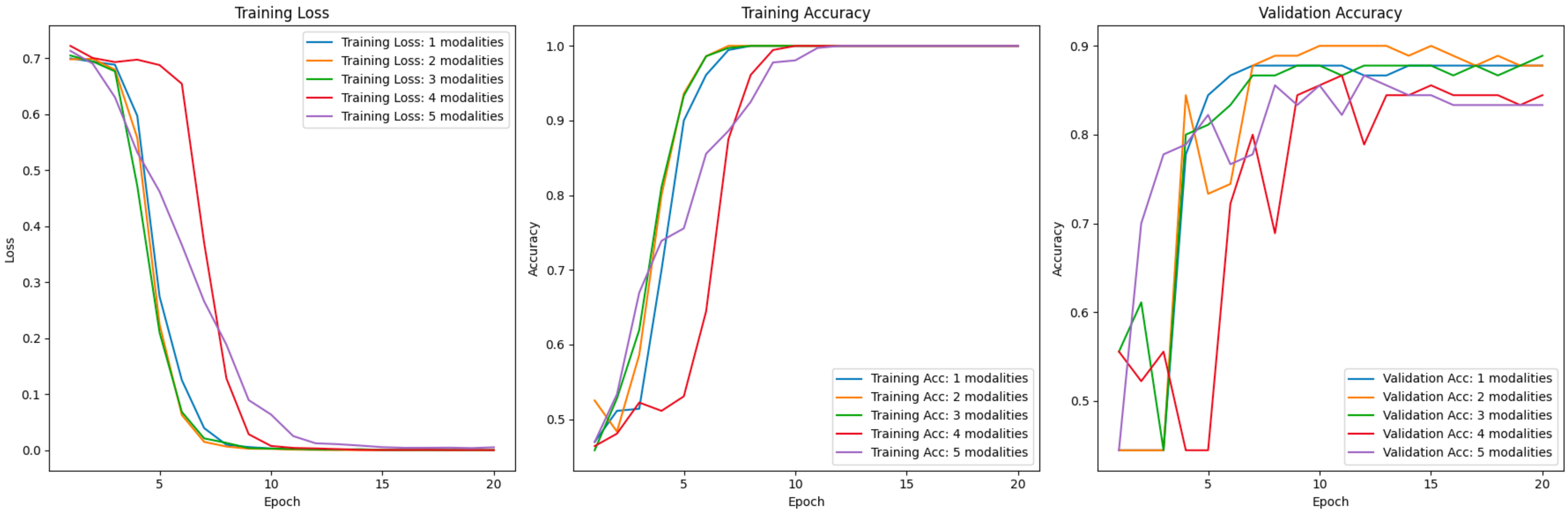
- Model Comparison: Best Accuracy & Computing Time

Modalities	Best Accuracy	Time Consumption	Saturation Epoch
GM	0.878	26 min	8
WM+GM	0.900	40 min	7
MD+WM+GM	0.889	47 min	8
FA+MD+WM+GM	0.867	34 min	10
DMN+FA+MD+WM+GM	0.833	43 min	12

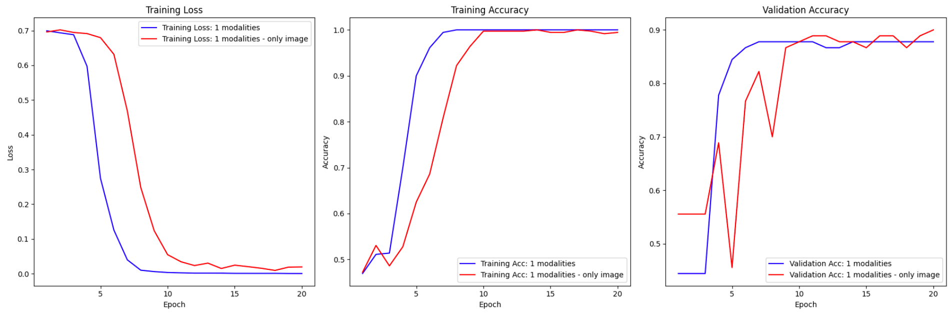


Result

- Comparison Metric: Loss & Accuracy



Sex Prediction



- Predicted Result
  - Predicted sex of first 10 subjects in the test data, after employing DL model.
    - Regardless of the number of modalities used in the models, the classification results remained consistent.
  - However, when using only the image data without age, the inference results differ from those obtained before.
    - When comparing the prediction results of the first 10 subjects, the image-only model predicted the 6th subject as male.

Subject	Age	Predictions
1	55	M
2	55	F
3	56	M
4	58	M
5	57	M
6	55	F
7	54	M
8	58	F
9	57	M
10	56	M

(0: Female, 1: Male)

Thank You