Animal_Classification-base

June 10, 2024

1 Base Model

This is initial model that was provided this no modification

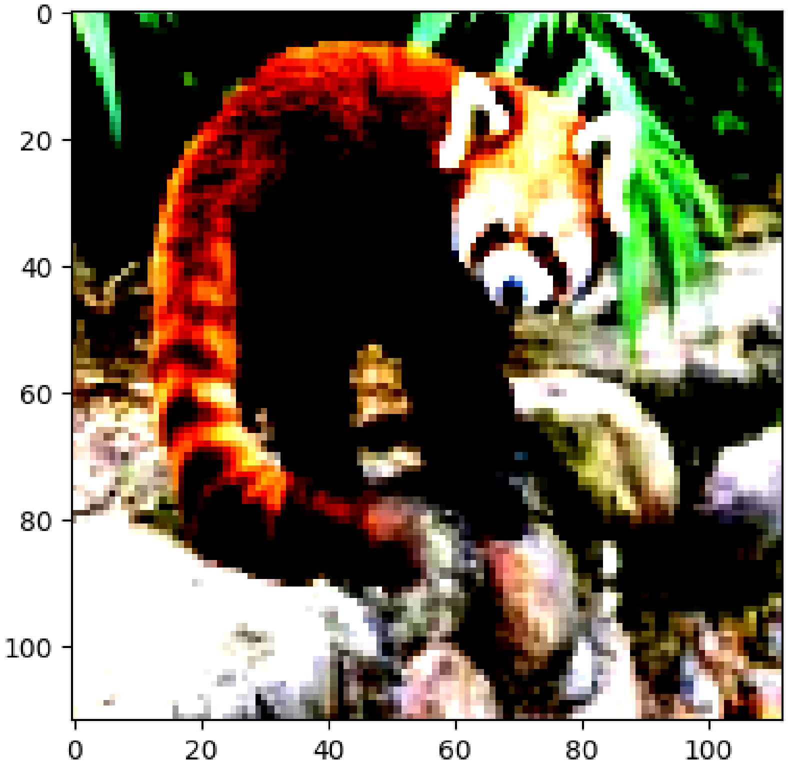
Validation Loss: 5.016 Validation Accuracy: 37.45% FLOPS: 0.69G

```
Size of training dataset : 6270
```

```
torch.Size([3, 112, 112])
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

```
Label:  ailurus-fulgens (5)
```



```
(5330, 313, 627)
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

```
ConvolutionalNetwork(
  (conv1): Conv2d(3, 64, kernel_size=(5, 5), stride=(1, 1))
  (conv2): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv3): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv4): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
  (fc1): Linear(in_features=3200, out_features=151, bias=True)
)
```

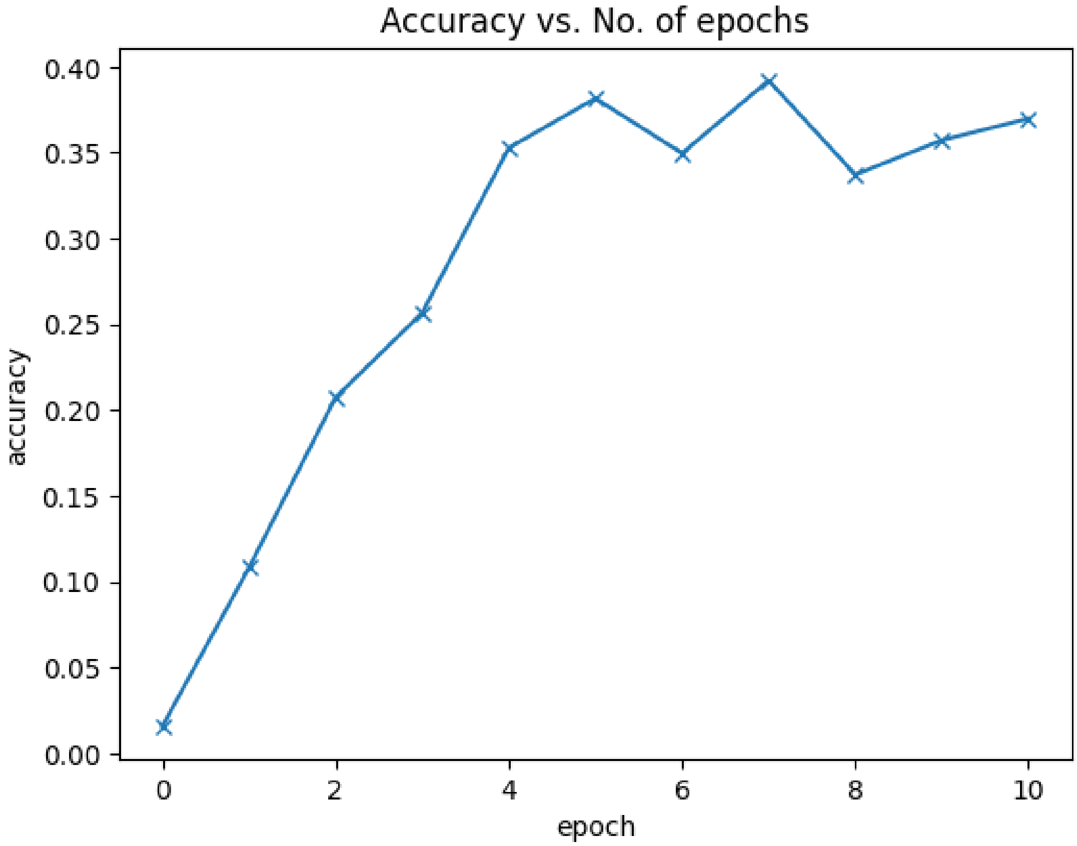
```
images.shape: torch.Size([16, 3, 112, 112])
out.shape: torch.Size([16, 151])
out[0]: tensor([-4.9810, -4.9628, -5.0004, -5.1166, -5.0207, -4.9754, -5.0287,
-5.0343,
        -4.9360, -5.0579, -4.9739, -5.0181, -5.0816, -4.9687, -4.9360, -4.9672,
        -5.0258, -4.9406, -5.0533, -5.0462, -5.0246, -4.9779, -4.9872, -5.0308,
        -5.0082, -5.0983, -5.0357, -5.0262, -5.0195, -4.9694, -4.9170, -5.1058,
        -5.0232, -5.0241, -5.0034, -5.0520, -4.9604, -5.0360, -5.0646, -4.9893,
        -4.9578, -5.0355, -5.0061, -5.0049, -5.0020, -5.0383, -5.0278, -5.0336,
        -5.0096, -4.9908, -5.0240, -5.0019, -4.9970, -5.0736, -5.0690, -5.0367,
        -4.9537, -5.0003, -5.0393, -5.0141, -4.9773, -4.9391, -4.9872, -4.9851,
        -5.0518, -5.1146, -5.0664, -4.9889, -5.0137, -5.0829, -4.9924, -4.9548,
        -5.0350, -4.9460, -5.0569, -5.0219, -5.0776, -5.0656, -5.0113, -5.0106,
        -5.0152, -4.9609, -5.0398, -4.9941, -4.9092, -5.0280, -5.0567, -4.9632,
        -5.0684, -4.9715, -5.0328, -5.0997, -5.0250, -5.0109, -5.0317, -5.0543,
        -5.0015, -4.9797, -4.9785, -5.0108, -5.0724, -5.0358, -5.1008, -4.9554,
        -5.0832, -5.0400, -4.9942, -5.0438, -5.0266, -5.0275, -5.0620, -5.0990,
        -5.0138, -5.0264, -5.0076, -4.9963, -4.9973, -4.9772, -5.0009, -5.0275,
        -5.0152, -5.0921, -5.0375, -5.0536, -5.0141, -5.0301, -4.9980, -5.0565,
        -5.0409, -5.0632, -5.0884, -5.0344, -5.0536, -4.9807, -5.0489, -5.0162,
        -4.9376, -5.0788, -5.0771, -4.9925, -5.0671, -5.0719, -5.0379, -4.9975,
        -4.9232, -4.9465, -4.9842, -4.9278, -5.0468, -5.0441, -5.0029],
       device='cuda:0', grad_fn=<SelectBackward0>)
```

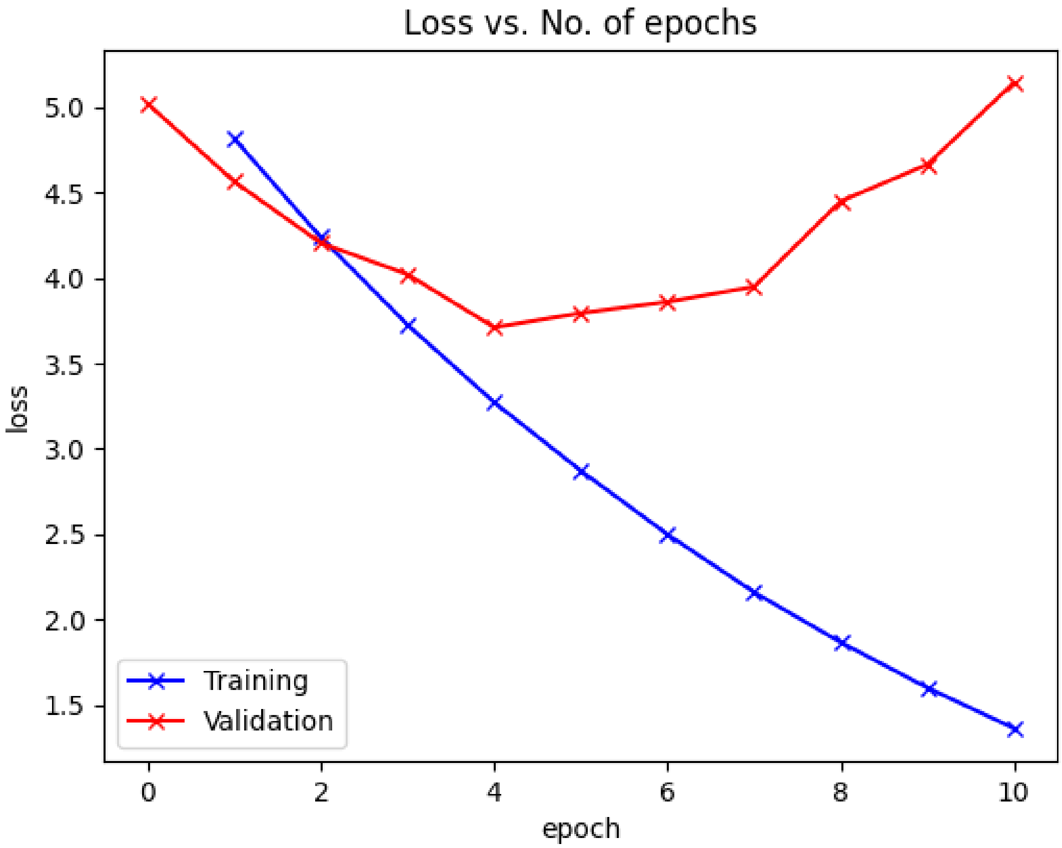
```
ConvolutionalNetwork(
  (conv1): Conv2d(3, 64, kernel_size=(5, 5), stride=(1, 1))
  (conv2): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv3): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv4): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
  (fc1): Linear(in_features=3200, out_features=151, bias=True)
)
```

```
[{'val_loss': 5.020401954650879, 'val_acc': 0.015625}]
```

```
627
```

```
  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [0], train_loss: 4.8155, val_loss: 4.5634, val_acc: 0.1080

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [1], train_loss: 4.2369, val_loss: 4.2035, val_acc: 0.2073

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [2], train_loss: 3.7248, val_loss: 4.0203, val_acc: 0.2566

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [3], train_loss: 3.2704, val_loss: 3.7099, val_acc: 0.3528

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [4], train_loss: 2.8699, val_loss: 3.7928, val_acc: 0.3816

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [5], train_loss: 2.4970, val_loss: 3.8591, val_acc: 0.3497

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [6], train_loss: 2.1576, val_loss: 3.9476, val_acc: 0.3920

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [7], train_loss: 1.8643, val_loss: 4.4483, val_acc: 0.3372

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [8], train_loss: 1.5983, val_loss: 4.6634, val_acc: 0.3573

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [9], train_loss: 1.3605, val_loss: 5.1438, val_acc: 0.3698
```


Accuracy vs. No. of epochs
accuracy
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
epoch
0
2
4
6
8
10



```
{'val_loss': 5.015594005584717, 'val_acc': 0.37447917461395264}
```

1.1 FLOPs

```
 + Number of FLOPs: 0.69G
```

Animal_Classification-data-aug

June 10, 2024

1 Data Augmentation

From the previous model, we have changed the data augmentation into the model with the aim to make the model more general.

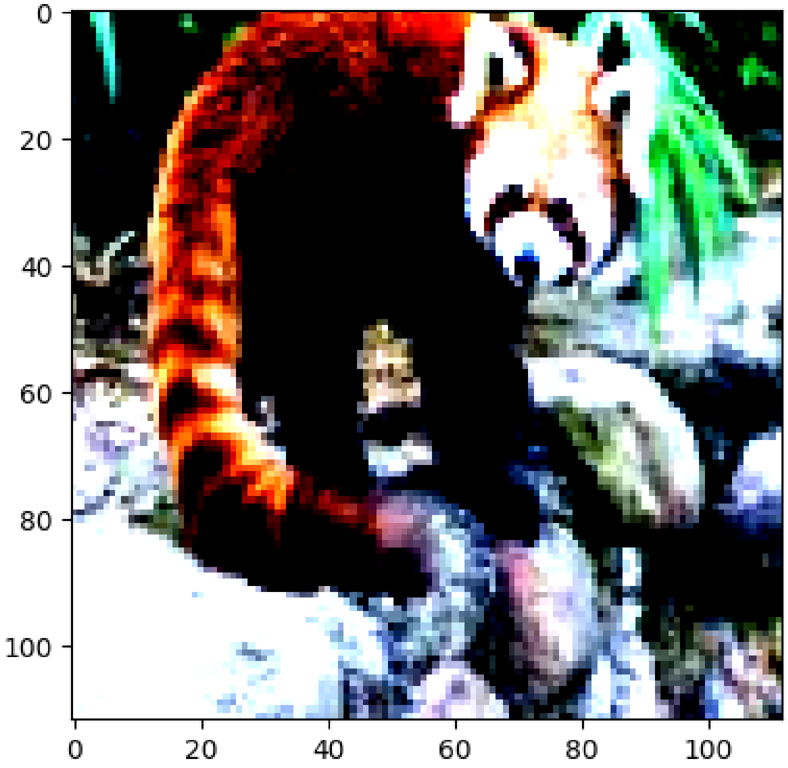
Validation Loss: 3.689 Validation Accuracy: 36.61% FLOPS: 0.69G

```
Size of training dataset : 6270
```

```
torch.Size([3, 112, 112])
```

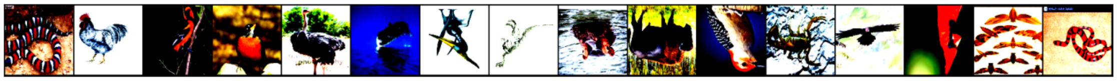
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

```
Label:  ailurus-fulgens (5)
```



```
 (5330, 313, 627)
```

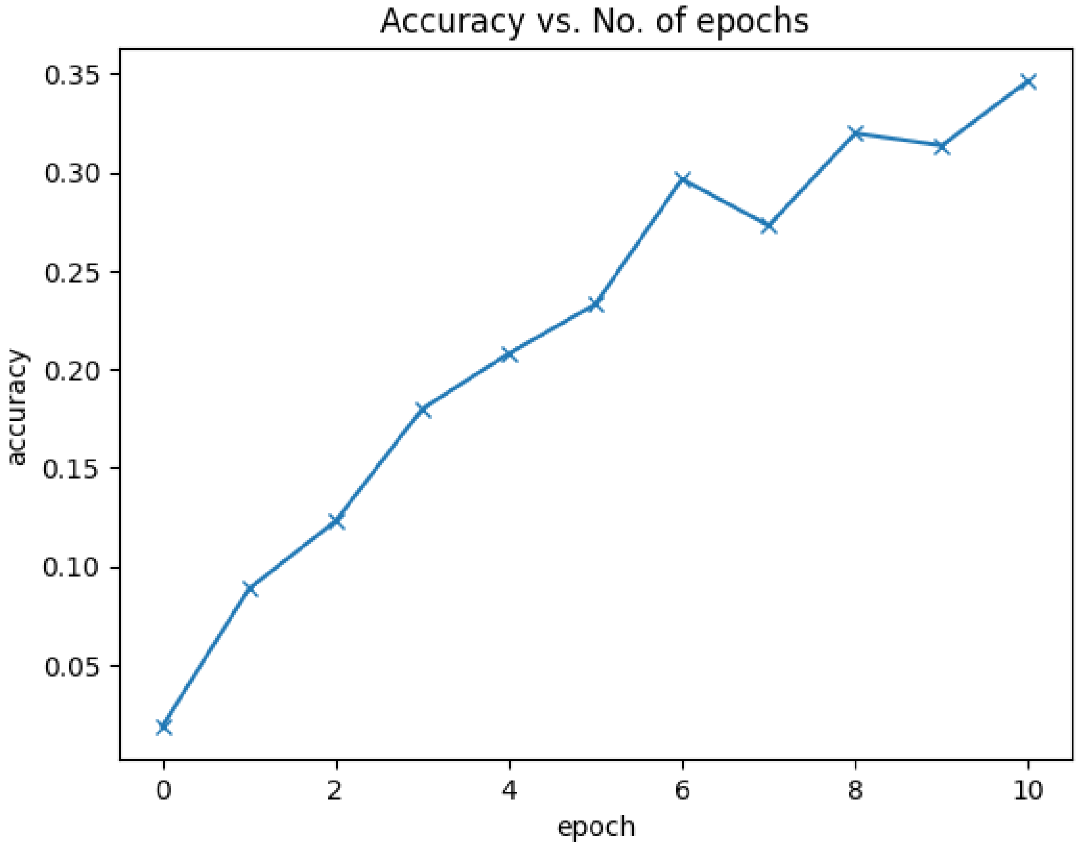
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

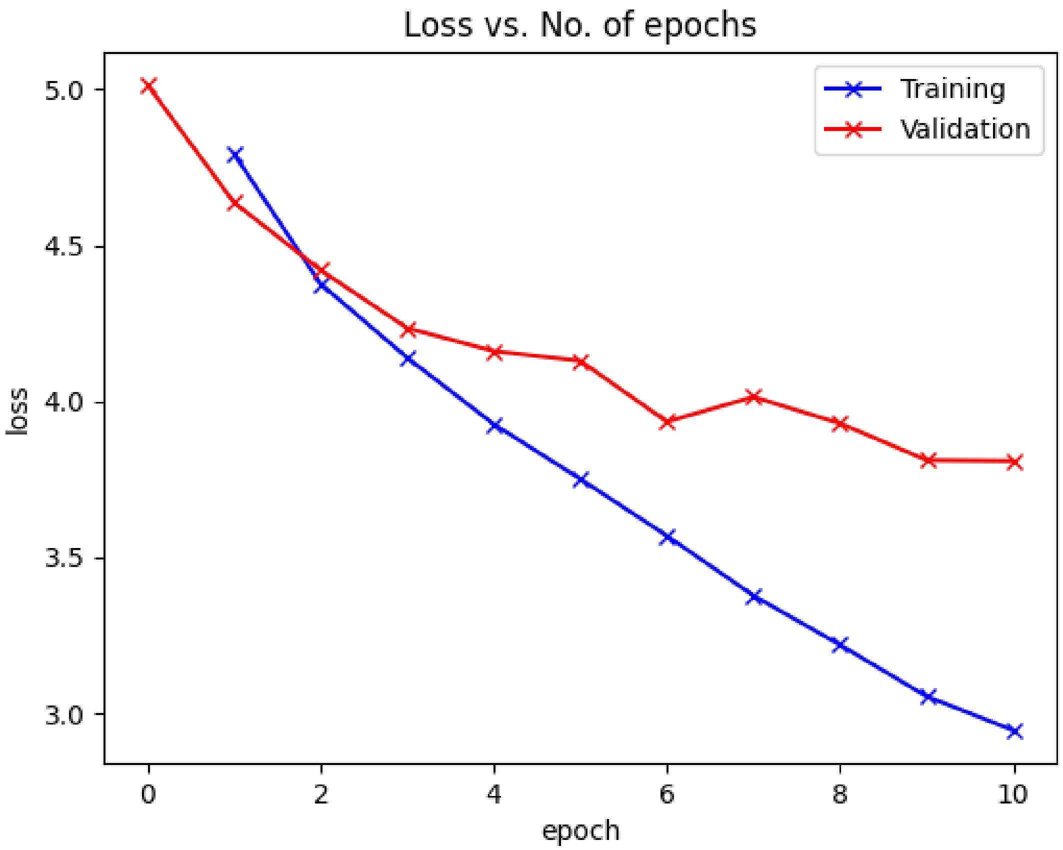


```
 ConvolutionalNetwork(
   (conv1): Conv2d(3, 64, kernel_size=(5, 5), stride=(1, 1))
   (conv2): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1))
   (conv3): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
   (conv4): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
   (fc1): Linear(in_features=3200, out_features=151, bias=True)
 )
```

```
images.shape: torch.Size([16, 3, 112, 112])
out.shape: torch.Size([16, 151])
out[0]: tensor([-5.0405, -5.0069, -4.9499, -5.0707, -5.0757, -5.0082, -4.9831,
-4.9833,
        -5.0559, -5.0269, -5.0913, -4.9888, -4.9934, -5.0182, -5.0519, -5.0426,
        -5.0444, -5.0217, -5.0342, -5.0866, -5.0165, -5.0140, -5.0279, -4.9752,
        -5.0075, -4.9683, -4.9756, -5.0821, -5.0339, -4.9775, -4.9766, -5.0211,
        -5.0789, -5.0153, -5.0031, -5.0059, -5.0231, -4.9917, -4.8929, -5.0270,
        -5.0481, -5.0200, -5.0160, -5.0715, -4.9473, -5.0549, -5.0045, -5.0938,
        -4.9971, -5.0012, -5.0303, -5.0332, -4.9954, -5.0794, -5.0283, -5.0726,
        -5.0624, -5.0386, -5.0130, -4.9753, -5.0669, -5.0276, -5.0438, -5.0435,
        -5.0622, -4.9776, -5.0274, -5.0404, -5.0266, -5.0178, -5.0310, -5.0147,
        -5.0273, -5.0788, -5.0491, -5.0703, -5.0287, -4.9993, -5.0185, -4.9626,
        -5.0880, -5.0043, -5.0573, -4.9818, -4.9979, -4.9563, -5.0507, -5.0166,
        -5.0372, -5.0336, -5.0458, -5.0125, -5.0072, -4.9287, -5.0209, -5.0372,
        -5.0294, -5.0109, -5.0749, -5.0774, -5.0551, -5.0230, -4.9546, -5.0355,
        -4.9514, -5.0215, -5.0329, -5.0142, -4.9523, -5.0537, -5.0671, -5.0318,
        -5.0894, -5.0019, -5.0205, -5.0277, -4.9608, -4.9479, -4.9819, -5.0188,
        -5.0346, -5.0617, -5.0043, -4.9750, -5.0060, -4.9762, -4.9555, -5.0211,
        -4.9650, -5.0130, -4.9971, -4.9691, -5.0214, -4.9848, -5.0402, -5.0176,
        -5.0021, -5.0755, -5.0118, -5.0288, -5.0153, -4.9983, -5.0189, -4.9564,
        -4.9752, -4.9709, -4.9840, -5.0463, -4.9700, -5.0691, -4.9562],
       device='cuda:0', grad_fn=<SelectBackward0>)
```

```
 ConvolutionalNetwork(
   (conv1): Conv2d(3, 64, kernel_size=(5, 5), stride=(1, 1))
   (conv2): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1))
   (conv3): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
   (conv4): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
   (fc1): Linear(in_features=3200, out_features=151, bias=True)
 )

 [{'val_loss': 5.015295505523682, 'val_acc': 0.01875000074505806}]

627

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [0], train_loss: 4.7936, val_loss: 4.6374, val_acc: 0.0885

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [1], train_loss: 4.3765, val_loss: 4.4203, val_acc: 0.1229

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [2], train_loss: 4.1393, val_loss: 4.2343, val_acc: 0.1799

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [3], train_loss: 3.9262, val_loss: 4.1602, val_acc: 0.2080

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [4], train_loss: 3.7502, val_loss: 4.1291, val_acc: 0.2330

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [5], train_loss: 3.5679, val_loss: 3.9339, val_acc: 0.2965

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [6], train_loss: 3.3759, val_loss: 4.0132, val_acc: 0.2729

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [7], train_loss: 3.2179, val_loss: 3.9284, val_acc: 0.3198

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [8], train_loss: 3.0526, val_loss: 3.8104, val_acc: 0.3135

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [9], train_loss: 2.9439, val_loss: 3.8080, val_acc: 0.3465
```


Accuracy vs. No. of epochs
accuracy
0.35
0.30
0.25
0.20
0.15
0.10
0.05
epoch
0
2
4
6
8
10



```
{'val_loss': 3.688917875289917, 'val_acc': 0.3661458492279053}
```

