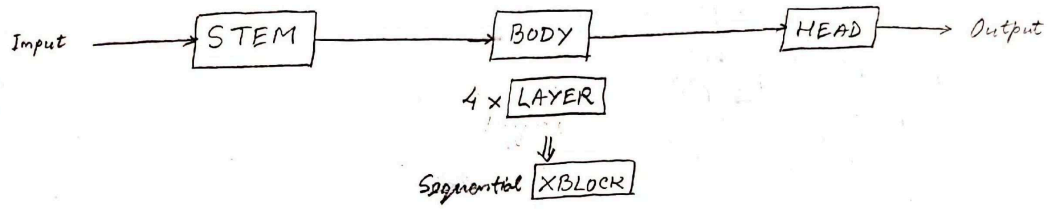
Project 02 – Networks

Rajarshi Chattopadhyay
RXC170010
rajarshi.c@utdallas.edu

1. Design

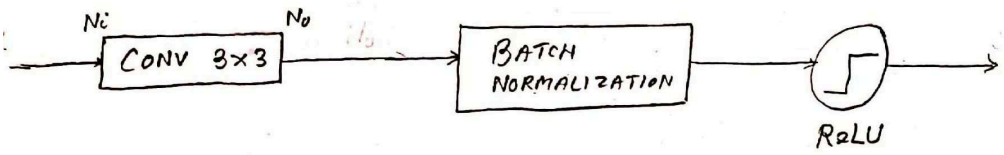
- Network Structure




```
Model(
      (stem): Stem()
      (body): Sequential(
            (0): Layer(Sequential(XBlock()…))
            (1): Layer(Sequential(XBlock()…))
            (2): Layer(Sequential(XBlock()…))
            (3): Layer(Sequential(XBlock()…)))
      (head): Head()
)
```

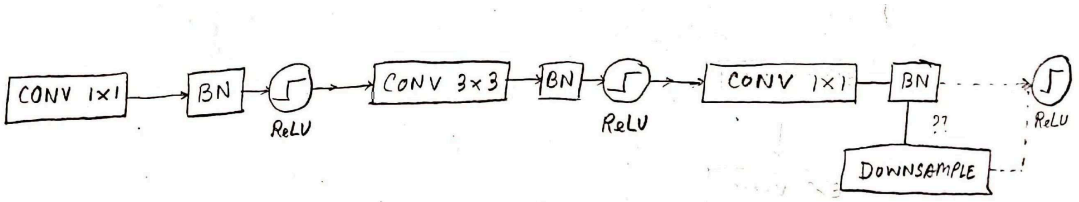
- Network standard building blocks

Stem



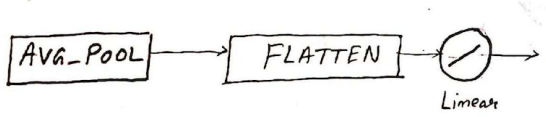

```
Stem(
      (conv_3x3): Conv2d
      (bn): BatchNorm2d
)
```

XBlock




```
XBlock(
      (conv1_1x1): Conv2d
      (bn1): BatchNorm2d
      (conv2_3x3): Conv2d
      (bn2): BatchNorm2d
      (downsample): Downsample
      (conv3_1x1): Conv2d
      (bn3): BatchNorm2d
)
```

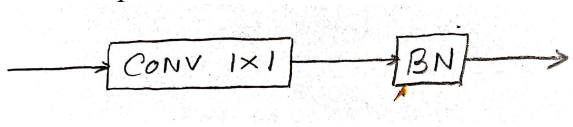
Head




```
Head(
      (avg_pool): AdaptiveAvgPool2d(output_size=1)
      (f_out): Linear(in_features=376, out_features=100, bias=True)
)
```

The network down sampling building block

Downsample




```
Downsample(
      (conv_1x1): Conv2d
      (bn): BatchNorm2d
)
```

