Computer Vision
Assignment-III

Use Of Transfer Learning to Classify Nature Images Through ResNet50 CNN

ABRAR ANWAR QURESHI

[317713]

***Abstract*---In this assignment a convolutional neural network(CNN) is trained, that is then used to classify a dataset of images from nature. This assignment will use the technique of transfer learning to classify six classes of images. The CNN used will be ResNet50 trained on ImageNet.**

I. INTRODUCTION

CNNs have started to gain lots of importance in the last few years owing to the efficiency with which they handle image processing tasks. Although the building blocks from which CNNs are made are pretty basic but the way that those building blocks are placed and used allows the CNN network to work with great speed and extract more useful information from an image. Every layer of a CNN extracts different levels of information from the image. Shallow layers extract extract more basic information such as corners, edges, etc. Whereas more deeper layers extract high level information such as windows, tires, faces, etc. This way CNNs have been very successful in image classification and object detection.

One of the most efficient and accurate CNN networks is the ResNet50. It is the winner of the 2015 ImageNet Large Scale Visual Recognition Challenge (ILSVRC). It consists of 152 layers of convolution, activation, pooling and fully connected layers. This is a very deep network which allows it to extract more information from an image and thus work better with less errors. However a common issue with considerably deep neural networks is that the gradient needed to optimize the weights of the network starts to vanish after some layers. To resolve this issue ResNet50 employs a very unique and straight forward approach. It injects the gradient from a previous layer into a layer ahead. This connection is known as the residual connection hence giving the CNN the name of 'Residual Network 50(ResNet50)'. This way the gradient is preserved and the weights can be updated with ease. For our purpose we employ the ResNet50 to classify images from nature that are divided into six classes.

II. METHODOLOGY

In order to use transfer learning technique we need a pre-trained model, whose weights are already optimized for image classification, and all we need to do is to train the last few fully connected layers.

Figure 1: Model Of ResNet50 with image of a residual block

We load the ResNet50 model with weights pre-trained from the ImageNet classification. However we do not load the upper layer of the model with the fully connected layers and instead create our own for classifying 6 classes. We add a flattening layer with a an ReLu activation layer and finally a fully connected softmax layer for 6 outputs corresponding to the 6 classes. Next we only make the last three layers we added to be trainable and then train the model.

For better generalization we also augment the images with shear, zooming, rotation, width shift, height shift and also horizontal flipping. This ensures better accuracy and performance. The loss function used is the categorical cross-entropy and the optimizer as the stochastic gradient descent. The training settings used are described in the adjoining table.

Table 1: Model and Training Parameters

Model Information	
Layers	152
Total Parameters	76,023,686
Trainable Parameters	52,440,070
Non-Trainable Parameters	23,583,616
Image Augmentation	
Rotation Range	0 to 90
Width Shift Range	0 to 0.2
Height Shift Range	0 to 0.2
Shear Range	0 to 0.2
Zoom Range	0 to 0.2
Horizontal Flip	True
Model Compilation	
Loss Method	Categorical Cross-entropy
Optimizer	Stochastic Gradient Descent(SGD)
Gradient Descent	
Learning Rate	0.001
Decay	1e-7
Momentum	0.9
Training Settings	
Epochs	8
Batch Size	64

III. RESULTS

The results of the training and the classification are described by the diagrams and tables below:

Computer Vision
Assignment-III

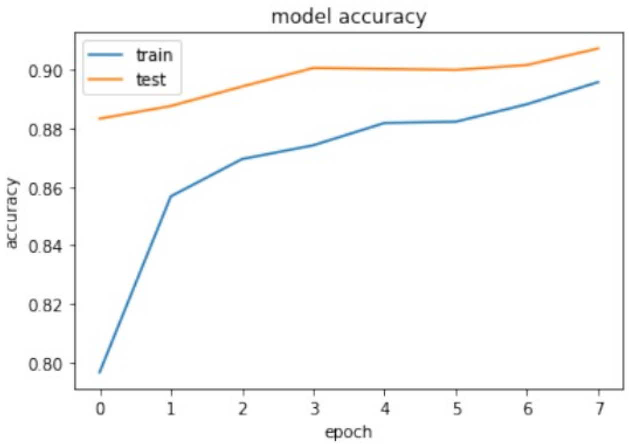

Figure 3: Model Accuracy History

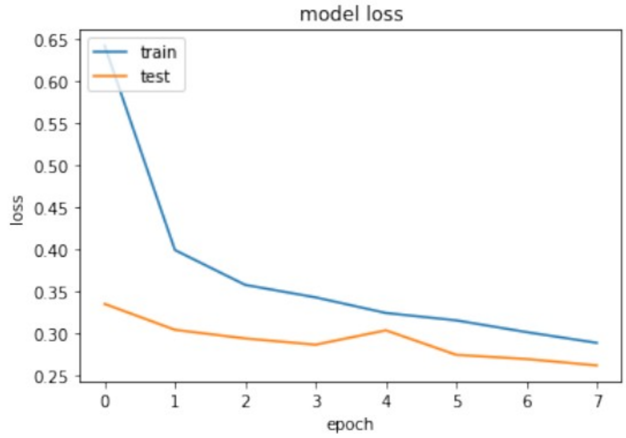

Figure 2: Model Loss History

```
Confusion Matrix:
[[1006    4    7    2   12  113]
 [   3 1144    3    7    4    5]
 [   3    7 1020  242   53    5]
 [   5    7   73 1140   70    2]
 [  11    7   35   73  993    9]
 [  67   10    9    4    9 1137]]
```

Figure 6: Confusion Matrix of Classification

Classification Report:

	precision	recall	f1-score	support
Buildings	0.92	0.88	0.90	1144
Forest	0.97	0.98	0.98	1166
Glacier	0.89	0.77	0.82	1330
Mountain	0.78	0.88	0.82	1297
Sea	0.87	0.88	0.88	1128
Street	0.89	0.92	0.91	1236
accuracy			0.88	7301
macro avg	0.89	0.88	0.88	7301
weighted avg	0.88	0.88	0.88	7301

For a visual result we choose 6 images all belonging to one class and then used our trained model to classify them into the 6 classes. The images we chose are:

Figure 4: Glacier Class

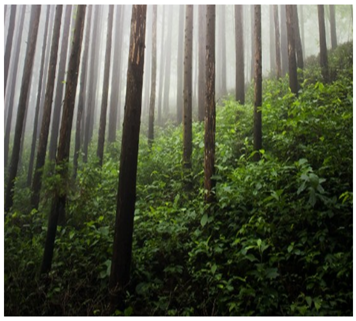

Figure 7: Building Class

Figure 8: Sea Class

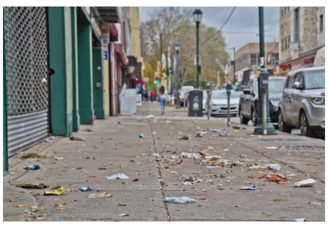

Figure 10: Street Class

Figure 9: Mountain Class

According to the following confusion matrix we see that all the images are classified correctly except for the sea image that is classified as a mountain class.

```
Confusion Matrix:
[[1 0 0 0 0 0]
 [0 1 0 0 0 0]
 [0 0 1 0 0 0]
 [0 0 0 1 0 0]
 [0 0 0 1 0 0]
 [0 0 0 0 0 1]]
```

Figure 12: Confusion Matrix for Visual Results

IV. CONCLUSION

We see that our approach of transfer learning has created very good results and the images have been classified with 89% accuracy. Now however if we slightly adjust the training settings we receive very different results. By increasing the learning rate our accuracy drops and we also get over-fitting. Also by increasing the number of epochs we also get some over-fitting. It is also observed that if we remove image augmentation then results again suffer. Naturally if we tuned more layers we have been able get to get better results for our case. The approach of image augmentation also helped our case as without it we would not have been able to get such high accuracy results.

V. GITHUB REPOSITORY LINK

All the code and weight files along with the images used for my visual results are available on the following GitHub repository link:

Abrar Qureshi GitHub Repository