Intel Image Classification with Transfer Learning

Abtin Mahyar

1. Data

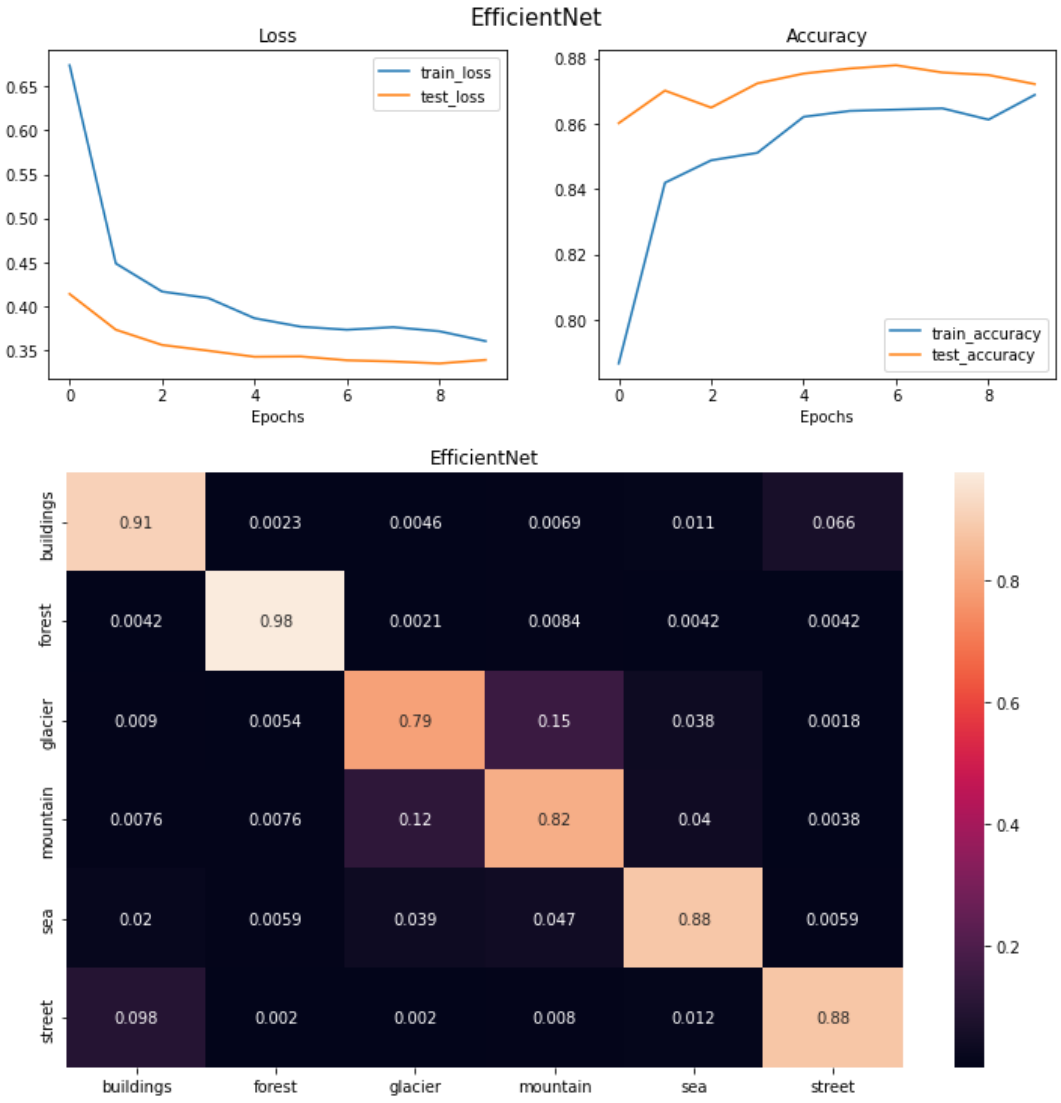
The process of aggregating and making the data ready for fitting into the models consists of following steps:

1. First of all, the dataset was downloaded from Kaggle. The data was available in three formats of train, test and prediction sets. Prediction set were dropped since images did not have label; therefore, no evaluation could not be performed on this set. There were 14,000 records in the training set, and 3,000 records in test set. There were 6 different classes as target labels for the records. The size of each input image is 150 * 150 pixels.
2. There was no need to augmentation because of size of the dataset, and the quality of dataset. Some transformations were used in order to convert raw inputs to Torch tensor for ease of computation in following works. Images were resized to desired input shape of backbone model in transfer learning process. Finally, images were normalized in order to stabilize gradient descent step.
3. 3,000 instances of training data were used as validation set for training phase.
4. Train, validation and test datasets were passed into their own data loader in order to make the data into batches of size 32. For training phase, I used shuffling for each epoch and test set was used for calculating performance of each model using different methods such as accuracy, F1-score, recall, precision and confusion matrix.

