**Assignment 2**

**Submitted by: Adarsh Ghimire**

**Student ID: 100058927**

**General summary of the assignment.**

In this assignment, the UNET architecture model is implemented, trained, and tested for the ISPRS Dataset segmentation task. The dataset comprises of 2400 images of size 300x300 in total. So, for the completeness and robustness of the model, the total data is divided into 3 sets:

1. Training Set of 2000 images
2. Validation Set of 200 images
3. Test Set of 200 images

The validation set has been chosen to find the optimal model parameters, and test set has been kept untouched until the model in finalized so that there is no information leakage.

The model has been trained in Kaggle for 20 epochs, with batch size of 10 only, and all the model parameters and optimizer states has been saved in regular manner to allow further training, if required. In addition, the best performing model on validation set is also saved so that it can later be used for inference purpose. Below is the summarized result of the best model:

|  |  |  |
| --- | --- | --- |
| **Dataset** | **Loss (cross entropy)** | **Accuracy** |
| Training set | 0.59 | 80.033 |
| Validation set | 0.63 | 79.312 |
| Test set | 0.54 | 79.992 |

The models are large thus are saved in google drive for sharing. Please find the model from the below link.

Model folder link :

<https://drive.google.com/drive/folders/13MimYs-8wyylYbM_q_kEvb4cmjGhPrqF?usp=sharing>

Comprises of two files:

1. model\_best.pth.tar 🡪 Is the best model
2. Checkpoint.pth.tar 🡪 is the current model state

**Detailed explanation of the assignment**

1. **Initialization**

Here the initial some parameters are initialized at beginning.

Text

Description automatically generated

1. Some necessary functions

Graphical user interface, text, application

Description automatically generated

This part of the code is responsible for conversion of the RGB color encoding of the labels of the data to categorical value from 0 to 6.

Graphical user interface, text, application, email

Description automatically generated

This is the custom dataset class for the custom ISPRS dataset so that later pytorch data loader can use it.

Graphical user interface, text, application

Description automatically generated

This function computes the cross-entropy loss for 2D data. Since the model outputs

*(batch, classes\_score, input\_widht, input\_height)* as the output, and the actual label is just (*batch, class\_number, input\_width, input\_height*). So, the above function computes the loss among them.

Graphical user interface, text, application, chat or text message

Description automatically generated

The function computes the confusion matrix and accuracy on per pixel basis.

1. **Selecting training, validation, and test data**

**Graphical user interface, text, application, email

Description automatically generated**

1. **Implementation of U-Net Model**

The implementation of U-Net model has been modularized for easy and faster implementation. And, to avoid repetitive coding.

First, is the DoubleConv class which performs two convolution operations followed by the Relu activation. In between the convolution and activation operation batch normalization layer has been added for making learning efficient and to avoid overfitting. As it can be seen from summarized result above that the model is able to generalize.

**Graphical user interface, text, application, email

Description automatically generated**

Second is the Down class which does maxpooling and then performs DoubleConv operation as described above.

**Graphical user interface, text, application

Description automatically generated**

Third is the Up class which performs the upscaling operation since we need to convert our encoded information back to original size. In this class, the upscaling can be performed in two ways has been implemented. First is using nn.Upsample for upscaling and another is using nn.ConvTranspose2d. Both works fine. After the upscaling operation feature map concatenation with feature map from down operation is performed followed by Double Convolution operation.

**Graphical user interface, text, application, email

Description automatically generated**

Below, is the overall architecture for the Unet.

Table

Description automatically generated with low confidence

1. **Data Visualization**

**Graphical user interface, text

Description automatically generated**

This function will convert the class labels to corresponding color palette so that visualization can be done.

**Text

Description automatically generated with medium confidence**

This function plots the image, its ground truth and the prediction from the model if provided.

Example of image and its corresponding label.

Graphical user interface, application

Description automatically generated

Visualizing the untrained model output, just to confirm that the process will work fine during training process. Checking the current untrained model loss, accuracy, and some visualization.

Randomly initialized untrained model summary :

Loss : 1.8572

Accuracy : 12.83%

**Graphical user interface

Description automatically generated**

**Graphical user interface

Description automatically generated**

**Graphical user interface

Description automatically generated**

1. **Training**

**Graphical user interface, text, application

Description automatically generated**

This function for calculating accuracy has been redefined so that the function do not keep on printing on every step of training

**Text

Description automatically generated with low confidence**

This function will save the model in regular manner as well as save the best performing model.

Graphical user interface, text, application, chat or text message

Description automatically generated

Initial parameters setup for training, and Unet model object instantiation.

**Graphical user interface, text, application

Description automatically generatedA picture containing text

Description automatically generatedTraining step**

**Graphical user interface, text, application, email

Description automatically generated**

Above code snippet shows the training step followed by validation step for verification of the trained model. The model verification is done every 100th step. And if the validation model performance is better than other previous performance then the optimal model parameters are saved as *model.pt.tar*. And latest model trained is saved as *checkpoint.pt.tar* in the home directory. So, the checkpoint is saved to allow further training later, and best model is saved for best inferencing model.

Output of the above step.

**Text

Description automatically generatedText

Description automatically generated**

**…………….**

1. **Model training summary plots**

Chart, histogram

Description automatically generatedGraphical user interface, text, application

Description automatically generated

From the plot we can see that the training loss and validation loss are decreasing swiftly during training of the model. The validation loss is changing within the entire process however the overall nature of the curve is decreasing. And the performance has not saturated yet, so we can still improve the performance if we train for more epoch or by changing the hyperparameters.

Text

Description automatically generatedGraphical user interface

Description automatically generated with medium confidence

The above curve shows the validation accuracy performance during the training process.

1. **Model Testing**

First the best model parameter is loaded for the inferencing.

**Graphical user interface, text, application, chat or text message

Description automatically generated**

**Test step**

**Text

Description automatically generated with medium confidenceGraphical user interface, text, application, email

Description automatically generated**

1. **Generating the prediction results and ground truth**

Code snippet for generating the prediction plots

Graphical user interface, text, application, email

Description automatically generated

Chart

Description automatically generated

Text

Description automatically generated

Chart

Description automatically generated

Text

Description automatically generated

Graphical user interface, chart

Description automatically generated with medium confidence