Report on Task 2

Resnet18 on EuroSAT dataset for Image Classification

Name: Ankit Kumar Verma
Roll Num: 21CS60A04

Directory Structure:

```
/resnet_21CS60A04_AnkitKumarVerma
     resnet_21CS60A04_AnkitKumarVerma.ipynb
     resnet_21CS60A04_AnkitKumarVerma.pdf
     /logs_resnet
          best_eurosat_pretrained_false.pth
          best_eurosat_pretrained_true.pth
          the above two *.pth files are downloaded from google drive
```

Dependencies:

Stable Internet Connection	The program downloads the model weights from GoogleDrive Repo. Link: https://drive.google.com/drive/u/0/folders/1GhHPvuxytPwZuuwoPaM1Ypu-gYEoJ75q
Python	>3, 3.9.6 (recommended)

Python Libraries	All the libraries mentioned under "**Importing required libraries**" should be installed on the system

Configurations:

(Configure the following variables under "**Setting filepaths**" and "**Setting Model Parameters**" section of the *.ipynb* file before execution):

Dataset Filepath `dataset_filepath`	This path should be a local machine path. You can also set the path to a google drive if you are working on Kaggle/Google Colab and have mounted your drive.
`logs_path`	Path set to store/save the models, and also to download the golden weights from the google drive
`is_wandb (True/False)`	Set to True if you want to connect to wandb webiste
`is_drive_mount (True/False)`	Set to True if you have set the dataset path to a google drive link
`num_epochs`	Number of epocs to run the model. Default value is set to 200.
`learning_rate`	Learning Rate for the model. Default value is set to 0.001. It was found during the experiment that 0.001 gives the best result.
`batch_size`	Batch size for the model. Default value is set to 128. This model was tested on Google Colab. Since the free version couldn't accommodate more than 128 for batch_size, the config value of used.

Results/Inferences
Detailed analysis and execution can be followed on the following links:

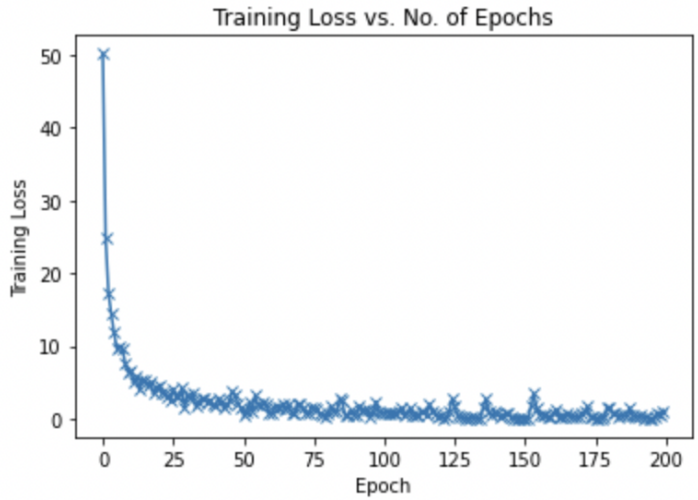
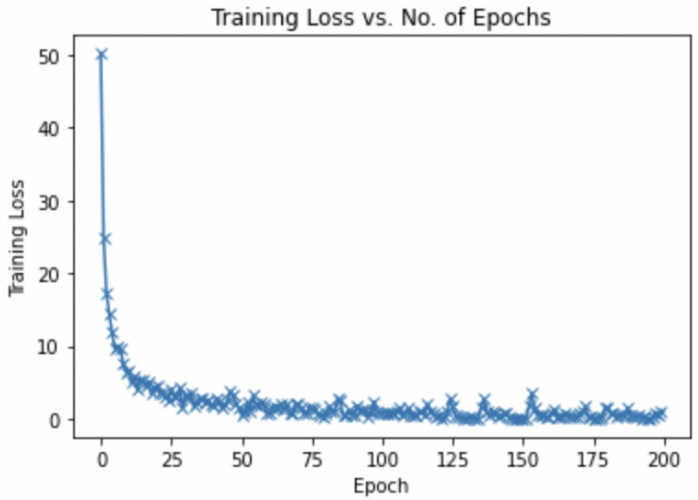
pretrained=True : https://www.kaggle.com/code/ankitverma8009/21cs60a04-ankitkumarverma-task2part1
pretraied=False:https://www.kaggle.com/code/ankitverma8009/21cs60a04-ankitkumarverma-task2part2