1.1 FLOPs

```
+ Number of FLOPs: 0.69G
```

Animal_Classification-epoch

June 10, 2024

1 Increasing epochs

From the previous model, we have changed the number of epochs

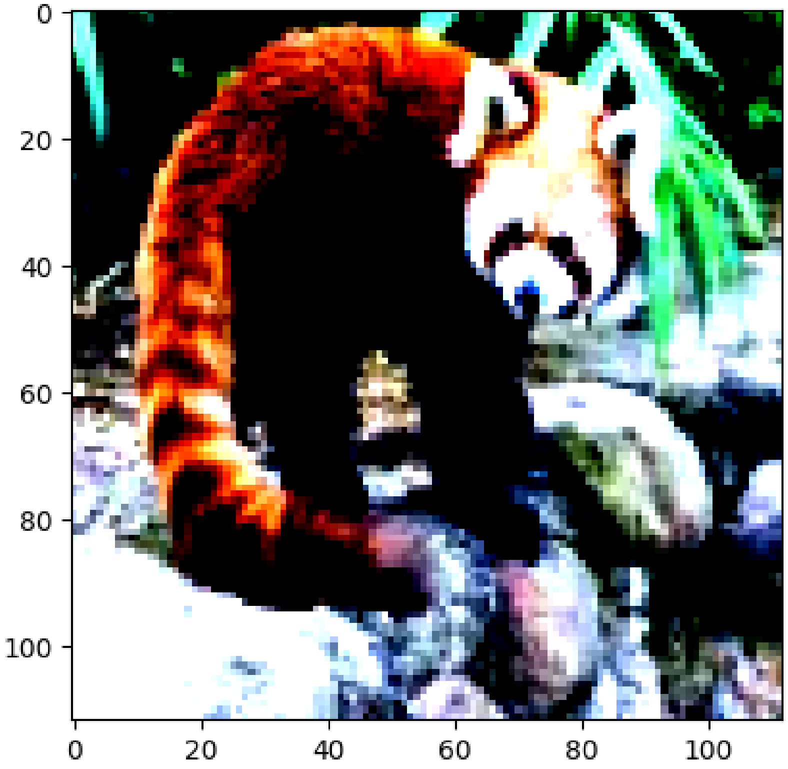
Validation Loss: 3.893 Validation Accuracy: 39.11% FLOPS: 0.69G

```
Size of training dataset : 6270
```

```
torch.Size([3, 112, 112])
```

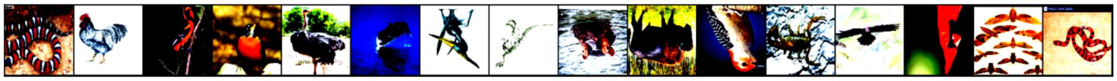
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

```
Label:  ailurus-fulgens (5)
```



```
(5330, 313, 627)
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

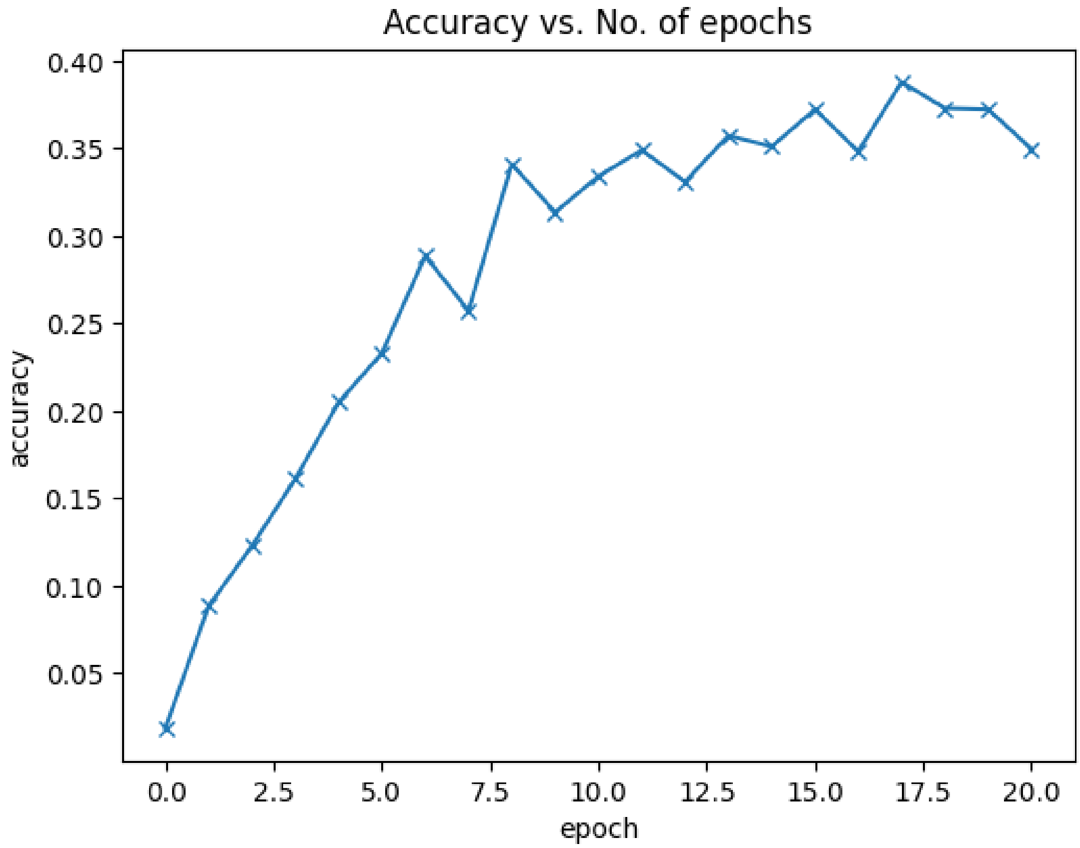


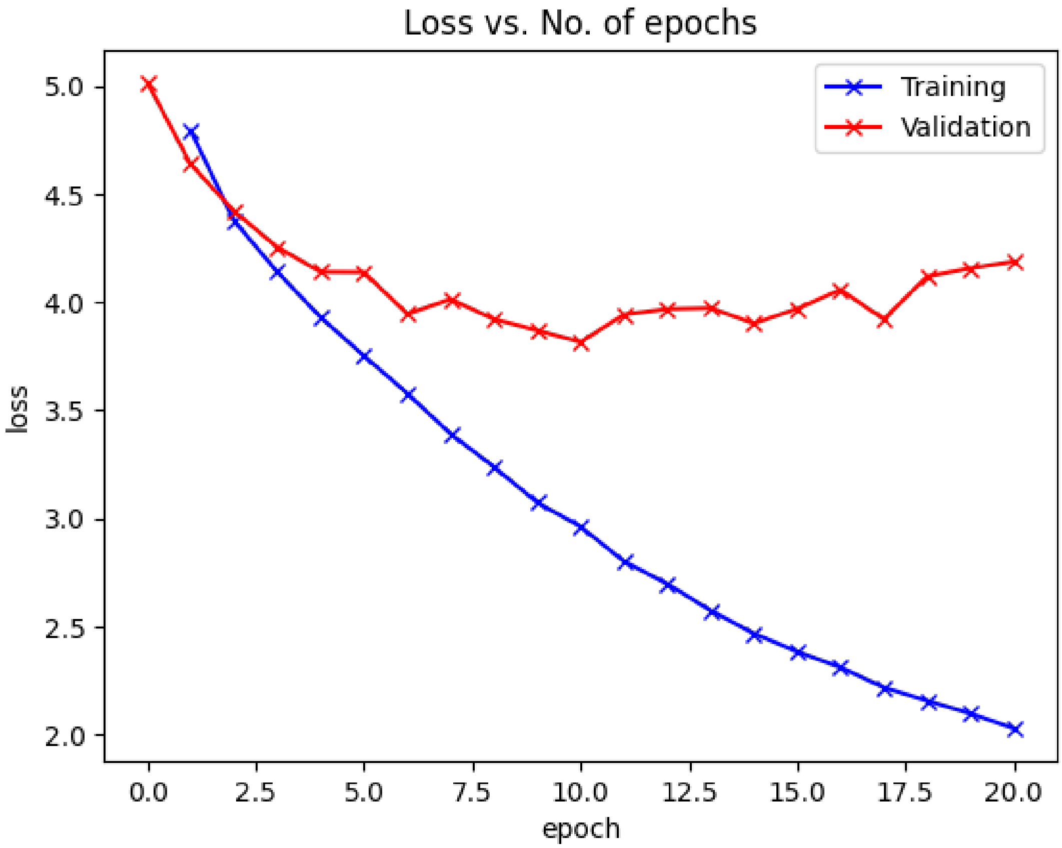
```
ConvolutionalNetwork(
  (conv1): Conv2d(3, 64, kernel_size=(5, 5), stride=(1, 1))
  (conv2): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv3): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv4): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
  (fc1): Linear(in_features=3200, out_features=151, bias=True)
)
```

```
images.shape: torch.Size([16, 3, 112, 112])
out.shape: torch.Size([16, 151])
out[0]: tensor([-5.0405, -5.0069, -4.9499, -5.0707, -5.0757, -5.0082, -4.9831,
-4.9833,
        -5.0559, -5.0269, -5.0913, -4.9888, -4.9934, -5.0182, -5.0519, -5.0426,
        -5.0444, -5.0217, -5.0342, -5.0866, -5.0165, -5.0140, -5.0279, -4.9752,
        -5.0075, -4.9683, -4.9756, -5.0821, -5.0339, -4.9775, -4.9766, -5.0211,
        -5.0789, -5.0153, -5.0031, -5.0059, -5.0231, -4.9917, -4.8929, -5.0270,
        -5.0481, -5.0200, -5.0160, -5.0715, -4.9473, -5.0549, -5.0045, -5.0938,
        -4.9971, -5.0012, -5.0303, -5.0332, -4.9954, -5.0794, -5.0283, -5.0726,
        -5.0624, -5.0386, -5.0130, -4.9753, -5.0669, -5.0276, -5.0438, -5.0435,
        -5.0622, -4.9776, -5.0274, -5.0404, -5.0266, -5.0178, -5.0310, -5.0147,
        -5.0273, -5.0788, -5.0491, -5.0703, -5.0287, -4.9993, -5.0185, -4.9626,
        -5.0880, -5.0043, -5.0573, -4.9818, -4.9979, -4.9563, -5.0507, -5.0166,
        -5.0372, -5.0336, -5.0458, -5.0125, -5.0072, -4.9287, -5.0209, -5.0372,
        -5.0294, -5.0109, -5.0749, -5.0774, -5.0551, -5.0230, -4.9546, -5.0355,
        -4.9514, -5.0215, -5.0329, -5.0142, -4.9523, -5.0537, -5.0671, -5.0318,
        -5.0894, -5.0019, -5.0205, -5.0277, -4.9608, -4.9479, -4.9819, -5.0188,
        -5.0346, -5.0617, -5.0043, -4.9750, -5.0060, -4.9762, -4.9555, -5.0211,
        -4.9650, -5.0130, -4.9971, -4.9691, -5.0214, -4.9848, -5.0402, -5.0176,
        -5.0021, -5.0755, -5.0118, -5.0288, -5.0153, -4.9983, -5.0189, -4.9564,
        -4.9752, -4.9709, -4.9840, -5.0463, -4.9700, -5.0691, -4.9562],
       device='cuda:0', grad_fn=<SelectBackward0>)
```

```
ConvolutionalNetwork(
  (conv1): Conv2d(3, 64, kernel_size=(5, 5), stride=(1, 1))
  (conv2): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv3): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv4): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
  (fc1): Linear(in_features=3200, out_features=151, bias=True)
)

[{'val_loss': 5.015295505523682, 'val_acc': 0.01875000074505806}]

627

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [0], train_loss: 4.7936, val_loss: 4.6374, val_acc: 0.0885

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [1], train_loss: 4.3765, val_loss: 4.4203, val_acc: 0.1229

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [2], train_loss: 4.1399, val_loss: 4.2541, val_acc: 0.1611

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [3], train_loss: 3.9299, val_loss: 4.1399, val_acc: 0.2049

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [4], train_loss: 3.7517, val_loss: 4.1383, val_acc: 0.2330

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [5], train_loss: 3.5783, val_loss: 3.9452, val_acc: 0.2885

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [6], train_loss: 3.3906, val_loss: 4.0134, val_acc: 0.2573

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [7], train_loss: 3.2382, val_loss: 3.9209, val_acc: 0.3410

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [8], train_loss: 3.0732, val_loss: 3.8694, val_acc: 0.3135

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [9], train_loss: 2.9617, val_loss: 3.8153, val_acc: 0.3340

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [10], train_loss: 2.8012, val_loss: 3.9421, val_acc: 0.3490

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [11], train_loss: 2.6973, val_loss: 3.9678, val_acc: 0.3309

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [12], train_loss: 2.5743, val_loss: 3.9724, val_acc: 0.3573

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [13], train_loss: 2.4676, val_loss: 3.9011, val_acc: 0.3510
```

```
  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [14], train_loss: 2.3833, val_loss: 3.9690, val_acc: 0.3722

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [15], train_loss: 2.3112, val_loss: 4.0574, val_acc: 0.3483

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [16], train_loss: 2.2180, val_loss: 3.9208, val_acc: 0.3878

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [17], train_loss: 2.1563, val_loss: 4.1189, val_acc: 0.3729

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [18], train_loss: 2.0974, val_loss: 4.1565, val_acc: 0.3722

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [19], train_loss: 2.0290, val_loss: 4.1874, val_acc: 0.3497
```





```
{'val_loss': 3.8934414386749268, 'val_acc': 0.39114585518836975}
```

1.1 FLOPs

```
+ Number of FLOPs: 0.69G
```

Animal_Classification-optimization-function

June 10, 2024

1 Changing Optimisation Function

From the previous model, we have changed the Optimisation function to RAdam

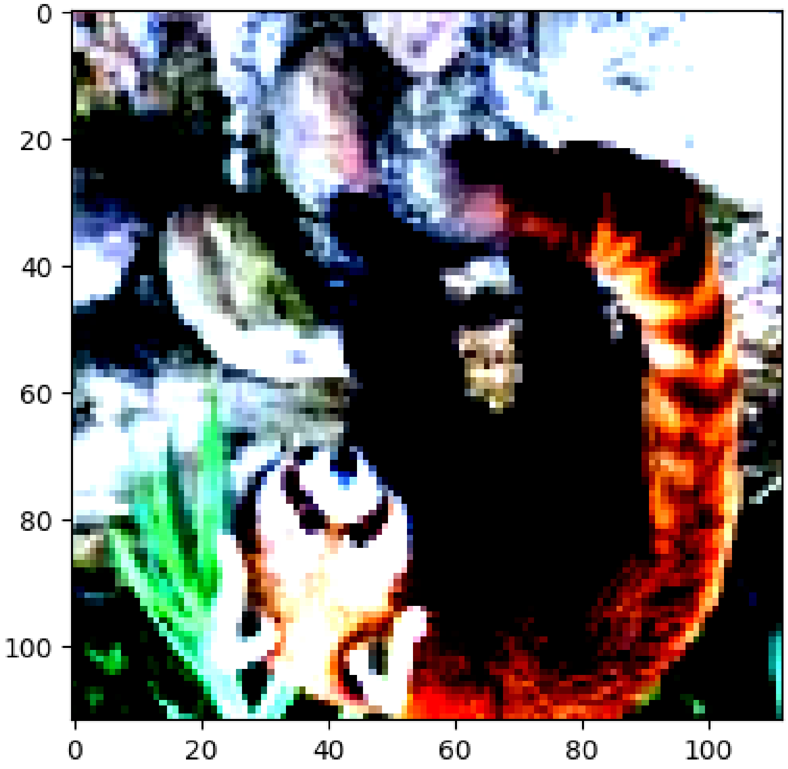
Validation Loss: 4.241 Validation Accuracy: 44.58% FLOPS: 0.69G

```
Size of training dataset : 6270
```

```
torch.Size([3, 112, 112])
```

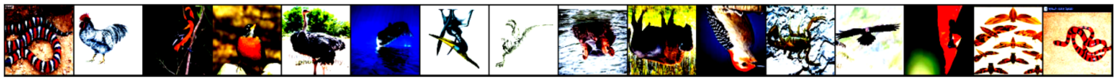
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

```
Label:  ailurus-fulgens (5)
```



```
(5330, 313, 627)
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```



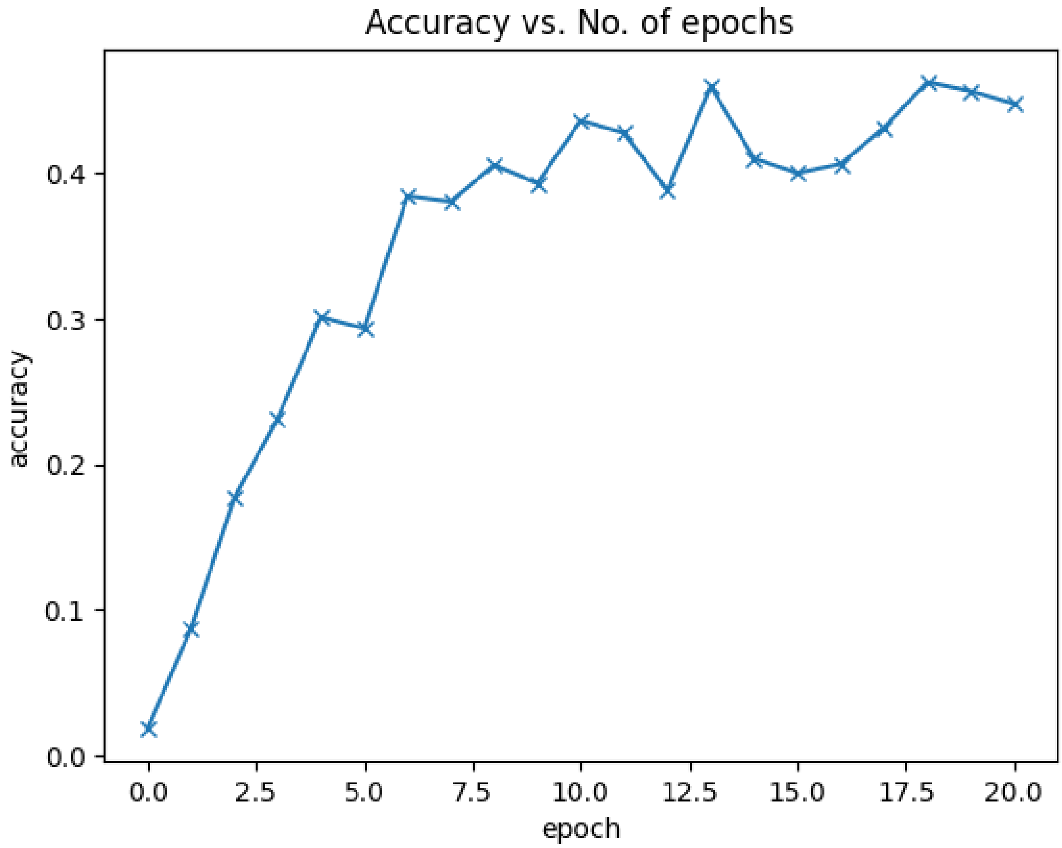
```
ConvolutionalNetwork(
  (conv1): Conv2d(3, 64, kernel_size=(5, 5), stride=(1, 1))
  (conv2): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv3): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv4): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
  (fc1): Linear(in_features=3200, out_features=151, bias=True)
)
```

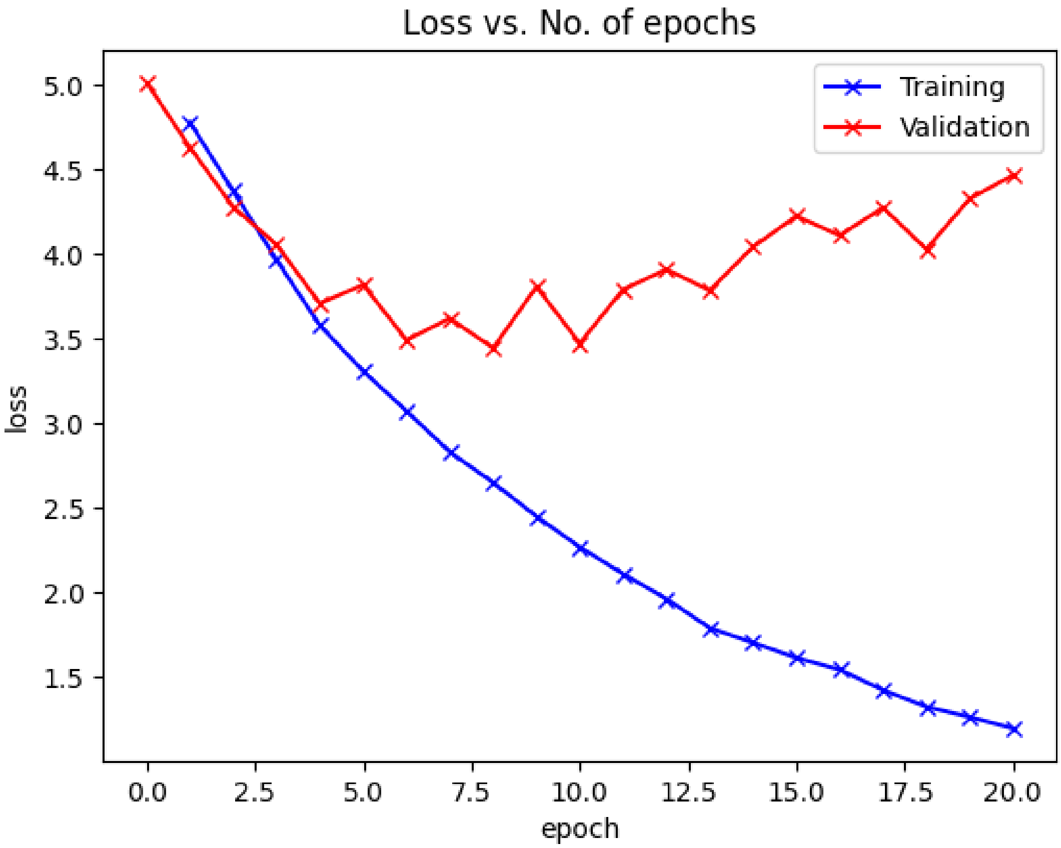
```
images.shape: torch.Size([16, 3, 112, 112])
out.shape: torch.Size([16, 151])
out[0]: tensor([-5.0405, -5.0069, -4.9499, -5.0707, -5.0757, -5.0082, -4.9831,
-4.9833,
        -5.0559, -5.0269, -5.0913, -4.9888, -4.9934, -5.0182, -5.0519, -5.0426,
        -5.0444, -5.0217, -5.0342, -5.0866, -5.0165, -5.0140, -5.0279, -4.9752,
        -5.0075, -4.9683, -4.9756, -5.0821, -5.0339, -4.9775, -4.9766, -5.0211,
        -5.0789, -5.0153, -5.0031, -5.0059, -5.0231, -4.9917, -4.8929, -5.0270,
        -5.0481, -5.0200, -5.0160, -5.0715, -4.9473, -5.0549, -5.0045, -5.0938,
        -4.9971, -5.0012, -5.0303, -5.0332, -4.9954, -5.0794, -5.0283, -5.0726,
        -5.0624, -5.0386, -5.0130, -4.9753, -5.0669, -5.0276, -5.0438, -5.0435,
        -5.0622, -4.9776, -5.0274, -5.0404, -5.0266, -5.0178, -5.0310, -5.0147,
        -5.0273, -5.0788, -5.0491, -5.0703, -5.0287, -4.9993, -5.0185, -4.9626,
        -5.0880, -5.0043, -5.0573, -4.9818, -4.9979, -4.9563, -5.0507, -5.0166,
        -5.0372, -5.0336, -5.0458, -5.0125, -5.0072, -4.9287, -5.0209, -5.0372,
        -5.0294, -5.0109, -5.0749, -5.0774, -5.0551, -5.0230, -4.9546, -5.0355,
        -4.9514, -5.0215, -5.0329, -5.0142, -4.9523, -5.0537, -5.0671, -5.0318,
        -5.0894, -5.0019, -5.0205, -5.0277, -4.9608, -4.9479, -4.9819, -5.0188,
        -5.0346, -5.0617, -5.0043, -4.9750, -5.0060, -4.9762, -4.9555, -5.0211,
        -4.9650, -5.0130, -4.9971, -4.9691, -5.0214, -4.9848, -5.0402, -5.0176,
        -5.0021, -5.0755, -5.0118, -5.0288, -5.0153, -4.9983, -5.0189, -4.9564,
        -4.9752, -4.9709, -4.9840, -5.0463, -4.9700, -5.0691, -4.9562],
       device='cuda:0', grad_fn=<SelectBackward0>)
```

```
ConvolutionalNetwork(
  (conv1): Conv2d(3, 64, kernel_size=(5, 5), stride=(1, 1))
  (conv2): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv3): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv4): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
```

```
   (fc1): Linear(in_features=3200, out_features=151, bias=True)
 )

 [{'val_loss': 5.015295505523682, 'val_acc': 0.01875000074505806}]

627

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [0], train_loss: 4.7799, val_loss: 4.6322, val_acc: 0.0868

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [1], train_loss: 4.3743, val_loss: 4.2798, val_acc: 0.1767

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [2], train_loss: 3.9671, val_loss: 4.0570, val_acc: 0.2309

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [3], train_loss: 3.5803, val_loss: 3.7100, val_acc: 0.3010

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [4], train_loss: 3.3091, val_loss: 3.8204, val_acc: 0.2931

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [5], train_loss: 3.0747, val_loss: 3.4936, val_acc: 0.3840

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [6], train_loss: 2.8316, val_loss: 3.6213, val_acc: 0.3802

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [7], train_loss: 2.6511, val_loss: 3.4462, val_acc: 0.4052

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [8], train_loss: 2.4510, val_loss: 3.8086, val_acc: 0.3927

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [9], train_loss: 2.2702, val_loss: 3.4743, val_acc: 0.4358

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [10], train_loss: 2.1108, val_loss: 3.7924, val_acc: 0.4271

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [11], train_loss: 1.9643, val_loss: 3.9103, val_acc: 0.3878

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [12], train_loss: 1.7916, val_loss: 3.7872, val_acc: 0.4590

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [13], train_loss: 1.7047, val_loss: 4.0440, val_acc: 0.4097
```

```
  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [14], train_loss: 1.6155, val_loss: 4.2254, val_acc: 0.3997

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [15], train_loss: 1.5454, val_loss: 4.1115, val_acc: 0.4059

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [16], train_loss: 1.4244, val_loss: 4.2762, val_acc: 0.4309

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [17], train_loss: 1.3263, val_loss: 4.0303, val_acc: 0.4622

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [18], train_loss: 1.2653, val_loss: 4.3288, val_acc: 0.4559

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [19], train_loss: 1.1986, val_loss: 4.4682, val_acc: 0.4472
```





```
{'val_loss': 4.240588665008545, 'val_acc': 0.44583335518836975}
```

