Results-Homework 4

Name: Anurag Kalapala

Student ID: 02080192

Optimizer Performance:

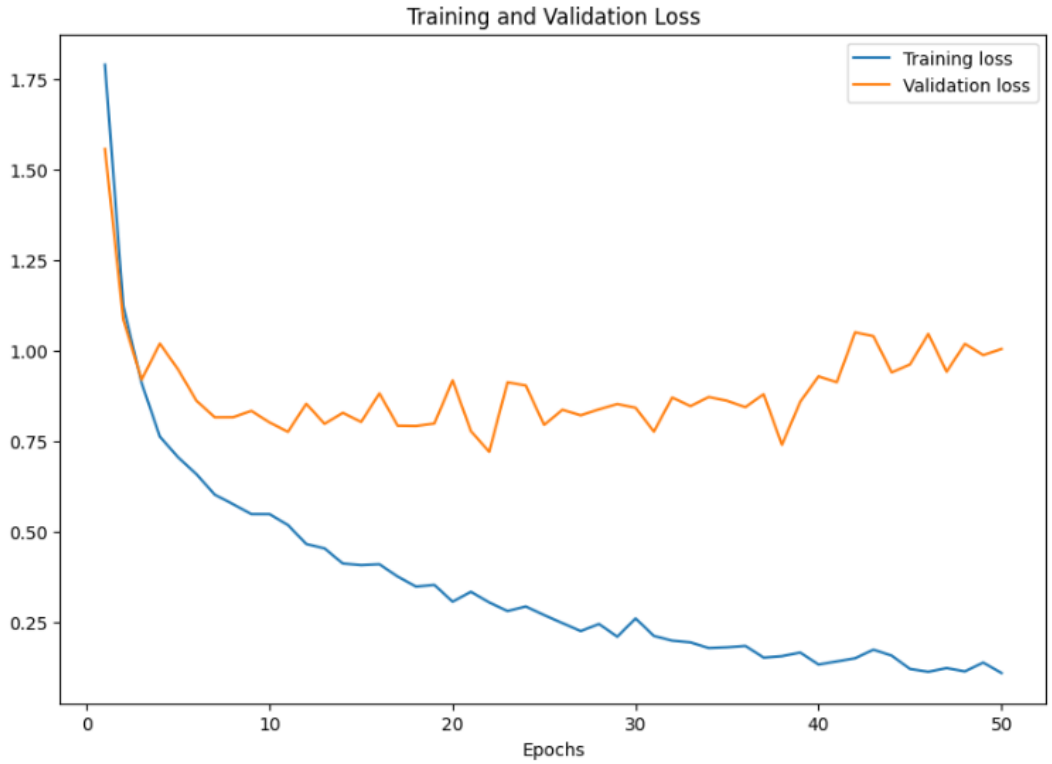
```
ResNet(
  (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)
  (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu): ReLU(inplace=True)
  (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
  (layer1): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    (1): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
```

These are some helper functions to evaluate the training process.