DataSet	**Accuracy (%) [pretrained=True]**
Validation Set	95.58
Test Set	95.53

DataSet	**Loss (%) [pretrained=False]**
Validation Set	91.95
Test Set	91.48

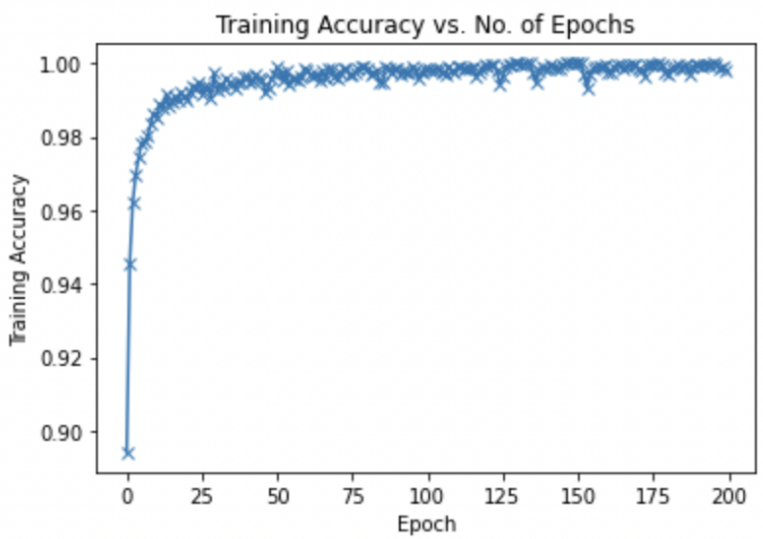
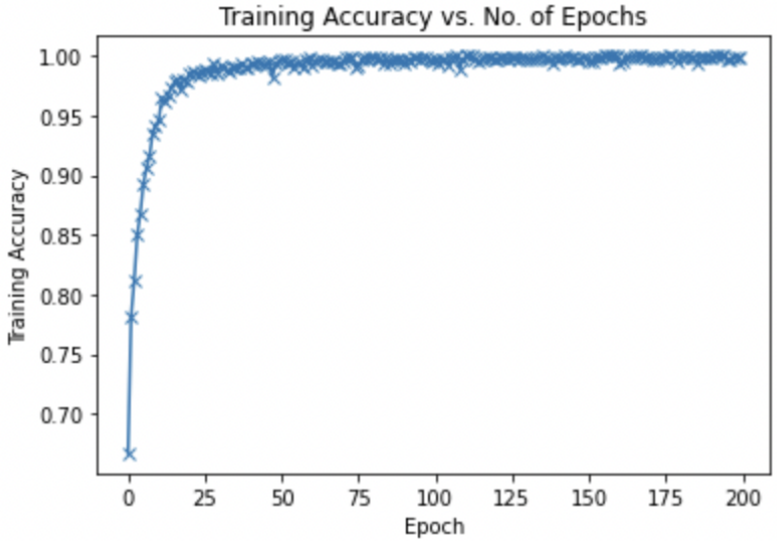
We can infer from that above data that ***model with pretrained set to True*** **performs better** than *model with pretrained set to False* on both te validation and test datasets.

Training Loss:

