CS543 – Assignment 4

Pallaw Kumar (pallawk2)

Question 1.

1.1

model by using Python print command:

```
BaseNet(
  (conv1): Conv2d(1, 6, kernel_size=(5, 5), stride=(1, 1))
  (relu1): ReLU()
  (pool1): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
  (conv2): Conv2d(6, 12, kernel_size=(5, 5), stride=(1, 1))
  (relu2): ReLU()
  (pool2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
  (fc1): Linear(in_features=192, out_features=12, bias=True)
  (relu3): ReLU()
  (fc2): Linear(in_features=12, out_features=10, bias=True)
)
```

Final accuracy on the validation set:

```
Accuracy of the final network on the val images: 87.0 %
Accuracy of T-shirt/top : 90.6 %
Accuracy of Trouser : 98.4 %
Accuracy of Pullover : 71.8 %
Accuracy of Dress : 83.3 %
Accuracy of  Coat : 80.6 %
Accuracy of Sandal : 97.6 %
Accuracy of Shirt : 61.0 %
Accuracy of Sneaker : 93.3 %
Accuracy of   Bag : 96.1 %
Accuracy of Ankle boot : 95.6 %
```

```
Question 1.2
```

Best model:

```
BaseNet(
  (conv1): Conv2d(1, 32, kernel_size=(5, 5), stride=(1, 1))
  (bn1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  (conv2): Conv2d(32, 64, kernel_size=(5, 5), stride=(1, 1))
  (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  (fc1): Linear(in_features=1024, out_features=512, bias=True)
  (bn3): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  (fc2): Linear(in_features=512, out_features=256, bias=True)
  (bn4): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  (fc3): Linear(in_features=256, out_features=128, bias=True)
  (bn5): BatchNorm1d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  (fc5): Linear(in_features=128, out_features=10, bias=True)
  (pool1): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
  (pool2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
  (drop1): Dropout(p=0.2, inplace=False)
  (drop2): Dropout(p=0.2, inplace=False)
  (relu): ReLU()
)
```

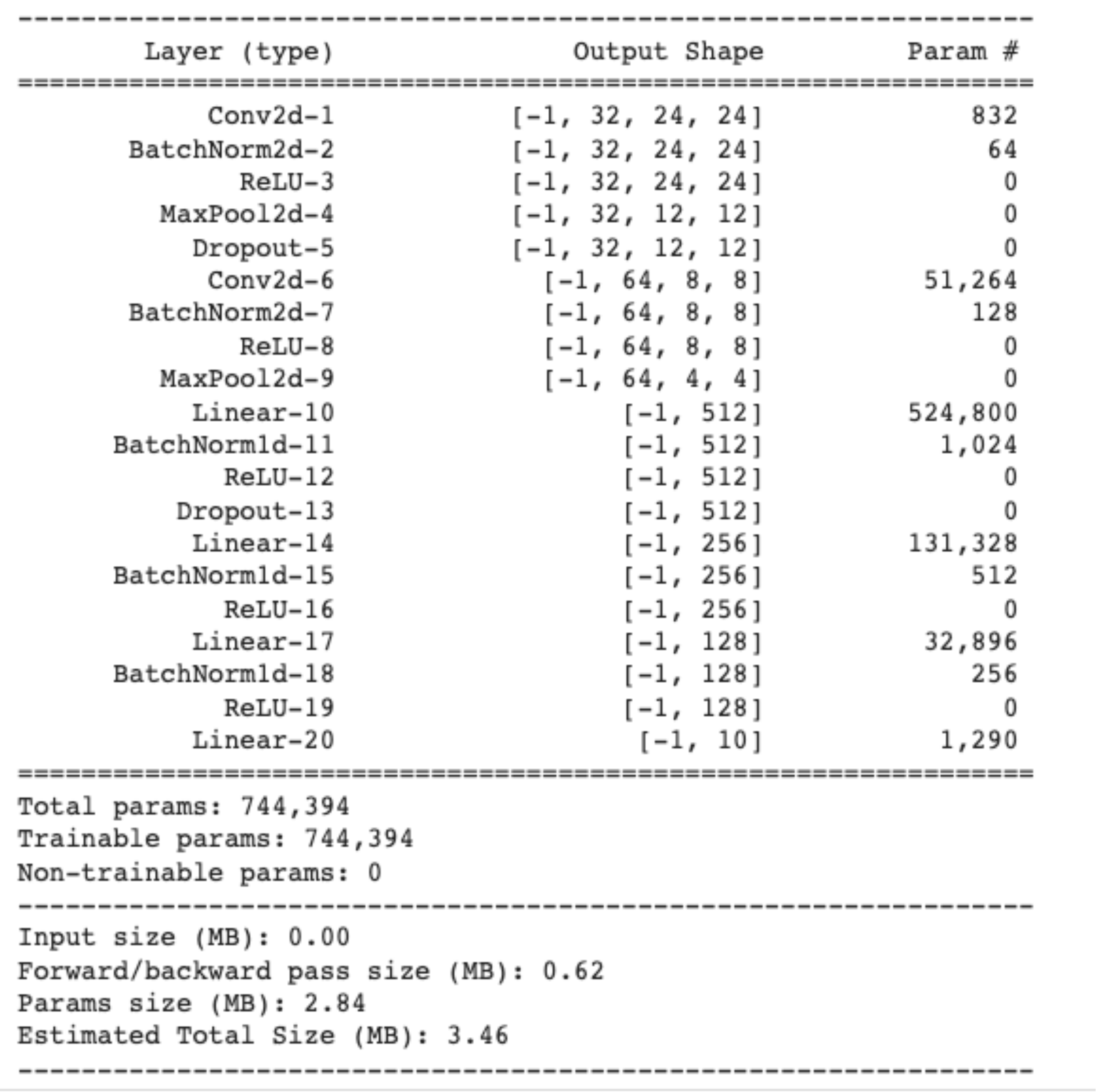
Final accuracy on validation set:

```
Accuracy of the final network on the val images: 92.1 %
Accuracy of T-shirt/top : 87.3 %
Accuracy of Trouser : 97.8 %
Accuracy of Pullover : 87.1 %
Accuracy of Dress : 93.3 %
Accuracy of  Coat : 86.3 %
Accuracy of Sandal : 97.6 %
Accuracy of Shirt : 79.4 %
Accuracy of Sneaker : 96.9 %
Accuracy of   Bag : 99.2 %
Accuracy of Ankle boot : 95.8 %
```