2. Experiments

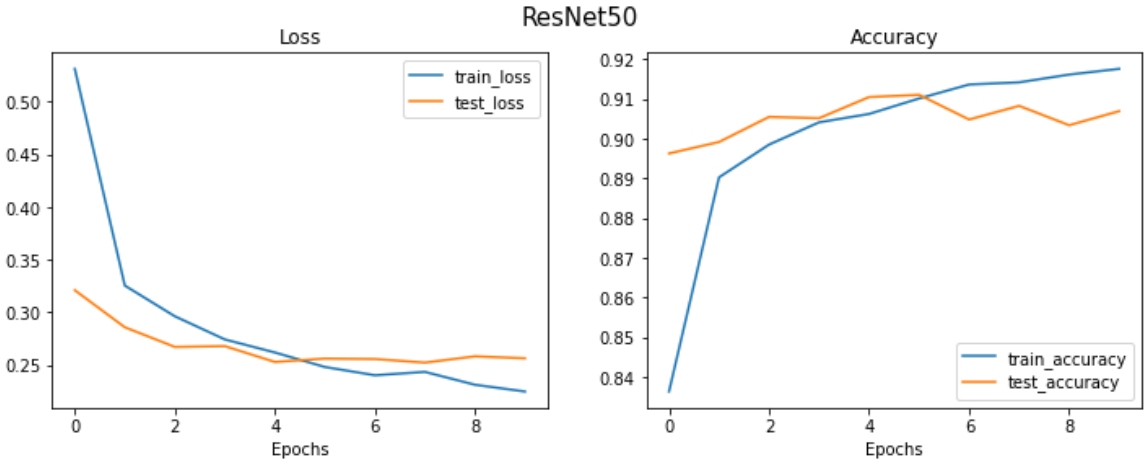
Lots of experiments was performed on the dataset using different architectures of pytorch pre-trained models which are going to discuss in the following section. I used the cross entropy as the loss function and Adam optimizer with learning rate set to 0.001 for models. All of the models were trained for 10 epochs.

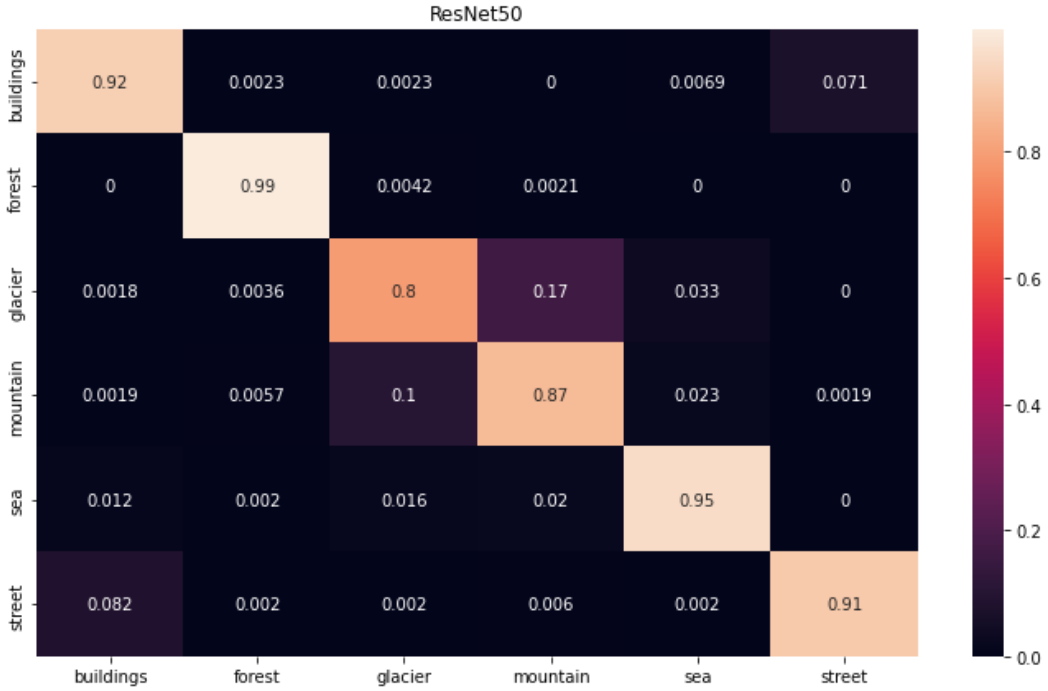
1. <u>EfficientNet</u>
   For experimenting this section, I used EfficientNet B0 with default weights which was trained with ImageNet. All of the parameters were freeze and their weights were consistent during training. Classifier block was changed in order to perform fine-tuning. This block has one dropout layer with probability 0.2 and a linear layer in order to perform the classification. The changes in loss and accuracy of the model on both train and test sets are illustrated in the following plot. As it can be seen from the plots, model extracts the important features from the image pretty well and achieve a good accuracy. Also, the model was started to overfit the training data at the last epochs. The model has the lowest accuracy on mountain and most likely misclassify these images with glacier.



