Agnik Saha(Roll: 21CS60A01)

I have followed steps:

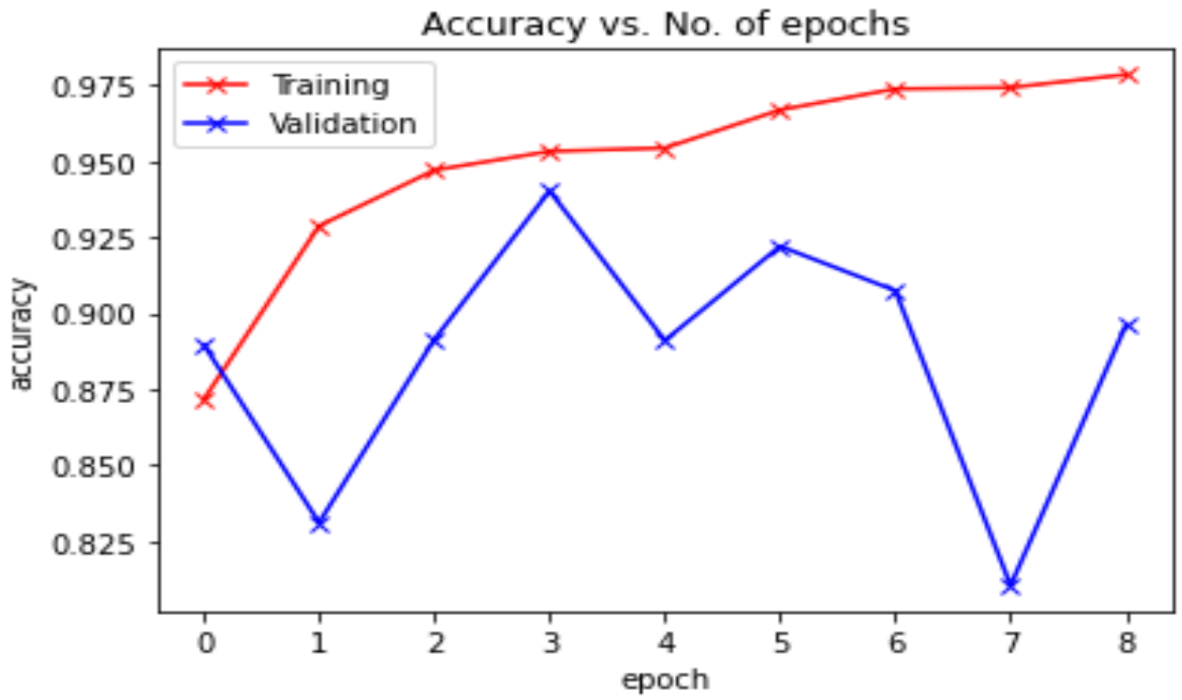
1. Loading training, validation and test dataset
2. Preparing train_loader , val_loader & test_loader
3. Creating a train_model function that trains the model and this function returns training accuracy and training loss
4. Creating validate_model function test the model and this function returns test accuracy and test loss
5. Define our model **resnet_eurosat_classifier**
6. Model fitting step -> Here we fit our model using train_model and validate_model function

Hyperparameters used in **resnet_eurosat_classifier** model

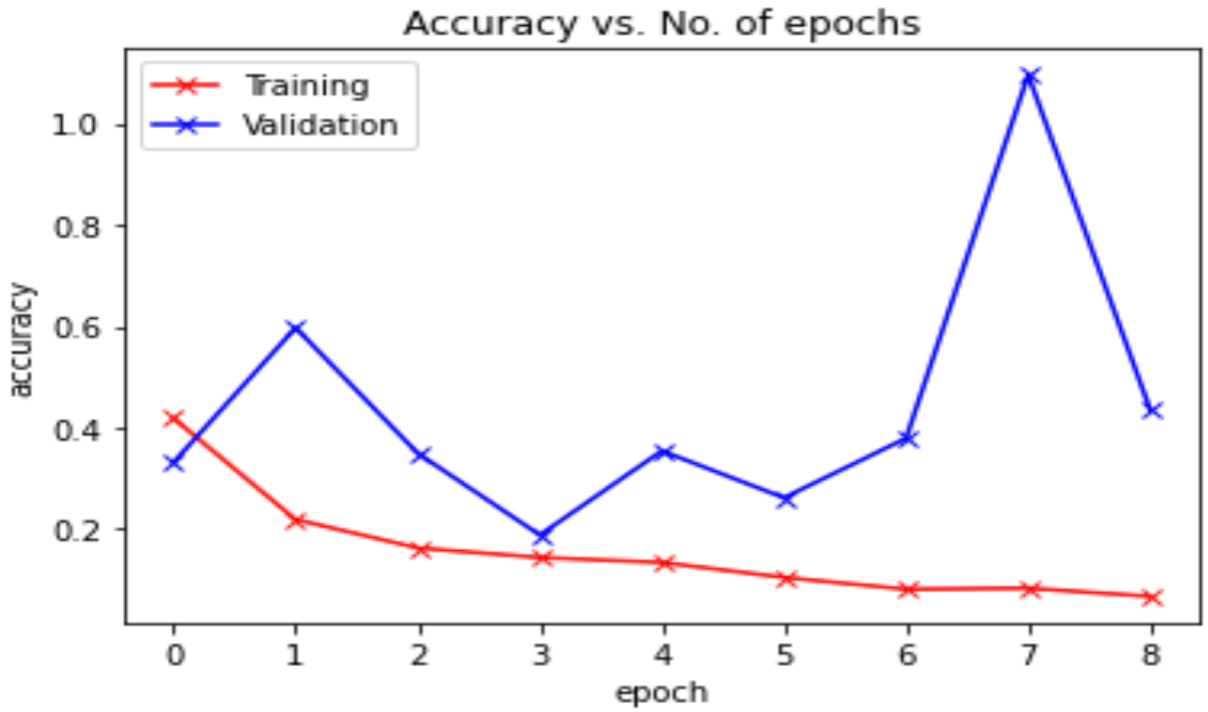
Hyperparameter	Value
Optimizer	Adam
Learning rate	0.001
Weight decay	0
Early stopping patience	5
epochs	15
Dropout rate	0.5