Table defining final architecture:

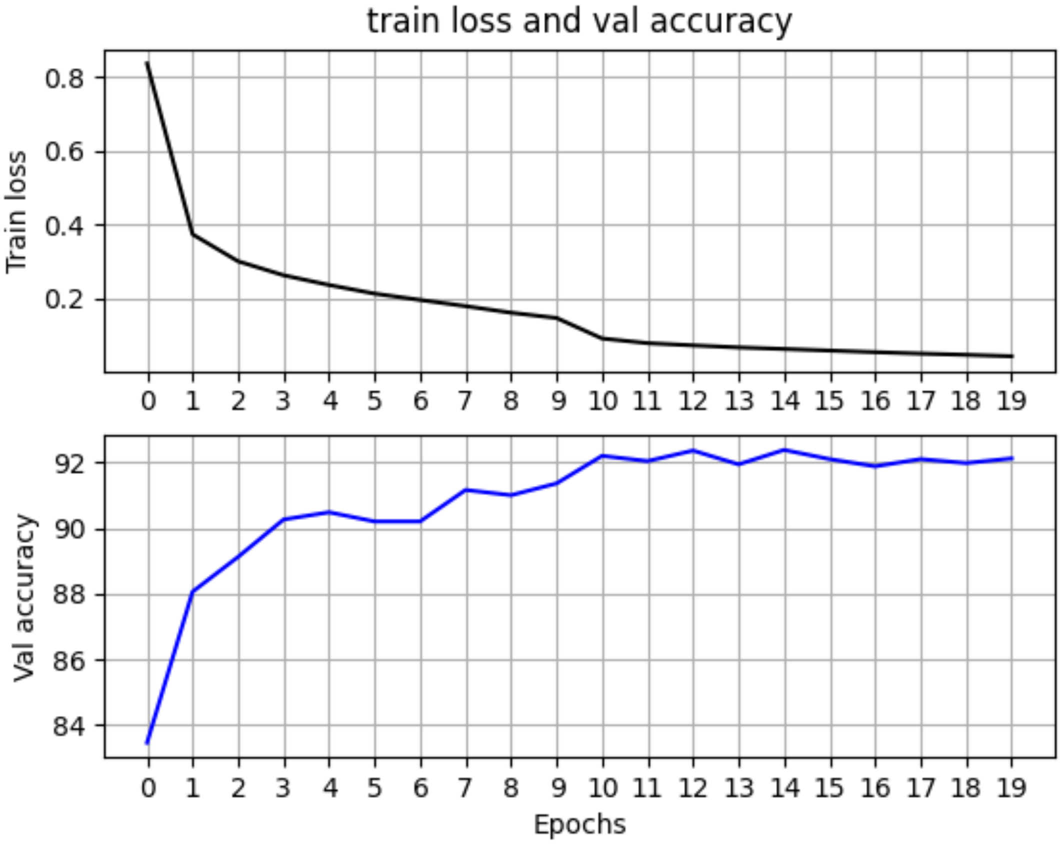
Layer No	Layer Type	Kernel size	Input Dim	Output Dim	Input Channels	Output Channels
1	Conv2d	5	28	24	1	32
2	BatchNorm2d	-	24	24	32	32
3	relu	-	24	24	32	32
4	Maxpool2d	2	24	12	32	32
5	Dropout	-	12	12	32	32
6	Conv2d	5	12	8	32	64
7	BatchNorm2d	-	8	8	64	64