Parameters

channels	3
classes	100
network depth	13
initial width	24
slope	36
quantization	2.5
bottleneck ratio	1
group width	8

```
Model(
  (stem): Stem(
    (conv_3x3): Conv2d(3, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
  (body): Sequential(
    (0): Layer(
      (layers): Sequential(
        (0): XBlock(
          (conv1_1x1): Conv2d(32, 32, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (conv2 3x3): Conv2d(32, 32, kernel size=(3, 3), stride=(1, 1), padding=(1, 1), groups=4, bias=False)
          (bn2): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (downsample): Downsample(
            (conv_1x1): Conv2d(32, 24, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (bn): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          )
          (conv3_1x1): Conv2d(32, 24, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn3): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        )
      )
    )
    (1): Layer(
      (layers): Sequential(
        (0): XBlock(
          (conv1_1x1): Conv2d(24, 24, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn1): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (conv2_3x3): Conv2d(24, 24, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=3, bias=False)
          (bn2): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (downsample): Downsample(
            (conv 1x1): Conv2d(24, 64, kernel size=(1, 1), stride=(2, 2), bias=False)
            (bn): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          )
          (conv3 1x1): Conv2d(24, 64, kernel size=(1, 1), stride=(1, 1), bias=False)
          (bn3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        )
      )
    )
    (2): Layer(
      (layers): Sequential(
        (0): XBlock(
          (conv1_1x1): 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_3x3): Conv2d(64, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=8, bias=False)
          (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (downsample): Downsample(
            (conv_1x1): Conv2d(64, 152, kernel_size=(1, 1), stride=(2, 2), bias=False)
            (bn): BatchNorm2d(152, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          )
          (conv3_1x1): Conv2d(64, 152, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn3): BatchNorm2d(152, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        )
        (1): XBlock(
          (conv1_1x1): Conv2d(152, 152, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn1): BatchNorm2d(152, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (conv2 3x3): Conv2d(152, 152, kernel size=(3, 3), stride=(1, 1), padding=(1, 1), groups=19, bias=False)
          (bn2): BatchNorm2d(152, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (conv3_1x1): Conv2d(152, 152, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn3): BatchNorm2d(152, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        )
        (2): XBlock(
          (conv1_1x1): Conv2d(152, 152, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn1): BatchNorm2d(152, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (conv2_3x3): Conv2d(152, 152, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=19, bias=False)
          (bn2): BatchNorm2d(152, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (conv3_1x1): Conv2d(152, 152, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn3): BatchNorm2d(152, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        )
```

```
        (3): XBlock(
          (conv1_1x1): Conv2d(152, 152, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn1): BatchNorm2d(152, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (conv2_3x3): Conv2d(152, 152, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=19, bias=False)
          (bn2): BatchNorm2d(152, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (conv3_1x1): Conv2d(152, 152, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn3): BatchNorm2d(152, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        )
      )
    )
    (3): Layer(
      (layers): Sequential(
        (0): XBlock(
          (conv1_1x1): Conv2d(152, 152, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn1): BatchNorm2d(152, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (conv2_3x3): Conv2d(152, 152, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), groups=19, bias=False)
          (bn2): BatchNorm2d(152, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (downsample): Downsample(
            (conv_1x1): Conv2d(152, 376, kernel_size=(1, 1), stride=(2, 2), bias=False)
            (bn): BatchNorm2d(376, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          )
          (conv3_1x1): Conv2d(152, 376, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn3): BatchNorm2d(376, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        )
        (1): XBlock(
          (conv1_1x1): Conv2d(376, 376, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn1): BatchNorm2d(376, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (conv2_3x3): Conv2d(376, 376, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=47, bias=False)
          (bn2): BatchNorm2d(376, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (conv3_1x1): Conv2d(376, 376, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn3): BatchNorm2d(376, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        )
        (2): XBlock(
          (conv1_1x1): Conv2d(376, 376, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn1): BatchNorm2d(376, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (conv2_3x3): Conv2d(376, 376, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=47, bias=False)
          (bn2): BatchNorm2d(376, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (conv3_1x1): Conv2d(376, 376, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn3): BatchNorm2d(376, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        )
        (3): XBlock(
          (conv1_1x1): Conv2d(376, 376, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn1): BatchNorm2d(376, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (conv2_3x3): Conv2d(376, 376, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=47, bias=False)
          (bn2): BatchNorm2d(376, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (conv3_1x1): Conv2d(376, 376, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn3): BatchNorm2d(376, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        )
        (4): XBlock(
          (conv1_1x1): Conv2d(376, 376, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn1): BatchNorm2d(376, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (conv2_3x3): Conv2d(376, 376, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=47, bias=False)
          (bn2): BatchNorm2d(376, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (conv3_1x1): Conv2d(376, 376, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn3): BatchNorm2d(376, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        )
        (5): XBlock(
          (conv1_1x1): Conv2d(376, 376, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn1): BatchNorm2d(376, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (conv2_3x3): Conv2d(376, 376, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=47, bias=False)
          (bn2): BatchNorm2d(376, eps=1e-05, momentum=0.1, affine=True, track running stats=True)
          (conv3_1x1): Conv2d(376, 376, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn3): BatchNorm2d(376, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        )
        (6): XBlock(
          (conv1_1x1): Conv2d(376, 376, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn1): BatchNorm2d(376, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (conv2_3x3): Conv2d(376, 376, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=47, bias=False)
          (bn2): BatchNorm2d(376, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
          (conv3_1x1): Conv2d(376, 376, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn3): BatchNorm2d(376, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        )
      )
    )
  )
  (head): Head(
    (avg_pool): AdaptiveAvgPool2d(output_size=1)
    (f_out): Linear(in_features=376, out_features=100, bias=True)
  )
)
```

