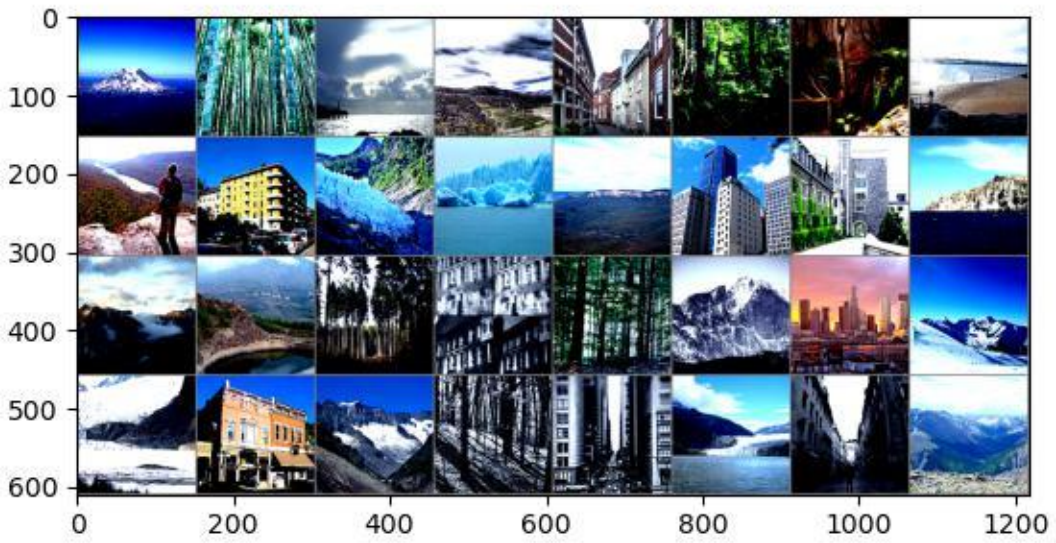
Experiment

The experiment used an implementation of Pytorch's Resnet18 architecture and pretrained weights, to train a model given a fixed dataset obtained from Kaggle. The dataset includes six classes: buildings, forest, glacier, mountain, sea, and street. The dataset included a train set count of 14034, with 3000 test samples. The optimizer was Adam with a learning rate of 0.001, trained at a total of 10 epochs. Two different networks, a fixed feature extractor, and finetune network were trained and tested with the start point of pretrained Resnet18 weights.

Figure 1. The below mosaic shows random samples of the dataset of input size 224, 224, with 3 channels.



Fixed Feature Extractor

The fixed feature extractor's layers were frozen except for the first and last layers, effectively making it a fixed network.

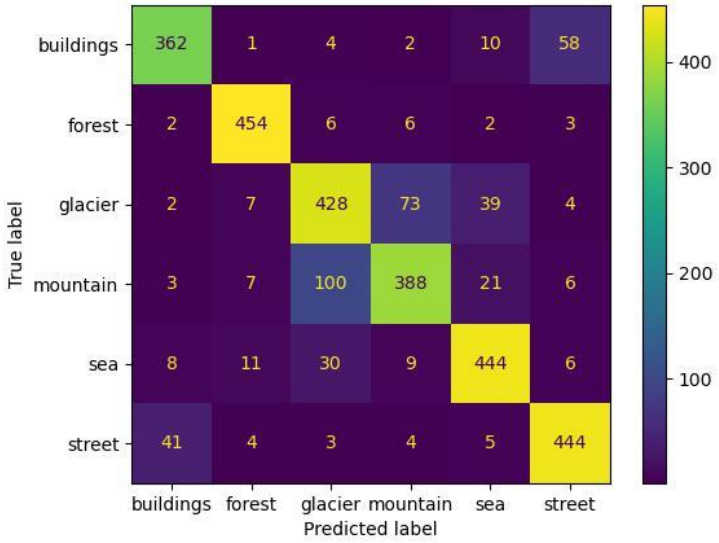
Finetune

The finetune network allowed all parameters to be updated during training.

Results

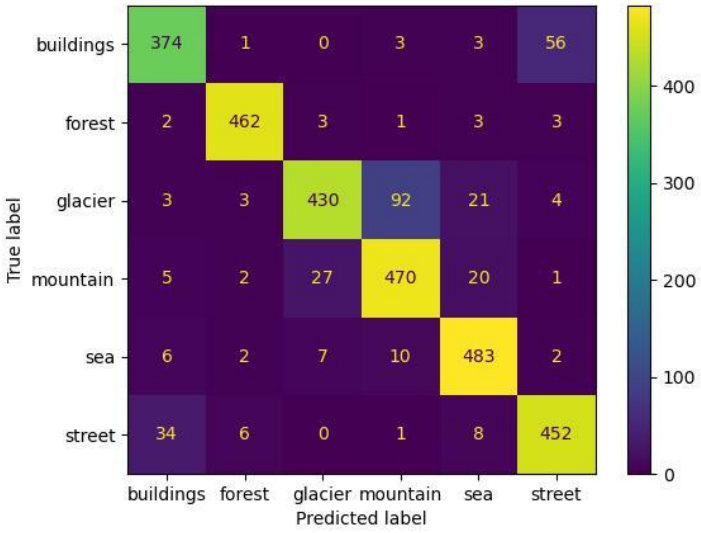
The fixed feature extraction network took a total of 402.1 seconds to train and test. It scored a normalized accuracy score of 0.84.

Figure 2. The confusion matrix of the fixed feature extractor shows good performance on all classes except for mountain.



The finetuned network took a total of 428.1 seconds to train and test. It scored a normalized accuracy score of 0.89.

Figure 3. The finetuned model's confusion matrix can be seen below.



Summary

The fixed feature extractor scored a higher accuracy score than I thought that it would. It also took much longer to train and test then I thought it would. Finetuning scored better accuracy on the test set. The confusion matrix also shows better test performance for every class in the dataset. This was a very interesting outcome.