8	relu	-	8	8	64	64
9	Maxpool2d	2	8	4	64	64
10	linear	-	1	1	1024	512
11	BatchNorm1d	-	1	1	1024	
12	relu	-	1	1	512	512
13	Dropout	-	1	1	512	512
14	linear	-	1	1	512	256
15	BatchNorm1d	-	1	1	256	256
16	relu	-	1	1	256	256
17	linear	-	1	1	256	128
18	BatchNorm1d	-	1	1	128	128
19	relu	-	1	1	128	128
20	linear	-	1	1	128	10



```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1           [-1, 32, 24, 24]             832
       BatchNorm2d-2           [-1, 32, 24, 24]              64
              ReLU-3           [-1, 32, 24, 24]               0
         MaxPool2d-4           [-1, 32, 12, 12]               0
           Dropout-5           [-1, 32, 12, 12]               0
            Conv2d-6             [-1, 64, 8, 8]          51,264
       BatchNorm2d-7             [-1, 64, 8, 8]             128
              ReLU-8             [-1, 64, 8, 8]               0
         MaxPool2d-9             [-1, 64, 4, 4]               0
           Linear-10                  [-1, 512]         524,800
      BatchNorm1d-11                  [-1, 512]           1,024
             ReLU-12                  [-1, 512]               0
          Dropout-13                  [-1, 512]               0
           Linear-14                  [-1, 256]         131,328
      BatchNorm1d-15                  [-1, 256]             512
             ReLU-16                  [-1, 256]               0
           Linear-17                  [-1, 128]          32,896
      BatchNorm1d-18                  [-1, 128]             256
             ReLU-19                  [-1, 128]               0
           Linear-20                   [-1, 10]           1,290
================================================================
Total params: 744,394
Trainable params: 744,394
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.62
Params size (MB): 2.84
Estimated Total Size (MB): 3.46
----------------------------------------------------------------
```

Training loss plot and test accuracy plot for final model:



Ablation table:

Model Components	Validation Accuracy
Baseline Model	87.0%
+ Data Augmentation	87.7 %
+ Batch Normalization	89.0 %
+ Dropout	88.8 %
+ Increased Model Depth	89.7 %
+ Increased Model Width	92.1 %

Test set accuracy (category-wise and aggregate) for best model from gradescope:

```
Accuracy: 91.2 %
Accuracy of T-shirt/top : 86.9 %
Accuracy of Trouser : 98.4 %
Accuracy of Pullover : 86.4 %
Accuracy of Dress : 88.4 %
Accuracy of  Coat : 86.6 %
Accuracy of Sandal : 96.9 %
Accuracy of Shirt : 76.0 %
Accuracy of Sneaker : 98.1 %
Accuracy of   Bag : 99.5 %
Accuracy of Ankle boot : 97.4 %
```

Question 2:

2.1

To implement upsampling I have used torch.nn.functional.interpolate function for resizing a tensor to a specified size using interpolation. In this case, the function is used to resize the output tensor x to a shape of (224, 224) using bilinear interpolation. The mode parameter 'bilinear' is used which performs linear interpolation between the 4 nearest neighbors of each output pixel.