2. <u>ResNet50</u>

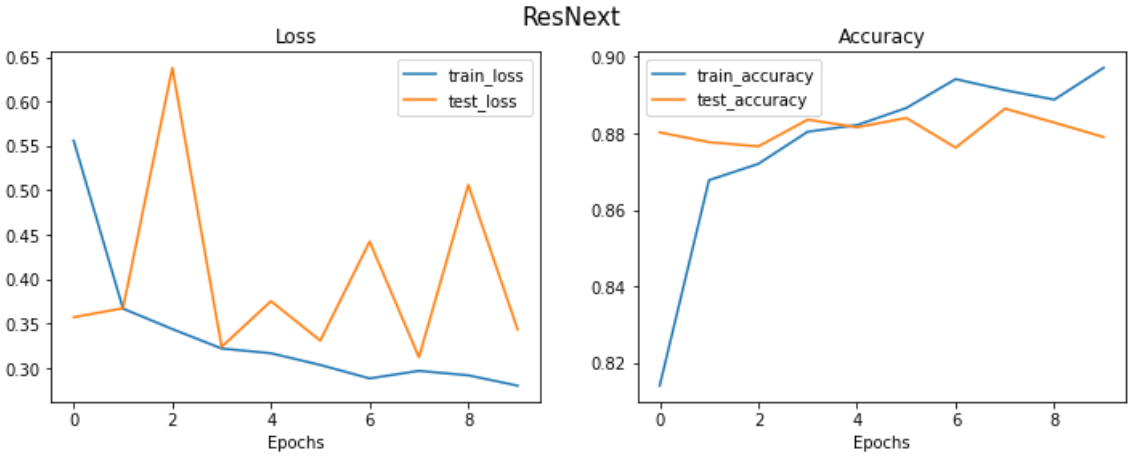
For experimenting this section, I used ResNet50 with default weights which was trained with ImageNet. All of the parameters were freeze and their weights were consistent during training. Classifier block was changed in order to perform fine-tuning. This block has one dropout layer with probability 0.2 and a linear layer in order to perform the classification. The changes in loss and accuracy of the model on both train and test sets are illustrated in the following plot. As it can be seen from the plots, model extracts the important features from the image pretty well and achieve a good accuracy. Also, there is a sign of overfitting to the training data which could be fixed by decreasing the learning rate. Same problem exits for this model too; model has the lowest accuracy on mountain and most likely misclassify these images with glacier. However, model has the higher accuracy than previous one.