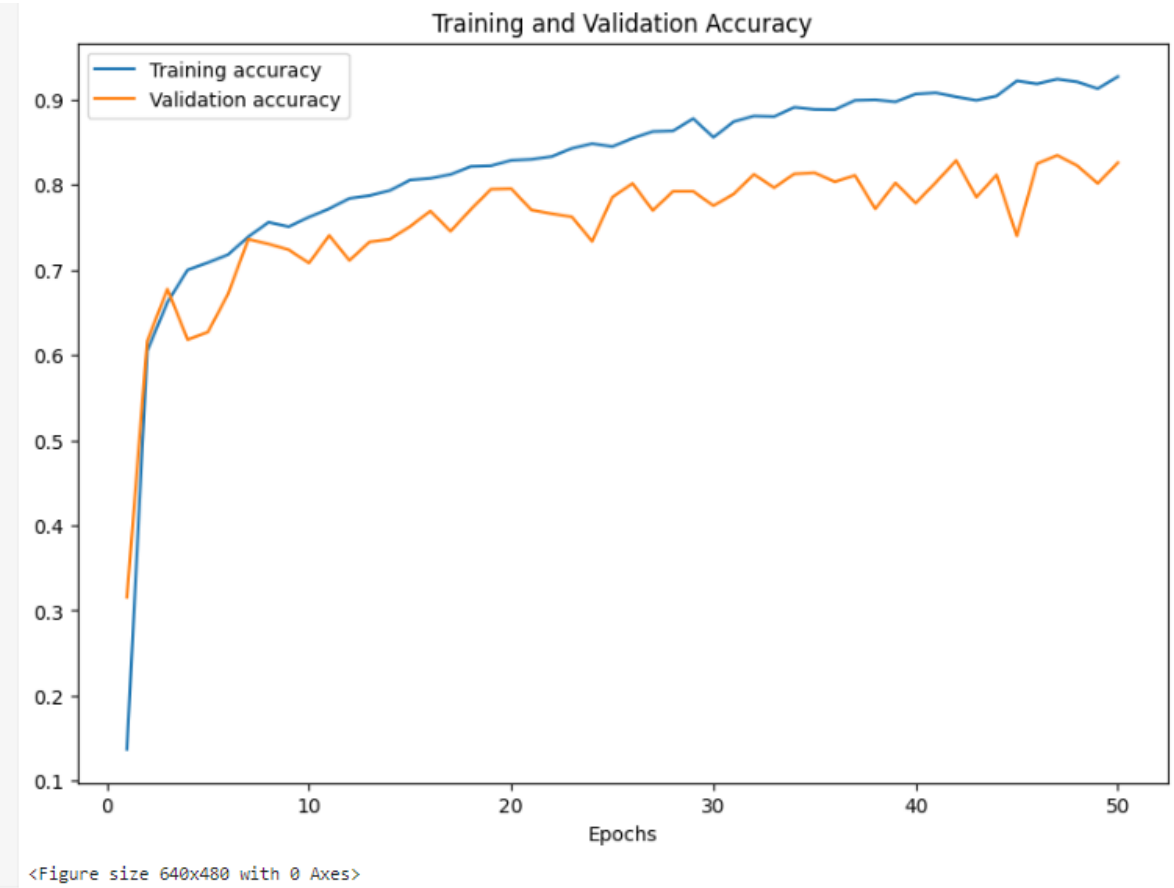
Training and validation of loss and accuracy:

```
Epoch: 40
Loss: 0.132  Accuracy:0.907
Validation Loss: 0.929  Val Accuracy: 0.779
Epoch: 41
Loss: 0.140  Accuracy:0.908
Validation Loss: 0.913  Val Accuracy: 0.803
Epoch: 42
Loss: 0.149  Accuracy:0.903
Validation Loss: 1.051  Val Accuracy: 0.828
Epoch: 43
Loss: 0.173  Accuracy:0.899
Validation Loss: 1.040  Val Accuracy: 0.785
Epoch: 44
Loss: 0.156  Accuracy:0.904
Validation Loss: 0.940  Val Accuracy: 0.812
Epoch: 45
Loss: 0.120  Accuracy:0.922
Validation Loss: 0.962  Val Accuracy: 0.740
Epoch: 46
Loss: 0.112  Accuracy:0.919
Validation Loss: 1.047  Val Accuracy: 0.825
Epoch: 47
Loss: 0.122  Accuracy:0.924
Validation Loss: 0.942  Val Accuracy: 0.835
Epoch: 48
Loss: 0.113  Accuracy:0.921
Validation Loss: 1.019  Val Accuracy: 0.822
Epoch: 49
Loss: 0.137  Accuracy:0.913
Validation Loss: 0.988  Val Accuracy: 0.802
Epoch: 50
Loss: 0.108  Accuracy:0.927
Validation Loss: 1.005  Val Accuracy: 0.826
Finished Training
```