2. Training

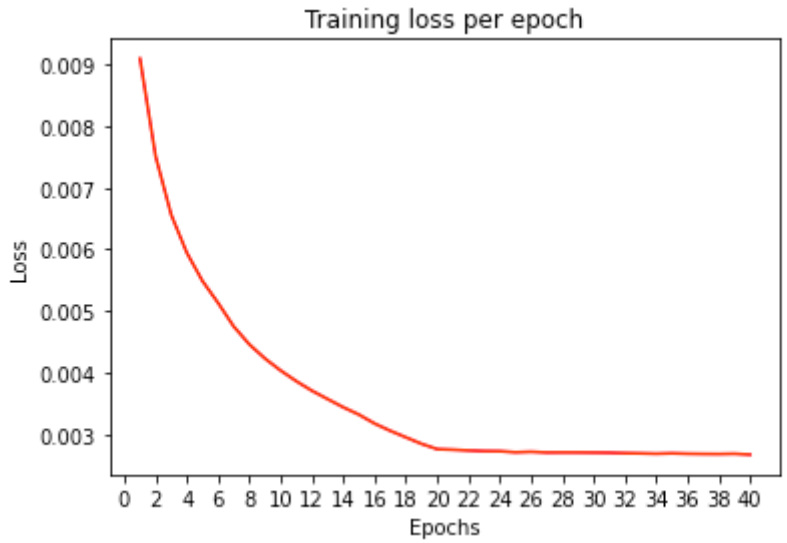
- Hyperparameters

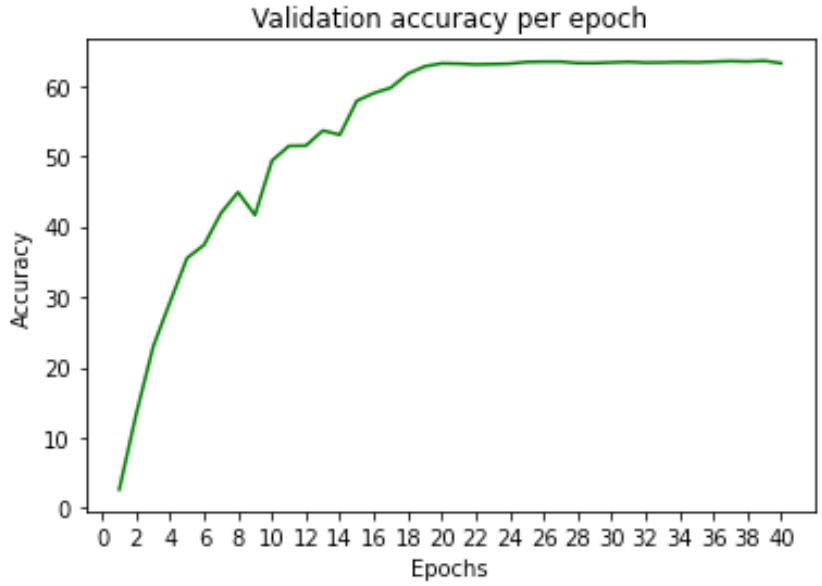
learning rate max value	0.001
learning rate initial	0.001*0.01
initial epochs	5
learning rate final	0.001*0.01
final epochs	35

Error: Cross Entropy Loss
Optimizer: Adam Optimizer(learning rate: 0.001)