Accuracy vs No of epochs plot:

Agnik Saha(Roll: 21CS60A01)



Loss vs No of epochs plot:



Agnik Saha(Roll: 21CS60A01)

Accuracy on test data:

Accuracy of the network: 94.48148148148148 %
Accuracy of AnnualCrop: 100.0 %
Accuracy of Forest: 97.43589743589743 %
Accuracy of HerbaceousVegetation: 93.93939393939394 %
Accuracy of Highway: 96.96969696969697 %
Accuracy of Industrial: 97.14285714285714 %
Accuracy of Pasture: 100.0 %
Accuracy of PermanentCrop: 84.0 %
Accuracy of Residential: 97.5 %
Accuracy of River: 90.3225806451613 %
Accuracy of SeaLake: 97.72727272727273 %

Accuracy of validation data is 0.9415954415954416

RECALL

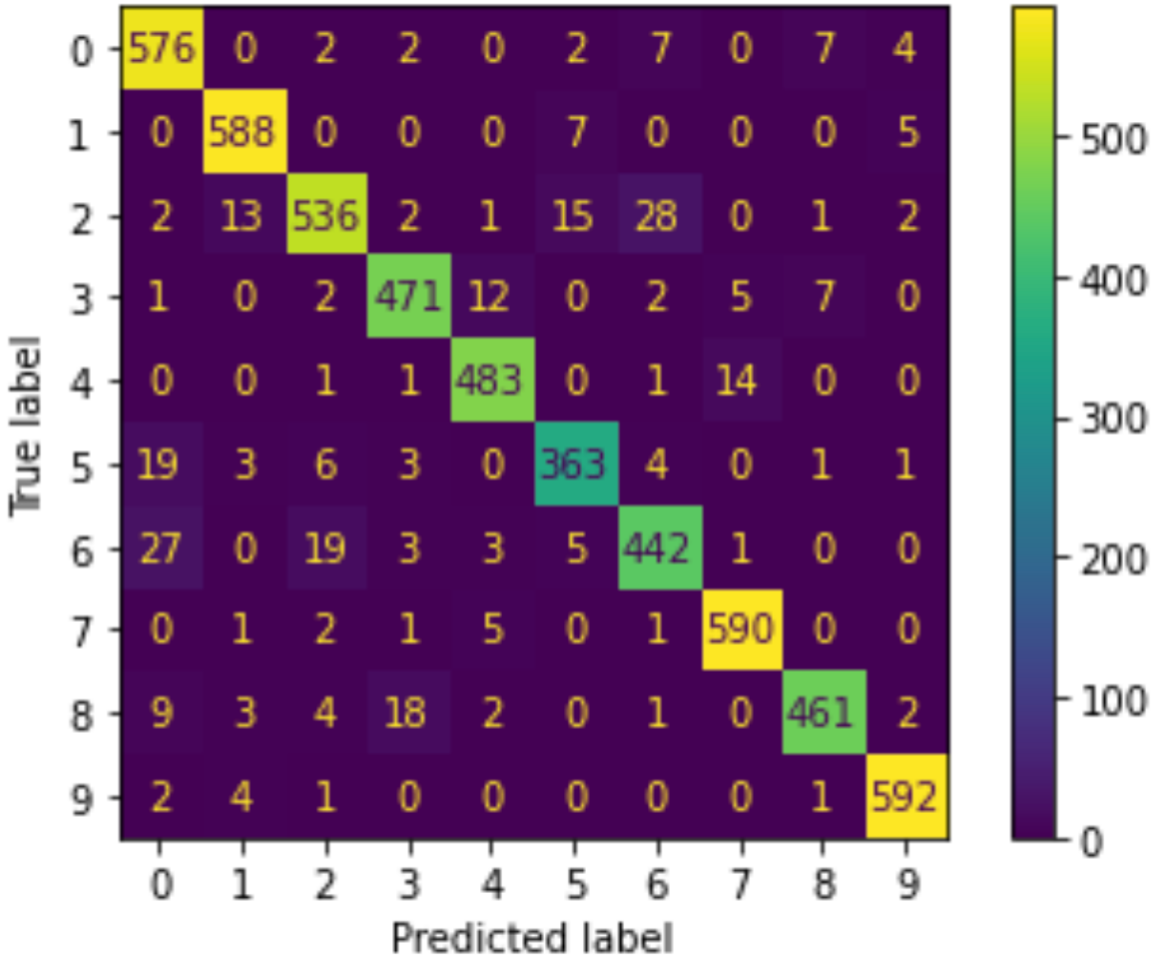
Recall value of classes = [0.96 0.98 0.89333333 0.942 0.966 0.9075
0.884 0.98333333 0.922 0.98666667]