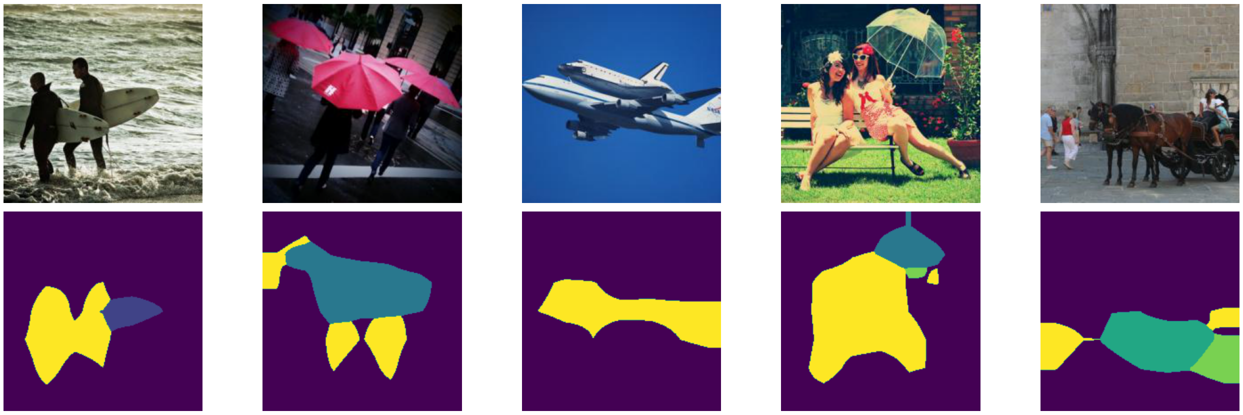
Layer No	Layer Type	Kernel size	Input Dim	Output Dim	Input Channels	Output Channels
1-17**	Pretrained ResNet18	-	224	7	3	512
19	Conv2d	-	7	7	512	6
20	Upsample	-	7	224	6	6

Training details:

a. The training loop was changed such that the training loss, mean AP, and iou (for the validation set) could be displayed or tracked for each iteration/epoch.

b. A 1e-4 learning rate was applied. Forty epochs were used to train the model. At epoch 30, the best model was discovered.

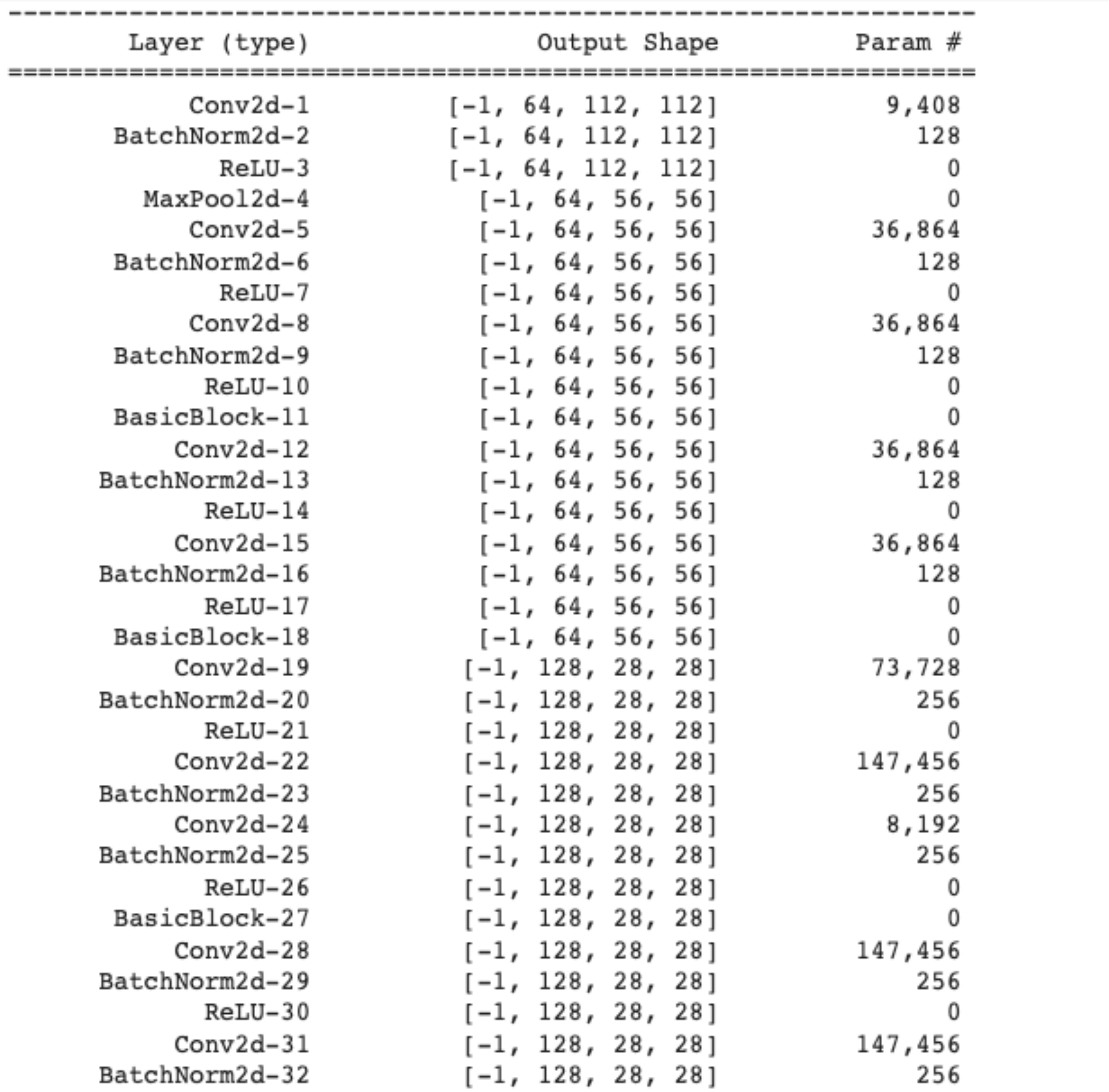
visualization of validation set:



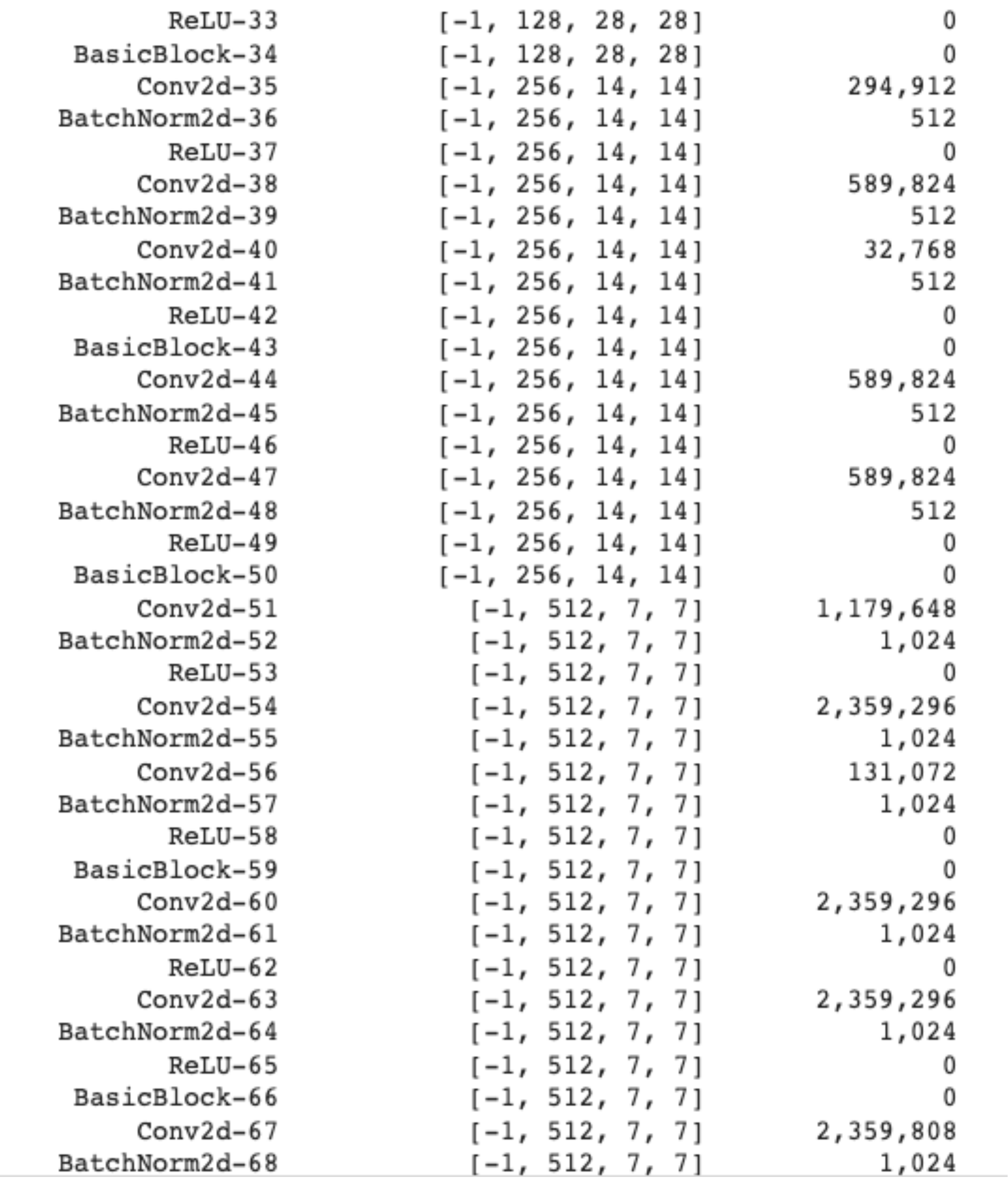
```
performance on validation set
background: AP: 0.93, IoU: 0.75
sports: AP: 0.51, IoU: 0.22
accessory: AP: 0.56, IoU: 0.33
animal: AP: 0.90, IoU: 0.67
vehicle: AP: 0.75, IoU: 0.50
person: AP: 0.83, IoU: 0.59
mean: AP: 0.75, IoU: 0.51
```