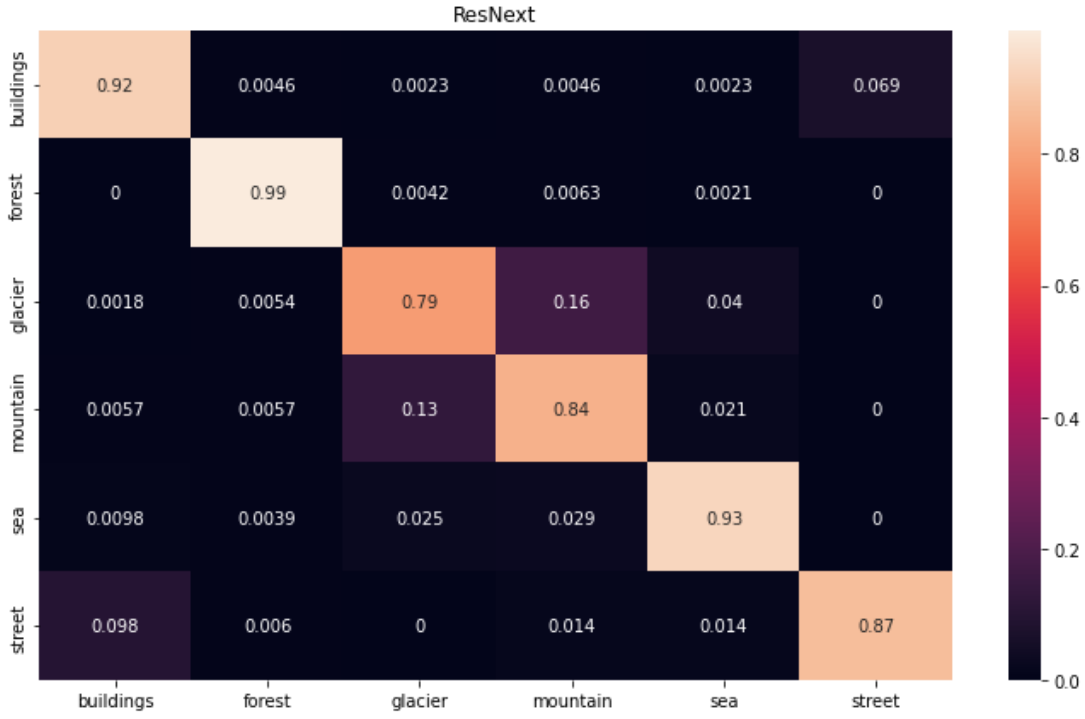




3. <u>ResNext</u>

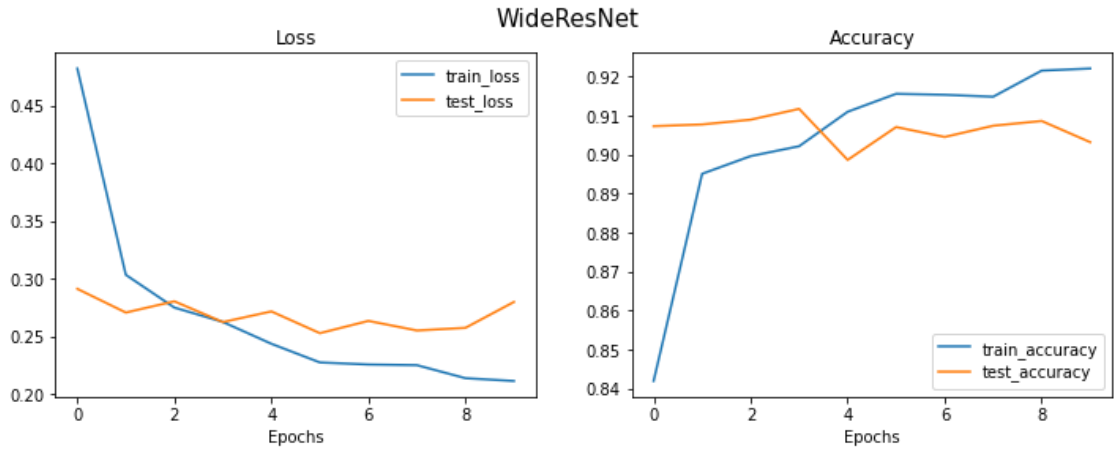
   For experimenting this section, I used ResNext50 with default weights which was trained with ImageNet. All of the parameters were freeze and their weights were consistent during training. Classifier block was changed in order to perform fine-tuning. This block has one dropout layer with probability 0.2 and a linear layer in order to perform the classification. The changes in loss and accuracy of the model on both train and test sets are illustrated in the following plot. As it can be seen from the plots, model does not perform as well as prior models. There is a notable sign of overfitting to the training data which could be tackled by a decrease in learning rate. Same problem exits for this model too; model has the lowest accuracy on mountain and most likely misclassify these images with glacier.