1.1 FLOPs

```
+ Number of FLOPs: 0.69G
```

Animal_Classification-learning-rate

June 10, 2024

1 Changing Learning Rate

From the previous model, we have changed the learning rate to 0.0001

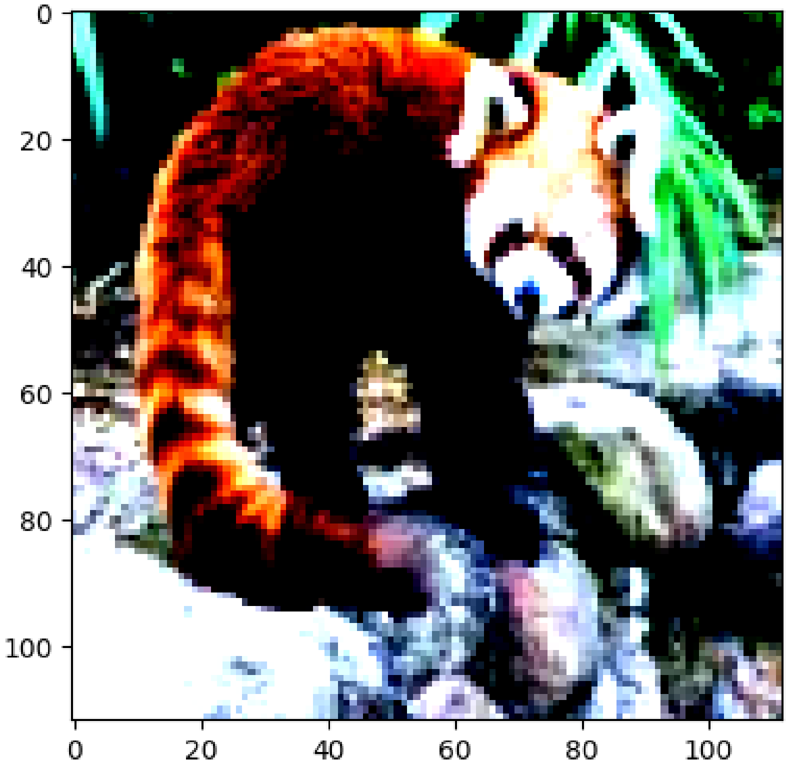
Validation Loss: 3.705 Validation Accuracy: 45.21% FLOPS: 0.69G

```
Size of training dataset : 6270
```

```
torch.Size([3, 112, 112])
```

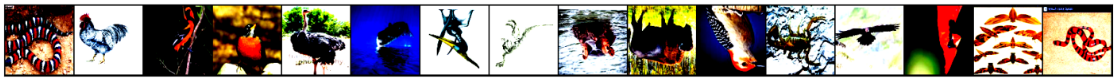
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

```
Label:  ailurus-fulgens (5)
```



```
(5330, 313, 627)
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```



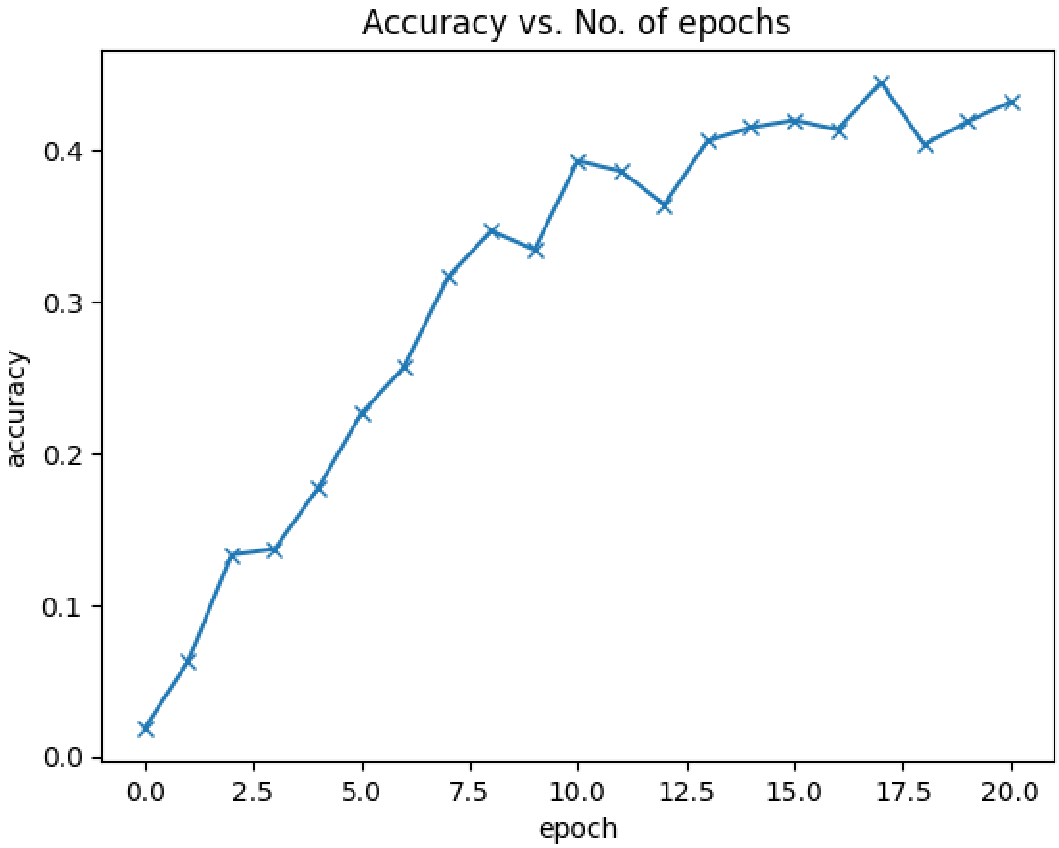
```
ConvolutionalNetwork(
  (conv1): Conv2d(3, 64, kernel_size=(5, 5), stride=(1, 1))
  (conv2): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv3): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv4): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
  (fc1): Linear(in_features=3200, out_features=151, bias=True)
)
```

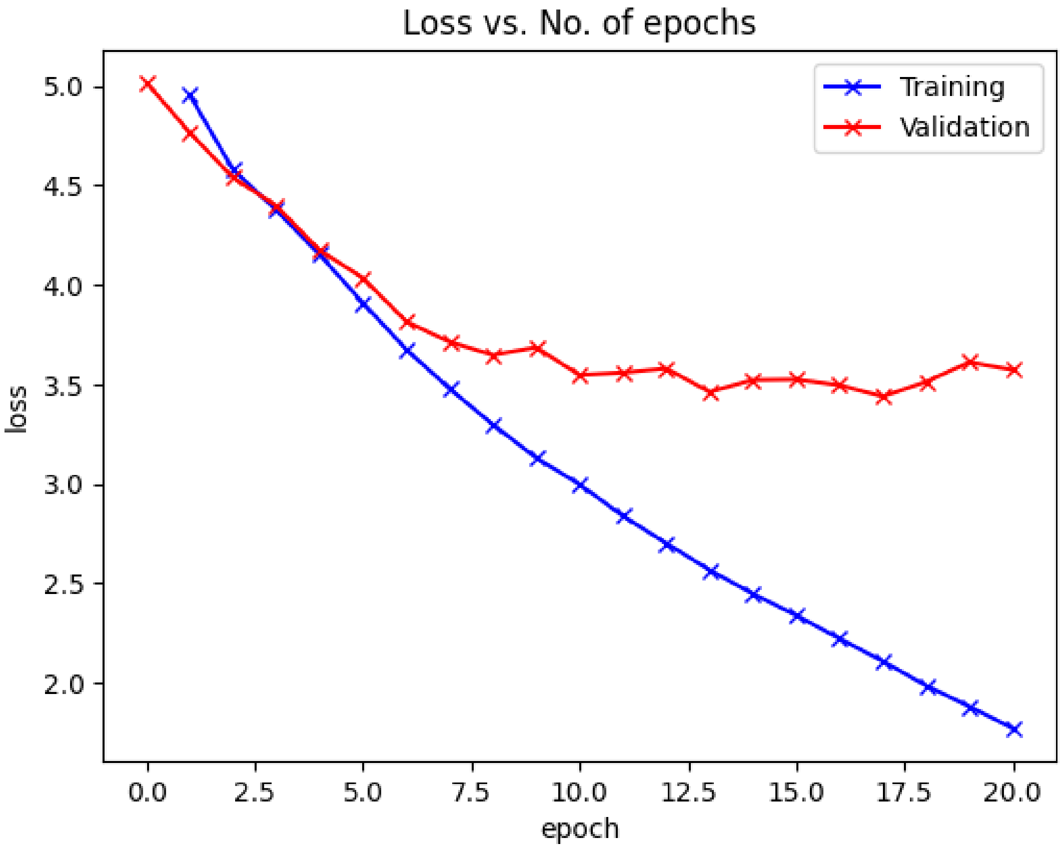
```
images.shape: torch.Size([16, 3, 112, 112])
out.shape: torch.Size([16, 151])
out[0]: tensor([-5.0405, -5.0069, -4.9499, -5.0707, -5.0757, -5.0082, -4.9831,
-4.9833,
        -5.0559, -5.0269, -5.0913, -4.9888, -4.9934, -5.0182, -5.0519, -5.0426,
        -5.0444, -5.0217, -5.0342, -5.0866, -5.0165, -5.0140, -5.0279, -4.9752,
        -5.0075, -4.9683, -4.9756, -5.0821, -5.0339, -4.9775, -4.9766, -5.0211,
        -5.0789, -5.0153, -5.0031, -5.0059, -5.0231, -4.9917, -4.8929, -5.0270,
        -5.0481, -5.0200, -5.0160, -5.0715, -4.9473, -5.0549, -5.0045, -5.0938,
        -4.9971, -5.0012, -5.0303, -5.0332, -4.9954, -5.0794, -5.0283, -5.0726,
        -5.0624, -5.0386, -5.0130, -4.9753, -5.0669, -5.0276, -5.0438, -5.0435,
        -5.0622, -4.9776, -5.0274, -5.0404, -5.0266, -5.0178, -5.0310, -5.0147,
        -5.0273, -5.0788, -5.0491, -5.0703, -5.0287, -4.9993, -5.0185, -4.9626,
        -5.0880, -5.0043, -5.0573, -4.9818, -4.9979, -4.9563, -5.0507, -5.0166,
        -5.0372, -5.0336, -5.0458, -5.0125, -5.0072, -4.9287, -5.0209, -5.0372,
        -5.0294, -5.0109, -5.0749, -5.0774, -5.0551, -5.0230, -4.9546, -5.0355,
        -4.9514, -5.0215, -5.0329, -5.0142, -4.9523, -5.0537, -5.0671, -5.0318,
        -5.0894, -5.0019, -5.0205, -5.0277, -4.9608, -4.9479, -4.9819, -5.0188,
        -5.0346, -5.0617, -5.0043, -4.9750, -5.0060, -4.9762, -4.9555, -5.0211,
        -4.9650, -5.0130, -4.9971, -4.9691, -5.0214, -4.9848, -5.0402, -5.0176,
        -5.0021, -5.0755, -5.0118, -5.0288, -5.0153, -4.9983, -5.0189, -4.9564,
        -4.9752, -4.9709, -4.9840, -5.0463, -4.9700, -5.0691, -4.9562],
       device='cuda:0', grad_fn=<SelectBackward0>)
```

```
ConvolutionalNetwork(
  (conv1): Conv2d(3, 64, kernel_size=(5, 5), stride=(1, 1))
  (conv2): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv3): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv4): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
```

```
   (fc1): Linear(in_features=3200, out_features=151, bias=True)
 )

 [{'val_loss': 5.015295505523682, 'val_acc': 0.01875000074505806}]

627

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [0], train_loss: 4.9533, val_loss: 4.7652, val_acc: 0.0625

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [1], train_loss: 4.5783, val_loss: 4.5370, val_acc: 0.1330

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [2], train_loss: 4.3737, val_loss: 4.3963, val_acc: 0.1368

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [3], train_loss: 4.1503, val_loss: 4.1720, val_acc: 0.1767

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [4], train_loss: 3.9079, val_loss: 4.0336, val_acc: 0.2260

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [5], train_loss: 3.6745, val_loss: 3.8139, val_acc: 0.2573

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [6], train_loss: 3.4768, val_loss: 3.7119, val_acc: 0.3160

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [7], train_loss: 3.2964, val_loss: 3.6458, val_acc: 0.3465

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [8], train_loss: 3.1289, val_loss: 3.6842, val_acc: 0.3340

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [9], train_loss: 2.9962, val_loss: 3.5448, val_acc: 0.3927

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [10], train_loss: 2.8369, val_loss: 3.5569, val_acc: 0.3858

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [11], train_loss: 2.7003, val_loss: 3.5783, val_acc: 0.3639

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [12], train_loss: 2.5653, val_loss: 3.4606, val_acc: 0.4059

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [13], train_loss: 2.4460, val_loss: 3.5204, val_acc: 0.4146
```

```
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [14], train_loss: 2.3380, val_loss: 3.5230, val_acc: 0.4194
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [15], train_loss: 2.2209, val_loss: 3.4943, val_acc: 0.4132
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [16], train_loss: 2.1072, val_loss: 3.4380, val_acc: 0.4444
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [17], train_loss: 1.9845, val_loss: 3.5110, val_acc: 0.4035
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [18], train_loss: 1.8794, val_loss: 3.6100, val_acc: 0.4184
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [19], train_loss: 1.7698, val_loss: 3.5710, val_acc: 0.4316
```





```
{'val_loss': 3.704805850982666, 'val_acc': 0.4520833492279053}
```

1.1 FLOPs

```
+ Number of FLOPs: 0.69G
```

Animal_Classification-activation

June 10, 2024

1 Changing Activation Function

From the previous model, we have changed the activiation function from relu to elu.

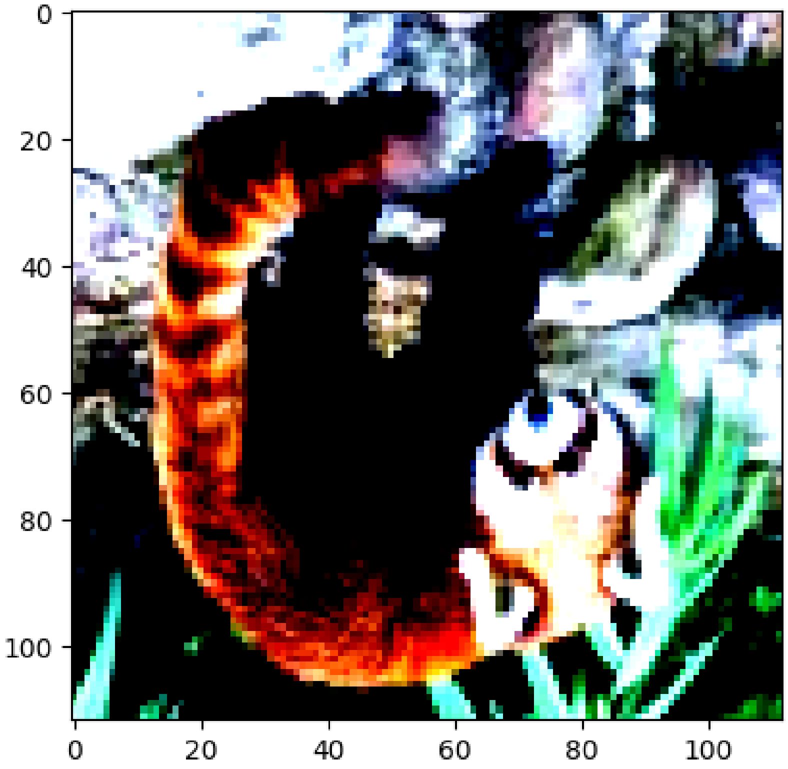
Validation Loss: 3.429 Validation Accuracy: 48.80% FLOPS: 0.69G

```
Size of training dataset : 6270
```

```
torch.Size([3, 112, 112])
```

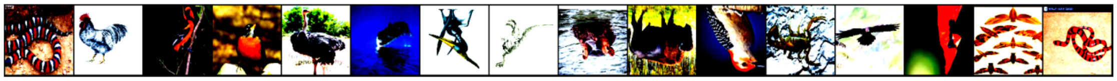
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

```
Label:  ailurus-fulgens (5)
```



```
(5330, 313, 627)
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```



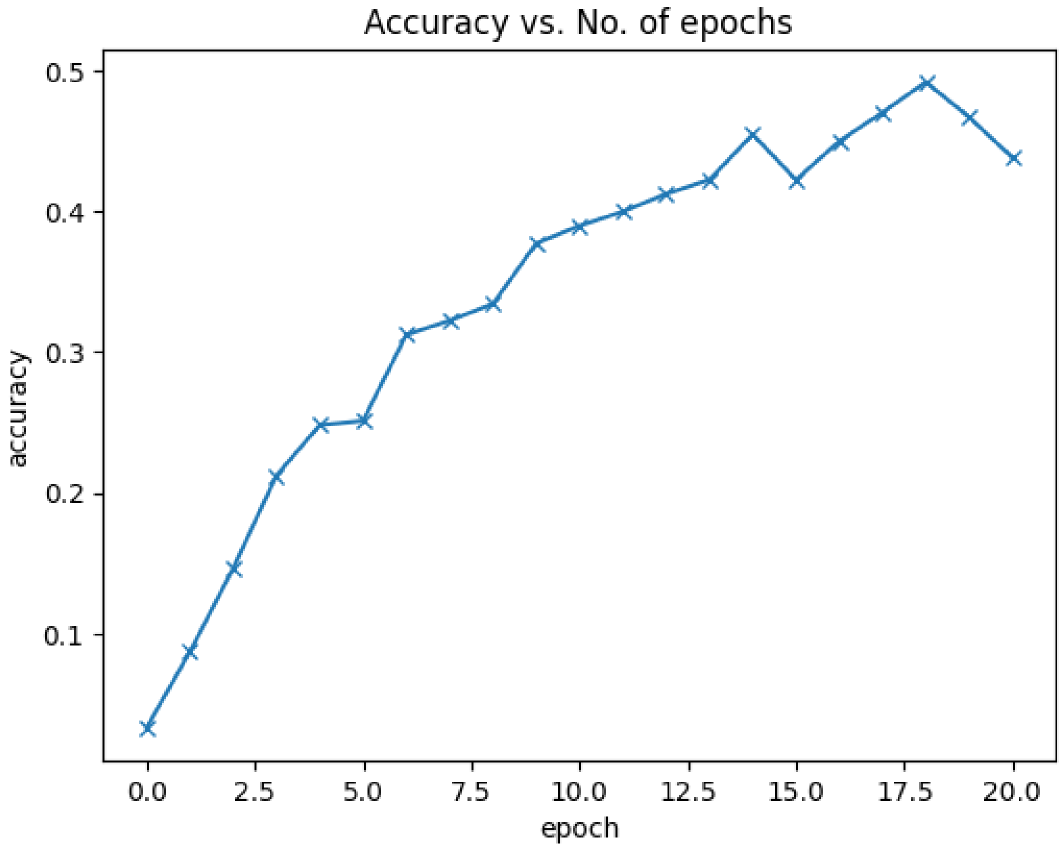
```
ConvolutionalNetwork(
  (conv1): Conv2d(3, 64, kernel_size=(5, 5), stride=(1, 1))
  (conv2): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv3): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv4): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
  (fc1): Linear(in_features=3200, out_features=151, bias=True)
)
```

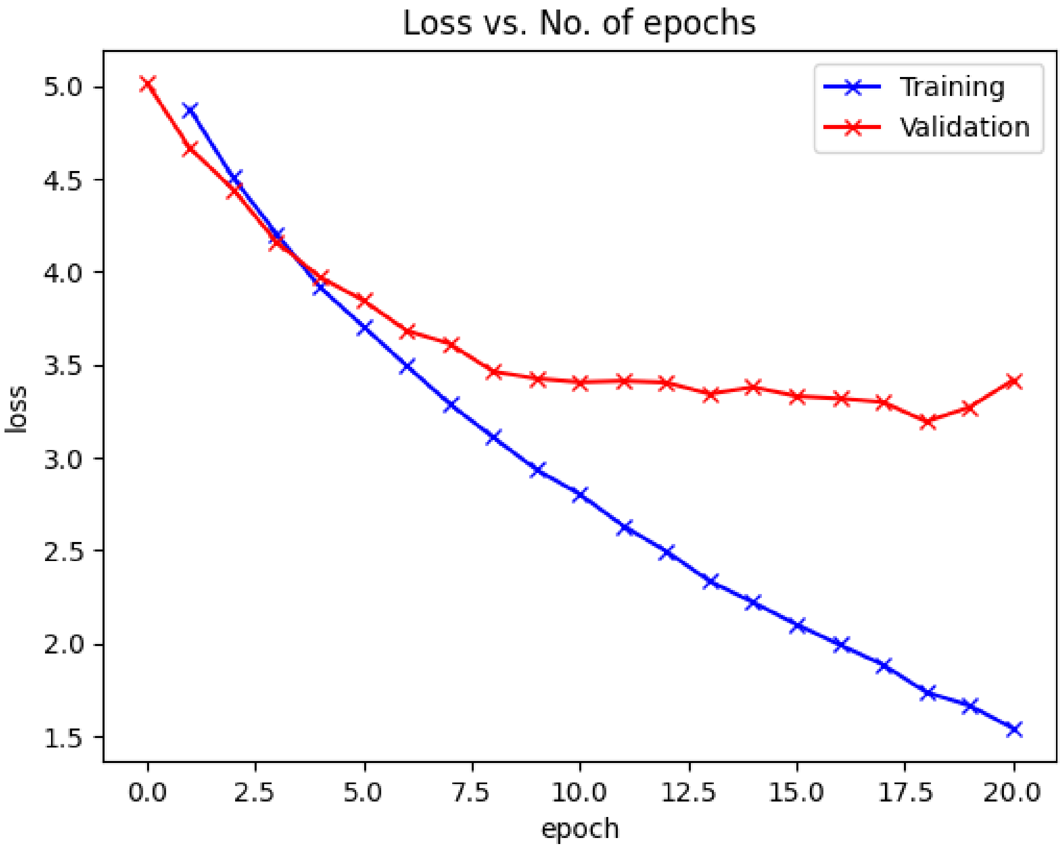
```
images.shape: torch.Size([16, 3, 112, 112])
out.shape: torch.Size([16, 151])
out[0]: tensor([-5.0489, -5.0781, -4.9266, -5.1204, -5.0625, -5.0855, -4.9174,
-4.9842,
        -5.1094, -5.0609, -5.1220, -5.0109, -4.9250, -5.0083, -5.0407, -5.1288,
        -5.0546, -5.0991, -5.0241, -5.1131, -5.0059, -5.0717, -5.0266, -4.9116,
        -5.0760, -4.9464, -5.0019, -5.1417, -5.0238, -4.9832, -5.0143, -4.9214,
        -5.0793, -5.0900, -4.9990, -4.9707, -4.9987, -4.9166, -4.8067, -5.0662,
        -5.0293, -5.0226, -4.9846, -5.0890, -4.8706, -5.0841, -5.0358, -5.0542,
        -5.0192, -5.0265, -5.0231, -5.0679, -5.0016, -5.0581, -5.0925, -5.0918,
        -5.0381, -5.0127, -5.0724, -4.9524, -5.1009, -5.0825, -5.0414, -5.0108,
        -5.1328, -5.0444, -4.9628, -5.0557, -5.0541, -5.0662, -5.0348, -4.9805,
        -5.0027, -5.0458, -5.1196, -5.0853, -5.0458, -5.0245, -4.9233, -4.9343,
        -5.1627, -4.9826, -5.0540, -5.0019, -4.9863, -4.9165, -5.0167, -4.9963,
        -5.0581, -5.0338, -5.0931, -4.9972, -4.9782, -4.8560, -4.9929, -4.9815,
        -5.0214, -5.0012, -5.0680, -5.1028, -5.1095, -5.0180, -4.9250, -5.0122,
        -4.9136, -5.0391, -5.0391, -5.0373, -4.8550, -5.0919, -5.1288, -4.9897,
        -5.2032, -4.9982, -5.0336, -5.0183, -4.9084, -4.9526, -4.9259, -5.1717,
        -5.0815, -5.0431, -5.0133, -4.9156, -5.0519, -4.9967, -4.9056, -4.9836,
        -4.9027, -4.9574, -5.0047, -5.0021, -5.0542, -5.0065, -5.0082, -5.0182,
        -4.9585, -5.1772, -5.0208, -5.0635, -5.1108, -4.9964, -5.0044, -4.8936,
        -4.9402, -4.9707, -4.9313, -5.0254, -4.9108, -5.0988, -4.9118],
       device='cuda:0', grad_fn=<SelectBackward0>)
```

```
ConvolutionalNetwork(
  (conv1): Conv2d(3, 64, kernel_size=(5, 5), stride=(1, 1))
  (conv2): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv3): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv4): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1))
```

```
   (fc1): Linear(in_features=3200, out_features=151, bias=True)
 )

 [{'val_loss': 5.017691135406494, 'val_acc': 0.0329861119389534}]

627

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [0], train_loss: 4.8735, val_loss: 4.6650, val_acc: 0.0868

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [1], train_loss: 4.5052, val_loss: 4.4427, val_acc: 0.1462

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [2], train_loss: 4.2044, val_loss: 4.1597, val_acc: 0.2122

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [3], train_loss: 3.9166, val_loss: 3.9708, val_acc: 0.2479

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [4], train_loss: 3.7063, val_loss: 3.8459, val_acc: 0.2510

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [5], train_loss: 3.4947, val_loss: 3.6830, val_acc: 0.3122

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [6], train_loss: 3.2883, val_loss: 3.6121, val_acc: 0.3222

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [7], train_loss: 3.1090, val_loss: 3.4627, val_acc: 0.3340

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [8], train_loss: 2.9350, val_loss: 3.4247, val_acc: 0.3771

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [9], train_loss: 2.8027, val_loss: 3.4052, val_acc: 0.3896

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [10], train_loss: 2.6328, val_loss: 3.4121, val_acc: 0.3997

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [11], train_loss: 2.4950, val_loss: 3.4036, val_acc: 0.4122

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [12], train_loss: 2.3345, val_loss: 3.3422, val_acc: 0.4222

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [13], train_loss: 2.2208, val_loss: 3.3792, val_acc: 0.4545
```

```
  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [14], train_loss: 2.1022, val_loss: 3.3305, val_acc: 0.4222

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [15], train_loss: 1.9924, val_loss: 3.3173, val_acc: 0.4497

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [16], train_loss: 1.8851, val_loss: 3.2981, val_acc: 0.4701

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [17], train_loss: 1.7369, val_loss: 3.1934, val_acc: 0.4917

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [18], train_loss: 1.6652, val_loss: 3.2711, val_acc: 0.4667

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [19], train_loss: 1.5429, val_loss: 3.4164, val_acc: 0.4378
```





```
{'val_loss': 3.429217576980591, 'val_acc': 0.4880208671092987}
```