- Training loss and accuracy per epoch

```
Epoch  1 lr = 0.000010 avg_loss = 0.009097 accuracy =  2.60% time = 178.71s
Epoch  2 lr = 0.000208 avg_loss = 0.007510 accuracy = 13.32% time = 179.47s
Epoch  3 lr = 0.000406 avg_loss = 0.006559 accuracy = 22.92% time = 179.63s
Epoch  4 lr = 0.000604 avg_loss = 0.005941 accuracy = 29.32% time = 179.19s
Epoch  5 lr = 0.000802 avg_loss = 0.005484 accuracy = 35.53% time = 178.96s
Epoch  6 lr = 0.001000 avg_loss = 0.005128 accuracy = 37.39% time = 179.47s
Epoch  7 lr = 0.000994 avg_loss = 0.004745 accuracy = 41.93% time = 179.42s
Epoch  8 lr = 0.000975 avg_loss = 0.004455 accuracy = 44.88% time = 179.25s
Epoch  9 lr = 0.000944 avg_loss = 0.004225 accuracy = 41.62% time = 178.97s
Epoch 10 lr = 0.000902 avg_loss = 0.004031 accuracy = 49.37% time = 179.24s
Epoch 11 lr = 0.000848 avg_loss = 0.003858 accuracy = 51.45% time = 179.34s
Epoch 12 lr = 0.000784 avg_loss = 0.003703 accuracy = 51.50% time = 179.25s
Epoch 13 lr = 0.000710 avg_loss = 0.003568 accuracy = 53.65% time = 179.17s
Epoch 14 lr = 0.000627 avg_loss = 0.003435 accuracy = 53.04% time = 179.21s
Epoch 15 lr = 0.000537 avg_loss = 0.003316 accuracy = 57.88% time = 179.16s
Epoch 16 lr = 0.000440 avg_loss = 0.003171 accuracy = 58.96% time = 179.17s
Epoch 17 lr = 0.000337 avg_loss = 0.003054 accuracy = 59.72% time = 179.55s
Epoch 18 lr = 0.000230 avg_loss = 0.002947 accuracy = 61.72% time = 179.24s
Epoch 19 lr = 0.000121 avg_loss = 0.002842 accuracy = 62.76% time = 179.35s
Epoch 20 lr = 0.000010 avg_loss = 0.002756 accuracy = 63.19% time = 179.19s
Epoch 21 lr = 0.000010 avg_loss = 0.002747 accuracy = 63.15% time = 179.57s
Epoch 22 lr = 0.000010 avg_loss = 0.002733 accuracy = 63.04% time = 179.52s
Epoch 23 lr = 0.000010 avg_loss = 0.002725 accuracy = 63.09% time = 179.41s
Epoch 24 lr = 0.000010 avg_loss = 0.002723 accuracy = 63.15% time = 179.43s
Epoch 25 lr = 0.000010 avg_loss = 0.002704 accuracy = 63.39% time = 179.57s
Epoch 26 lr = 0.000010 avg_loss = 0.002713 accuracy = 63.43% time = 179.56s
Epoch 27 lr = 0.000010 avg_loss = 0.002700 accuracy = 63.43% time = 179.36s
Epoch 28 lr = 0.000010 avg_loss = 0.002702 accuracy = 63.24% time = 179.30s
Epoch 29 lr = 0.000010 avg_loss = 0.002701 accuracy = 63.24% time = 179.76s
Epoch 30 lr = 0.000010 avg_loss = 0.002699 accuracy = 63.32% time = 179.49s
Epoch 31 lr = 0.000010 avg_loss = 0.002697 accuracy = 63.41% time = 179.67s
Epoch 32 lr = 0.000010 avg_loss = 0.002691 accuracy = 63.30% time = 179.61s
Epoch 33 lr = 0.000010 avg_loss = 0.002687 accuracy = 63.32% time = 179.46s
Epoch 34 lr = 0.000010 avg_loss = 0.002681 accuracy = 63.39% time = 179.36s
Epoch 35 lr = 0.000010 avg_loss = 0.002688 accuracy = 63.35% time = 178.65s
Epoch 36 lr = 0.000010 avg_loss = 0.002681 accuracy = 63.43% time = 179.29s
Epoch 37 lr = 0.000010 avg_loss = 0.002678 accuracy = 63.52% time = 179.02s
Epoch 38 lr = 0.000010 avg_loss = 0.002676 accuracy = 63.45% time = 179.11s
Epoch 39 lr = 0.000010 avg_loss = 0.002681 accuracy = 63.56% time = 178.97s
Epoch 40 lr = 0.000010 avg_loss = 0.002667 accuracy = 63.22% time = 179.32s
```