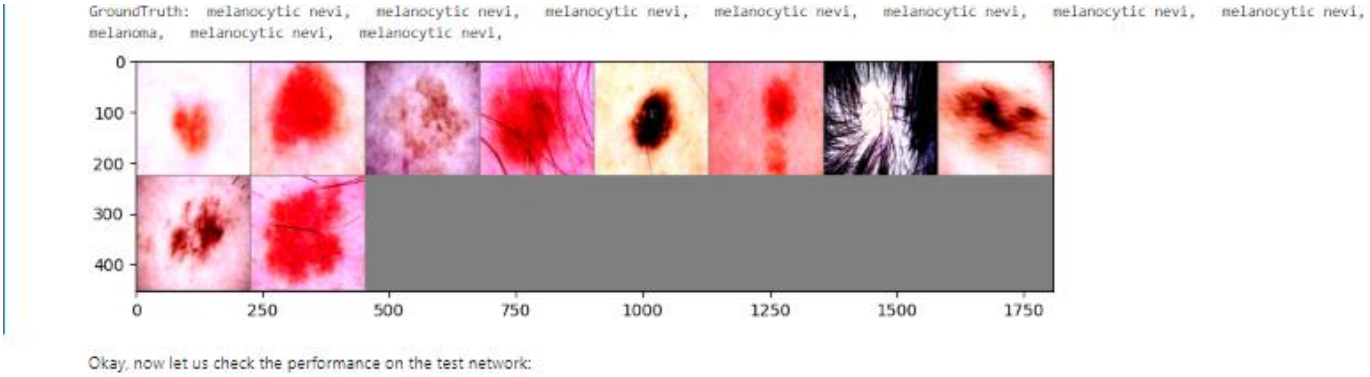
Training and validation of Loss plots:



Training and validation Accuracy:



Displaying Images from test set:



Okay, now let us check the performance on the test network:

Accuracy on test images:

```
Accuracy of the network on the test images: 83 %
```

Accuracy of all classes:

```
Accuracy of actinic keratoses : 66 %
Accuracy of basal cell carcinoma : 84 %
Accuracy of benign keratosis-like lesions : 60 %
Accuracy of dermatofibroma : 71 %
Accuracy of melanoma : 54 %
Accuracy of melanocytic nevi : 91 %
Accuracy of vascular lesions : 99 %
```

```
201: # Initializing a list to store the number of correctly classified
```

With evaluation mode of network:

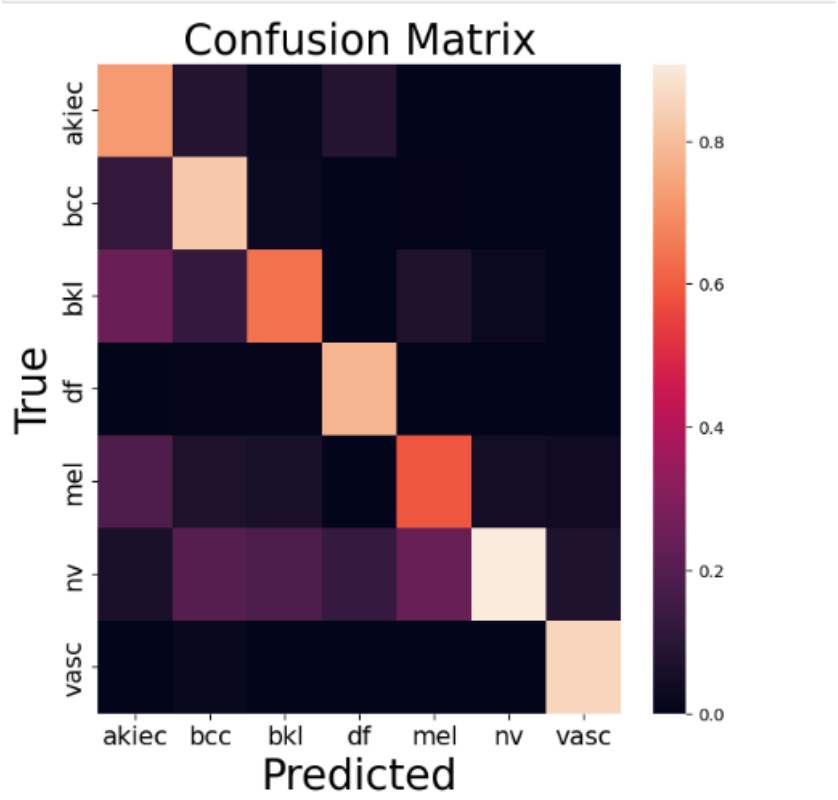
```
    print('Accuracy of %5s : %2d %%' % (
        classes[i], 100 * class_correct[i] / class_total[i]))
```

```
Accuracy of actinic keratoses : 64 %
Accuracy of basal cell carcinoma : 90 %
Accuracy of benign keratosis-like lesions : 70 %
Accuracy of dermatofibroma : 49 %
Accuracy of melanoma : 47 %
Accuracy of melanocytic nevi : 90 %
Accuracy of vascular lesions : 99 %
```