1.1 FLOPs

```
+ Number of FLOPs: 0.69G
```

Animal_Classification-transfer-VGG16

June 10, 2024

1 Changing to transfer learning with VGG16

From the previous model, we have swapped out the model for VGG16 that has been pretrained and tweaking it to use our data and classes.

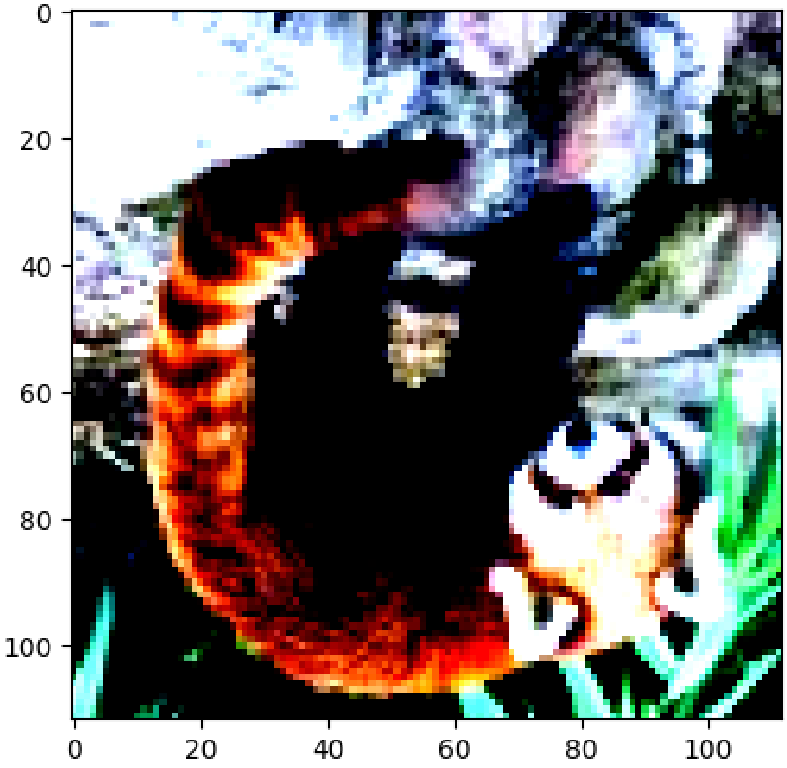
Validation Loss: 1.177 Validation Accuracy: 85.00% FLOPS: 7.89G

```
Size of training dataset : 6270
```

```
torch.Size([3, 112, 112])
```

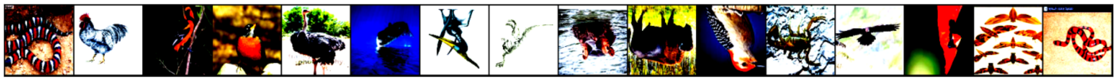
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

```
Label:  ailurus-fulgens (5)
```



```
(5330, 313, 627)
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

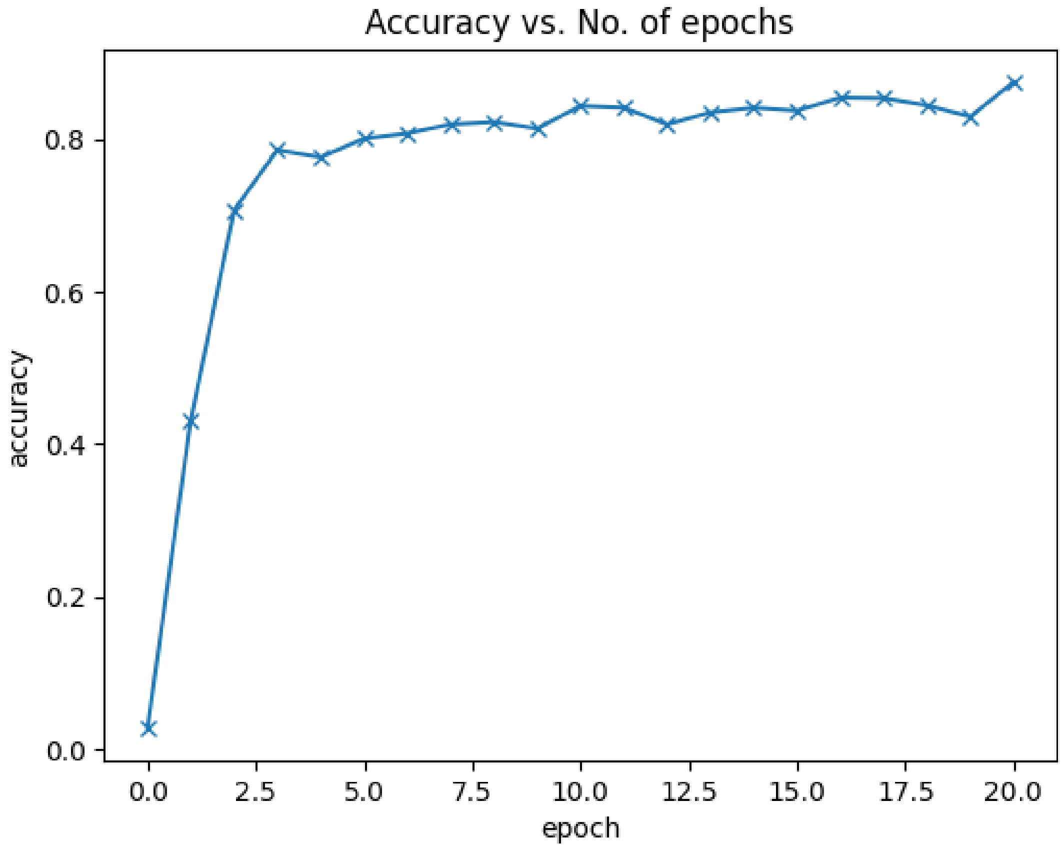


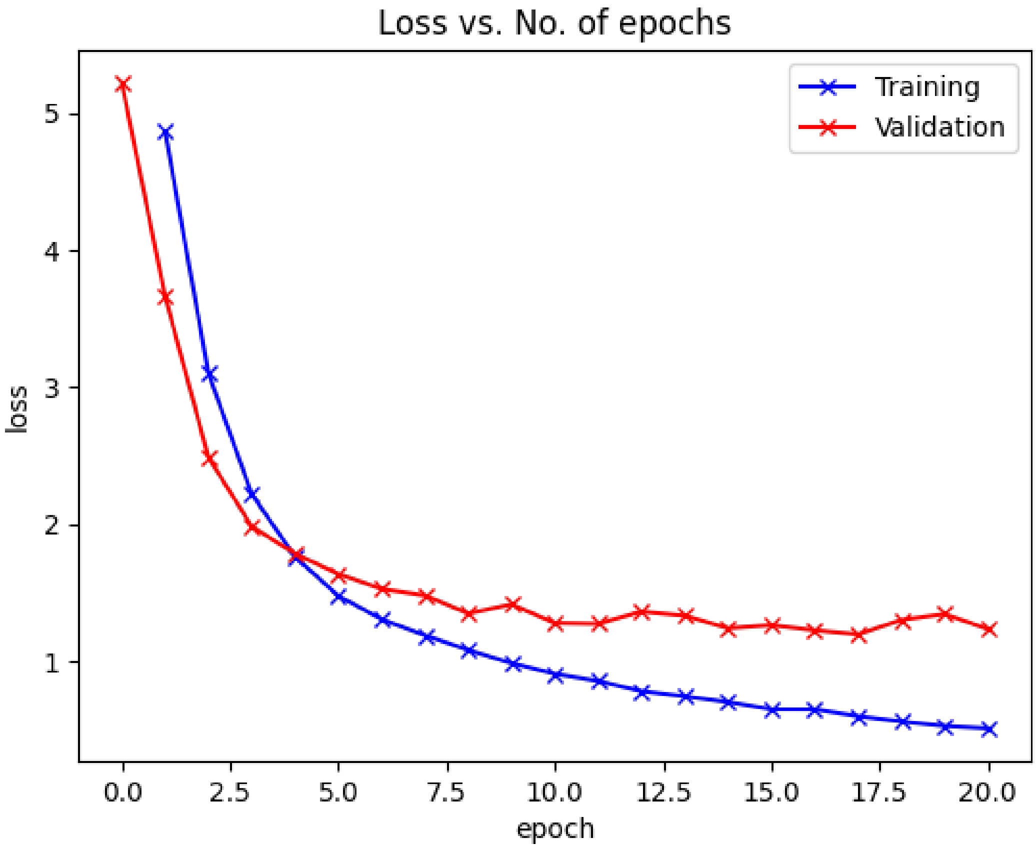
```
ConvolutionalNetwork(
  (model): VGG(
    (features): Sequential(
      (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): ReLU(inplace=True)
      (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (3): ReLU(inplace=True)
      (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
      (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (6): ReLU(inplace=True)
      (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (8): ReLU(inplace=True)
      (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
      (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (11): ReLU(inplace=True)
      (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (13): ReLU(inplace=True)
      (14): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (15): ReLU(inplace=True)
      (16): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
      (17): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (18): ReLU(inplace=True)
      (19): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (20): ReLU(inplace=True)
      (21): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (22): ReLU(inplace=True)
      (23): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
      (24): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (25): ReLU(inplace=True)
      (26): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (27): ReLU(inplace=True)
      (28): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (29): ReLU(inplace=True)
      (30): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    )
    (avgpool): AdaptiveAvgPool2d(output_size=(7, 7))
    (classifier): Sequential(
      (0): Linear(in_features=25088, out_features=4096, bias=True)
      (1): ReLU(inplace=True)
      (2): Dropout(p=0.5, inplace=False)
      (3): Linear(in_features=4096, out_features=4096, bias=True)
      (4): ReLU(inplace=True)
      (5): Dropout(p=0.5, inplace=False)
      (6): Linear(in_features=4096, out_features=151, bias=True)
    )
  )
)

images.shape: torch.Size([16, 3, 112, 112])
out.shape: torch.Size([16, 151])
out[0]: tensor([-5.0155, -4.8244, -6.4500, -5.2882, -5.6761, -5.2723, -8.1427,
-6.9423,
        -6.1285, -3.6968, -4.2547, -7.4687, -4.7189, -4.3489, -5.8139, -3.6071,
        -6.2837, -5.6112, -6.3119, -6.4560, -5.2747, -6.3577, -5.3859, -6.2020,
        -6.2545, -6.1579, -5.7563, -6.2114, -6.6234, -4.3613, -5.2160, -4.1965,
        -6.5862, -5.6447, -6.0422, -3.9137, -5.6003, -7.0435, -5.8277, -5.6527,
        -5.5874, -4.9604, -4.8894, -3.4264, -5.5862, -6.0568, -6.7974, -5.1861,
        -5.1507, -4.9369, -6.1578, -3.7941, -5.7377, -6.2363, -7.3798, -5.4343,
        -5.1280, -5.1943, -6.9009, -4.4677, -5.4308, -4.5975, -4.8791, -7.1803,
        -3.5253, -6.1256, -6.0084, -8.1544, -7.0820, -5.6563, -6.5758, -4.9197,
        -5.3123, -6.0809, -5.0418, -5.8404, -2.6726, -2.7620, -6.7194, -6.2637,
        -5.9573, -6.1571, -7.9727, -4.3419, -5.5299, -5.2522, -4.0395, -5.0515,
        -4.6345, -5.2396, -6.7284, -6.3737, -5.1581, -5.6935, -6.9317, -4.8901,
        -6.6558, -4.2015, -6.1732, -3.9284, -5.0238, -5.1499, -6.3977, -4.0645,
        -5.7820, -4.9934, -5.7307, -4.7646, -6.9689, -7.1588, -5.8584, -8.3142,
        -5.6302, -4.2608, -4.5197, -5.6761, -6.1620, -5.3299, -5.2093, -5.0674,
        -5.9995, -7.9803, -6.6908, -4.9153, -7.8850, -7.6099, -6.4608, -7.0354,
        -3.6432, -5.3385, -6.1655, -4.8455, -5.6094, -5.3136, -3.9510, -6.8466,
        -4.6274, -4.3193, -5.6198, -5.2383, -6.9402, -7.3852, -3.6571, -6.4589,
        -5.5141, -7.1587, -5.3708, -5.7735, -5.2962, -4.1481, -6.7642],
       device='cuda:0', grad_fn=<SelectBackward0>)

ConvolutionalNetwork(
  (model): VGG(
    (features): Sequential(
      (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): ReLU(inplace=True)
      (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (3): ReLU(inplace=True)
      (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
      (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (6): ReLU(inplace=True)
      (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (8): ReLU(inplace=True)
      (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
      (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (11): ReLU(inplace=True)
      (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (13): ReLU(inplace=True)
      (14): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (15): ReLU(inplace=True)
      (16): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
      (17): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (18): ReLU(inplace=True)
      (19): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (20): ReLU(inplace=True)
      (21): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (22): ReLU(inplace=True)
      (23): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
      (24): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (25): ReLU(inplace=True)
      (26): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (27): ReLU(inplace=True)
      (28): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (29): ReLU(inplace=True)
      (30): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    )
    (avgpool): AdaptiveAvgPool2d(output_size=(7, 7))
    (classifier): Sequential(
      (0): Linear(in_features=25088, out_features=4096, bias=True)
      (1): ReLU(inplace=True)
      (2): Dropout(p=0.5, inplace=False)
      (3): Linear(in_features=4096, out_features=4096, bias=True)
      (4): ReLU(inplace=True)
      (5): Dropout(p=0.5, inplace=False)
      (6): Linear(in_features=4096, out_features=151, bias=True)
    )
  )
)
```

```
[{'val_loss': 5.219048500061035, 'val_acc': 0.02743055485188961}]
```

```
627
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [0], train_loss: 4.8645, val_loss: 3.6620, val_acc: 0.4309
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [1], train_loss: 3.1019, val_loss: 2.4879, val_acc: 0.7069
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [2], train_loss: 2.2207, val_loss: 1.9842, val_acc: 0.7858
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [3], train_loss: 1.7636, val_loss: 1.7860, val_acc: 0.7764
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [4], train_loss: 1.4785, val_loss: 1.6382, val_acc: 0.8007
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [5], train_loss: 1.3082, val_loss: 1.5291, val_acc: 0.8076
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [6], train_loss: 1.1912, val_loss: 1.4818, val_acc: 0.8194
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [7], train_loss: 1.0833, val_loss: 1.3501, val_acc: 0.8226
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [8], train_loss: 0.9882, val_loss: 1.4161, val_acc: 0.8139
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [9], train_loss: 0.9103, val_loss: 1.2807, val_acc: 0.8438
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [10], train_loss: 0.8582, val_loss: 1.2756, val_acc: 0.8413
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [11], train_loss: 0.7840, val_loss: 1.3637, val_acc: 0.8194
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [12], train_loss: 0.7465, val_loss: 1.3341, val_acc: 0.8351
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [13], train_loss: 0.7050, val_loss: 1.2432, val_acc: 0.8413
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [14], train_loss: 0.6524, val_loss: 1.2659, val_acc: 0.8375
  0%|          | 0/334 [00:00<?, ?it/s]
```

```
Epoch [15], train_loss: 0.6518, val_loss: 1.2282, val_acc: 0.8545
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [16], train_loss: 0.6015, val_loss: 1.1983, val_acc: 0.8538
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [17], train_loss: 0.5634, val_loss: 1.3028, val_acc: 0.8444
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [18], train_loss: 0.5312, val_loss: 1.3459, val_acc: 0.8295
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [19], train_loss: 0.5133, val_loss: 1.2373, val_acc: 0.8750
```





```
{'val_loss': 1.1766126155853271, 'val_acc': 0.8500000238418579}
```

1.1 FLOPs

```
+ Number of FLOPs: 7.89G
```

Animal_Classification-transfer-enet-s

June 13, 2024

1 Changing to transfer learning with efficientnet v2 s

From the previous model, we have swapped out the model for efficientnet v2 s that has been pretrained and tweaking it to use our data and classes.

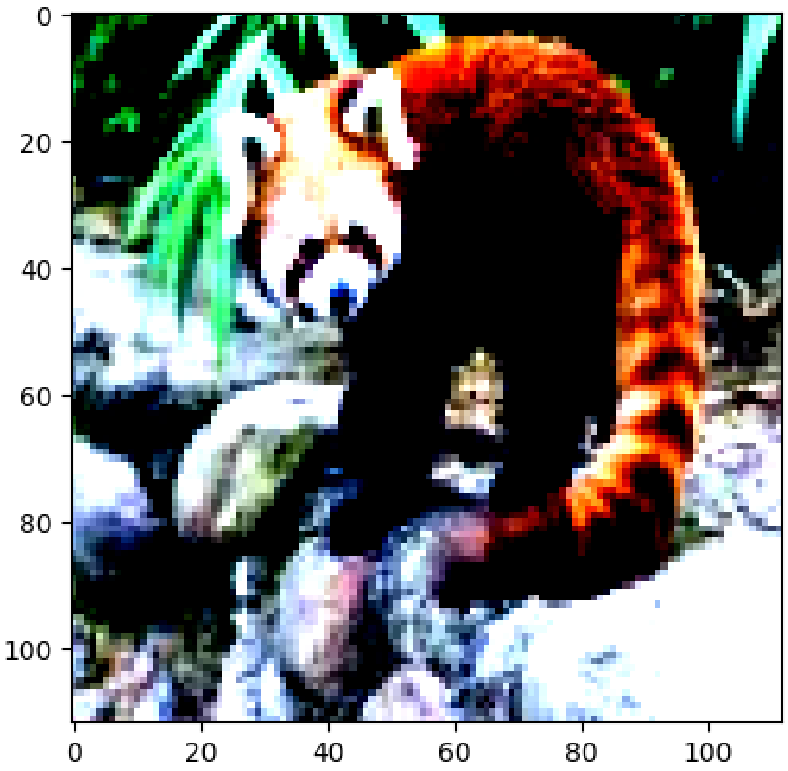
Validation Loss: 0.6571 Validation Accuracy: 93.44% FLOPS: 1.51G

```
Size of training dataset : 6270
```

```
torch.Size([3, 112, 112])
```

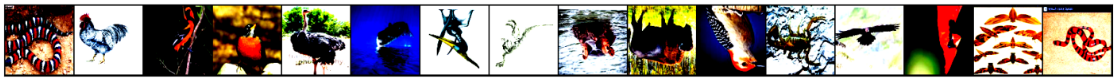
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