- Testing Accuracy

```
Testing Accuracy of test set = 63.22%

Class accuracy:
Accuracy of plane = 82.00
Accuracy of car   = 70.00
Accuracy of bird  = 66.00
Accuracy of cat   = 68.00
Accuracy of deer  = 54.00
Accuracy of dog   = 48.00
Accuracy of frog  = 58.00
Accuracy of horse = 48.00
Accuracy of ship  = 94.00
Accuracy of truck = 64.00
```

3. Implementation

MACs and parameters per operator

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 32, 56, 56]	864
BatchNorm2d-2	[-1, 32, 56, 56]	64
Stem-3	[-1, 32, 56, 56]	0
Conv2d-4	[-1, 32, 56, 56]	1,024
BatchNorm2d-5	[-1, 32, 56, 56]	64
Conv2d-6	[-1, 32, 56, 56]	2,304
BatchNorm2d-7	[-1, 32, 56, 56]	64
Conv2d-8	[-1, 24, 56, 56]	768
BatchNorm2d-9	[-1, 24, 56, 56]	48
Conv2d-10	[-1, 24, 56, 56]	768
BatchNorm2d-11	[-1, 24, 56, 56]	48
Downsample-12	[-1, 24, 56, 56]	0
XBlock-13	[-1, 24, 56, 56]	0
Layer-14	[-1, 24, 56, 56]	0
Conv2d-15	[-1, 24, 56, 56]	576
BatchNorm2d-16	[-1, 24, 56, 56]	48
Conv2d-17	[-1, 24, 28, 28]	1,728
BatchNorm2d-18	[-1, 24, 28, 28]	48
Conv2d-19	[-1, 64, 28, 28]	1,536
BatchNorm2d-20	[-1, 64, 28, 28]	128
Conv2d-21	[-1, 64, 28, 28]	1,536
BatchNorm2d-22	[-1, 64, 28, 28]	128
Downsample-23	[-1, 64, 28, 28]	0
XBlock-24	[-1, 64, 28, 28]	0
Layer-25	[-1, 64, 28, 28]	0
Conv2d-26	[-1, 64, 28, 28]	4,096
BatchNorm2d-27	[-1, 64, 28, 28]	128
Conv2d-28	[-1, 64, 14, 14]	4,608
BatchNorm2d-29	[-1, 64, 14, 14]	128
Conv2d-30	[-1, 152, 14, 14]	9,728
BatchNorm2d-31	[-1, 152, 14, 14]	304
Conv2d-32	[-1, 152, 14, 14]	9,728
BatchNorm2d-33	[-1, 152, 14, 14]	304
Downsample-34	[-1, 152, 14, 14]	0
XBlock-35	[-1, 152, 14, 14]	0
Conv2d-36	[-1, 152, 14, 14]	23,104
BatchNorm2d-37	[-1, 152, 14, 14]	304
Conv2d-38	[-1, 152, 14, 14]	10,944
BatchNorm2d-39	[-1, 152, 14, 14]	304
Conv2d-40	[-1, 152, 14, 14]	23,104
BatchNorm2d-41	[-1, 152, 14, 14]	304
XBlock-42	[-1, 152, 14, 14]	0
Conv2d-43	[-1, 152, 14, 14]	23,104
BatchNorm2d-44	[-1, 152, 14, 14]	304
Conv2d-45	[-1, 152, 14, 14]	10,944