`pretrained=True`	`pretrained=False`
	

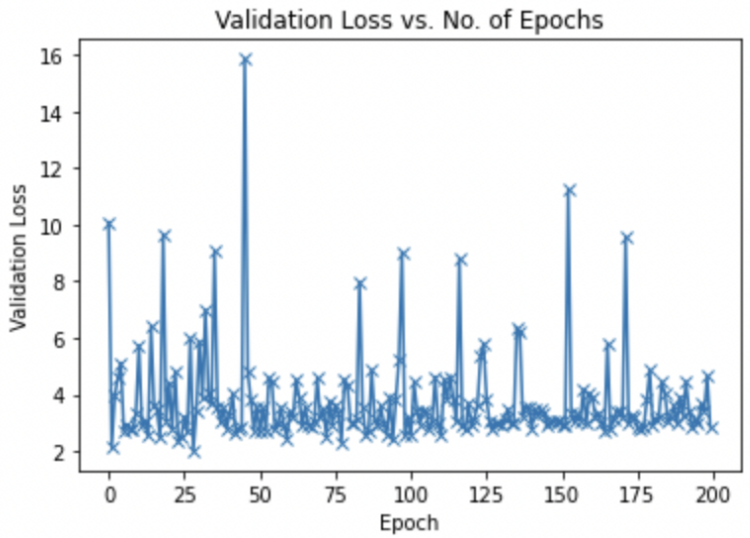
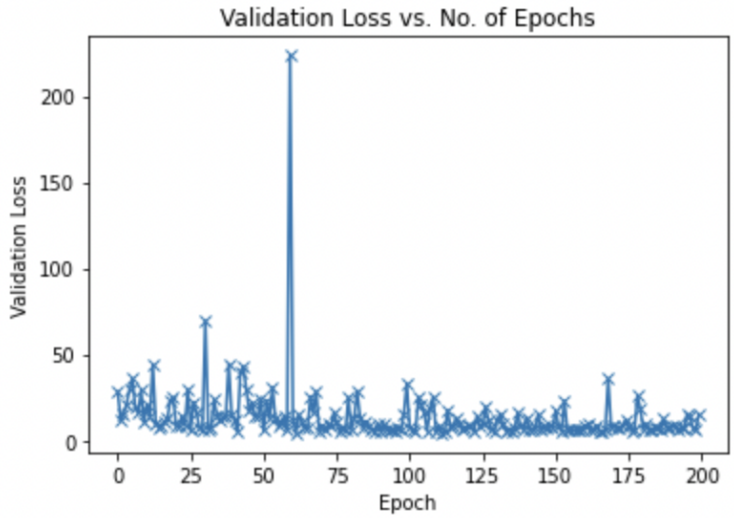
We can infer from that above graphs that both the models(with/without pretrained set to True) perform *similar* when a high epoch is considered. However, pretrained=True model outperforms pretrained=False by a margin of (~3%)

Training Accuracy:

`pretrained=True`	`pretrained=False`
	

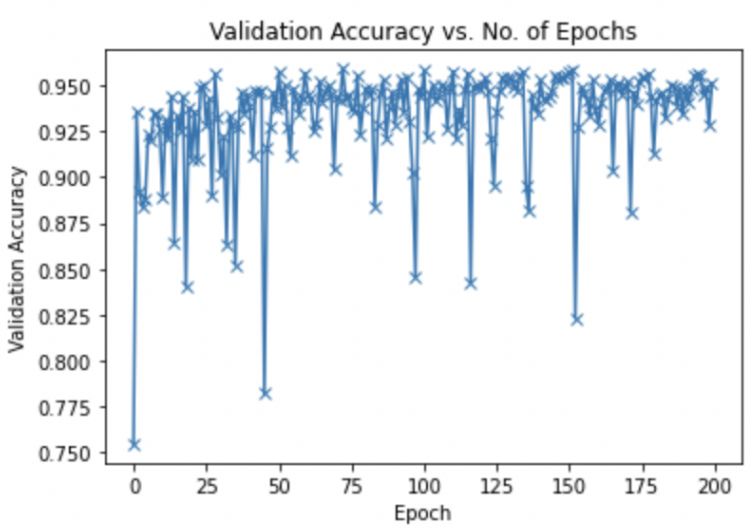
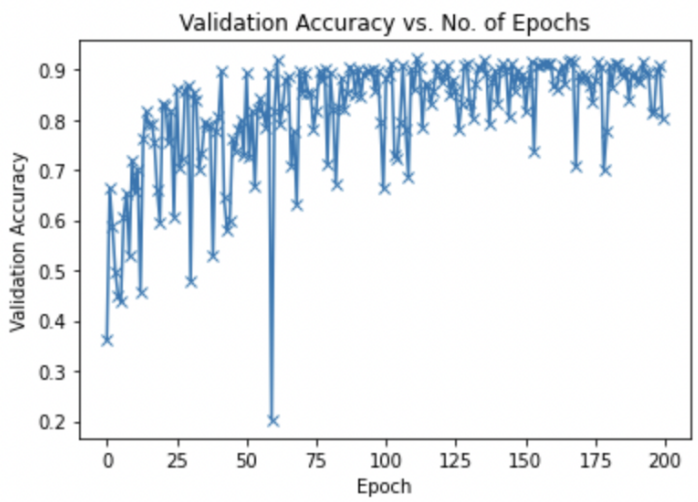
We can infer from that above graphs that both the models(with/without pretrained set to True) perform *similar* when a high epoch is considered. However, pretrained=True model outperforms pretrained=False by a margin of (~3%)

Validation Loss:

`pretrained=True`	`pretrained=False`
	

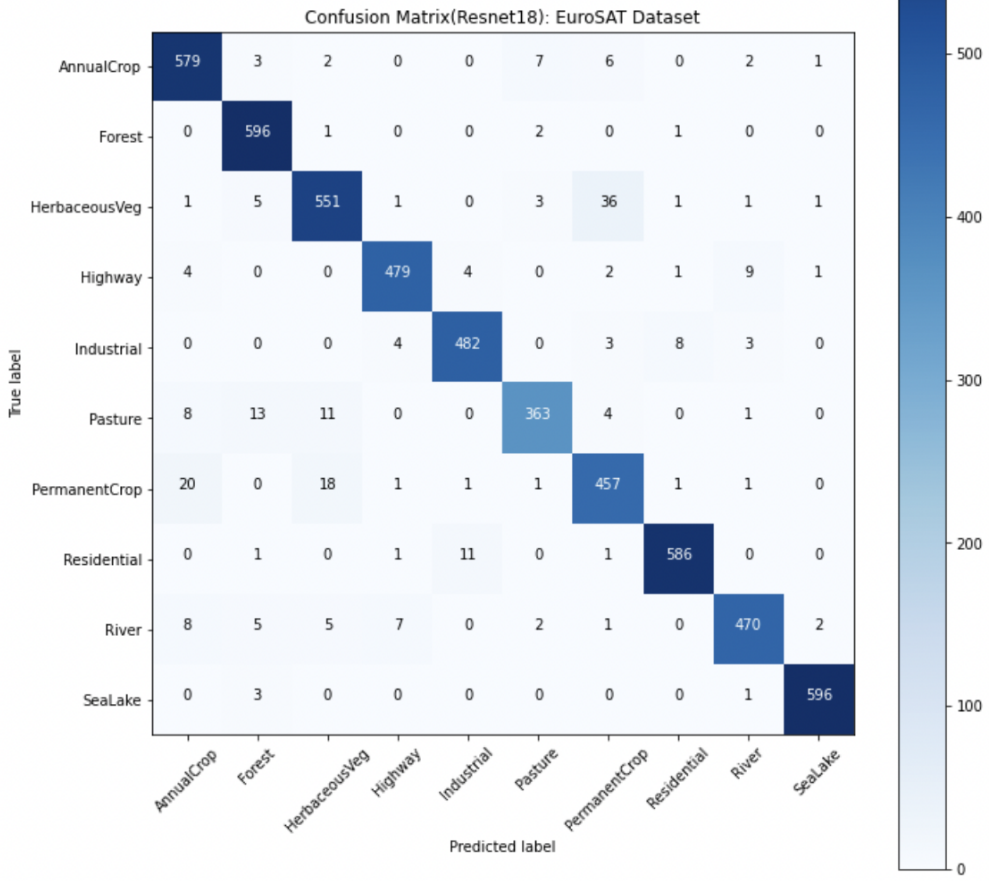
We can infer from that above graphs that both the models(with/without pretrained set to True) perform *similar* when a high epoch is considered. We also observe that there are more spikes in the pretrained=True graph than pretrained=Fasle.

Validation Accuracy:

`pretrained=True`	`pretrained=False`
	

We can infer from that above graphs that both the models(with/without pretrained set to True) perform *similar* when a high epoch is considered. We also observe that there are more spikes in the pretrained=True graph than pretrained=Fasle.

Confusion Matrix (pretrained=True)

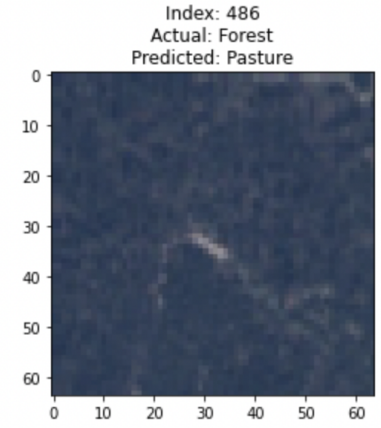


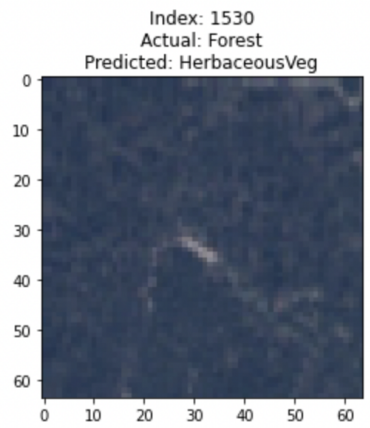
Class	Recall Value
AnnualCrop	0.965
Forest	0.9933333333333333
HerbaceousVeg	0.9183333333333333
Highway	0.958
Industrial	0.964
Pasture	0.9075
PermanentCrop	0.914
Residential	0.9766666666666667
River	0.94
SeaLake	0.9933333333333333