```
Label:  ailurus-fulgens (5)
```



```
(5330, 313, 627)
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```



```
ConvolutionalNetwork(
  (model): EfficientNet(
    (features): Sequential(
      (0): Conv2dNormActivation(
        (0): Conv2d(3, 24, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
        (1): BatchNorm2d(24, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
        (2): SiLU(inplace=True)
      )
      (1): Sequential(
        (0): FusedMBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(24, 24, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
              (1): BatchNorm2d(24, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.0, mode=row)
        )
        (1): FusedMBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(24, 24, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
              (1): BatchNorm2d(24, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.005, mode=row)
        )
      )
      (2): Sequential(
        (0): FusedMBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(24, 96, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
              (1): BatchNorm2d(96, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(96, 48, kernel_size=(1, 1), stride=(1, 1), bias=False)
              (1): BatchNorm2d(48, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.01, mode=row)
        )
        (1): FusedMBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(48, 192, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
              (1): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(192, 48, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(48, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.015000000000000003, mode=row)
        )
        (2): FusedMBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(48, 192, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
              (1): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(192, 48, kernel_size=(1, 1), stride=(1, 1),
bias=False)
```

```
              (1): BatchNorm2d(48, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.02, mode=row)
        )
        (3): FusedMBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(48, 192, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
              (1): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(192, 48, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(48, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.025, mode=row)
        )
      )
      (3): Sequential(
        (0): FusedMBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(48, 192, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1), bias=False)
              (1): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(192, 64, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(64, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.030000000000000006, mode=row)
        )
        (1): FusedMBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(64, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(64, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.035, mode=row)
        )
        (2): FusedMBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(64, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(64, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.04, mode=row)
        )
        (3): FusedMBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(64, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(64, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
```

```
            )
          )
          (stochastic_depth): StochasticDepth(p=0.045, mode=row)
        )
      )
      (4): Sequential(
        (0): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(256, 256, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1), groups=256, bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(256, 16, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(16, 256, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.05, mode=row)
        )
        (1): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(512, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=512, bias=False)
              (1): BatchNorm2d(512, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(512, 32, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(32, 512, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.05500000000000001, mode=row)
        )
        (2): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(512, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=512, bias=False)
              (1): BatchNorm2d(512, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(512, 32, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(32, 512, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.06000000000000001, mode=row)
        )
        (3): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(512, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=512, bias=False)
              (1): BatchNorm2d(512, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(512, 32, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(32, 512, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.065, mode=row)
        )
        (4): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(512, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
```

```
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=512, bias=False)
              (1): BatchNorm2d(512, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(512, 32, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(32, 512, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.07, mode=row)
        )
        (5): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(512, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=512, bias=False)
              (1): BatchNorm2d(512, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(512, 32, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(32, 512, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.075, mode=row)
        )
      )
      (5): Sequential(
        (0): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(128, 768, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(768, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(768, 768, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=768, bias=False)
              (1): BatchNorm2d(768, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(768, 32, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(32, 768, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(768, 160, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(160, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.08, mode=row)
        )
        (1): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=960, bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(960, 40, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(40, 960, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(160, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.085, mode=row)
        )
        (2): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=960, bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(960, 40, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(40, 960, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(160, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.09, mode=row)
        )
        (3): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=960, bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(960, 40, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(40, 960, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(160, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.095, mode=row)
        )
        (4): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=960, bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(960, 40, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(40, 960, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(160, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.1, mode=row)
        )
        (5): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=960, bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(960, 40, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(40, 960, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(160, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.10500000000000001, mode=row)
        )
        (6): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=960, bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(960, 40, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(40, 960, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(160, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.11000000000000001, mode=row)
        )
        (7): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=960, bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(960, 40, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(40, 960, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(160, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.11500000000000002, mode=row)
        )
        (8): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=960, bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
```

```
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(960, 40, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(40, 960, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(160, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.12000000000000002, mode=row)
        )
      )
      (6): Sequential(
        (0): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1), groups=960, bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(960, 40, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(40, 960, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(960, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.125, mode=row)
        )
        (1): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.13, mode=row)
        )
        (2): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.135, mode=row)
        )
        (3): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.14, mode=row)
        )
        (4): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.14500000000000002, mode=row)
        )
        (5): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.15, mode=row)
        )
        (6): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.155, mode=row)
        )
        (7): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.16, mode=row)
        )
        (8): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.165, mode=row)
        )
        (9): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.17, mode=row)
        )
        (10): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.175, mode=row)
        )
        (11): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.18, mode=row)
        )
        (12): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.185, mode=row)
        )
        (13): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.19, mode=row)
        )
        (14): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.195, mode=row)
        )
      )
      (7): Conv2dNormActivation(
        (0): Conv2d(256, 1280, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (1): BatchNorm2d(1280, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
        (2): SiLU(inplace=True)
      )
    )
    (avgpool): AdaptiveAvgPool2d(output_size=1)
    (classifier): Sequential(
      (0): Dropout(p=0.2, inplace=True)
      (1): Linear(in_features=1280, out_features=151, bias=True)
    )
  )
)
```

```
images.shape: torch.Size([16, 3, 112, 112])
out.shape: torch.Size([16, 151])
```

```
out[0]: tensor([-4.7994, -5.1829, -5.0027, -4.9127, -4.6998, -4.8673, -5.1724,
-4.9495,
        -4.7703, -4.9698, -5.1741, -5.1912, -5.0215, -4.8155, -5.0213, -5.2369,
        -4.8082, -5.2048, -5.1846, -4.8295, -4.9353, -5.2770, -4.9934, -5.1266,
        -4.6548, -4.9796, -5.0709, -5.0723, -4.7300, -5.0411, -5.0543, -4.9714,
        -5.1474, -4.8505, -4.9984, -5.0507, -4.8566, -5.0439, -4.9996, -5.1005,
        -5.1156, -5.0616, -5.0866, -5.0087, -5.2058, -5.1470, -4.8633, -5.0163,
        -4.9701, -4.8954, -4.9969, -5.2226, -5.0340, -5.0524, -5.0602, -4.9395,
        -5.2437, -4.9152, -4.7581, -4.9960, -5.0949, -5.0790, -4.9998, -5.0557,
        -5.1472, -5.0925, -5.0727, -4.8898, -4.9881, -4.9445, -5.0515, -5.1039,
        -4.9463, -4.9599, -4.9802, -5.0435, -5.2297, -4.9317, -5.1567, -4.9077,
        -5.1739, -4.7727, -4.9432, -4.9620, -5.1138, -5.1272, -4.9757, -5.2533,
        -5.0020, -4.8290, -5.0521, -5.1993, -4.9562, -5.3153, -4.9967, -5.0638,
        -4.8776, -5.0088, -4.6869, -5.0928, -5.4245, -5.0838, -5.0627, -5.0189,
        -5.1908, -5.1645, -4.9533, -5.1461, -5.1133, -5.0064, -5.1321, -4.9497,
        -5.1428, -5.0813, -4.9845, -5.0299, -4.8336, -4.9603, -5.1759, -5.0610,
        -4.9300, -4.6662, -4.9472, -4.9394, -5.1029, -4.7909, -5.1264, -5.1141,
        -4.9729, -4.8760, -5.1422, -5.1973, -4.8764, -5.1530, -5.2282, -4.9745,
        -5.1271, -5.0359, -4.9405, -5.0989, -5.3952, -5.2781, -4.9934, -4.8897,
        -5.0299, -5.2391, -5.0535, -4.9889, -5.0309, -4.9962, -4.9986],
       device='cuda:0', grad_fn=<SelectBackward0>)
```

```
ConvolutionalNetwork(
  (model): EfficientNet(
    (features): Sequential(
      (0): Conv2dNormActivation(
        (0): Conv2d(3, 24, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
        (1): BatchNorm2d(24, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
        (2): SiLU(inplace=True)
      )
      (1): Sequential(
        (0): FusedMBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(24, 24, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
              (1): BatchNorm2d(24, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.0, mode=row)
        )
        (1): FusedMBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(24, 24, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
              (1): BatchNorm2d(24, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.005, mode=row)
        )
      )
      (2): Sequential(
        (0): FusedMBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(24, 96, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
              (1): BatchNorm2d(96, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(96, 48, kernel_size=(1, 1), stride=(1, 1), bias=False)
              (1): BatchNorm2d(48, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.01, mode=row)
        )
        (1): FusedMBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(48, 192, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
              (1): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(192, 48, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(48, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.015000000000000003, mode=row)
        )
        (2): FusedMBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(48, 192, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
              (1): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(192, 48, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(48, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.02, mode=row)
        )
        (3): FusedMBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(48, 192, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
              (1): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(192, 48, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(48, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.025, mode=row)
        )
      )
      (3): Sequential(
        (0): FusedMBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(48, 192, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1), bias=False)
              (1): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(192, 64, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(64, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.030000000000000006, mode=row)
        )
        (1): FusedMBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(64, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(64, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.035, mode=row)
        )
        (2): FusedMBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(64, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(64, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.04, mode=row)
        )
        (3): FusedMBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(64, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(64, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.045, mode=row)
        )
      )
      (4): Sequential(
        (0): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(256, 256, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1), groups=256, bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(256, 16, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(16, 256, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.05, mode=row)
        )
        (1): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(512, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=512, bias=False)
              (1): BatchNorm2d(512, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(512, 32, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(32, 512, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.05500000000000001, mode=row)
        )
        (2): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(512, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=512, bias=False)
```

```
              (1): BatchNorm2d(512, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(512, 32, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(32, 512, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.06000000000000001, mode=row)
        )
        (3): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(512, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=512, bias=False)
              (1): BatchNorm2d(512, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(512, 32, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(32, 512, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.065, mode=row)
        )
        (4): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(512, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=512, bias=False)
              (1): BatchNorm2d(512, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(512, 32, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(32, 512, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.07, mode=row)
        )
        (5): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(512, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=512, bias=False)
              (1): BatchNorm2d(512, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(512, 32, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(32, 512, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.075, mode=row)
        )
      )
      (5): Sequential(
        (0): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(128, 768, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(768, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(768, 768, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=768, bias=False)
              (1): BatchNorm2d(768, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(768, 32, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(32, 768, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(768, 160, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(160, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.08, mode=row)
        )
        (1): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=960, bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(960, 40, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(40, 960, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(160, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.085, mode=row)
        )
        (2): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
       (2): SiLU(inplace=True)
      )
      (1): Conv2dNormActivation(
       (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=960, bias=False)
       (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
       (2): SiLU(inplace=True)
      )
      (2): SqueezeExcitation(
       (avgpool): AdaptiveAvgPool2d(output_size=1)
       (fc1): Conv2d(960, 40, kernel_size=(1, 1), stride=(1, 1))
       (fc2): Conv2d(40, 960, kernel_size=(1, 1), stride=(1, 1))
       (activation): SiLU(inplace=True)
       (scale_activation): Sigmoid()
      )
      (3): Conv2dNormActivation(
       (0): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1),
bias=False)
       (1): BatchNorm2d(160, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
      )
     )
     (stochastic_depth): StochasticDepth(p=0.09, mode=row)
    )
    (3): MBConv(
     (block): Sequential(
      (0): Conv2dNormActivation(
       (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1),
bias=False)
       (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
       (2): SiLU(inplace=True)
      )
      (1): Conv2dNormActivation(
       (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=960, bias=False)
       (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
       (2): SiLU(inplace=True)
      )
      (2): SqueezeExcitation(
       (avgpool): AdaptiveAvgPool2d(output_size=1)
       (fc1): Conv2d(960, 40, kernel_size=(1, 1), stride=(1, 1))
       (fc2): Conv2d(40, 960, kernel_size=(1, 1), stride=(1, 1))
       (activation): SiLU(inplace=True)
```

```
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(160, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.095, mode=row)
        )
        (4): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=960, bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(960, 40, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(40, 960, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(160, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.1, mode=row)
        )
        (5): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=960, bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(960, 40, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(40, 960, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(160, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.10500000000000001, mode=row)
        )
        (6): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=960, bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(960, 40, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(40, 960, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(160, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.11000000000000001, mode=row)
        )
        (7): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=960, bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(960, 40, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(40, 960, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(160, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.11500000000000002, mode=row)
        )
        (8): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=960, bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(960, 40, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(40, 960, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(160, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.12000000000000002, mode=row)
        )
      )
      (6): Sequential(
        (0): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(960, 960, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1), groups=960, bias=False)
              (1): BatchNorm2d(960, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(960, 40, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(40, 960, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(960, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.125, mode=row)
        )
        (1): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.13, mode=row)
        )
        (2): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.135, mode=row)
        )
        (3): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.14, mode=row)
        )
        (4): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
```

```
            )
          )
          (stochastic_depth): StochasticDepth(p=0.14500000000000002, mode=row)
        )
        (5): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.15, mode=row)
        )
        (6): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.155, mode=row)
        )
        (7): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
```

```
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.16, mode=row)
        )
        (8): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.165, mode=row)
        )
        (9): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.17, mode=row)
        )
        (10): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.175, mode=row)
        )
        (11): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.18, mode=row)
        )
        (12): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
```

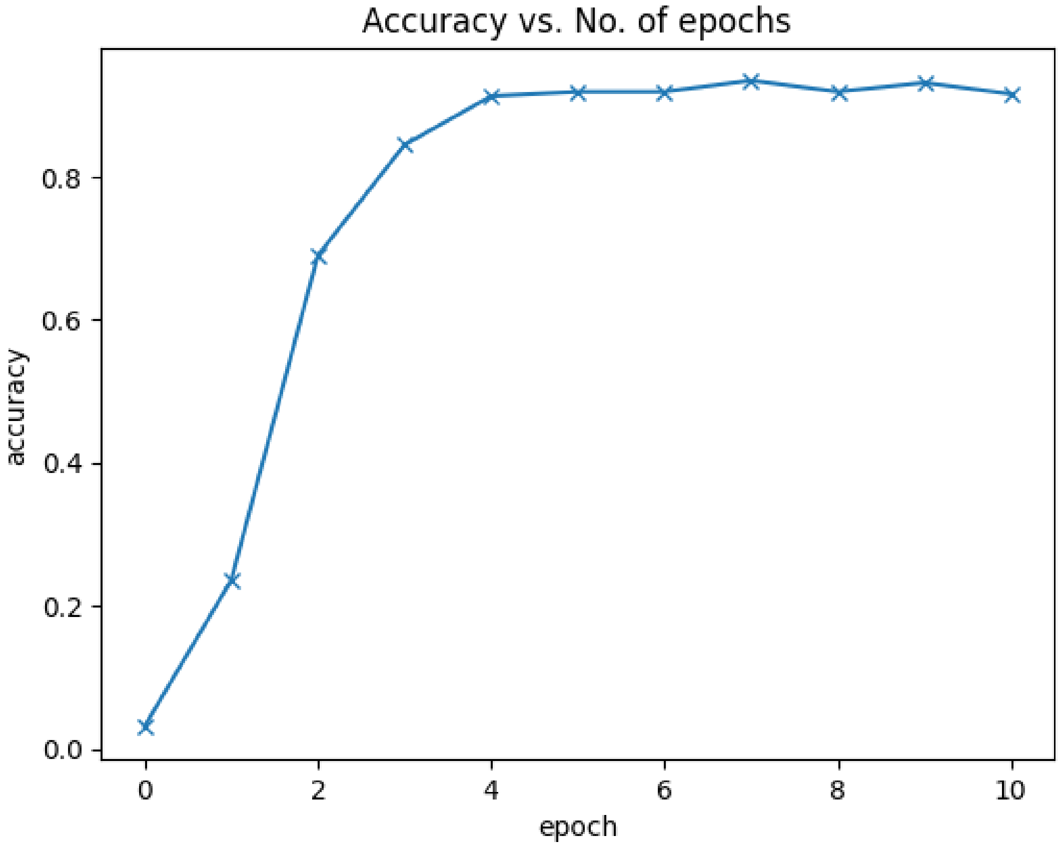
```
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.185, mode=row)
        )
        (13): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.19, mode=row)
        )
        (14): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(256, 1536, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1536, 1536, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1536, bias=False)
              (1): BatchNorm2d(1536, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1536, 64, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(64, 1536, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1536, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(256, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.195, mode=row)
        )
      )
      (7): Conv2dNormActivation(
        (0): Conv2d(256, 1280, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (1): BatchNorm2d(1280, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
```

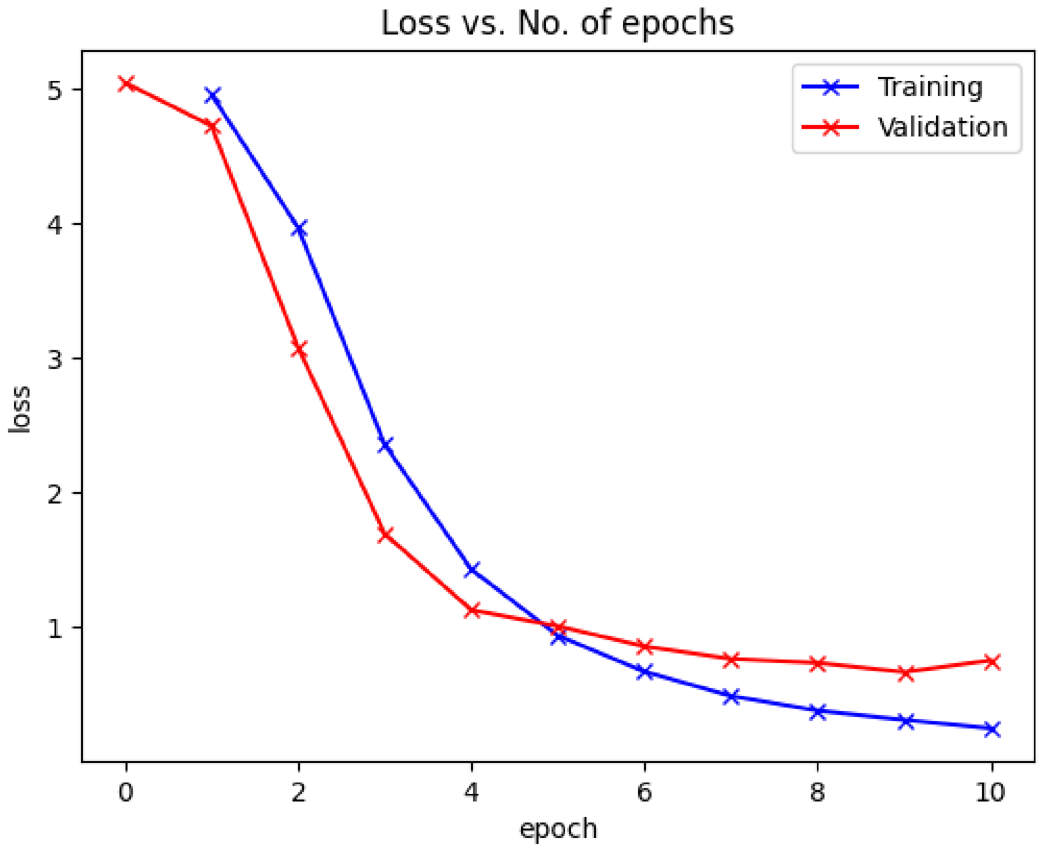
```
        (2): SiLU(inplace=True)
      )
    )
    (avgpool): AdaptiveAvgPool2d(output_size=1)
    (classifier): Sequential(
      (0): Dropout(p=0.2, inplace=True)
      (1): Linear(in_features=1280, out_features=151, bias=True)
    )
  )
)
```

```
[{'val_loss': 5.047817230224609, 'val_acc': 0.03125}]
```

```
627
```

```
  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [0], train_loss: 4.9595, val_loss: 4.7259, val_acc: 0.2354

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [1], train_loss: 3.9699, val_loss: 3.0797, val_acc: 0.6896

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [2], train_loss: 2.3567, val_loss: 1.6923, val_acc: 0.8444

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [3], train_loss: 1.4267, val_loss: 1.1240, val_acc: 0.9125

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [4], train_loss: 0.9333, val_loss: 1.0034, val_acc: 0.9187

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [5], train_loss: 0.6650, val_loss: 0.8530, val_acc: 0.9187

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [6], train_loss: 0.4843, val_loss: 0.7602, val_acc: 0.9344

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [7], train_loss: 0.3747, val_loss: 0.7302, val_acc: 0.9187

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [8], train_loss: 0.3062, val_loss: 0.6615, val_acc: 0.9313

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [9], train_loss: 0.2433, val_loss: 0.7486, val_acc: 0.9156
```


Accuracy vs. No. of epochs
accuracy
0.0
0.2
0.4
0.6
0.8
epoch
0
2
4
6
8
10



```
{'val_loss': 0.6571369171142578, 'val_acc': 0.934374988079071}
```

1.1 FLOPs

```
+ Number of FLOPs: 1.51G
```

Animal_Classification-transfer-enet

June 13, 2024

1 Changing to transfer learning with efficientnet b0

From the previous model, we have swapped out the model for efficientnet b0 that has been pretrained and tweaking it to use our data and classes.

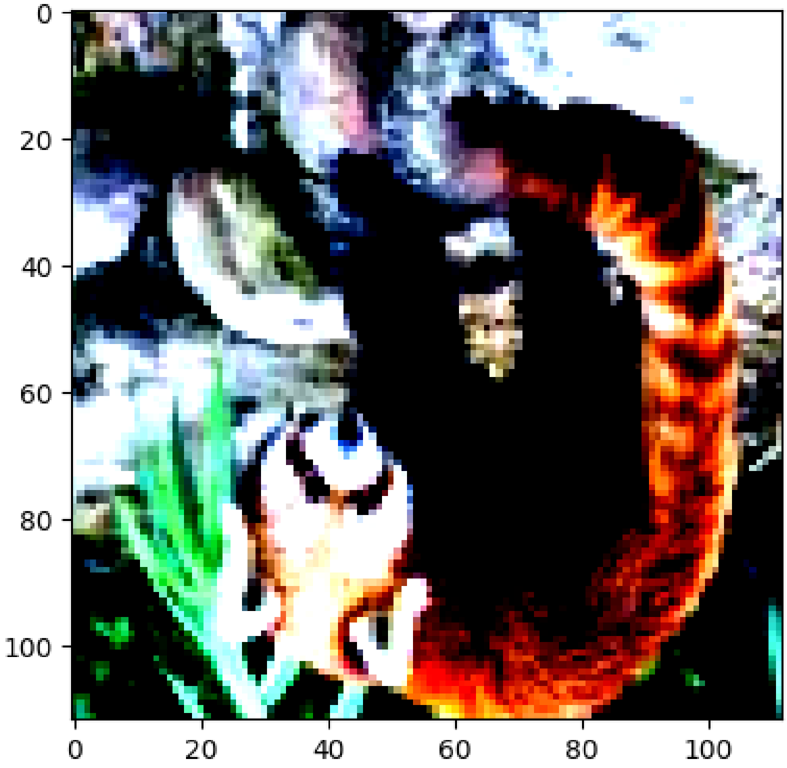
Validation Loss: 0.8605 Validation Accuracy: 91.30% FLOPS: 0.21G

```
Size of training dataset : 6270
```

```
torch.Size([3, 112, 112])
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