BatchNorm2d-46	[-1, 152, 14, 14]	304
Conv2d-47	[-1, 152, 14, 14]	23,104
BatchNorm2d-48	[-1, 152, 14, 14]	304
XBlock-49	[-1, 152, 14, 14]	0
Conv2d-50	[-1, 152, 14, 14]	23,104
BatchNorm2d-51	[-1, 152, 14, 14]	304
Conv2d-52	[-1, 152, 14, 14]	10,944
BatchNorm2d-53	[-1, 152, 14, 14]	304
Conv2d-54	[-1, 152, 14, 14]	23,104
BatchNorm2d-55	[-1, 152, 14, 14]	304
XBlock-56	[-1, 152, 14, 14]	0
Layer-57	[-1, 152, 14, 14]	0
Conv2d-58	[-1, 152, 14, 14]	23,104
BatchNorm2d-59	[-1, 152, 14, 14]	304
Conv2d-60	[-1, 152, 7, 7]	10,944
BatchNorm2d-61	[-1, 152, 7, 7]	304
Conv2d-62	[-1, 376, 7, 7]	57,152
BatchNorm2d-63	[-1, 376, 7, 7]	752
Conv2d-64	[-1, 376, 7, 7]	57,152
BatchNorm2d-65	[-1, 376, 7, 7]	752
Downsample-66	[-1, 376, 7, 7]	0
XBlock-67	[-1, 376, 7, 7]	0
Conv2d-68	[-1, 376, 7, 7]	141,376
BatchNorm2d-69	[-1, 376, 7, 7]	752
Conv2d-70	[-1, 376, 7, 7]	27,072
BatchNorm2d-71	[-1, 376, 7, 7]	752
Conv2d-72	[-1, 376, 7, 7]	141,376
BatchNorm2d-73	[-1, 376, 7, 7]	752
XBlock-74	[-1, 376, 7, 7]	0
Conv2d-75	[-1, 376, 7, 7]	141,376
BatchNorm2d-76	[-1, 376, 7, 7]	752
Conv2d-77	[-1, 376, 7, 7]	27,072
BatchNorm2d-78	[-1, 376, 7, 7]	752
Conv2d-79	[-1, 376, 7, 7]	141,376
BatchNorm2d-80	[-1, 376, 7, 7]	752
XBlock-81	[-1, 376, 7, 7]	0
Conv2d-82	[-1, 376, 7, 7]	141,376
BatchNorm2d-83	[-1, 376, 7, 7]	752
Conv2d-84	[-1, 376, 7, 7]	27,072
BatchNorm2d-85	[-1, 376, 7, 7]	752
Conv2d-86	[-1, 376, 7, 7]	141,376
BatchNorm2d-87	[-1, 376, 7, 7]	752
XBlock-88	[-1, 376, 7, 7]	0
Conv2d-89	[-1, 376, 7, 7]	141,376
BatchNorm2d-90	[-1, 376, 7, 7]	752
Conv2d-91	[-1, 376, 7, 7]	27,072
BatchNorm2d-92	[-1, 376, 7, 7]	752
Conv2d-93	[-1, 376, 7, 7]	141,376
BatchNorm2d-94	[-1, 376, 7, 7]	752
XBlock-95	[-1, 376, 7, 7]	0
Conv2d-96	[-1, 376, 7, 7]	141,376
BatchNorm2d-97	[-1, 376, 7, 7]	752
Conv2d-98	[-1, 376, 7, 7]	27,072
BatchNorm2d-99	[-1, 376, 7, 7]	752

```
          Conv2d-100            [-1, 376, 7, 7]         141,376
     BatchNorm2d-101            [-1, 376, 7, 7]             752
          XBlock-102            [-1, 376, 7, 7]               0
          Conv2d-103            [-1, 376, 7, 7]         141,376
     BatchNorm2d-104            [-1, 376, 7, 7]             752
          Conv2d-105            [-1, 376, 7, 7]          27,072
     BatchNorm2d-106            [-1, 376, 7, 7]             752
          Conv2d-107            [-1, 376, 7, 7]         141,376
     BatchNorm2d-108            [-1, 376, 7, 7]             752
          XBlock-109            [-1, 376, 7, 7]               0
           Layer-110            [-1, 376, 7, 7]               0
AdaptiveAvgPool2d-111            [-1, 376, 1, 1]               0
          Linear-112                  [-1, 100]          37,700
            Head-113                  [-1, 100]               0
================================================================
Total params: 2,275,604
Trainable params: 2,275,604
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.04
Forward/backward pass size (MB): 28.28
Params size (MB): 8.68
Estimated Total Size (MB): 36.99
----------------------------------------------------------------
```

4. Extra

- **Learning rate scheduling** has been done with a linear increase for the first 5 epochs and then a Cosine decay for the final 35 epochs. (cnn.py)
- **Checkpointing** of the trained model have been done after every epoch so that in case RAM crashes, or execution is interrupted, the last saved model can be retrieved, and training can be continued. (cnn.py)
- **Early stopping** functionality has been implemented to stop training if 5 consecutive epochs do not lead to a decrease in average training loss. (cnn.py)
- A function has been created that also **prints the weights in each operator** in the network in the model summary. (extra.py)
- The model has been **visualized using torchviz** by forward propagating a zero input through the model. (extra.py)
- The model visualization has been done and exported as ***cnn.onnx*** file which can be viewed using ***Netron***. (extra.py)