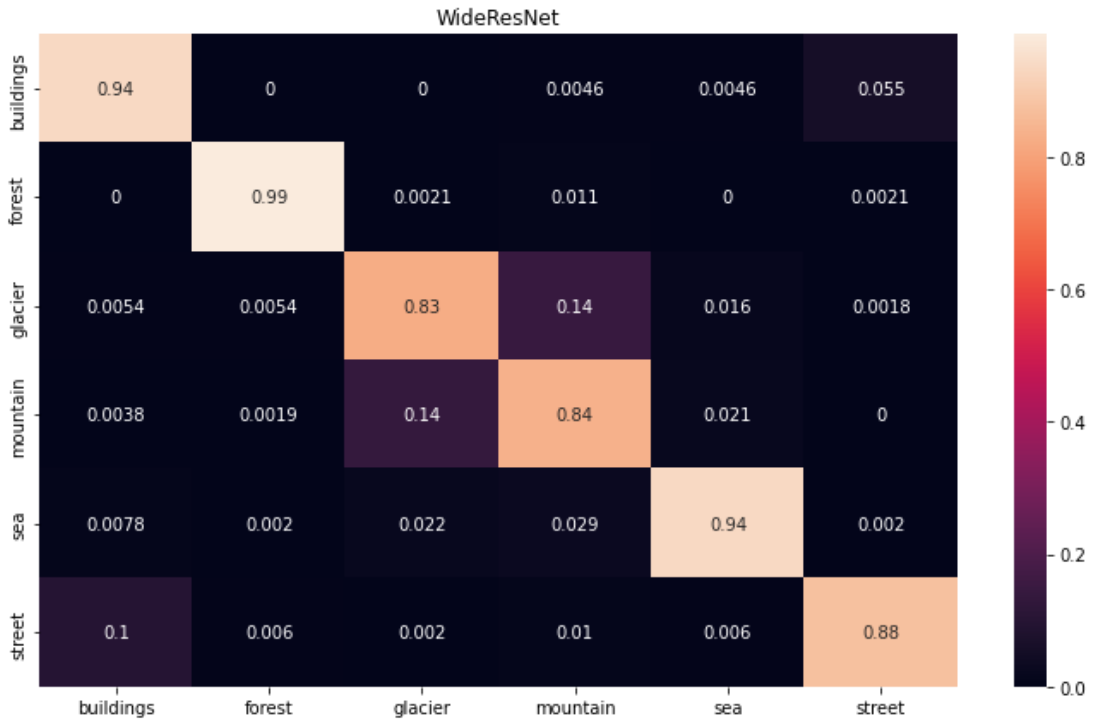




4. <u>WideResNet</u>

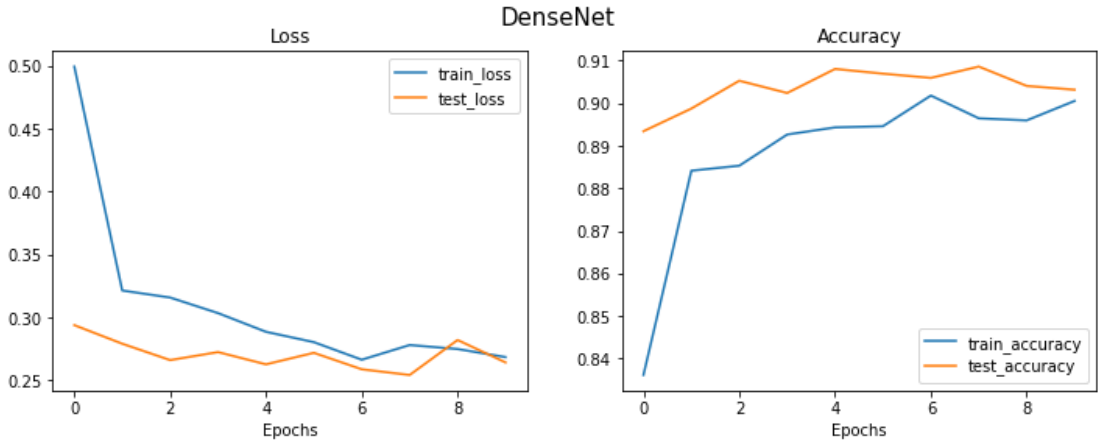
   For experimenting this section, I used Wide ResNet 50 with default weights which was trained with ImageNet. All of the parameters were freeze and their weights were consistent during training. Classifier block was changed in order to perform fine-tuning. This block has one dropout layer with probability 0.2 and a linear layer in order to perform the classification. The changes in loss and accuracy of the model on both train and test sets are illustrated in the following plot. As it can be seen from the plots, model does not perform as well as prior models on the test data but has good accuracy on training data. In other words, there is a notable sign of overfitting to the training data which could be tackled by a decrease in learning rate. Same problem exits for this model too; model has the lowest accuracy on mountain and most likely misclassify these images with glacier.