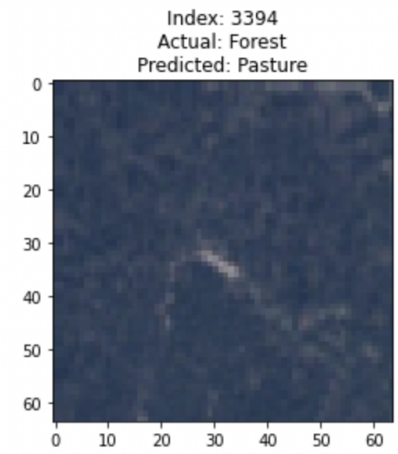
Model's most confident classes: [Forest, Sealake]

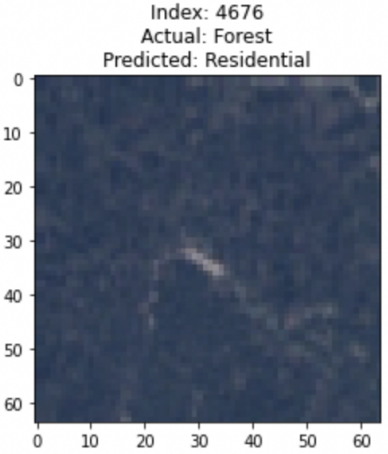
Model's least confident class: [Pasture]

Four examples the model got wrong and was most confident about. (pretrained = True)