question 2.2

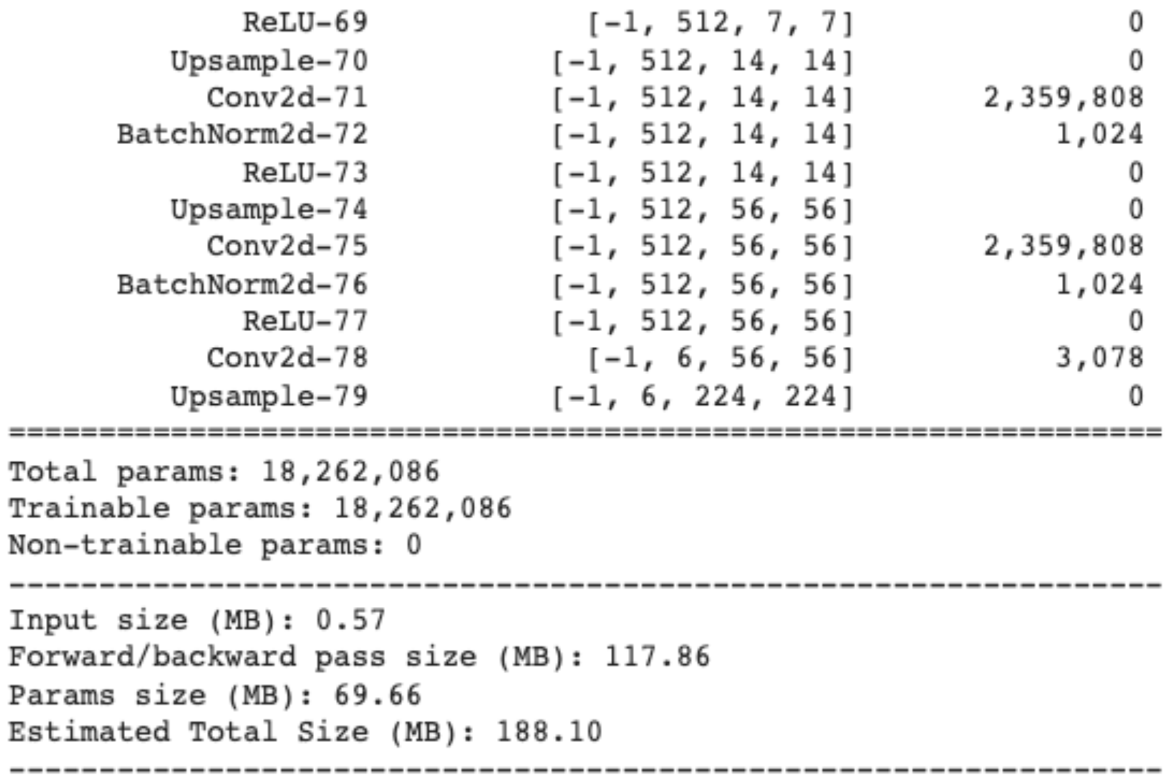
Best model:



```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1        [-1, 64, 112, 112]           9,408
       BatchNorm2d-2        [-1, 64, 112, 112]             128
              ReLU-3        [-1, 64, 112, 112]               0
         MaxPool2d-4          [-1, 64, 56, 56]               0
            Conv2d-5          [-1, 64, 56, 56]          36,864
       BatchNorm2d-6          [-1, 64, 56, 56]             128
              ReLU-7          [-1, 64, 56, 56]               0
            Conv2d-8          [-1, 64, 56, 56]          36,864
       BatchNorm2d-9          [-1, 64, 56, 56]             128
             ReLU-10          [-1, 64, 56, 56]               0
       BasicBlock-11          [-1, 64, 56, 56]               0
           Conv2d-12          [-1, 64, 56, 56]          36,864
      BatchNorm2d-13          [-1, 64, 56, 56]             128
             ReLU-14          [-1, 64, 56, 56]               0
           Conv2d-15          [-1, 64, 56, 56]          36,864
      BatchNorm2d-16          [-1, 64, 56, 56]             128
             ReLU-17          [-1, 64, 56, 56]               0
       BasicBlock-18          [-1, 64, 56, 56]               0
           Conv2d-19         [-1, 128, 28, 28]          73,728
      BatchNorm2d-20         [-1, 128, 28, 28]             256
             ReLU-21         [-1, 128, 28, 28]               0
           Conv2d-22         [-1, 128, 28, 28]         147,456
      BatchNorm2d-23         [-1, 128, 28, 28]             256
           Conv2d-24         [-1, 128, 28, 28]           8,192
      BatchNorm2d-25         [-1, 128, 28, 28]             256
             ReLU-26         [-1, 128, 28, 28]               0
       BasicBlock-27         [-1, 128, 28, 28]               0
           Conv2d-28         [-1, 128, 28, 28]         147,456
      BatchNorm2d-29         [-1, 128, 28, 28]             256
             ReLU-30         [-1, 128, 28, 28]               0
           Conv2d-31         [-1, 128, 28, 28]         147,456
      BatchNorm2d-32         [-1, 128, 28, 28]             256
```