Recall means the percentage of a certain class correctly identified.
Model was confident about the class whose recall value is maximum.

Here recall value is maximum for 9TH class ie SeaLake.
In the question it is asked to plot four examples the model got wrong and was most confident about and also add the true class and the predicted class along with the image.

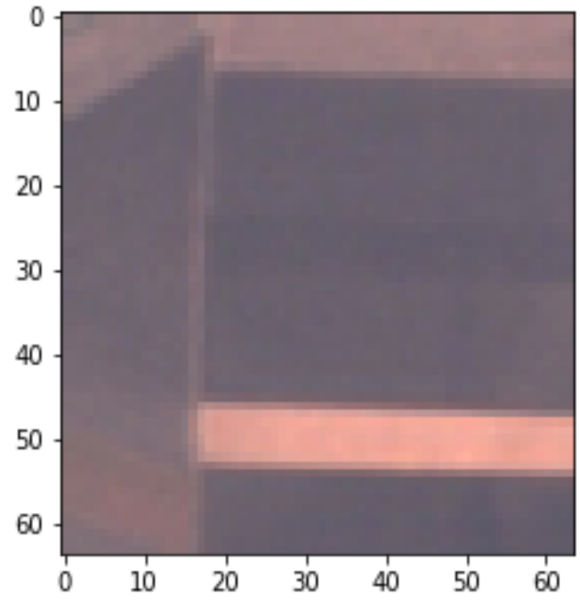
Agnik Saha(Roll: 21CS60A01)

Confusion matrix

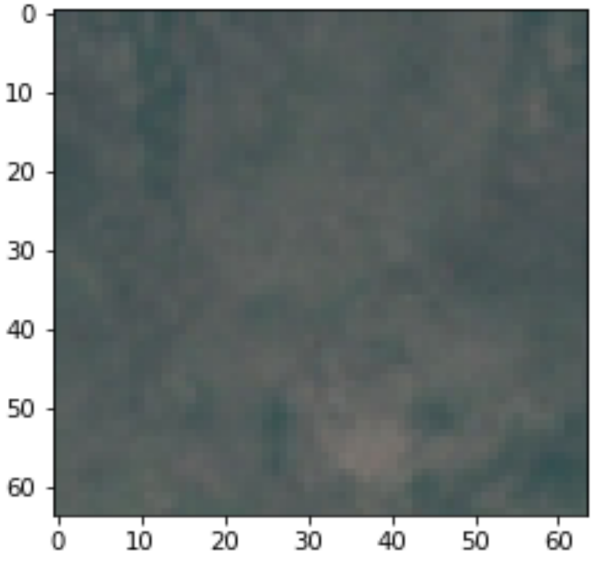


Here from the confusion matrix we have seen that we have got only 8 mispredicted images for label 9

```
True: SeaLake    Predicted: AnnualCrop
```

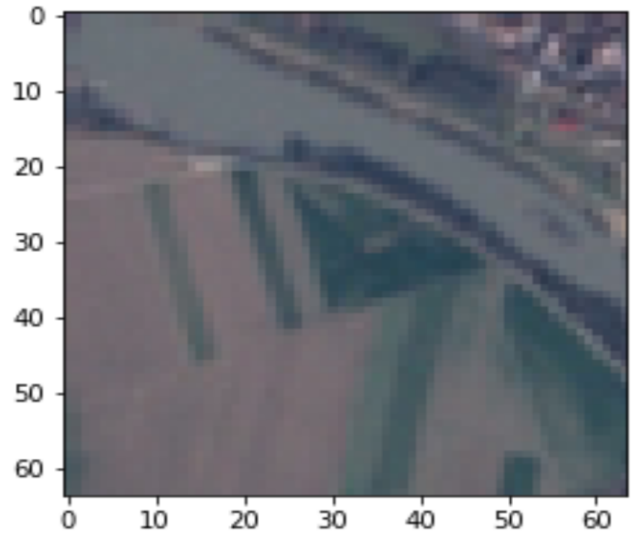



```
True: SeaLake    Predicted: Forest
```