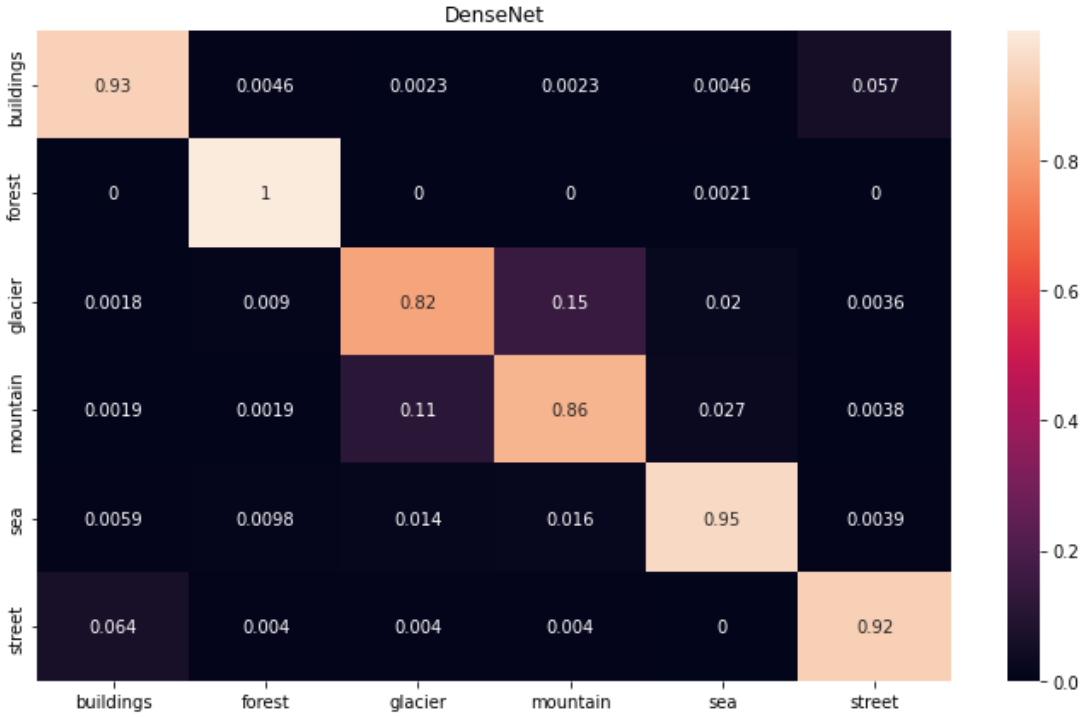




5. <u>DenseNet</u>

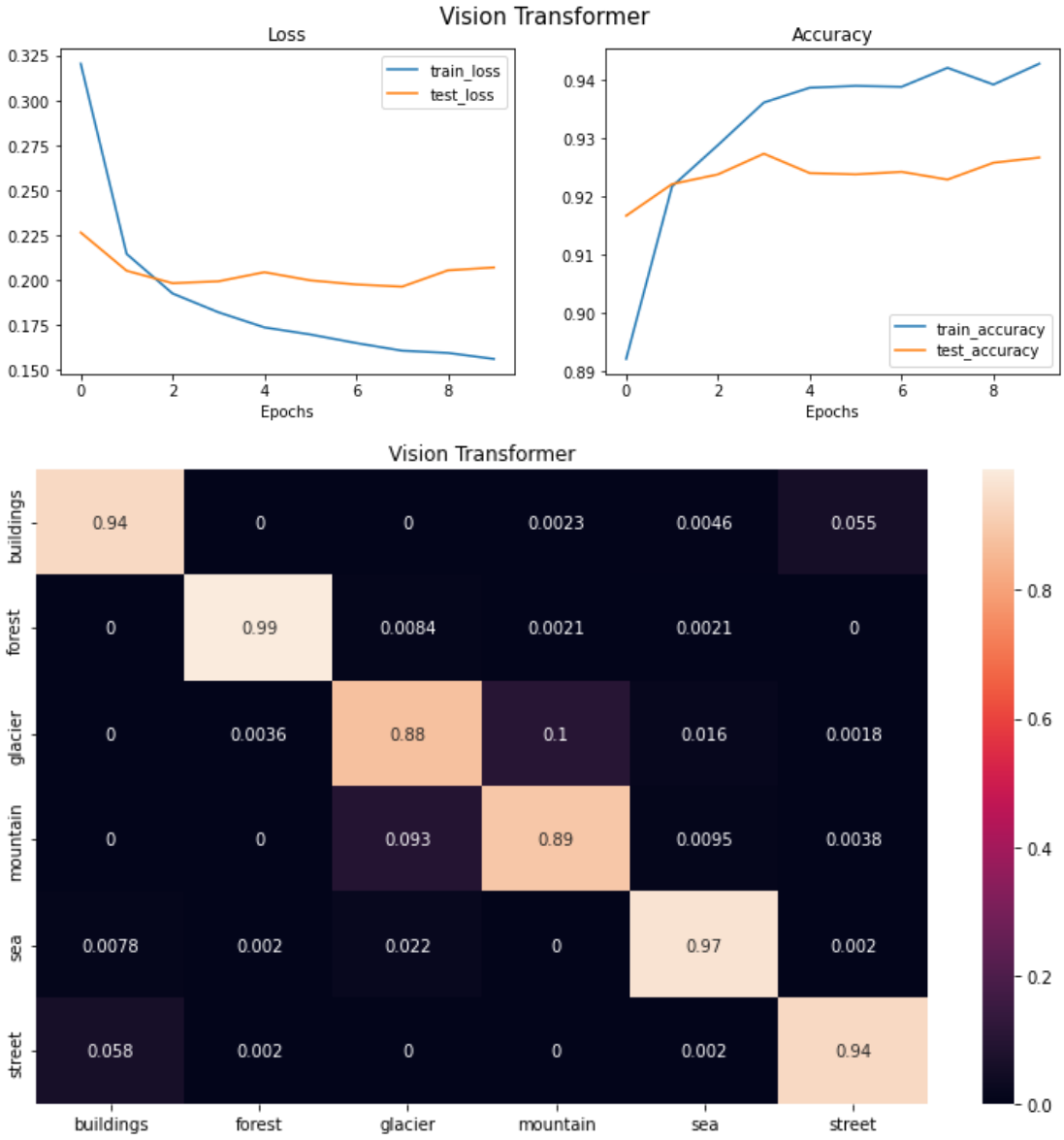
For experimenting this section, I used DenseNet 161 with default weights which was trained with ImageNet. All of the parameters were freeze and their weights were consistent during training. Classifier block was changed in order to perform fine-tuning. This block has one dropout layer with probability 0.2 and a linear layer in order to perform the classification. The changes in loss and accuracy of the model on both train and test sets are illustrated in the following plot. As it can be seen from the plots, model extracts the important features from the image pretty well and achieve a good accuracy. Also, the model was started to overfit the training data at the last epochs. The model has the lowest accuracy on mountain and most likely misclassify these images with glacier.





6. <u>Vision Transformer</u>

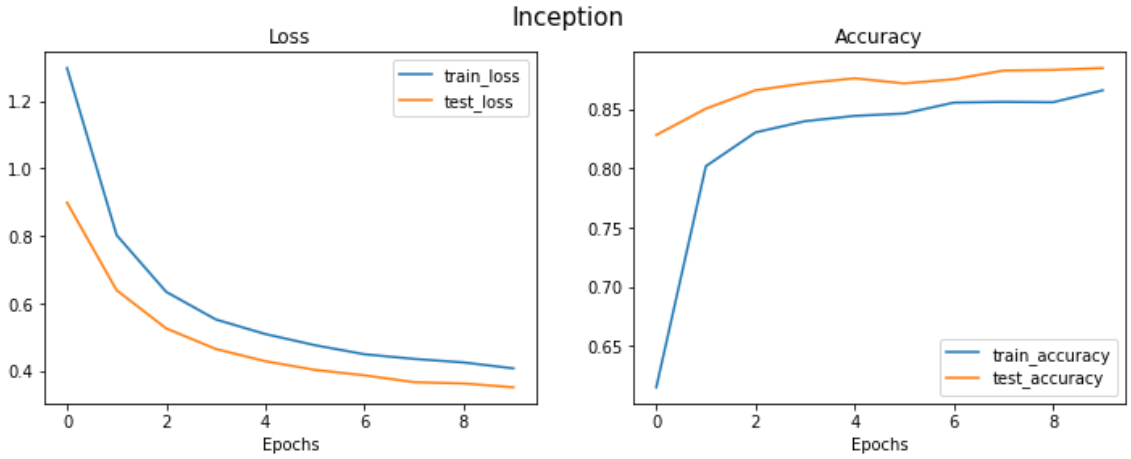
For experimenting this section, I used VisionTransformer-B-32 with default weights which was trained with ImageNet. Before fitting the data into this model, images should be resized to (224, 224). All of the parameters were freeze and their weights were consistent during training. Classifier block was changed in order to perform fine-tuning. This block has one dropout layer with probability 0.2 and a linear layer in order to perform the classification. The changes in loss and accuracy of the model on both train and test sets are illustrated in the following plot. As it can be seen from the plots, model extracts the most important features from the image pretty well and achieve a best accuracy among the other networks. Also, prior problem which was misclassifying mountain and glacier does not exist with the outputs of this model. With this observation, we can conclude that ViT can extract the most valuable features which are suitable for the classification of this dataset compare to the other models.