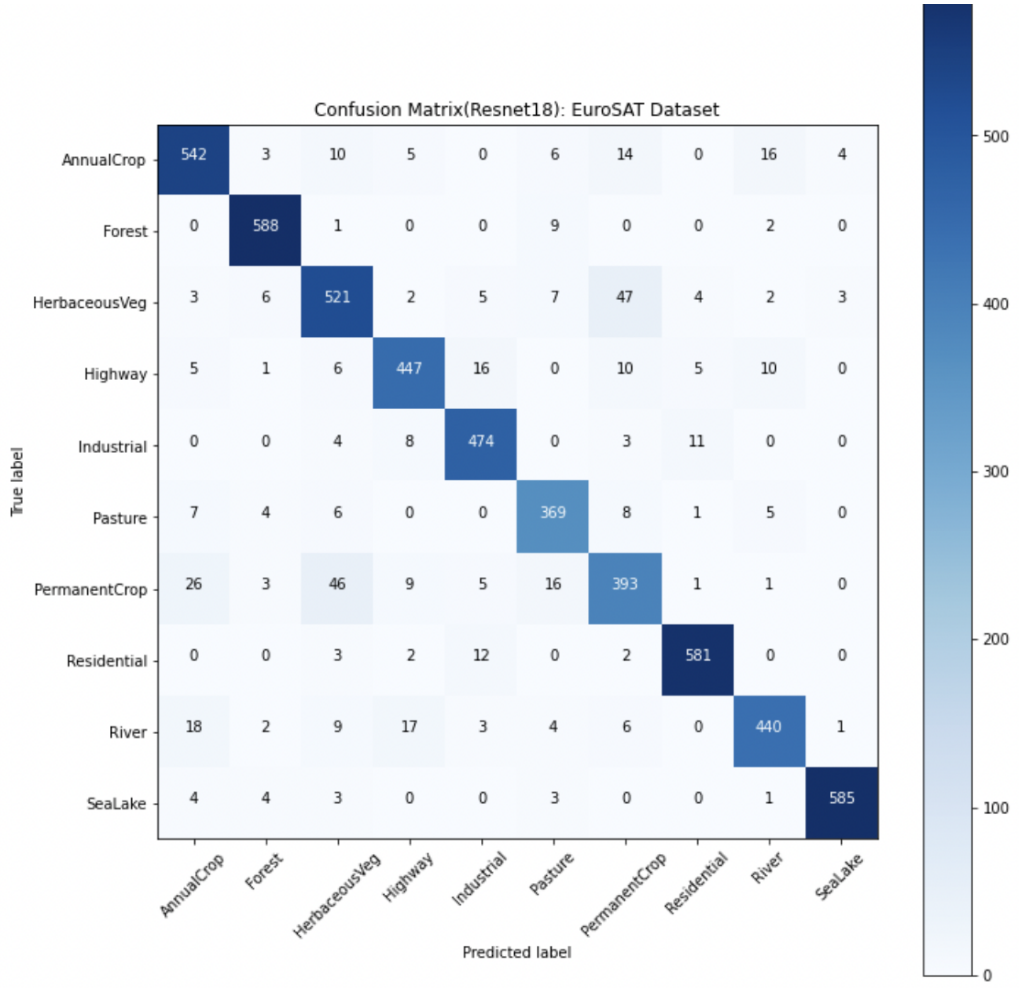








Confusion Matrix (pretrained=False):

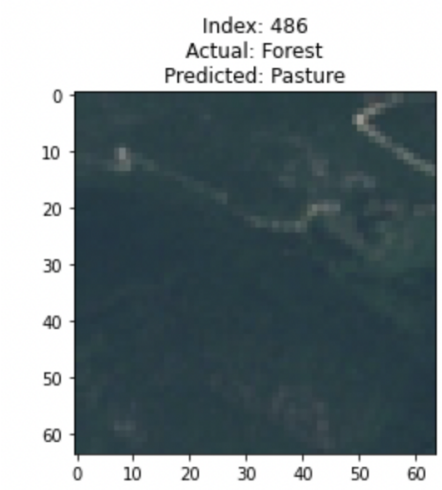


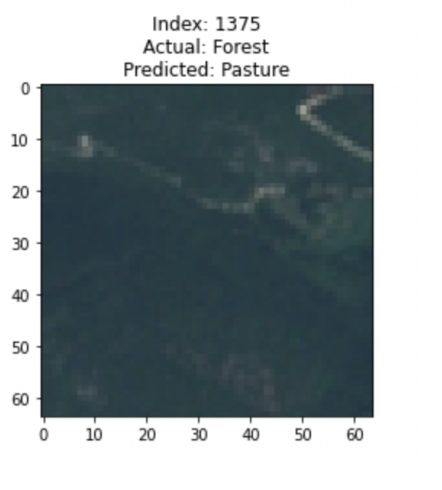
Class	Recall Value
AnnualCrop	0.9033333333333333
Forest	0.98
HerbaceousVeg	0.8683333333333333
Highway	0.894
Industrial	0.948
Pasture	0.9225
PermanentCrop	0.786
Residential	0.9683333333333334
River	0.88
SeaLake	0.975

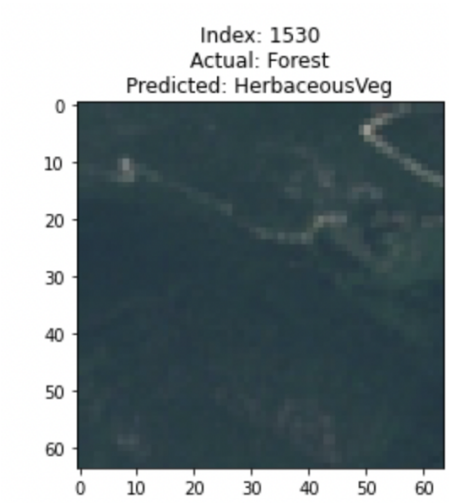
Model's most confident classes: [Forest]

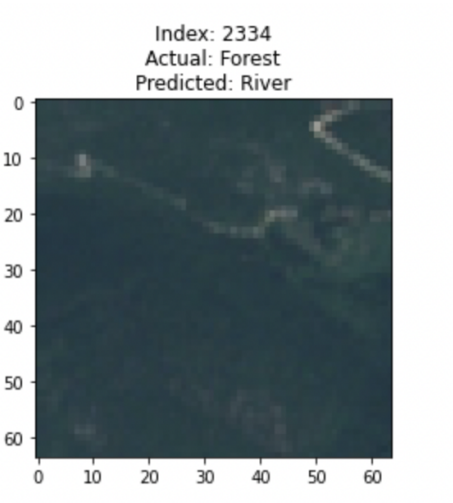
Model's least confident class: [Permanent Crop]

Four examples the model got wrong and was most confident about. (pretrained = False)









End of Report