Project #3: Encoder-Decoder Implementation

2019170540 Wonjun, Oh (오원준)

1. Description of your code

Main Skeleton In the model class invocation section, I configured the type of model and the address to load the model differently based on the Option. Specifically, it was crucial to set the number of classes to 22 instead of 20 during the invocation. An error occurred in the path configuration section, and to address the issue, I aligned the error-causing path with the local path. Additionally, I made modifications to ensure that the saved path varies depending on the type of the invoked model.

Modules Skeleton For the code related to "Validation," only the loss for the model's output is calculated. However, for "Training," additional code is required for loss backward propagation, optimizer step, and scheduler step. Particularly, it is crucial to call "Zero_grad()" since the optimizer, by default, accumulates gradients. In the code that converts the model's output to an RGB file, for each image, the RGB values corresponding to the indices in a pre-specified dictionary for each element (HxW elements) were found and replaced.

UNet I aligned the dimensions of the Convolution Layer according to the instructions, and during the forward operation, I performed UpSampling followed by concatenation using the torch cat operation. It was important to set dim to 1 so that concatenation could be done along the Channel dimension.

Resnet Encoder UNet I used the same code as the previous assignment, with the only difference being that I did not perform downsampling in the last ResidualBlock of Layer3. In contrast to the previous UNet, where different-dimensional outputs were concatenated, this time I concatenated outputs with the same dimensions.

2. Results

In the case of UNet, 1Epoch learning as indicated in the task resulted in the following Figure.

```
epoch 1 train loss :  0.8927034417243853 train acc :  0.8538018715930905
epoch 1 val loss :  1.28409982210881 val acc :  0.7658124872156091
Finish Training
Fin
PS F:\Projects>
```

(Figure 1) Results of UNet

In the case of UNet using a Resnet encoder, 1Epoch learning as indicated in the task resulted in the following Figure.

```
epoch 1 train loss :  0.9242609863820141 train acc :  0.7380200192462166
epoch 1 val loss :  1.0557180876667436 val acc :  0.7145449062725445
Finish Training
Fin
PS F:\Projects>
```

(Figure 1) Results of ResNet Encoder UNet

3. Discussions

Class Numbers Although the number of classes is shown as 20, I was wondering why there was an error when the class number was set to 20. In the case of PASCAL VOC2012, in addition to the 20 classes that should be classified basically, 21 classes were required, including background. Nevertheless, the code required 22 classes, and after examining the items corresponding to index 21 in the result**, it was found that it meant a dividing line separating the boundary between the class and the class.**

Up-sampling comparison between the two models Unlike the original paper, UNet code used “Bilinear interpolation” rather than Up-convolution in the Up-sampling process. On the other hand, in the case of the Resnet encoder UNet, “ConvTranspose2d” was used for Up-sampling.

End.