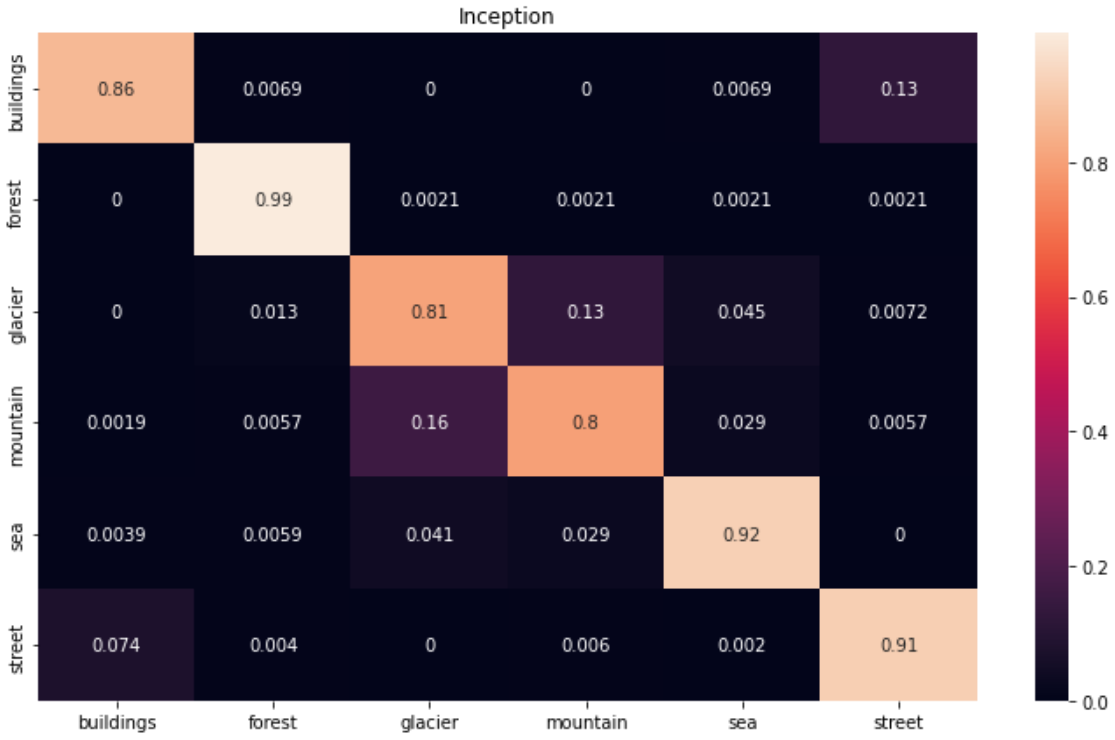


7. Inception

For experimenting this section, I used InceptionV3 with default weights which was trained with ImageNet. Before fitting the data into this model, images should be resized to (299, 299) and learning rate decreased to 0.0001. All of the parameters were freeze and their weights were consistent during training. Classifier block was changed in order to perform fine-tuning. This block has one dropout layer with probability 0.2 and a linear layer in order to perform the classification. The changes in loss and accuracy of the model on both train and test sets are illustrated in the following plot. As it can be seen from the plots, model does not perform as well as other models and could not extract the most important features from the input images in compare to other models. However, there is no significant sign of overfitting; whilst, all some of other models have overfitted to the training data. Same

problem exits for this model too; model has the lowest accuracy on mountain and most likely misclassify these images with glacier.