```
Label:  ailurus-fulgens (5)
```



```
(5330, 313, 627)
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

```
ConvolutionalNetwork(
  (model): EfficientNet(
    (features): Sequential(
      (0): Conv2dNormActivation(
        (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
        (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (2): SiLU(inplace=True)
      )
      (1): Sequential(
        (0): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=32, bias=False)
              (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(32, 8, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(8, 32, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (2): Conv2dNormActivation(
              (0): Conv2d(32, 16, kernel_size=(1, 1), stride=(1, 1), bias=False)
              (1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.0, mode=row)
        )
      )
      (2): Sequential(
        (0): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(16, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
              (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(96, 96, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), groups=96, bias=False)
              (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(96, 4, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(4, 96, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(96, 24, kernel_size=(1, 1), stride=(1, 1), bias=False)
              (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.0125, mode=row)
        )
        (1): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(24, 144, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(144, 144, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=144, bias=False)
              (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(144, 6, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(6, 144, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(144, 24, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.025, mode=row)
        )
      )
      (3): Sequential(
        (0): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(24, 144, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(144, 144, kernel_size=(5, 5), stride=(2, 2),
padding=(2, 2), groups=144, bias=False)
              (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(144, 6, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(6, 144, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(144, 40, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
```

```
            )
          )
          (stochastic_depth): StochasticDepth(p=0.037500000000000006, mode=row)
        )
        (1): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(40, 240, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(240, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(240, 240, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=240, bias=False)
              (1): BatchNorm2d(240, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(240, 10, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(10, 240, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(240, 40, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.05, mode=row)
        )
      )
      (4): Sequential(
        (0): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(40, 240, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(240, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(240, 240, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1), groups=240, bias=False)
              (1): BatchNorm2d(240, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(240, 10, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(10, 240, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(240, 80, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.0625, mode=row)
        )
        (1): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(480, 480, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=480, bias=False)
              (1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(480, 20, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(20, 480, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(480, 80, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.07500000000000001, mode=row)
        )
        (2): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(480, 480, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=480, bias=False)
              (1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(480, 20, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(20, 480, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(480, 80, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.08750000000000001, mode=row)
        )
      )
      (5): Sequential(
        (0): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1),
bias=False)
```

```
              (1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(480, 480, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=480, bias=False)
              (1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(480, 20, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(20, 480, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(480, 112, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(112, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.1, mode=row)
        )
        (1): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(112, 672, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(672, 672, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=672, bias=False)
              (1): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(672, 28, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(28, 672, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(672, 112, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(112, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.1125, mode=row)
        )
        (2): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(112, 672, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(672, 672, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=672, bias=False)
              (1): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(672, 28, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(28, 672, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(672, 112, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(112, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.125, mode=row)
        )
      )
      (6): Sequential(
        (0): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(112, 672, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(672, 672, kernel_size=(5, 5), stride=(2, 2),
padding=(2, 2), groups=672, bias=False)
              (1): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(672, 28, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(28, 672, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(672, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.1375, mode=row)
        )
        (1): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1152, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
              (1): BatchNorm2d(1152, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1152, 48, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(48, 1152, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.15000000000000002, mode=row)
        )
        (2): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1152, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
              (1): BatchNorm2d(1152, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1152, 48, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(48, 1152, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.1625, mode=row)
        )
        (3): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1152, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
              (1): BatchNorm2d(1152, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1152, 48, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(48, 1152, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.17500000000000002, mode=row)
        )
      )
      (7): Sequential(
        (0): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1152, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1152, 1152, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1152, bias=False)
```

```
       (1): BatchNorm2d(1152, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
       (2): SiLU(inplace=True)
      )
      (2): SqueezeExcitation(
       (avgpool): AdaptiveAvgPool2d(output_size=1)
       (fc1): Conv2d(1152, 48, kernel_size=(1, 1), stride=(1, 1))
       (fc2): Conv2d(48, 1152, kernel_size=(1, 1), stride=(1, 1))
       (activation): SiLU(inplace=True)
       (scale_activation): Sigmoid()
      )
      (3): Conv2dNormActivation(
       (0): Conv2d(1152, 320, kernel_size=(1, 1), stride=(1, 1),
bias=False)
       (1): BatchNorm2d(320, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
     )
     (stochastic_depth): StochasticDepth(p=0.1875, mode=row)
    )
   )
   (8): Conv2dNormActivation(
    (0): Conv2d(320, 1280, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (1): BatchNorm2d(1280, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (2): SiLU(inplace=True)
   )
  )
  (avgpool): AdaptiveAvgPool2d(output_size=1)
  (classifier): Sequential(
   (0): Dropout(p=0.2, inplace=True)
   (1): Linear(in_features=1280, out_features=151, bias=True)
  )
 )
)

images.shape: torch.Size([8, 3, 112, 112])
out.shape: torch.Size([8, 151])
out[0]: tensor([-5.3306, -5.1478, -4.6661, -4.6892, -4.8743, -5.4414, -5.1203,
-5.2616,
    -5.0763, -5.2150, -4.8569, -4.2328, -4.2651, -4.9740, -5.8967, -4.8240,
    -4.7153, -5.2438, -5.9391, -4.8895, -4.1349, -5.4026, -4.4439, -5.4200,
    -4.7755, -5.4014, -4.2853, -5.0213, -5.3200, -5.0363, -4.8848, -5.0683,
    -4.9752, -4.8050, -5.2937, -4.3433, -5.4060, -4.4473, -4.7193, -5.4616,
    -4.9635, -5.3621, -5.7129, -5.0883, -4.8706, -5.0065, -4.6918, -4.7496,
    -4.8032, -5.0112, -4.9375, -5.1954, -5.3248, -4.8421, -5.0990, -5.6386,
    -5.3078, -5.0376, -5.0106, -5.0876, -5.7190, -5.3046, -5.2521, -4.9910,
    -5.1990, -5.4892, -4.5919, -5.4890, -5.5808, -5.3988, -5.6106, -5.2343,
    -5.4663, -4.3562, -5.1467, -5.4533, -5.1481, -5.3110, -4.9424, -5.7410,
    -5.0600, -4.8960, -4.8630, -4.1390, -4.9727, -4.9667, -4.8241, -5.3833,
    -4.6636, -4.5892, -5.3594, -5.4599, -5.1801, -5.2848, -5.1190, -4.6956,
    -4.9888, -5.2416, -4.7839, -5.1666, -5.3740, -4.7411, -5.1656, -4.9770,
    -6.0877, -4.8717, -5.1582, -4.1725, -4.7421, -4.4821, -5.4239, -5.0960,
    -5.3757, -4.8542, -4.7736, -5.5021, -5.3699, -5.4960, -4.9947, -4.6696,
    -5.9240, -4.9215, -4.9380, -5.1339, -5.5515, -5.0422, -5.4240, -4.8855,
    -4.8572, -5.9668, -4.8704, -5.6482, -5.2565, -5.0988, -5.1251, -5.1164,
    -5.1848, -4.5638, -5.3515, -5.2576, -5.5184, -5.0843, -5.3026, -4.8263,
    -4.7056, -4.9838, -5.7830, -5.6214, -5.4560, -4.9965, -4.8960],
   device='cuda:0', grad_fn=<SelectBackward0>)

ConvolutionalNetwork(
 (model): EfficientNet(
  (features): Sequential(
   (0): Conv2dNormActivation(
    (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
    (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (2): SiLU(inplace=True)
   )
   (1): Sequential(
    (0): MBConv(
     (block): Sequential(
      (0): Conv2dNormActivation(
       (0): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=32, bias=False)
       (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
       (2): SiLU(inplace=True)
      )
      (1): SqueezeExcitation(
       (avgpool): AdaptiveAvgPool2d(output_size=1)
       (fc1): Conv2d(32, 8, kernel_size=(1, 1), stride=(1, 1))
       (fc2): Conv2d(8, 32, kernel_size=(1, 1), stride=(1, 1))
       (activation): SiLU(inplace=True)
       (scale_activation): Sigmoid()
      )
      (2): Conv2dNormActivation(
       (0): Conv2d(32, 16, kernel_size=(1, 1), stride=(1, 1), bias=False)
       (1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
     )
     (stochastic_depth): StochasticDepth(p=0.0, mode=row)
    )
   )
   (2): Sequential(
    (0): MBConv(
     (block): Sequential(
      (0): Conv2dNormActivation(
       (0): Conv2d(16, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
       (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
       (2): SiLU(inplace=True)
      )
      (1): Conv2dNormActivation(
       (0): Conv2d(96, 96, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), groups=96, bias=False)
       (1): BatchNorm2d(96, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
       (2): SiLU(inplace=True)
      )
      (2): SqueezeExcitation(
       (avgpool): AdaptiveAvgPool2d(output_size=1)
       (fc1): Conv2d(96, 4, kernel_size=(1, 1), stride=(1, 1))
       (fc2): Conv2d(4, 96, kernel_size=(1, 1), stride=(1, 1))
       (activation): SiLU(inplace=True)
       (scale_activation): Sigmoid()
      )
      (3): Conv2dNormActivation(
       (0): Conv2d(96, 24, kernel_size=(1, 1), stride=(1, 1), bias=False)
       (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
     )
     (stochastic_depth): StochasticDepth(p=0.0125, mode=row)
    )
    (1): MBConv(
     (block): Sequential(
      (0): Conv2dNormActivation(
       (0): Conv2d(24, 144, kernel_size=(1, 1), stride=(1, 1),
bias=False)
       (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
       (2): SiLU(inplace=True)
      )
      (1): Conv2dNormActivation(
       (0): Conv2d(144, 144, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=144, bias=False)
       (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
       (2): SiLU(inplace=True)
      )
      (2): SqueezeExcitation(
       (avgpool): AdaptiveAvgPool2d(output_size=1)
       (fc1): Conv2d(144, 6, kernel_size=(1, 1), stride=(1, 1))
       (fc2): Conv2d(6, 144, kernel_size=(1, 1), stride=(1, 1))
       (activation): SiLU(inplace=True)
       (scale_activation): Sigmoid()
      )
      (3): Conv2dNormActivation(
       (0): Conv2d(144, 24, kernel_size=(1, 1), stride=(1, 1),
bias=False)
       (1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
     )
     (stochastic_depth): StochasticDepth(p=0.025, mode=row)
    )
   )
   (3): Sequential(
    (0): MBConv(
     (block): Sequential(
      (0): Conv2dNormActivation(
       (0): Conv2d(24, 144, kernel_size=(1, 1), stride=(1, 1),
bias=False)
       (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
       (2): SiLU(inplace=True)
      )
      (1): Conv2dNormActivation(
       (0): Conv2d(144, 144, kernel_size=(5, 5), stride=(2, 2),
padding=(2, 2), groups=144, bias=False)
       (1): BatchNorm2d(144, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
       (2): SiLU(inplace=True)
      )
      (2): SqueezeExcitation(
       (avgpool): AdaptiveAvgPool2d(output_size=1)
       (fc1): Conv2d(144, 6, kernel_size=(1, 1), stride=(1, 1))
       (fc2): Conv2d(6, 144, kernel_size=(1, 1), stride=(1, 1))
       (activation): SiLU(inplace=True)
       (scale_activation): Sigmoid()
      )
      (3): Conv2dNormActivation(
       (0): Conv2d(144, 40, kernel_size=(1, 1), stride=(1, 1),
bias=False)
       (1): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
```

```
            )
          )
          (stochastic_depth): StochasticDepth(p=0.037500000000000006, mode=row)
        )
        (1): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(40, 240, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(240, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(240, 240, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=240, bias=False)
              (1): BatchNorm2d(240, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(240, 10, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(10, 240, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(240, 40, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.05, mode=row)
        )
      )
      (4): Sequential(
        (0): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(40, 240, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(240, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(240, 240, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1), groups=240, bias=False)
              (1): BatchNorm2d(240, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(240, 10, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(10, 240, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(240, 80, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.0625, mode=row)
        )
        (1): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(480, 480, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=480, bias=False)
              (1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(480, 20, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(20, 480, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(480, 80, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.07500000000000001, mode=row)
        )
        (2): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(480, 480, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=480, bias=False)
              (1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(480, 20, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(20, 480, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(480, 80, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(80, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.08750000000000001, mode=row)
        )
      )
      (5): Sequential(
        (0): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1),
bias=False)
```

```
              (1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(480, 480, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=480, bias=False)
              (1): BatchNorm2d(480, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(480, 20, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(20, 480, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(480, 112, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(112, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.1, mode=row)
        )
        (1): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(112, 672, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(672, 672, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=672, bias=False)
              (1): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(672, 28, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(28, 672, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(672, 112, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(112, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.1125, mode=row)
        )
        (2): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(112, 672, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(672, 672, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=672, bias=False)
              (1): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(672, 28, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(28, 672, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(672, 112, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(112, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.125, mode=row)
        )
      )
      (6): Sequential(
        (0): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(112, 672, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(672, 672, kernel_size=(5, 5), stride=(2, 2),
padding=(2, 2), groups=672, bias=False)
              (1): BatchNorm2d(672, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(672, 28, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(28, 672, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(672, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.1375, mode=row)
        )
        (1): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1152, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
              (1): BatchNorm2d(1152, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1152, 48, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(48, 1152, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.15000000000000002, mode=row)
        )
        (2): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1152, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
              (1): BatchNorm2d(1152, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1152, 48, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(48, 1152, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.1625, mode=row)
        )
        (3): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1152, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1152, 1152, kernel_size=(5, 5), stride=(1, 1),
padding=(2, 2), groups=1152, bias=False)
              (1): BatchNorm2d(1152, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1152, 48, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(48, 1152, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1152, 192, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(192, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.17500000000000002, mode=row)
        )
      )
      (7): Sequential(
        (0): MBConv(
          (block): Sequential(
            (0): Conv2dNormActivation(
              (0): Conv2d(192, 1152, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(1152, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (1): Conv2dNormActivation(
              (0): Conv2d(1152, 1152, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), groups=1152, bias=False)
```

```
              (1): BatchNorm2d(1152, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
              (2): SiLU(inplace=True)
            )
            (2): SqueezeExcitation(
              (avgpool): AdaptiveAvgPool2d(output_size=1)
              (fc1): Conv2d(1152, 48, kernel_size=(1, 1), stride=(1, 1))
              (fc2): Conv2d(48, 1152, kernel_size=(1, 1), stride=(1, 1))
              (activation): SiLU(inplace=True)
              (scale_activation): Sigmoid()
            )
            (3): Conv2dNormActivation(
              (0): Conv2d(1152, 320, kernel_size=(1, 1), stride=(1, 1),
bias=False)
              (1): BatchNorm2d(320, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
            )
          )
          (stochastic_depth): StochasticDepth(p=0.1875, mode=row)
        )
      )
      (8): Conv2dNormActivation(
        (0): Conv2d(320, 1280, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (1): BatchNorm2d(1280, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (2): SiLU(inplace=True)
      )
    )
    (avgpool): AdaptiveAvgPool2d(output_size=1)
    (classifier): Sequential(
      (0): Dropout(p=0.2, inplace=True)
      (1): Linear(in_features=1280, out_features=151, bias=True)
    )
  )
)
```

```
[{'val_loss': 5.034629821777344, 'val_acc': 0.01875000074505806}]
```

```
627
```

```
  0%|          | 0/667 [00:00<?, ?it/s]
```

```
Epoch [0], train_loss: 4.4882, val_loss: 3.5946, val_acc: 0.5719
```

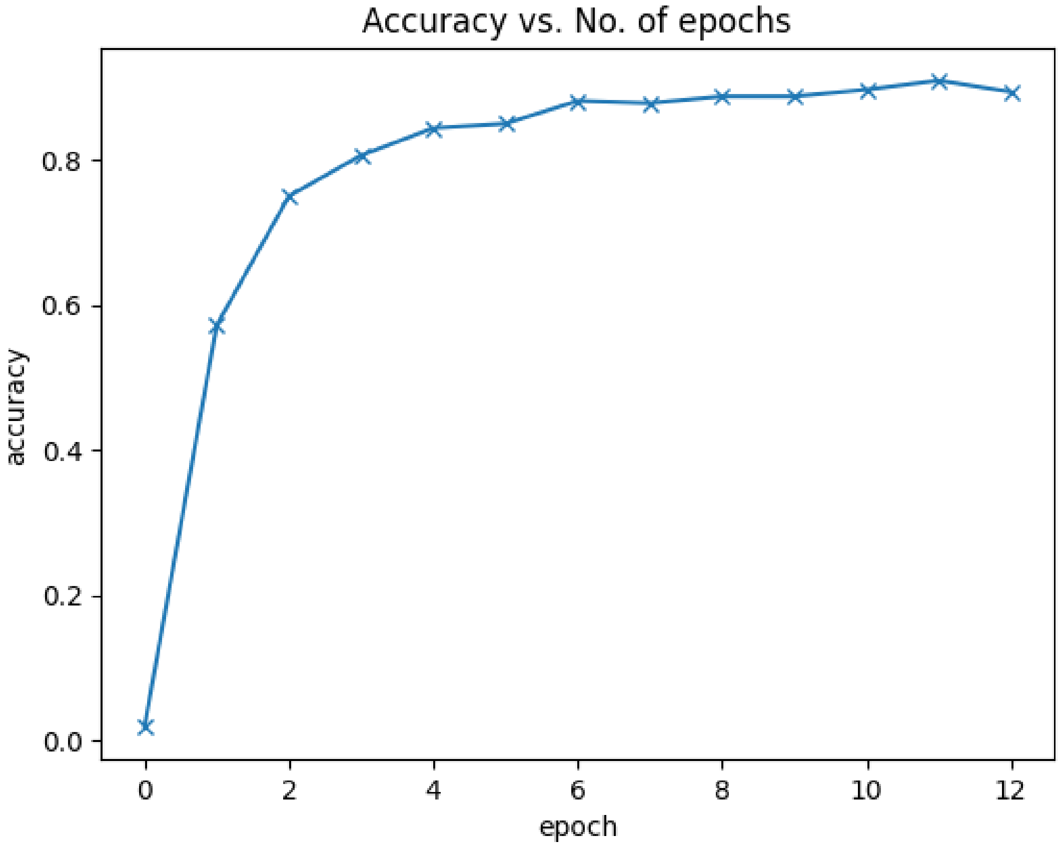
```
  0%|          | 0/667 [00:00<?, ?it/s]
```

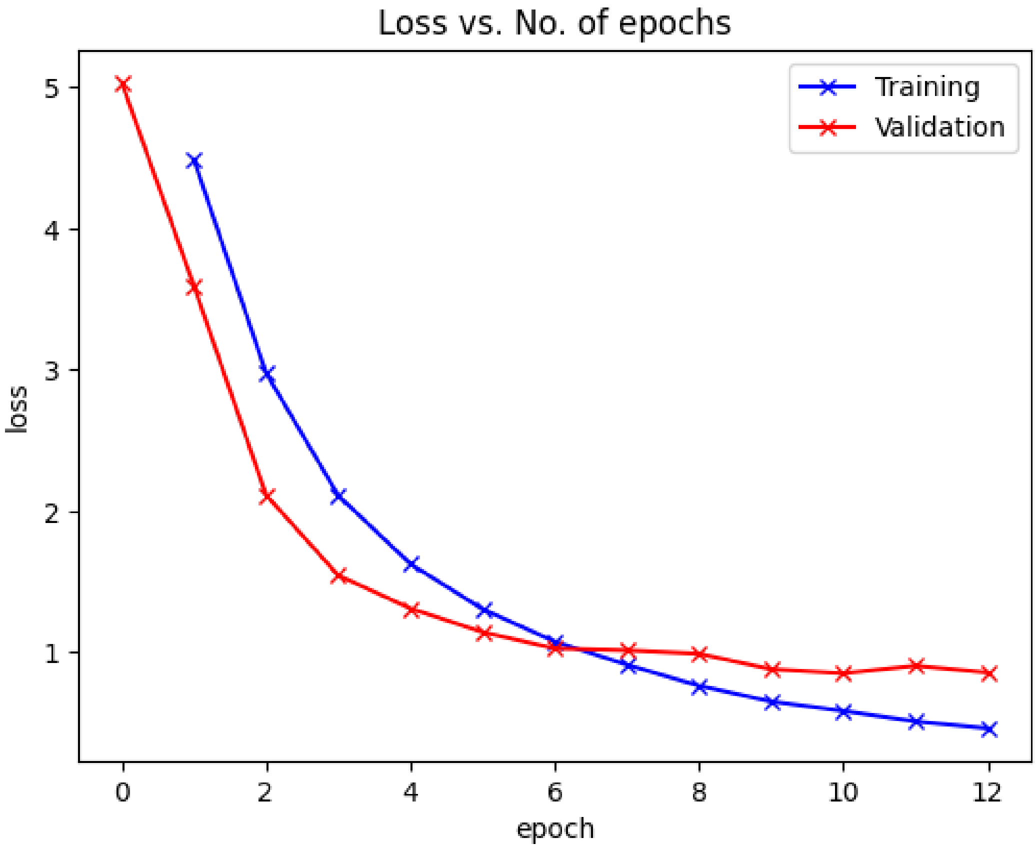
```
Epoch [1], train_loss: 2.9778, val_loss: 2.1140, val_acc: 0.7500
```

```
  0%|          | 0/667 [00:00<?, ?it/s]
```

```
Epoch [2], train_loss: 2.1107, val_loss: 1.5449, val_acc: 0.8063
```

```
  0%|          | 0/667 [00:00<?, ?it/s]
Epoch [3], train_loss: 1.6258, val_loss: 1.3110, val_acc: 0.8438
  0%|          | 0/667 [00:00<?, ?it/s]
Epoch [4], train_loss: 1.3064, val_loss: 1.1435, val_acc: 0.8500
  0%|          | 0/667 [00:00<?, ?it/s]
Epoch [5], train_loss: 1.0760, val_loss: 1.0273, val_acc: 0.8813
  0%|          | 0/667 [00:00<?, ?it/s]
Epoch [6], train_loss: 0.9121, val_loss: 1.0137, val_acc: 0.8781
  0%|          | 0/667 [00:00<?, ?it/s]
Epoch [7], train_loss: 0.7634, val_loss: 0.9898, val_acc: 0.8875
  0%|          | 0/667 [00:00<?, ?it/s]
Epoch [8], train_loss: 0.6517, val_loss: 0.8805, val_acc: 0.8875
  0%|          | 0/667 [00:00<?, ?it/s]
Epoch [9], train_loss: 0.5850, val_loss: 0.8516, val_acc: 0.8969
  0%|          | 0/667 [00:00<?, ?it/s]
Epoch [10], train_loss: 0.5112, val_loss: 0.9048, val_acc: 0.9094
  0%|          | 0/667 [00:00<?, ?it/s]
Epoch [11], train_loss: 0.4621, val_loss: 0.8567, val_acc: 0.8938
```


Accuracy vs. No. of epochs
accuracy
0.0
0.2
0.4
0.6
0.8
epoch
0
2
4
6
8
10
12



```
{'val_loss': 0.8605183959007263, 'val_acc': 0.9129747152328491}
```

1.1 FLOPs

```
 + Number of FLOPs: 0.21G
```

Animal_Classification-transfer-mobile-v3-large

June 13, 2024

1 Changing to transfer learning with Mobile Net V3 Large

From the previous model, we have swapped out the model for Mobile Net V3 Large that has been pretrained and tweaking it to use our data and classes.

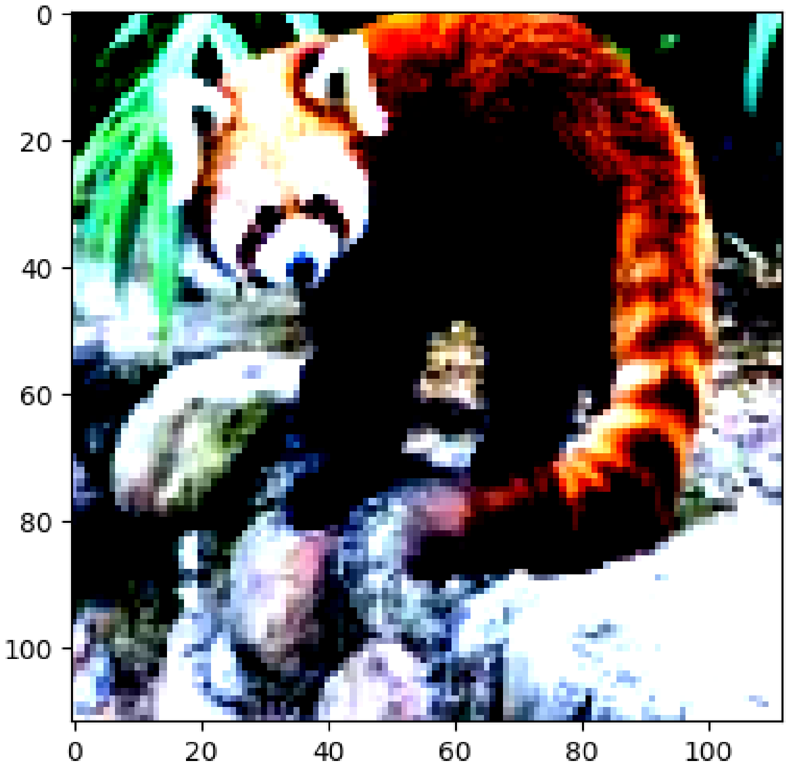
Validation Loss: 1.1079 Validation Accuracy: 89.72% FLOPS: 0.12G

```
Size of training dataset : 6270
```

```
torch.Size([3, 112, 112])
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

```
Label:  ailurus-fulgens (5)
```



```
(5330, 313, 627)
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

```
ConvolutionalNetwork(
  (model): MobileNetV3(
    (features): Sequential(
      (0): Conv2dNormActivation(
        (0): Conv2d(3, 16, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
        (1): BatchNorm2d(16, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
        (2): Hardswish()
      )
      (1): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(16, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=16, bias=False)
            (1): BatchNorm2d(16, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(16, 16, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(16, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (2): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(16, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(64, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (1): Conv2dNormActivation(
      (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), groups=64, bias=False)
      (1): BatchNorm2d(64, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
      (2): ReLU(inplace=True)
     )
     (2): Conv2dNormActivation(
      (0): Conv2d(64, 24, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(24, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
     )
    )
   )
   (3): InvertedResidual(
    (block): Sequential(
     (0): Conv2dNormActivation(
      (0): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(72, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
      (2): ReLU(inplace=True)
     )
     (1): Conv2dNormActivation(
      (0): Conv2d(72, 72, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=72, bias=False)
      (1): BatchNorm2d(72, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
      (2): ReLU(inplace=True)
     )
     (2): Conv2dNormActivation(
      (0): Conv2d(72, 24, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(24, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
     )
    )
   )
   (4): InvertedResidual(
    (block): Sequential(
     (0): Conv2dNormActivation(
      (0): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (1): BatchNorm2d(72, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
      (2): ReLU(inplace=True)
     )
     (1): Conv2dNormActivation(
      (0): Conv2d(72, 72, kernel_size=(5, 5), stride=(2, 2), padding=(2,
2), groups=72, bias=False)
      (1): BatchNorm2d(72, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(72, 24, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(72, 40, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(40, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (5): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(40, 120, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(120, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(120, 120, kernel_size=(5, 5), stride=(1, 1), padding=(2,
2), groups=120, bias=False)
            (1): BatchNorm2d(120, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(120, 32, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(32, 120, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(120, 40, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(40, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (6): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(40, 120, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(120, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(120, 120, kernel_size=(5, 5), stride=(1, 1), padding=(2,
2), groups=120, bias=False)
            (1): BatchNorm2d(120, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(120, 32, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(32, 120, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(120, 40, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(40, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (7): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(40, 240, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(240, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(240, 240, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), groups=240, bias=False)
            (1): BatchNorm2d(240, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): Conv2dNormActivation(
            (0): Conv2d(240, 80, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(80, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
```

```
          )
        )
      )
      (8): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(80, 200, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(200, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(200, 200, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=200, bias=False)
            (1): BatchNorm2d(200, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): Conv2dNormActivation(
            (0): Conv2d(200, 80, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(80, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (9): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(80, 184, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(184, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(184, 184, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=184, bias=False)
            (1): BatchNorm2d(184, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): Conv2dNormActivation(
            (0): Conv2d(184, 80, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(80, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (10): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(80, 184, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(184, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(184, 184, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=184, bias=False)
            (1): BatchNorm2d(184, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): Conv2dNormActivation(
            (0): Conv2d(184, 80, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(80, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (11): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(480, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(480, 480, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=480, bias=False)
            (1): BatchNorm2d(480, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(480, 120, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(120, 480, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(480, 112, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(112, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (12): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(112, 672, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(672, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(672, 672, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=672, bias=False)
            (1): BatchNorm2d(672, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(672, 168, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(168, 672, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(672, 112, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(112, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (13): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(112, 672, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(672, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(672, 672, kernel_size=(5, 5), stride=(2, 2), padding=(2,
2), groups=672, bias=False)
            (1): BatchNorm2d(672, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(672, 168, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(168, 672, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(672, 160, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(160, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (14): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(960, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(960, 960, kernel_size=(5, 5), stride=(1, 1), padding=(2,
2), groups=960, bias=False)
            (1): BatchNorm2d(960, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(960, 240, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(240, 960, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(160, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (15): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(960, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(960, 960, kernel_size=(5, 5), stride=(1, 1), padding=(2,
2), groups=960, bias=False)
            (1): BatchNorm2d(960, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(960, 240, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(240, 960, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(160, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (16): Conv2dNormActivation(
        (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (1): BatchNorm2d(960, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
        (2): Hardswish()
      )
    )
    (avgpool): AdaptiveAvgPool2d(output_size=1)
    (classifier): Sequential(
      (0): Linear(in_features=960, out_features=1280, bias=True)
      (1): Hardswish()
      (2): Dropout(p=0.2, inplace=True)
      (3): Linear(in_features=1280, out_features=151, bias=True)
    )
  )
)
```