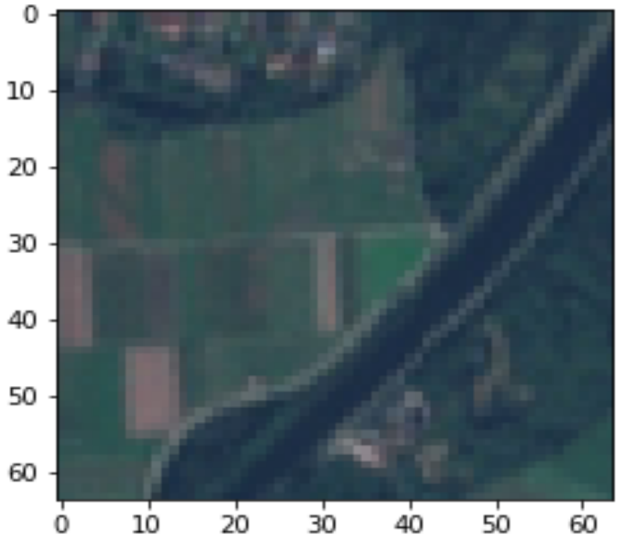


Agnik Saha(Roll: 21CS60A01)

```
True: SeaLake    Predicted: Forest
```




```
True: SeaLake    Predicted: AnnualCrop
```



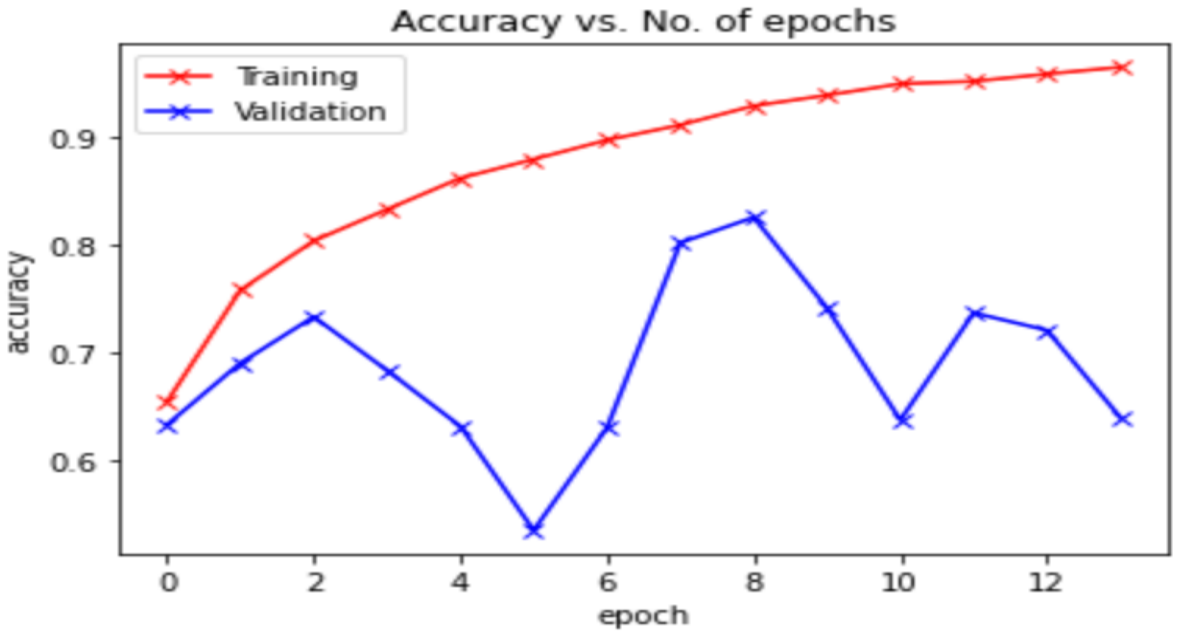
I retrain the model with the same hyper parameters, but keep pretrained=False. Keeping the same parameters, model performance drastically decreased.

Accuracy of test data is 0.875 whereas previous accuracy was 0.9448148148148148.

Accuracy of validation data is 94.48148148148 whereas previous accuracy was 0.9415954415954416

Agnik Saha(Roll: 21CS60A01)

I have plotted training and test accuracy for the resnet(pretrained = False)




```
plot_losses(train_loss_false,val_loss_false)
```