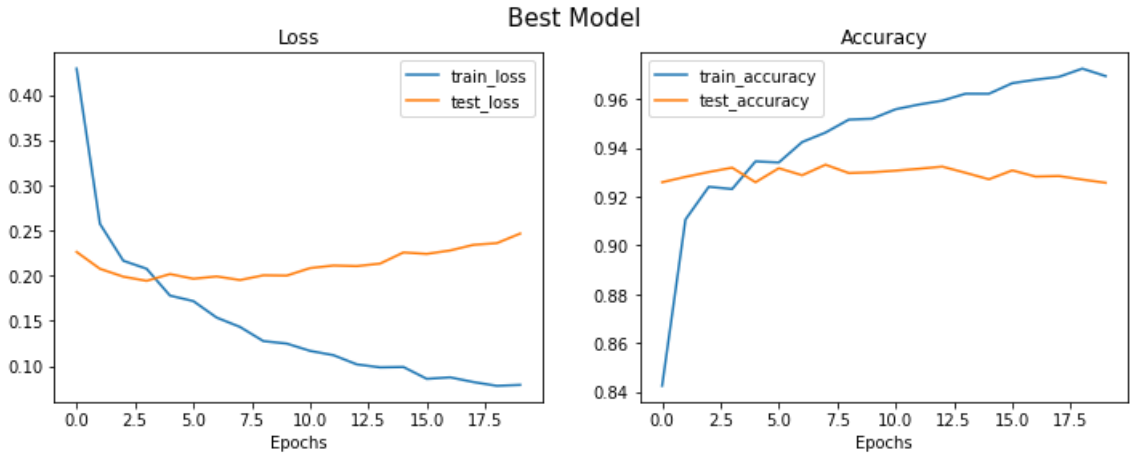


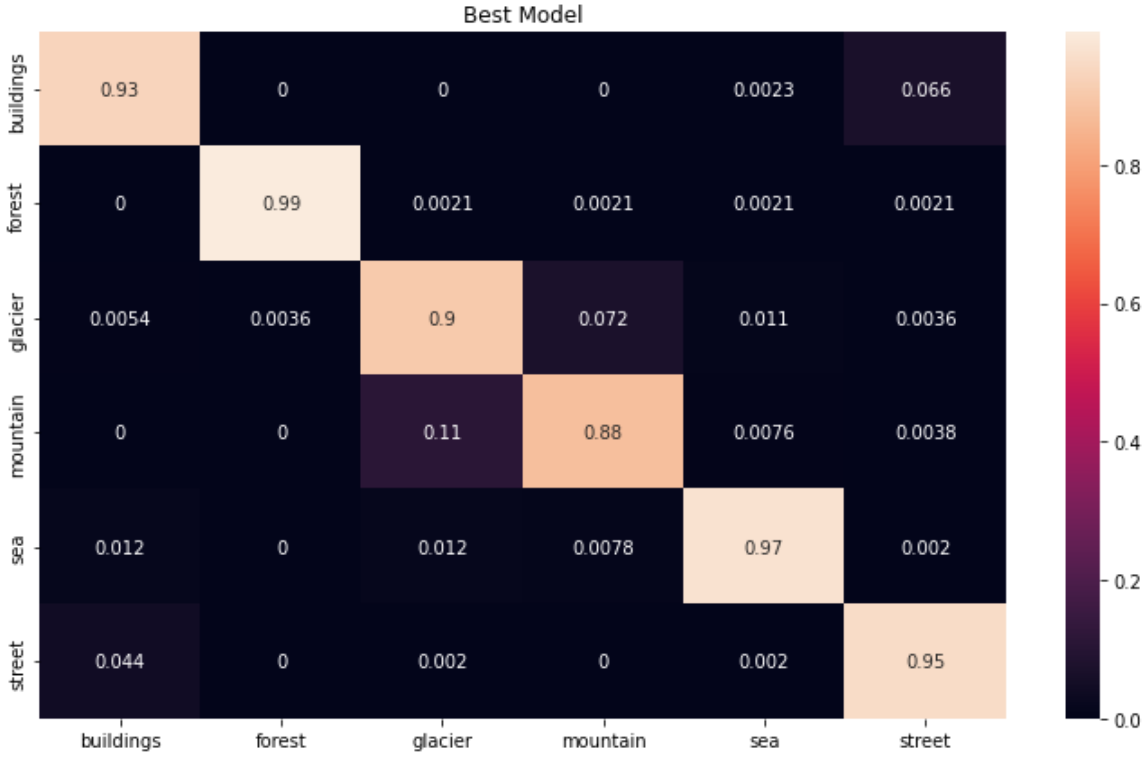


8. <u>Best Model</u>

Since ViT has the best performance among the other models that were mentioned in this report, I used VisionTransformer-B-32 with default weights which was trained with ImageNet and change its classifier and made it a bit deeper in order to use features extracted from this model as best as possible. Before fitting the data into this model, images should be resized to (224, 224), learning rate decreased to 0.0001 and model was trained for 20 epochs. All of the parameters were freeze and their weights were consistent during training. Classifier block was changed in order to perform fine-tuning. This block has 3 linear layers with respectively 512, 1024, and 6 neurons which is used for taking the calculated representation from previous blocks and doing the classification. There are also batch

normalization and dropout layers in this block in order to prevent overfitting and increase generalization. The changes in loss and accuracy of the model on both train and test sets are illustrated in the following plot. As it can be seen from the plots, model extracts the most important features from the image pretty well and achieve a best accuracy among the other networks. Also, prior problem which was misclassifying mountain and glacier does not exist with the outputs of this model. With this observation, we can conclude that ViT can extract the most valuable features which are suitable for the classification of this dataset compare to the other models. There is a notable sign of overfitting to the training data.





3. Results

Results of above experiments are aggregated in the following table and sorted based on the accuracy of the model on test data. I used different metrics such as f1-score, precision, recall for comparing the models with each other. Since the task was multi-class classification, and number of different records in each class was equal, I used macro averaging for calculating these scores.

	Model Name	accuracy	f1 score	precision	recall
7	Best Model	0.936059	0.937351	0.937566	0.937321
5	Vision Transformer	0.931848	0.933293	0.933158	0.933452
4	DenseNet	0.909464	0.911508	0.911008	0.912447
1	ResNet	0.901817	0.904158	0.904232	0.905131
3	WideResNet	0.896166	0.899042	0.899442	0.899542
2	ResNext	0.885195	0.887988	0.888307	0.888837
6	Inception	0.879765	0.881762	0.881935	0.882028
0	EfficientNet	0.873005	0.875718	0.875811	0.876475

As it can be concluded from the above table, the best model which uses ViT architecture as the feature extractor and a 3-layer classifier, has the best performance on the test data compare to other models.