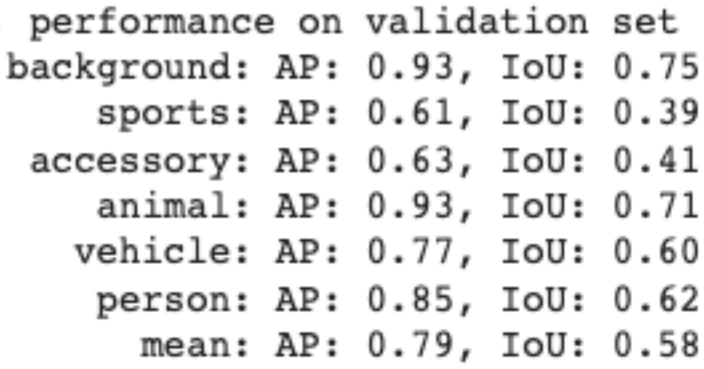


```
           ReLU-33          [-1, 128, 28, 28]               0
     BasicBlock-34          [-1, 128, 28, 28]               0
         Conv2d-35          [-1, 256, 14, 14]         294,912
    BatchNorm2d-36          [-1, 256, 14, 14]             512
           ReLU-37          [-1, 256, 14, 14]               0
         Conv2d-38          [-1, 256, 14, 14]         589,824
    BatchNorm2d-39          [-1, 256, 14, 14]             512
         Conv2d-40          [-1, 256, 14, 14]          32,768
    BatchNorm2d-41          [-1, 256, 14, 14]             512
           ReLU-42          [-1, 256, 14, 14]               0
     BasicBlock-43          [-1, 256, 14, 14]               0
         Conv2d-44          [-1, 256, 14, 14]         589,824
    BatchNorm2d-45          [-1, 256, 14, 14]             512
           ReLU-46          [-1, 256, 14, 14]               0
         Conv2d-47          [-1, 256, 14, 14]         589,824
    BatchNorm2d-48          [-1, 256, 14, 14]             512
           ReLU-49          [-1, 256, 14, 14]               0
     BasicBlock-50          [-1, 256, 14, 14]               0
         Conv2d-51            [-1, 512, 7, 7]       1,179,648
    BatchNorm2d-52            [-1, 512, 7, 7]           1,024
           ReLU-53            [-1, 512, 7, 7]               0
         Conv2d-54            [-1, 512, 7, 7]       2,359,296
    BatchNorm2d-55            [-1, 512, 7, 7]           1,024
         Conv2d-56            [-1, 512, 7, 7]         131,072
    BatchNorm2d-57            [-1, 512, 7, 7]           1,024
           ReLU-58            [-1, 512, 7, 7]               0
     BasicBlock-59            [-1, 512, 7, 7]               0
         Conv2d-60            [-1, 512, 7, 7]       2,359,296
    BatchNorm2d-61            [-1, 512, 7, 7]           1,024
           ReLU-62            [-1, 512, 7, 7]               0
         Conv2d-63            [-1, 512, 7, 7]       2,359,296
    BatchNorm2d-64            [-1, 512, 7, 7]           1,024
           ReLU-65            [-1, 512, 7, 7]               0
     BasicBlock-66            [-1, 512, 7, 7]               0
         Conv2d-67            [-1, 512, 7, 7]       2,359,808
    BatchNorm2d-68            [-1, 512, 7, 7]           1,024
```



```
           ReLU-69          [-1, 512, 7, 7]               0
       Upsample-70        [-1, 512, 14, 14]               0
         Conv2d-71        [-1, 512, 14, 14]       2,359,808
    BatchNorm2d-72        [-1, 512, 14, 14]           1,024
           ReLU-73        [-1, 512, 14, 14]               0
       Upsample-74        [-1, 512, 56, 56]               0
         Conv2d-75        [-1, 512, 56, 56]       2,359,808
    BatchNorm2d-76        [-1, 512, 56, 56]           1,024
           ReLU-77        [-1, 512, 56, 56]               0
         Conv2d-78          [-1, 6, 56, 56]           3,078
       Upsample-79        [-1, 6, 224, 224]               0
================================================================
Total params: 18,262,086
Trainable params: 18,262,086
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.57
Forward/backward pass size (MB): 117.86
Params size (MB): 69.66
Estimated Total Size (MB): 188.10
----------------------------------------------------------------
```