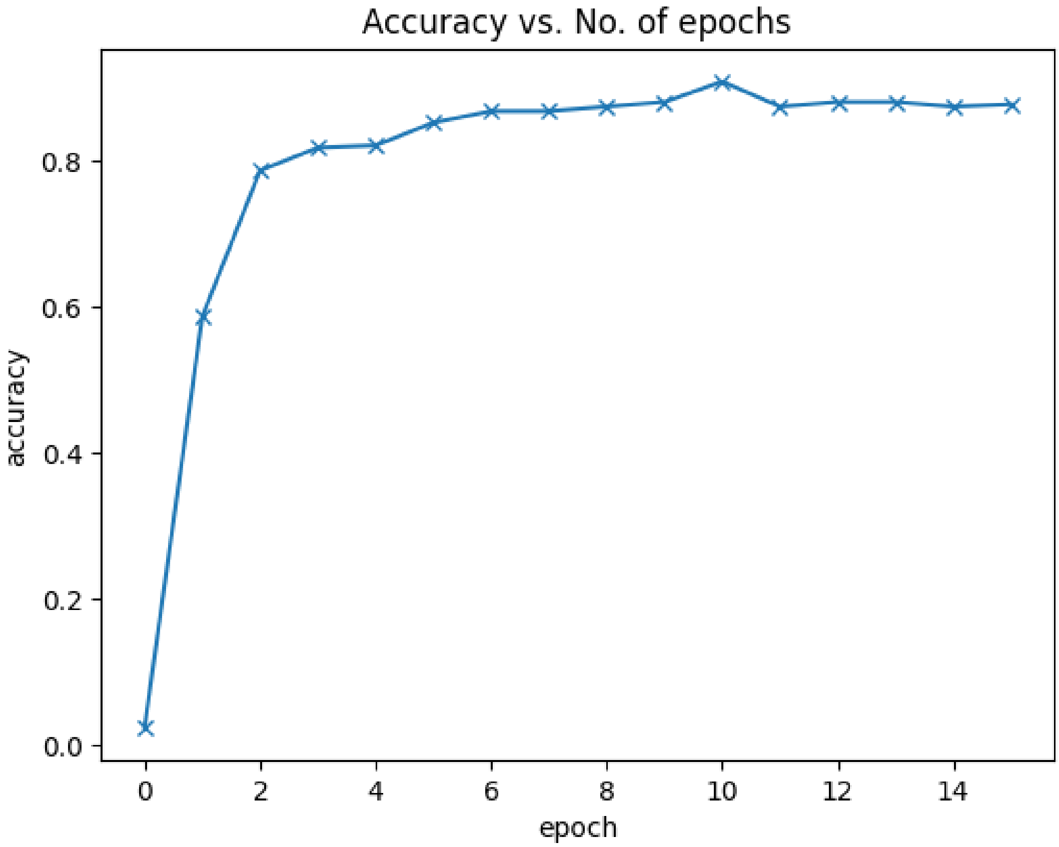
```
images.shape: torch.Size([8, 3, 112, 112])
out.shape: torch.Size([8, 151])
out[0]: tensor([-4.7812, -4.4831, -5.3164, -5.1257, -4.5402, -5.5535, -4.8652,
-5.8874,
        -5.0361, -4.3467, -5.1253, -4.2328, -5.1439, -5.7179, -4.6263, -5.0127,
        -4.7240, -4.5692, -5.9683, -5.2047, -5.8117, -5.1798, -5.3363, -5.0413,
        -5.1740, -4.7090, -5.5426, -4.7905, -5.1183, -5.6845, -5.2732, -4.7975,
        -5.8939, -5.3481, -5.0668, -5.2674, -5.8295, -5.5214, -5.5249, -4.5756,
        -4.9349, -4.3446, -5.0958, -5.7592, -5.0068, -4.8165, -5.7177, -5.6110,
        -5.9156, -5.3247, -5.0197, -4.9382, -5.1333, -5.6891, -5.0468, -4.6456,
        -5.6181, -5.3155, -5.4917, -4.9752, -4.8466, -5.4030, -5.6401, -4.9278,
        -4.8315, -5.3491, -5.7371, -4.6582, -4.7203, -5.2830, -4.7965, -5.4060,
        -6.0145, -5.0599, -4.6089, -4.8897, -5.7236, -4.0639, -5.2115, -5.1297,
        -5.0582, -5.1172, -4.7781, -5.1480, -4.8607, -4.8927, -4.0642, -4.6462,
        -5.7286, -5.5381, -5.2825, -4.1856, -5.3057, -5.7258, -4.8991, -4.7629,
        -4.9660, -4.9021, -5.8596, -4.8411, -5.3522, -4.5745, -5.0632, -5.3313,
        -5.3129, -5.0528, -4.5861, -5.0233, -5.0781, -5.6057, -5.0709, -5.6290,
        -4.9182, -4.3021, -5.2826, -4.6825, -4.5921, -5.1634, -5.2189, -5.2730,
        -4.8037, -4.9847, -5.3225, -5.4399, -4.4496, -5.4737, -5.4246, -4.5623,
        -5.0154, -5.2249, -5.0212, -4.2939, -4.9801, -5.2743, -4.5334, -6.1248,
        -5.0221, -4.8258, -4.4920, -5.4247, -5.3156, -5.4661, -5.5334, -4.4338,
        -4.9518, -5.1836, -5.2404, -4.6790, -5.5185, -4.9575, -5.4492],
       device='cuda:0', grad_fn=<SelectBackward0>)

ConvolutionalNetwork(
  (model): MobileNetV3(
    (features): Sequential(
      (0): Conv2dNormActivation(
        (0): Conv2d(3, 16, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
        (1): BatchNorm2d(16, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
        (2): Hardswish()
      )
      (1): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(16, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=16, bias=False)
            (1): BatchNorm2d(16, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(16, 16, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(16, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (2): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(16, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(64, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), groups=64, bias=False)
            (1): BatchNorm2d(64, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (2): Conv2dNormActivation(
            (0): Conv2d(64, 24, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(24, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (3): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(72, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(72, 72, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=72, bias=False)
            (1): BatchNorm2d(72, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (2): Conv2dNormActivation(
            (0): Conv2d(72, 24, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(24, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (4): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(72, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(72, 72, kernel_size=(5, 5), stride=(2, 2), padding=(2,
2), groups=72, bias=False)
            (1): BatchNorm2d(72, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(72, 24, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(24, 72, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(72, 40, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(40, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (5): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(40, 120, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(120, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(120, 120, kernel_size=(5, 5), stride=(1, 1), padding=(2,
2), groups=120, bias=False)
            (1): BatchNorm2d(120, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(120, 32, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(32, 120, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(120, 40, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(40, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (6): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(40, 120, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(120, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(120, 120, kernel_size=(5, 5), stride=(1, 1), padding=(2,
2), groups=120, bias=False)
            (1): BatchNorm2d(120, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(120, 32, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(32, 120, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(120, 40, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(40, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (7): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(40, 240, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(240, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(240, 240, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), groups=240, bias=False)
            (1): BatchNorm2d(240, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): Conv2dNormActivation(
            (0): Conv2d(240, 80, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(80, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (8): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(80, 200, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(200, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(200, 200, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=200, bias=False)
            (1): BatchNorm2d(200, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): Conv2dNormActivation(
            (0): Conv2d(200, 80, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(80, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (9): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(80, 184, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(184, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(184, 184, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=184, bias=False)
            (1): BatchNorm2d(184, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): Conv2dNormActivation(
            (0): Conv2d(184, 80, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(80, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (10): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(80, 184, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(184, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(184, 184, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=184, bias=False)
            (1): BatchNorm2d(184, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): Conv2dNormActivation(
            (0): Conv2d(184, 80, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(80, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (11): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(80, 480, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(480, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(480, 480, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=480, bias=False)
            (1): BatchNorm2d(480, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(480, 120, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(120, 480, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(480, 112, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(112, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (12): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(112, 672, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(672, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(672, 672, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=672, bias=False)
            (1): BatchNorm2d(672, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(672, 168, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(168, 672, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(672, 112, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(112, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (13): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(112, 672, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(672, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(672, 672, kernel_size=(5, 5), stride=(2, 2), padding=(2,
2), groups=672, bias=False)
            (1): BatchNorm2d(672, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(672, 168, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(168, 672, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(672, 160, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(160, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (14): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(960, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(960, 960, kernel_size=(5, 5), stride=(1, 1), padding=(2,
2), groups=960, bias=False)
            (1): BatchNorm2d(960, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(960, 240, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(240, 960, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(160, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (15): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(960, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(960, 960, kernel_size=(5, 5), stride=(1, 1), padding=(2,
2), groups=960, bias=False)
            (1): BatchNorm2d(960, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(960, 240, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(240, 960, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(960, 160, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(160, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (16): Conv2dNormActivation(
        (0): Conv2d(160, 960, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (1): BatchNorm2d(960, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
        (2): Hardswish()
      )
    )
    (avgpool): AdaptiveAvgPool2d(output_size=1)
    (classifier): Sequential(
      (0): Linear(in_features=960, out_features=1280, bias=True)
      (1): Hardswish()
      (2): Dropout(p=0.2, inplace=True)
      (3): Linear(in_features=1280, out_features=151, bias=True)
    )
```

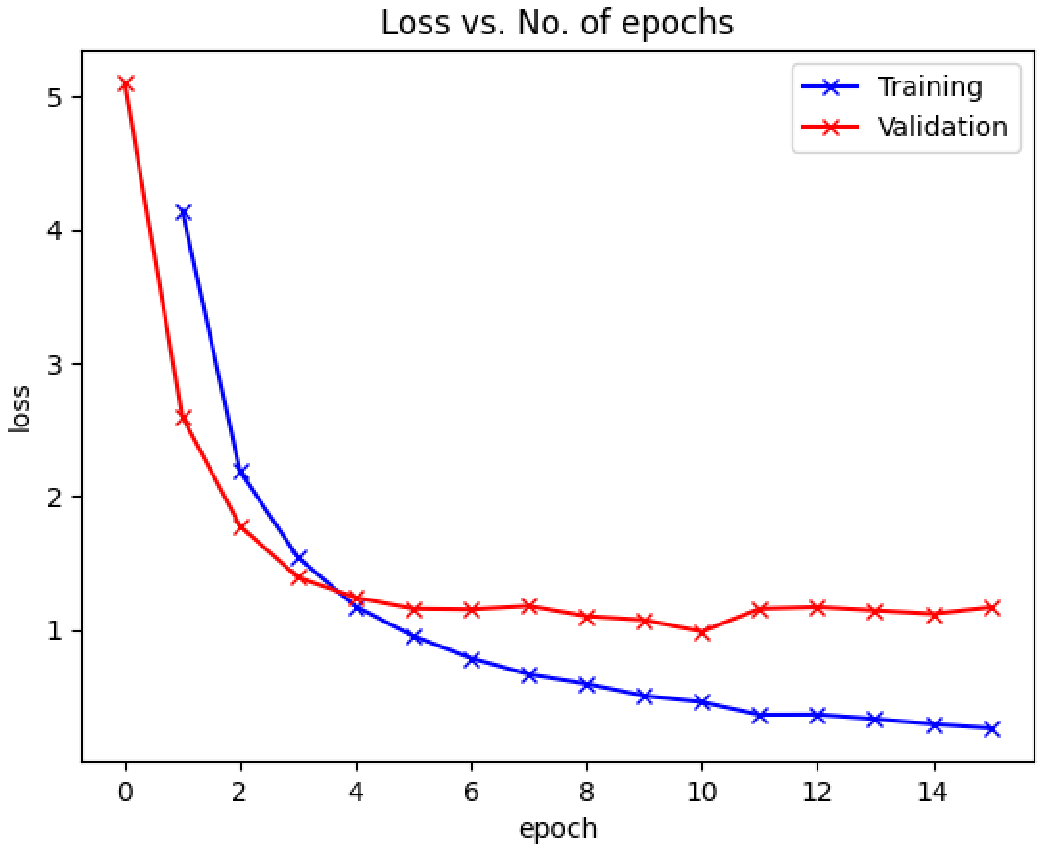
```
  )
)
```

```
[{'val_loss': 5.10817813873291, 'val_acc': 0.02500000037252903}]
```

```
627

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [0], train_loss: 4.1468, val_loss: 2.5995, val_acc: 0.5875

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [1], train_loss: 2.1886, val_loss: 1.7783, val_acc: 0.7875

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [2], train_loss: 1.5442, val_loss: 1.3930, val_acc: 0.8188

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [3], train_loss: 1.1747, val_loss: 1.2415, val_acc: 0.8219

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [4], train_loss: 0.9517, val_loss: 1.1572, val_acc: 0.8531

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [5], train_loss: 0.7862, val_loss: 1.1514, val_acc: 0.8688

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [6], train_loss: 0.6645, val_loss: 1.1776, val_acc: 0.8688

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [7], train_loss: 0.5927, val_loss: 1.1007, val_acc: 0.8750

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [8], train_loss: 0.5039, val_loss: 1.0720, val_acc: 0.8813

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [9], train_loss: 0.4561, val_loss: 0.9861, val_acc: 0.9094

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [10], train_loss: 0.3608, val_loss: 1.1562, val_acc: 0.8750

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [11], train_loss: 0.3625, val_loss: 1.1689, val_acc: 0.8813

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [12], train_loss: 0.3294, val_loss: 1.1434, val_acc: 0.8813

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [13], train_loss: 0.2910, val_loss: 1.1179, val_acc: 0.8750
```

```
  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [14], train_loss: 0.2611, val_loss: 1.1665, val_acc: 0.8781
```





```
{'val_loss': 1.1078855991363525, 'val_acc': 0.8971519470214844}
```

1.1 FLOPs

```
+ Number of FLOPs: 0.12G
```

Animal_Classification-transfer-mobile-v3-small

June 13, 2024

1 Changing to transfer learning with Mobile Net V3 Small

From the previous model, we have swapped out the model for Mobile Net V3 Small that has been pretrained and tweaking it to use our data and classes.

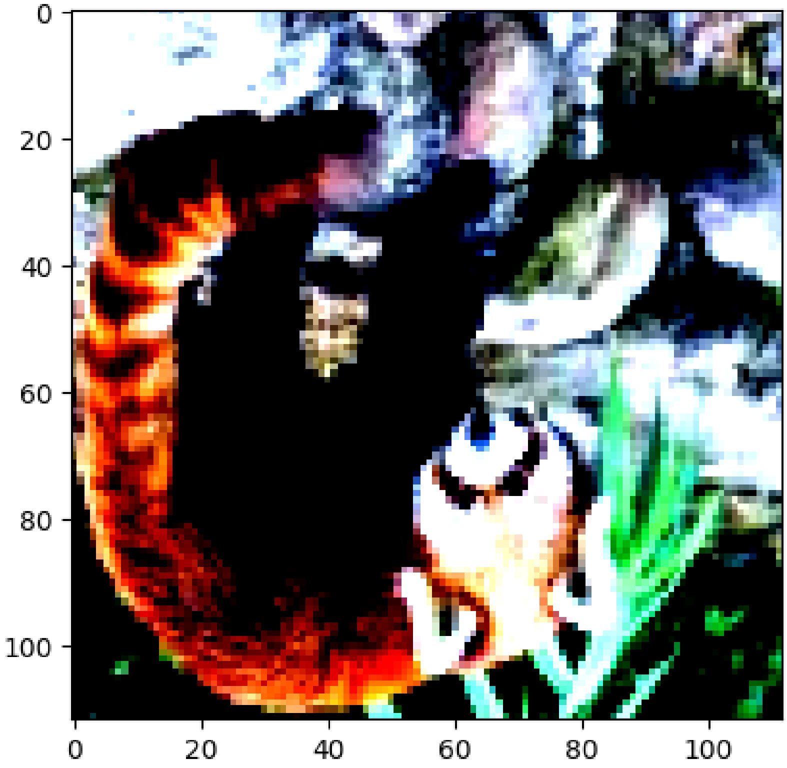
Validation Loss: 1.2289 Validation Accuracy: 85.76% FLOPS: 0.03G

```
Size of training dataset : 6270
```

```
torch.Size([3, 112, 112])
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

```
Label:  ailurus-fulgens (5)
```



```
(5330, 313, 627)
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

```
ConvolutionalNetwork(
  (model): MobileNetV3(
    (features): Sequential(
      (0): Conv2dNormActivation(
        (0): Conv2d(3, 16, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
        (1): BatchNorm2d(16, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
        (2): Hardswish()
      )
      (1): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(16, 16, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), groups=16, bias=False)
            (1): BatchNorm2d(16, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (1): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(16, 8, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(8, 16, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (2): Conv2dNormActivation(
            (0): Conv2d(16, 16, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(16, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (2): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(16, 72, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(72, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(72, 72, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), groups=72, bias=False)
            (1): BatchNorm2d(72, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (2): Conv2dNormActivation(
            (0): Conv2d(72, 24, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(24, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (3): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(24, 88, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(88, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(88, 88, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=88, bias=False)
            (1): BatchNorm2d(88, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (2): Conv2dNormActivation(
            (0): Conv2d(88, 24, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(24, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (4): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(24, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(96, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(96, 96, kernel_size=(5, 5), stride=(2, 2), padding=(2,
2), groups=96, bias=False)
            (1): BatchNorm2d(96, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(96, 24, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(24, 96, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(96, 40, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(40, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (5): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(40, 240, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(240, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(240, 240, kernel_size=(5, 5), stride=(1, 1), padding=(2,
2), groups=240, bias=False)
            (1): BatchNorm2d(240, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(240, 64, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(64, 240, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(240, 40, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(40, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (6): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(40, 240, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(240, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(240, 240, kernel_size=(5, 5), stride=(1, 1), padding=(2,
2), groups=240, bias=False)
            (1): BatchNorm2d(240, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(240, 64, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(64, 240, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(240, 40, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(40, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (7): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(40, 120, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(120, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(120, 120, kernel_size=(5, 5), stride=(1, 1), padding=(2,
2), groups=120, bias=False)
            (1): BatchNorm2d(120, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(120, 32, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(32, 120, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(120, 48, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(48, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (8): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(48, 144, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(144, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(144, 144, kernel_size=(5, 5), stride=(1, 1), padding=(2,
2), groups=144, bias=False)
            (1): BatchNorm2d(144, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(144, 40, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(40, 144, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(144, 48, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(48, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (9): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(48, 288, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(288, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(288, 288, kernel_size=(5, 5), stride=(2, 2), padding=(2,
2), groups=288, bias=False)
            (1): BatchNorm2d(288, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(288, 72, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(72, 288, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(288, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(96, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (10): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(576, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(576, 576, kernel_size=(5, 5), stride=(1, 1), padding=(2,
2), groups=576, bias=False)
            (1): BatchNorm2d(576, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(576, 144, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(144, 576, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(96, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (11): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(576, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(576, 576, kernel_size=(5, 5), stride=(1, 1), padding=(2,
2), groups=576, bias=False)
            (1): BatchNorm2d(576, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(576, 144, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(144, 576, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(96, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (12): Conv2dNormActivation(
        (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (1): BatchNorm2d(576, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
        (2): Hardswish()
      )
    )
    (avgpool): AdaptiveAvgPool2d(output_size=1)
    (classifier): Sequential(
      (0): Linear(in_features=576, out_features=1024, bias=True)
      (1): Hardswish()
      (2): Dropout(p=0.2, inplace=True)
      (3): Linear(in_features=1024, out_features=151, bias=True)
    )
  )
)

images.shape: torch.Size([8, 3, 112, 112])
out.shape: torch.Size([8, 151])
out[0]: tensor([-5.5542, -5.6103, -5.4379, -5.2414, -5.1411, -4.6883, -4.7928,
-4.9843,
        -4.9865, -5.5290, -4.4779, -5.4457, -5.3733, -4.3677, -5.6185, -5.5864,
        -4.5083, -5.0144, -4.3929, -4.8658, -4.8815, -5.8586, -4.5566, -5.1401,
        -4.9144, -5.6852, -5.5693, -5.1761, -5.0683, -5.5963, -5.0061, -4.9665,
        -5.8103, -4.6418, -5.4931, -4.1183, -4.8317, -4.8607, -5.1162, -4.8572,
        -5.9464, -4.8994, -4.7951, -4.9792, -4.8258, -5.0329, -5.0679, -5.1438,
        -5.2102, -5.3552, -4.8719, -5.9381, -5.1105, -5.4190, -5.4374, -5.0009,
        -5.0370, -5.5993, -5.2528, -4.5962, -6.0117, -5.0390, -4.9475, -4.4735,
        -5.4557, -4.3037, -5.4275, -5.0196, -5.1799, -6.4949, -4.8761, -5.4855,
        -4.9137, -4.8761, -5.1473, -5.3224, -4.8004, -4.7817, -5.6682, -6.2449,
        -4.9177, -5.0530, -4.7397, -5.5145, -5.3851, -4.7552, -5.6719, -5.3272,
        -5.0901, -4.3154, -6.2621, -4.5276, -5.3719, -4.6722, -5.6334, -4.7404,
        -5.1805, -4.7256, -4.8417, -4.6307, -5.3082, -5.1648, -4.6274, -4.5309,
        -5.7943, -5.1004, -5.0038, -5.3724, -4.9499, -5.3513, -5.0583, -4.0588,
        -5.5525, -4.7955, -5.0639, -4.8760, -5.2460, -5.7086, -4.4823, -5.0263,
        -5.4103, -5.0922, -5.9388, -4.5565, -5.2087, -5.8740, -5.4014, -4.6751,
        -5.0326, -4.9352, -5.4580, -5.4378, -5.6487, -5.2868, -4.9246, -4.8195,
        -5.3997, -5.3548, -4.6796, -5.0489, -4.6602, -4.8843, -4.2516, -5.2222,
        -5.9748, -4.5397, -4.6895, -4.2863, -5.2250, -4.9740, -4.8543],
       device='cuda:0', grad_fn=<SelectBackward0>)

ConvolutionalNetwork(
  (model): MobileNetV3(
    (features): Sequential(
      (0): Conv2dNormActivation(
        (0): Conv2d(3, 16, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
        (1): BatchNorm2d(16, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
        (2): Hardswish()
      )
      (1): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(16, 16, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), groups=16, bias=False)
            (1): BatchNorm2d(16, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (1): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(16, 8, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(8, 16, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (2): Conv2dNormActivation(
            (0): Conv2d(16, 16, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(16, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (2): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(16, 72, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(72, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(72, 72, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), groups=72, bias=False)
            (1): BatchNorm2d(72, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (2): Conv2dNormActivation(
            (0): Conv2d(72, 24, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(24, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (3): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(24, 88, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(88, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
```

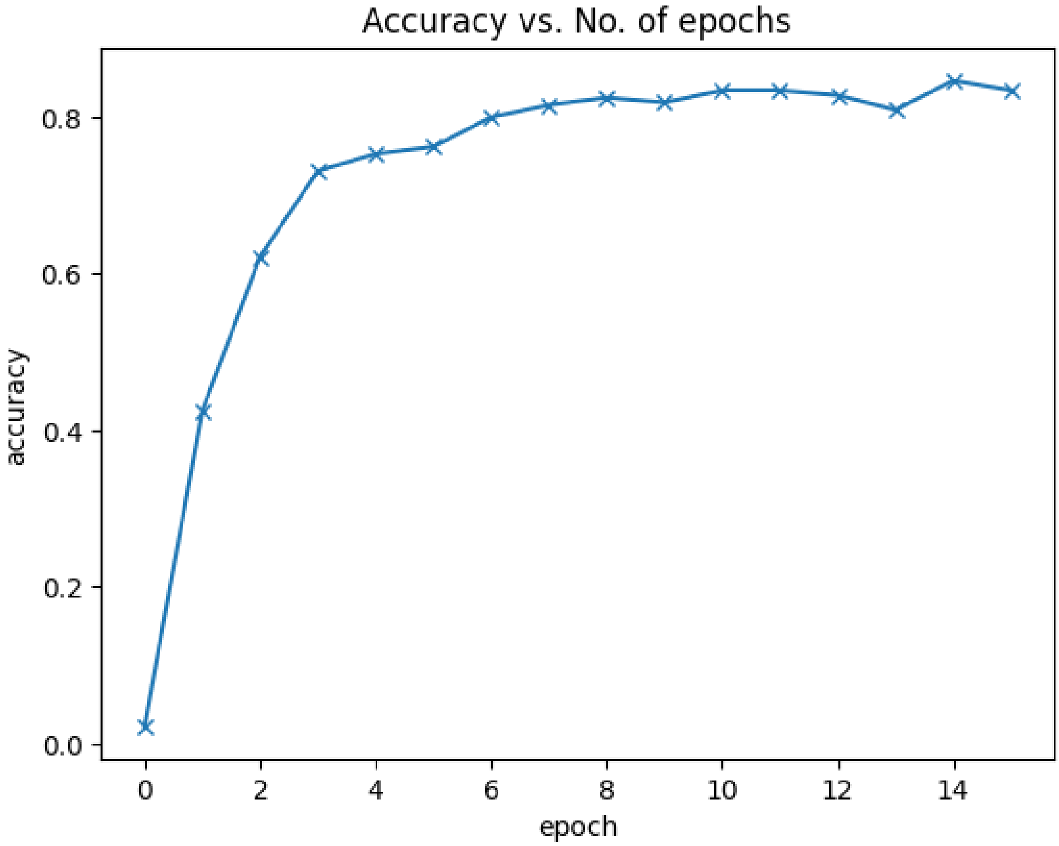
```
            (2): ReLU(inplace=True)
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(88, 88, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), groups=88, bias=False)
            (1): BatchNorm2d(88, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): ReLU(inplace=True)
          )
          (2): Conv2dNormActivation(
            (0): Conv2d(88, 24, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(24, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (4): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(24, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(96, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(96, 96, kernel_size=(5, 5), stride=(2, 2), padding=(2,
2), groups=96, bias=False)
            (1): BatchNorm2d(96, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(96, 24, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(24, 96, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(96, 40, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(40, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (5): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(40, 240, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(240, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(240, 240, kernel_size=(5, 5), stride=(1, 1), padding=(2,
2), groups=240, bias=False)
            (1): BatchNorm2d(240, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(240, 64, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(64, 240, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(240, 40, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(40, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (6): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(40, 240, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(240, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(240, 240, kernel_size=(5, 5), stride=(1, 1), padding=(2,
2), groups=240, bias=False)
            (1): BatchNorm2d(240, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(240, 64, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(64, 240, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(240, 40, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(40, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (7): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(40, 120, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(120, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(120, 120, kernel_size=(5, 5), stride=(1, 1), padding=(2,
2), groups=120, bias=False)
            (1): BatchNorm2d(120, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(120, 32, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(32, 120, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(120, 48, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(48, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (8): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(48, 144, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(144, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(144, 144, kernel_size=(5, 5), stride=(1, 1), padding=(2,
2), groups=144, bias=False)
            (1): BatchNorm2d(144, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(144, 40, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(40, 144, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(144, 48, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(48, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (9): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(48, 288, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(288, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(288, 288, kernel_size=(5, 5), stride=(2, 2), padding=(2,
2), groups=288, bias=False)
            (1): BatchNorm2d(288, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(288, 72, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(72, 288, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(288, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(96, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (10): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(576, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(576, 576, kernel_size=(5, 5), stride=(1, 1), padding=(2,
2), groups=576, bias=False)
            (1): BatchNorm2d(576, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(576, 144, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(144, 576, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(96, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (11): InvertedResidual(
        (block): Sequential(
          (0): Conv2dNormActivation(
            (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(576, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (1): Conv2dNormActivation(
            (0): Conv2d(576, 576, kernel_size=(5, 5), stride=(1, 1), padding=(2,
2), groups=576, bias=False)
            (1): BatchNorm2d(576, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
            (2): Hardswish()
          )
          (2): SqueezeExcitation(
            (avgpool): AdaptiveAvgPool2d(output_size=1)
            (fc1): Conv2d(576, 144, kernel_size=(1, 1), stride=(1, 1))
            (fc2): Conv2d(144, 576, kernel_size=(1, 1), stride=(1, 1))
            (activation): ReLU()
            (scale_activation): Hardsigmoid()
          )
          (3): Conv2dNormActivation(
            (0): Conv2d(576, 96, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): BatchNorm2d(96, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
          )
        )
      )
      (12): Conv2dNormActivation(
        (0): Conv2d(96, 576, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (1): BatchNorm2d(576, eps=0.001, momentum=0.01, affine=True,
track_running_stats=True)
        (2): Hardswish()
      )
    )
    (avgpool): AdaptiveAvgPool2d(output_size=1)
    (classifier): Sequential(
      (0): Linear(in_features=576, out_features=1024, bias=True)
      (1): Hardswish()
      (2): Dropout(p=0.2, inplace=True)
      (3): Linear(in_features=1024, out_features=151, bias=True)
    )
  )
)
```

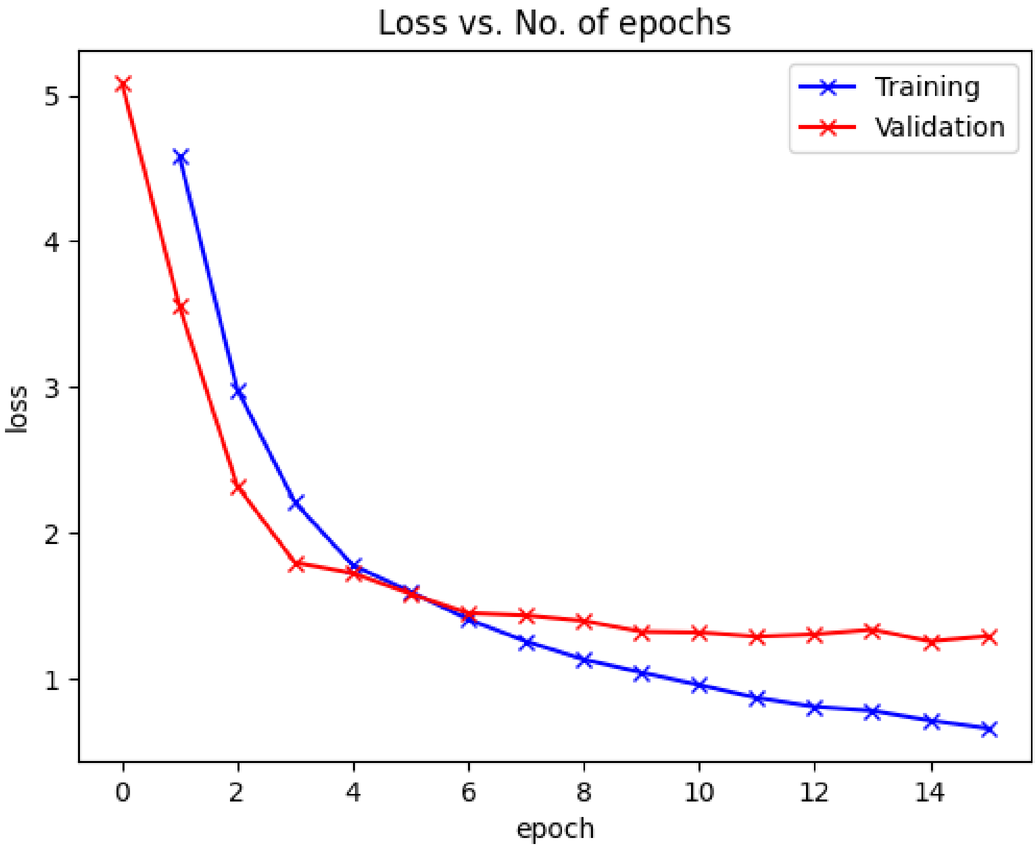
```
[{'val_loss': 5.08850622177124, 'val_acc': 0.021874999627470970}]
```

```
627
```

```
  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [0], train_loss: 4.5812, val_loss: 3.5535, val_acc: 0.4250

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [1], train_loss: 2.9841, val_loss: 2.3200, val_acc: 0.6219

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [2], train_loss: 2.2071, val_loss: 1.7945, val_acc: 0.7312

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [3], train_loss: 1.7772, val_loss: 1.7229, val_acc: 0.7531

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [4], train_loss: 1.5954, val_loss: 1.5807, val_acc: 0.7625
```

```
  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [5], train_loss: 1.4059, val_loss: 1.4487, val_acc: 0.8000

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [6], train_loss: 1.2548, val_loss: 1.4326, val_acc: 0.8156

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [7], train_loss: 1.1310, val_loss: 1.3956, val_acc: 0.8250

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [8], train_loss: 1.0410, val_loss: 1.3212, val_acc: 0.8188

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [9], train_loss: 0.9561, val_loss: 1.3160, val_acc: 0.8344

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [10], train_loss: 0.8672, val_loss: 1.2879, val_acc: 0.8344

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [11], train_loss: 0.8060, val_loss: 1.3030, val_acc: 0.8281

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [12], train_loss: 0.7782, val_loss: 1.3350, val_acc: 0.8094

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [13], train_loss: 0.7123, val_loss: 1.2550, val_acc: 0.8469

  0%|          | 0/667 [00:00<?, ?it/s]

Epoch [14], train_loss: 0.6580, val_loss: 1.2911, val_acc: 0.8344
```


Accuracy vs. No. of epochs
accuracy
epoch
0.0
0.2
0.4
0.6
0.8
0
2
4
6
8
10
12
14



```
{'val_loss': 1.2289390563964844, 'val_acc': 0.8575949668884277}
```

1.1 FLOPs

```
+ Number of FLOPs: 0.03G
```

Animal_Classification-transfer-resnet

June 10, 2024

1 Changing to transfer learning with resnet

From the previous model, we have swapped out the model for resnet that has been pretrained and tweaking it to use our data and classes.

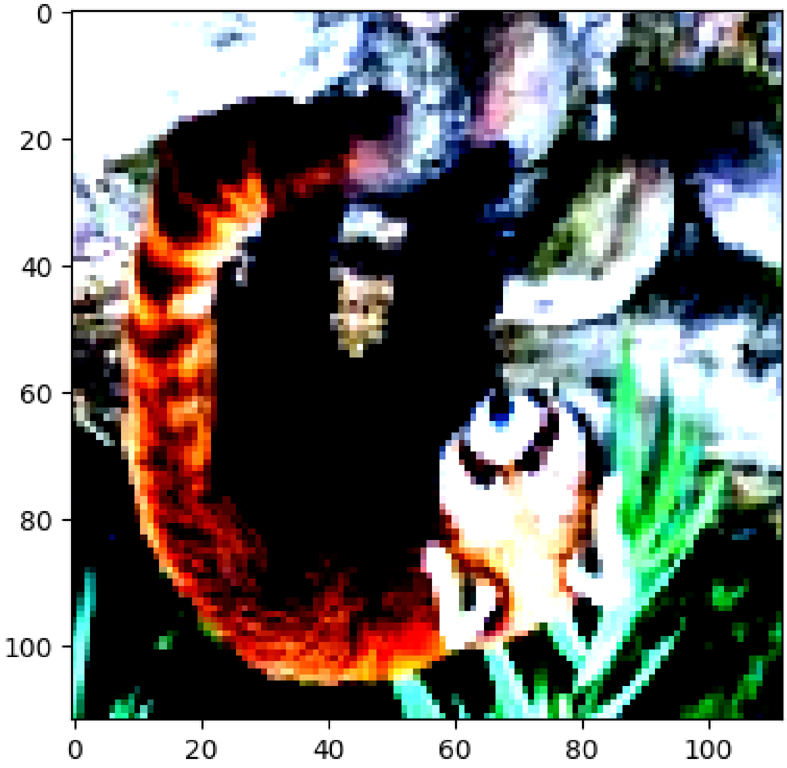
Validation Loss: 1.417 Validation Accuracy: 81.99% FLOPS: 2.15G

```
Size of training dataset : 6270
```

```
torch.Size([3, 112, 112])
```

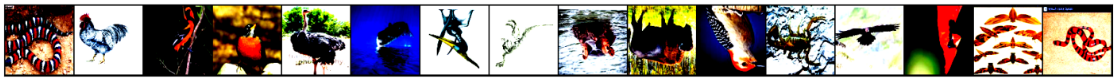
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```

```
Label:  ailurus-fulgens (5)
```



```
(5330, 313, 627)
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```



```
Downloading: "https://download.pytorch.org/models/resnet50-0676ba61.pth" to
C:\Users\Nathaniel/.cache\torch\hub\checkpoints\resnet50-0676ba61.pth
100%|      | 97.8M/97.8M [00:08<00:00, 12.3MB/s]
```

```
ConvolutionalNetwork(
  (model): 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): Bottleneck(
        (conv1): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=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)
        (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
        (downsample): Sequential(
          (0): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        )
      )
      (1): Bottleneck(
        (conv1): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=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)
        (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
      )
      (2): Bottleneck(
        (conv1): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=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)
        (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
      )
    )
    (layer2): Sequential(
      (0): Bottleneck(
        (conv1): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
        (downsample): Sequential(
          (0): Conv2d(256, 512, kernel_size=(1, 1), stride=(2, 2), bias=False)
          (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        )
      )
      (1): Bottleneck(
        (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
```

```
        (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
      )
      (2): Bottleneck(
        (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
      )
      (3): Bottleneck(
        (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
      )
    )
    (layer3): Sequential(
      (0): Bottleneck(
        (conv1): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
```

```
        (relu): ReLU(inplace=True)
        (downsample): Sequential(
          (0): Conv2d(512, 1024, kernel_size=(1, 1), stride=(2, 2), bias=False)
          (1): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        )
      )
      (1): Bottleneck(
        (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
      )
      (2): Bottleneck(
        (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
      )
      (3): Bottleneck(
        (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
      )
      (4): Bottleneck(
        (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
      )
      (5): Bottleneck(
        (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
      )
    )
    (layer4): Sequential(
      (0): Bottleneck(
        (conv1): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(512, 2048, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn3): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
        (downsample): Sequential(
          (0): Conv2d(1024, 2048, kernel_size=(1, 1), stride=(2, 2), bias=False)
          (1): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        )
      )
      (1): Bottleneck(
        (conv1): Conv2d(2048, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(512, 2048, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn3): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
      )
      (2): Bottleneck(
        (conv1): Conv2d(2048, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(512, 2048, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn3): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
      )
    )
    (avgpool): AdaptiveAvgPool2d(output_size=(1, 1))
    (fc): Linear(in_features=2048, out_features=151, bias=True)
  )
)
```

```
images.shape: torch.Size([16, 3, 112, 112])
out.shape: torch.Size([16, 151])
```

```
out[0]: tensor([-4.8035, -5.4346, -4.9056, -5.2217, -4.8687, -5.0862, -5.2110,
-4.7170,
        -5.1203, -4.1530, -4.9783, -4.8200, -5.3803, -5.0957, -5.0043, -4.7594,
        -4.7490, -4.5283, -5.1798, -5.4866, -5.3665, -5.2403, -5.1532, -4.9642,
        -5.3161, -4.8780, -4.8628, -5.3597, -4.9670, -4.6837, -5.4355, -4.3187,
        -4.8319, -4.6025, -4.7260, -5.4119, -5.1961, -4.4779, -5.4957, -4.8368,
        -4.8110, -5.2743, -5.1362, -4.3503, -6.3225, -6.0512, -5.2784, -4.4603,
        -5.2558, -5.1267, -5.1013, -4.9857, -5.8806, -5.0303, -4.9887, -4.6971,
        -4.9067, -5.3940, -5.6316, -5.1791, -5.0291, -4.7752, -5.4904, -5.0896,
        -4.9409, -5.4047, -4.8629, -5.3676, -4.6756, -4.6326, -5.1445, -4.3950,
        -5.1615, -5.3045, -5.6603, -5.4345, -4.7248, -4.4626, -5.4798, -5.0188,
        -4.5190, -4.9779, -5.3858, -5.3624, -4.7533, -5.5111, -4.7871, -5.6075,
        -5.2479, -5.0409, -4.8156, -5.0107, -5.4288, -5.6537, -5.5920, -4.5416,
        -4.7345, -5.7733, -5.6739, -5.6597, -4.8400, -5.2330, -4.7147, -5.0707,
        -6.2022, -4.0836, -5.5255, -5.3723, -4.8276, -5.6728, -5.5321, -4.6315,
        -4.4020, -5.3991, -5.7593, -4.8768, -5.0624, -5.3508, -5.6107, -4.8140,
        -5.0628, -5.0972, -5.0505, -5.3999, -5.5225, -4.1837, -5.2019, -4.9234,
        -5.1628, -5.1805, -4.8499, -4.5938, -5.3550, -5.0081, -4.4907, -5.6279,
        -5.5163, -4.8735, -4.9069, -5.2587, -5.5043, -5.4321, -5.2317, -4.6302,
        -5.2208, -5.7738, -5.2098, -4.8862, -4.4524, -5.5090, -5.0694],
       device='cuda:0', grad_fn=<SelectBackward0>)
```

```
ConvolutionalNetwork(
  (model): 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): Bottleneck(
        (conv1): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=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)
        (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
        (downsample): Sequential(
          (0): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        )
      )
      (1): Bottleneck(
        (conv1): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=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)
        (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
      )
      (2): Bottleneck(
        (conv1): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=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)
        (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
      )
    )
    (layer2): Sequential(
      (0): Bottleneck(
        (conv1): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
        (downsample): Sequential(
          (0): Conv2d(256, 512, kernel_size=(1, 1), stride=(2, 2), bias=False)
          (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
```

```
        )
      )
      (1): Bottleneck(
        (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
      )
      (2): Bottleneck(
        (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
      )
      (3): Bottleneck(
        (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
      )
    )
    (layer3): Sequential(
      (0): Bottleneck(
        (conv1): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
```

```
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
        (downsample): Sequential(
          (0): Conv2d(512, 1024, kernel_size=(1, 1), stride=(2, 2), bias=False)
          (1): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        )
      )
      (1): Bottleneck(
        (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
      )
      (2): Bottleneck(
        (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
      )
      (3): Bottleneck(
        (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
      )
      (4): Bottleneck(
        (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
      )
      (5): Bottleneck(
        (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
      )
    )
    (layer4): Sequential(
      (0): Bottleneck(
        (conv1): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(512, 2048, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn3): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
        (downsample): Sequential(
          (0): Conv2d(1024, 2048, kernel_size=(1, 1), stride=(2, 2), bias=False)
          (1): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        )
      )
      (1): Bottleneck(
        (conv1): Conv2d(2048, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(512, 2048, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn3): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
      )
      (2): Bottleneck(
        (conv1): Conv2d(2048, 512, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (conv3): Conv2d(512, 2048, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn3): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=True)
```

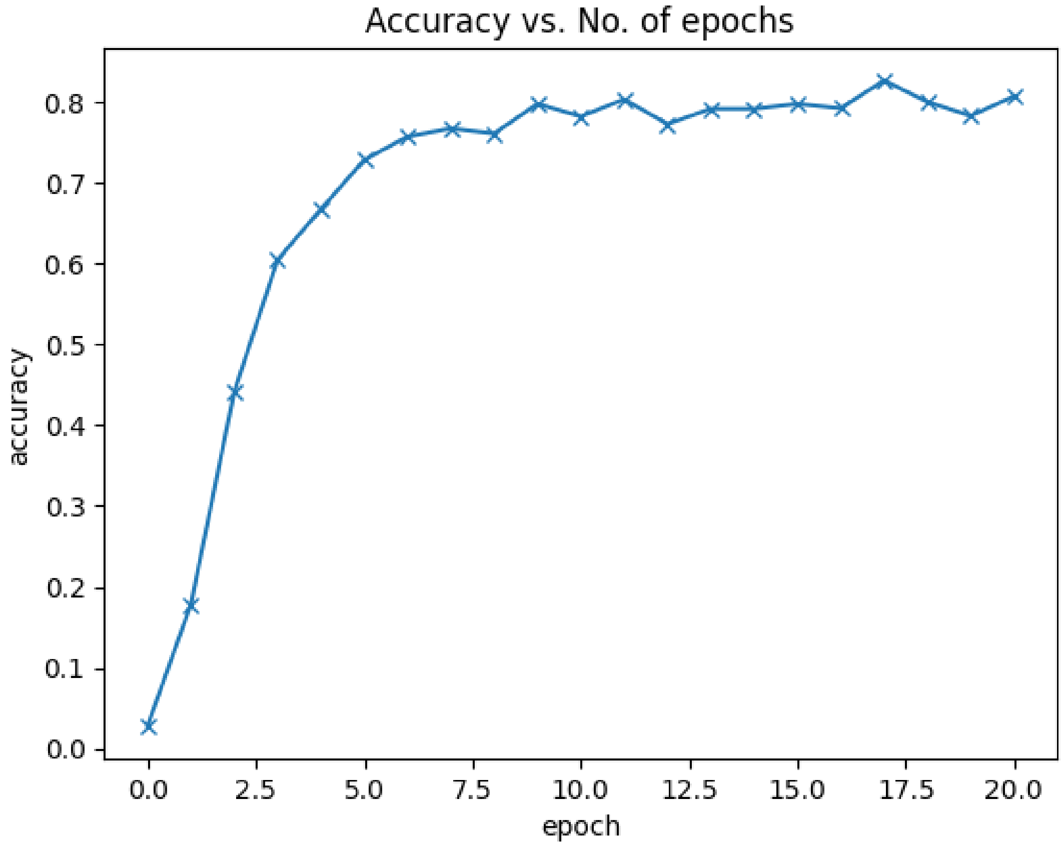
```
       )
     )
     (avgpool): AdaptiveAvgPool2d(output_size=(1, 1))
     (fc): Linear(in_features=2048, out_features=151, bias=True)
   )
 )
```

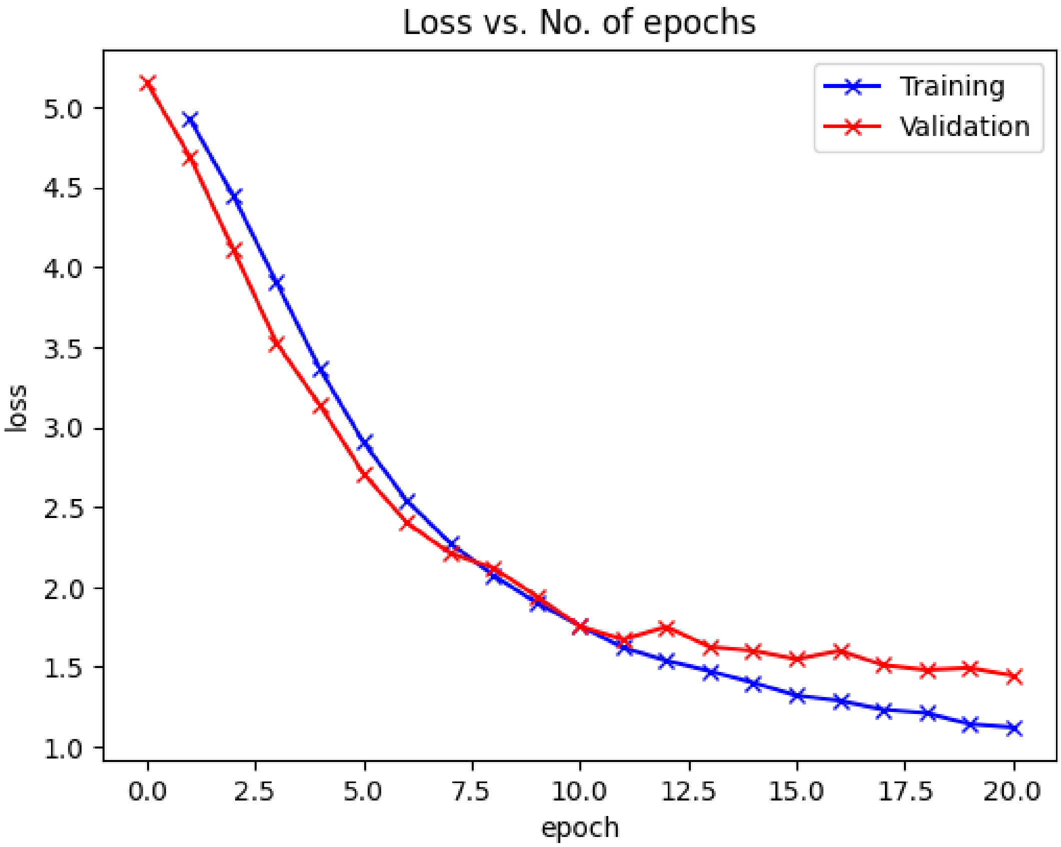
```
 [{'val_loss': 5.157966613769531, 'val_acc': 0.028125001117587089}]
```

```
627
```

```
  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [0], train_loss: 4.9248, val_loss: 4.6912, val_acc: 0.1781

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [1], train_loss: 4.4483, val_loss: 4.1113, val_acc: 0.4410

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [2], train_loss: 3.9039, val_loss: 3.5254, val_acc: 0.6045

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [3], train_loss: 3.3648, val_loss: 3.1376, val_acc: 0.6670

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [4], train_loss: 2.9085, val_loss: 2.7114, val_acc: 0.7281

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [5], train_loss: 2.5424, val_loss: 2.4042, val_acc: 0.7569

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [6], train_loss: 2.2765, val_loss: 2.2119, val_acc: 0.7670

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [7], train_loss: 2.0725, val_loss: 2.1190, val_acc: 0.7608

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [8], train_loss: 1.9021, val_loss: 1.9431, val_acc: 0.7976

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [9], train_loss: 1.7576, val_loss: 1.7518, val_acc: 0.7819

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [10], train_loss: 1.6205, val_loss: 1.6727, val_acc: 0.8031

  0%|          | 0/334 [00:00<?, ?it/s]

Epoch [11], train_loss: 1.5389, val_loss: 1.7498, val_acc: 0.7726

  0%|          | 0/334 [00:00<?, ?it/s]
```

```
Epoch [12], train_loss: 1.4735, val_loss: 1.6254, val_acc: 0.7913
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [13], train_loss: 1.4014, val_loss: 1.6008, val_acc: 0.7913
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [14], train_loss: 1.3214, val_loss: 1.5484, val_acc: 0.7976
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [15], train_loss: 1.2885, val_loss: 1.6005, val_acc: 0.7920
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [16], train_loss: 1.2336, val_loss: 1.5111, val_acc: 0.8264
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [17], train_loss: 1.2116, val_loss: 1.4810, val_acc: 0.8000
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [18], train_loss: 1.1422, val_loss: 1.4941, val_acc: 0.7826
  0%|          | 0/334 [00:00<?, ?it/s]
Epoch [19], train_loss: 1.1215, val_loss: 1.4455, val_acc: 0.8069
```





```
{'val_loss': 1.4172265529632568, 'val_acc': 0.8197917342185974}
```

1.1 FLOPs

```
+ Number of FLOPs: 2.15G
```