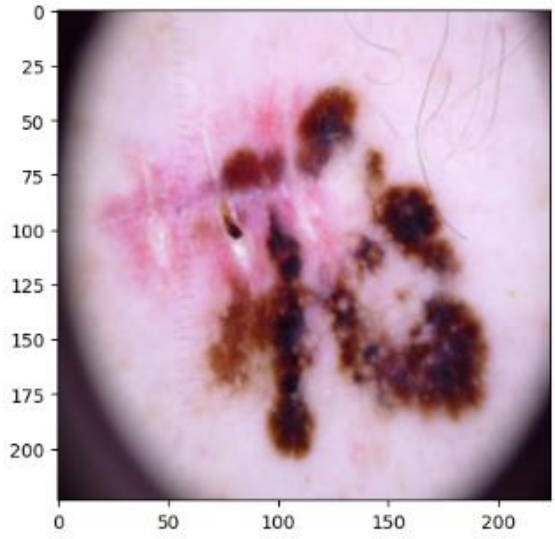
Confusion Matix:

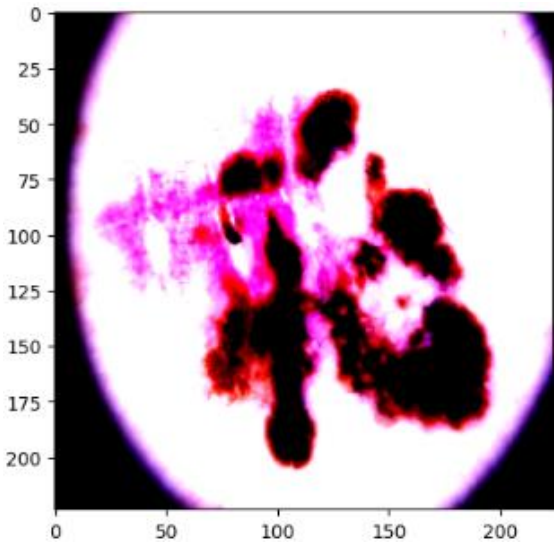
```
tensor([[4.7000e+01, 9.0000e+00, 4.0000e+00, 2.0000e+00, 0.0000e+00, 3.0000e+00,
         0.0000e+00],
        [8.0000e+00, 8.5000e+01, 5.0000e+00, 0.0000e+00, 1.0000e+00, 4.0000e+00,
         0.0000e+00],
        [1.6000e+01, 1.3000e+01, 1.4100e+02, 0.0000e+00, 1.6000e+01, 3.4000e+01,
         0.0000e+00],
        [0.0000e+00, 1.0000e+00, 2.0000e+00, 1.8000e+01, 0.0000e+00, 2.0000e+00,
         0.0000e+00],
        [1.2000e+01, 7.0000e+00, 1.3000e+01, 0.0000e+00, 1.3100e+02, 5.9000e+01,
         1.0000e+00],
        [4.0000e+00, 2.1000e+01, 4.0000e+01, 3.0000e+00, 5.4000e+01, 1.2170e+03,
         2.0000e+00],
        [0.0000e+00, 2.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 2.0000e+00,
         2.4000e+01]])
```

Heatmap of normalized confusion matrix:



Grad Cam:






```
Generating Grad-CAM @layer4
        #0: melanoma (0.98846)
```