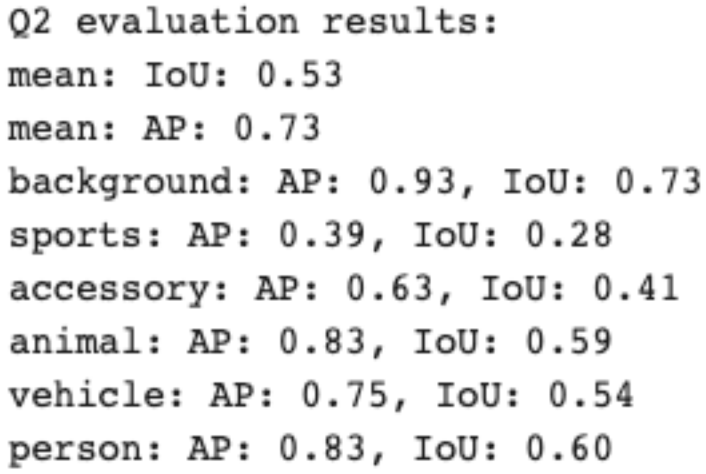
Final performance on validation set:
The final performance on the validation set was 0.79 mAP and 0.58 mIoU



```
 performance on validation set
background: AP: 0.93, IoU: 0.75
    sports: AP: 0.61, IoU: 0.39
 accessory: AP: 0.63, IoU: 0.41
    animal: AP: 0.93, IoU: 0.71
   vehicle: AP: 0.77, IoU: 0.60
    person: AP: 0.85, IoU: 0.62
      mean: AP: 0.79, IoU: 0.58
```

Model Components	Mean AP	mean IoU
Baseline Model	0.74	0.51
+ Data Augmentation	0.75	0.52
+ dilated convolution layers upsampling by 32	0.76	0.54
+ Batch Normalization	0.76	0.55
+ Increased Model Depth by adding more dilated convolution layers and upsampling step by step	0.79	0.58

Test set accuracy of best model from gradescope:



```
Q2 evaluation results:
mean: IoU: 0.53
mean: AP: 0.73
background: AP: 0.93, IoU: 0.73
sports: AP: 0.39, IoU: 0.28
accessory: AP: 0.63, IoU: 0.41
animal: AP: 0.83, IoU: 0.59
vehicle: AP: 0.75, IoU: 0.54
person: AP: 0.83, IoU: 0.60
```

Visualization: