Assignment 6

Vishal Batvia

CMPT 762 Computer Vision

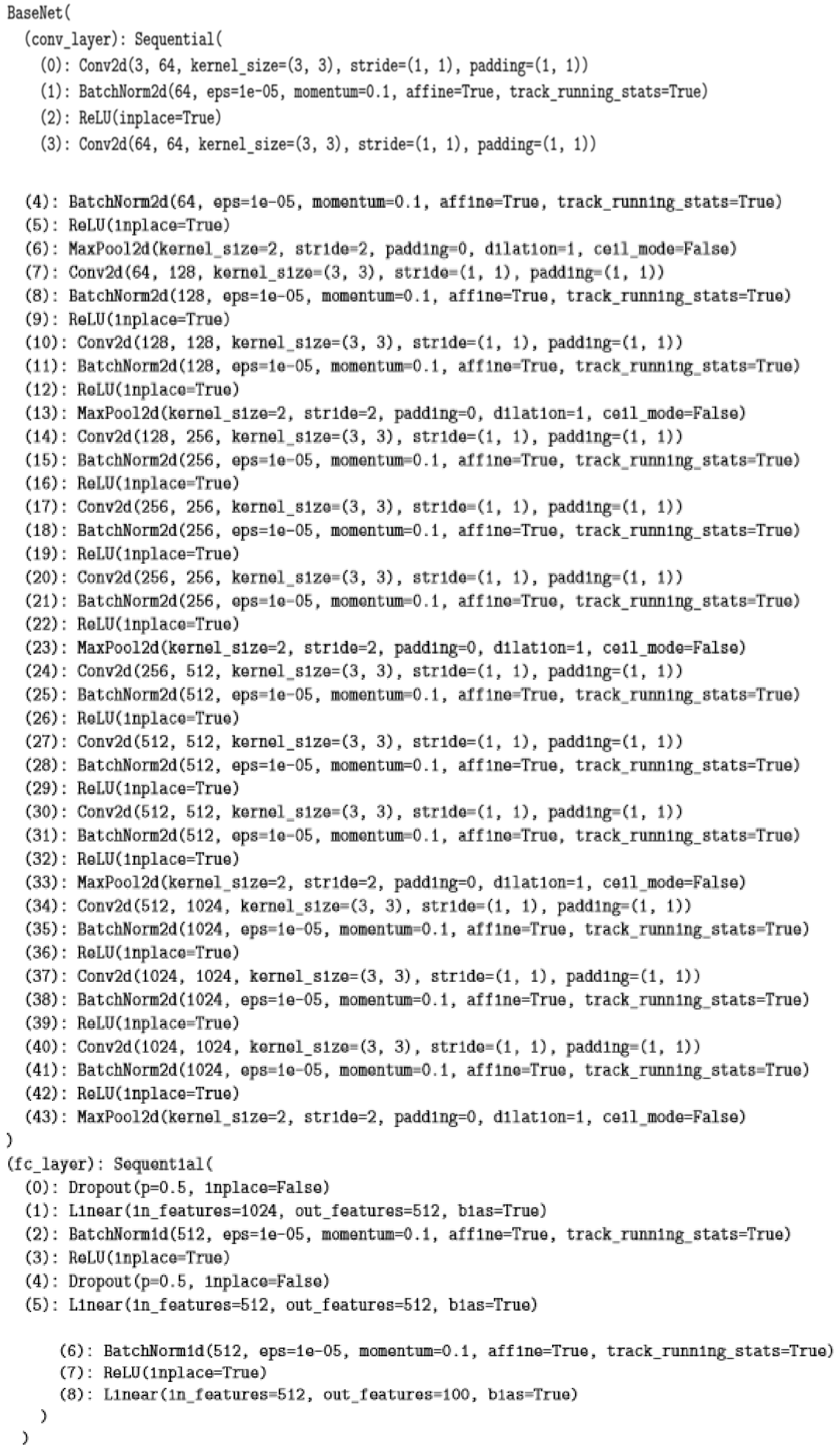
Author Note

Kaggle Name: - VishalBatavia

Assignment 6

1. Part 1 BaseNet on CIFAR 100

1.1. Improved Network Architecture



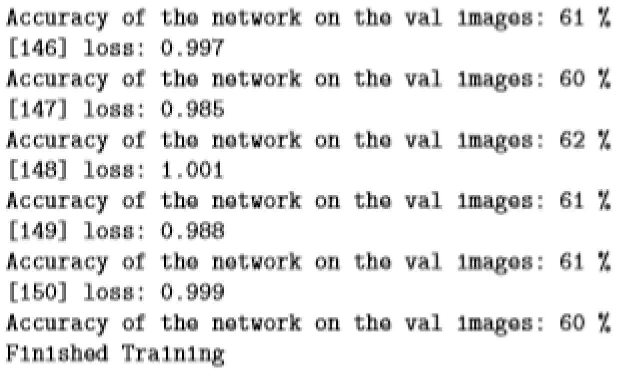
```
BaseNet(
  (conv_layer): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))

    (4): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (7): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (8): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (9): ReLU(inplace=True)
    (10): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (12): ReLU(inplace=True)
    (13): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (14): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (15): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (16): ReLU(inplace=True)
    (17): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (18): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (19): ReLU(inplace=True)
    (20): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (21): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (22): ReLU(inplace=True)
    (23): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (24): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (25): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (26): ReLU(inplace=True)
    (27): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (28): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (29): ReLU(inplace=True)
    (30): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (31): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (32): ReLU(inplace=True)
    (33): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (34): Conv2d(512, 1024, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (35): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (36): ReLU(inplace=True)
    (37): Conv2d(1024, 1024, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (38): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (39): ReLU(inplace=True)
    (40): Conv2d(1024, 1024, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (41): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (42): ReLU(inplace=True)
    (43): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (fc_layer): Sequential(
    (0): Dropout(p=0.5, inplace=False)
    (1): Linear(in_features=1024, out_features=512, bias=True)
    (2): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (3): ReLU(inplace=True)
    (4): Dropout(p=0.5, inplace=False)
    (5): Linear(in_features=512, out_features=512, bias=True)

    (6): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): ReLU(inplace=True)
    (8): Linear(in_features=512, out_features=100, bias=True)
  )
)
```

1.2. Improvements in accuracy

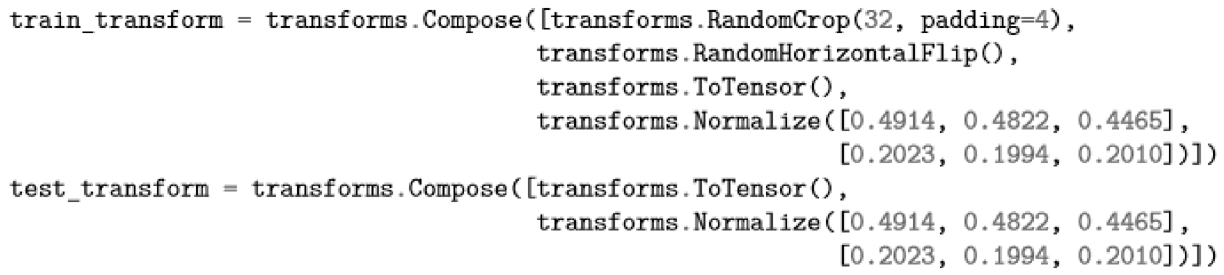
The accuracy with the vanilla model Basenet was 21% on the test set on Kaggle while the accuracy of the final improved network was 64.5% on test set on Kaggle and 62% on validation set while training.



```
Accuracy of the network on the val images: 61 %
[146] loss: 0.997
Accuracy of the network on the val images: 60 %
[147] loss: 0.985
Accuracy of the network on the val images: 62 %
[148] loss: 1.001
Accuracy of the network on the val images: 61 %
[149] loss: 0.988
Accuracy of the network on the val images: 61 %
[150] loss: 0.999
Accuracy of the network on the val images: 60 %
Finished Training
```

1.3. Changes in network

1.3.1. Data Augmentation:



```
train_transform = transforms.Compose([transforms.RandomCrop(32, padding=4),
                                      transforms.RandomHorizontalFlip(),
                                      transforms.ToTensor(),
                                      transforms.Normalize([0.4914, 0.4822, 0.4465],
                                                           [0.2023, 0.1994, 0.2010])])
test_transform = transforms.Compose([transforms.ToTensor(),
                                      transforms.Normalize([0.4914, 0.4822, 0.4465],
                                                           [0.2023, 0.1994, 0.2010])])
```

1.3.2. Network:

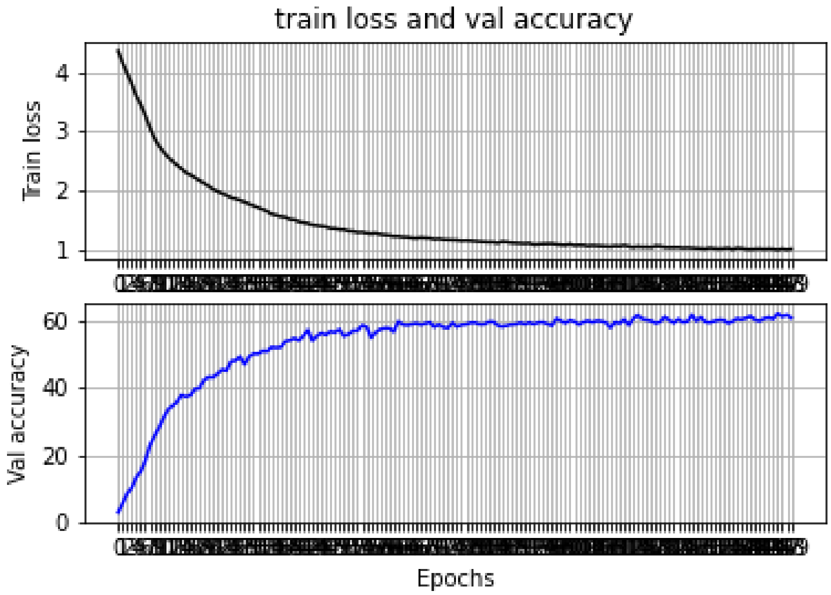
As shown in above network 13 conv layer and 3 fully connected linear layers were used.

After each conv layer and linear layer there was a layer of Batch Normalization except the last linear layer.

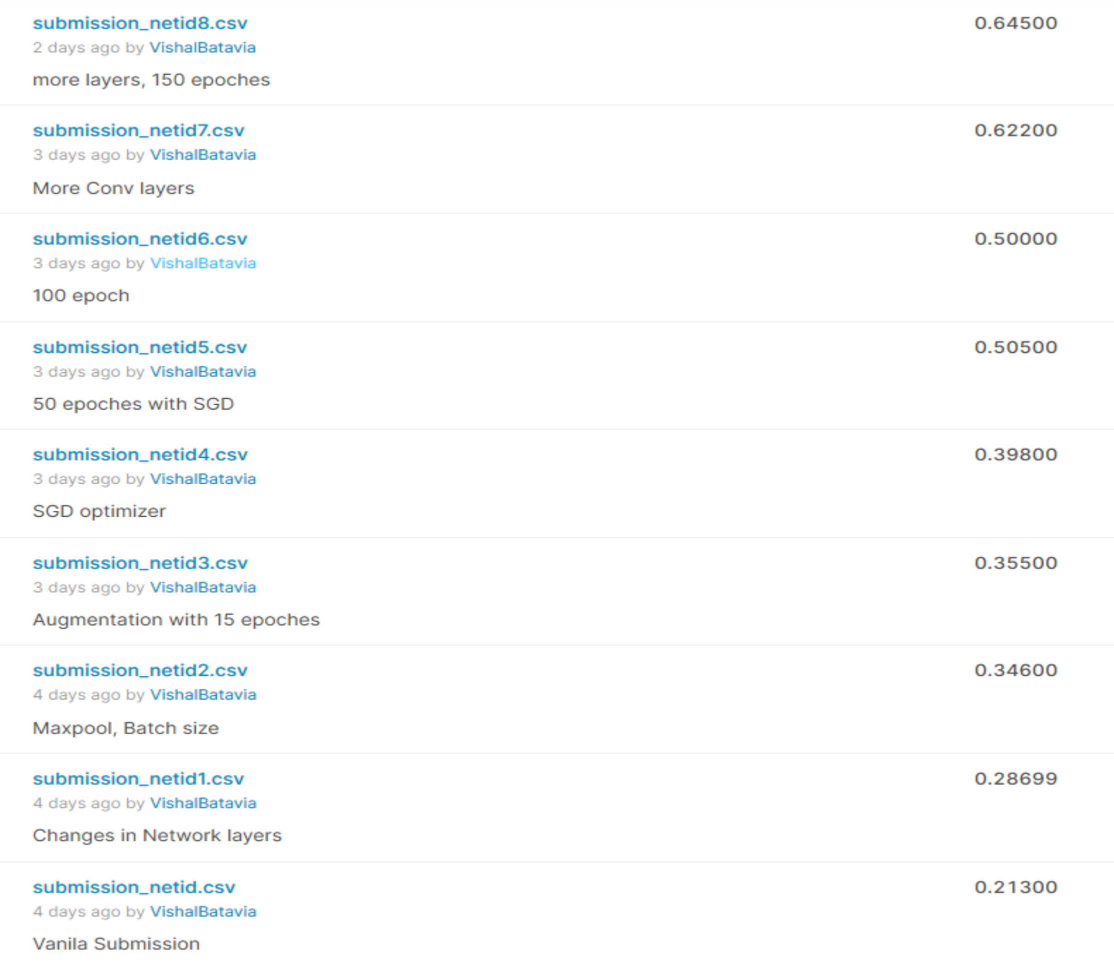
A dropout of 50% was used before first 2 linear layer to avoid overfitting.

Relu was used as activation throughout the network.

Crossentropy loss with SGD optimizer was used with 0.01 learning rate for 150 epoch and train batch of 32.



1.3.3. Ablation Study



submission_netid8.csv 0.64500
2 days ago by VishalBatavia
more layers, 150 epoches

submission_netid7.csv 0.62200
3 days ago by VishalBatavia
More Conv layers

submission_netid6.csv 0.50000
3 days ago by VishalBatavia
100 epoch

submission_netid5.csv 0.50500
3 days ago by VishalBatavia
50 epoches with SGD

submission_netid4.csv 0.39800
3 days ago by VishalBatavia
SGD optimizer

submission_netid3.csv 0.35500
3 days ago by VishalBatavia
Augmentation with 15 epoches

submission_netid2.csv 0.34600
4 days ago by VishalBatavia
Maxpool, Batch size

submission_netid1.csv 0.28699
4 days ago by VishalBatavia
Changes in Network layers

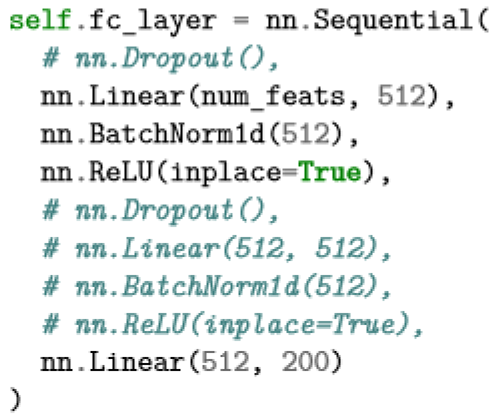
submission_netid.csv 0.21300
4 days ago by VishalBatavia
Vanila Submission

As shown from the above submissions with 7 conv layer, 2 linear layer and 2 dropout with Batch normalization after each conc and linear layer and with Relu activation the max test accuracy obtained was 50%.

In between Vanilla and the network with 50% accuracy as described above, various other options were used like different epochs, learning rate, momentum, Adam optimizer, different percentage of dropout as well as no dropout.

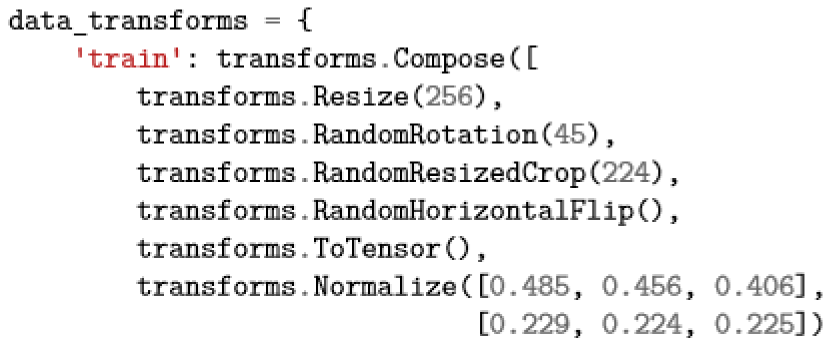
2. Part 2 Transfer learning

- Epoch = 50
- Batch = 32
- LR = 0.001
- Pretrained ResNet18 with FC layer replaced as shown below



```
self.fc_layer = nn.Sequential(
  # nn.Dropout(),
  nn.Linear(num_feats, 512),
  nn.BatchNorm1d(512),
  nn.ReLU(inplace=True),
  # nn.Dropout(),
  # nn.Linear(512, 512),
  # nn.BatchNorm1d(512),
  # nn.ReLU(inplace=True),
  nn.Linear(512, 200)
)
```

- Data augmentation as shown below

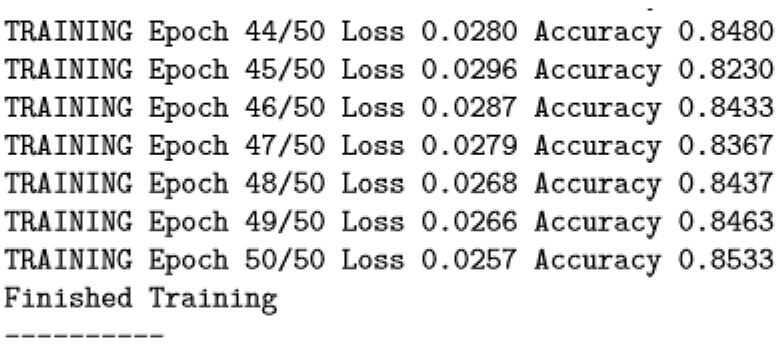


```
data_transforms = {
    'train': transforms.Compose([
        transforms.Resize(256),
        transforms.RandomRotation(45),
        transforms.RandomResizedCrop(224),
        transforms.RandomHorizontalFlip(),
        transforms.ToTensor(),
        transforms.Normalize([0.485, 0.456, 0.406],
                             [0.229, 0.224, 0.225])
```

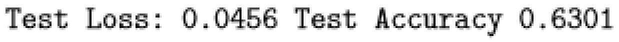
- Optimizer SGD

2.1. ResNet as a fixed feature Extractor.

Training accuracy was 85% while testing accuracy was 63%



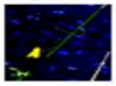
```
TRAINING Epoch 44/50 Loss 0.0280 Accuracy 0.8480
TRAINING Epoch 45/50 Loss 0.0296 Accuracy 0.8230
TRAINING Epoch 46/50 Loss 0.0287 Accuracy 0.8433
TRAINING Epoch 47/50 Loss 0.0279 Accuracy 0.8367
TRAINING Epoch 48/50 Loss 0.0268 Accuracy 0.8437
TRAINING Epoch 49/50 Loss 0.0266 Accuracy 0.8463
TRAINING Epoch 50/50 Loss 0.0257 Accuracy 0.8533
Finished Training
----------
```



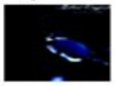
```
Test Loss: 0.0456 Test Accuracy 0.6301
```

- Visualization

class: 047.American_Goldfinch predicted: 047.American_Goldfinch



class: 079.Belted_Kingfisher predicted: 079.Belted_Kingfisher



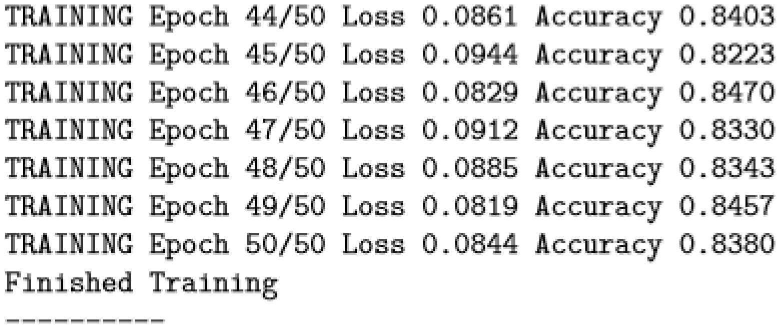
class: 099.Ovenbird predicted: 099.Ovenbird

class: 146.Forsters_Tern predicted: 111.Loggerhead_Shrike

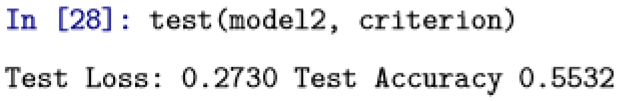
class: 188.Pileated_Woodpecker predicted: 188.Pileated_Woodpecker

2.2. Finetuning the ResNet

Training accuracy was 83% while testing accuracy was 55%



```
TRAINING Epoch 44/50 Loss 0.0861 Accuracy 0.8403
TRAINING Epoch 45/50 Loss 0.0944 Accuracy 0.8223
TRAINING Epoch 46/50 Loss 0.0829 Accuracy 0.8470
TRAINING Epoch 47/50 Loss 0.0912 Accuracy 0.8330
TRAINING Epoch 48/50 Loss 0.0885 Accuracy 0.8343
TRAINING Epoch 49/50 Loss 0.0819 Accuracy 0.8457
TRAINING Epoch 50/50 Loss 0.0844 Accuracy 0.8380
Finished Training
----------
```



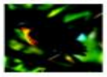
```
In [28]: test(model2, criterion)

Test Loss: 0.2730 Test Accuracy 0.5532
```

- Visualization

class: 133.White_throated_Sparrow predicted: 133.White_throated_Sparrow

class: 020.Yellow_breasted_Chat predicted: 020.Yellow_breasted_Chat



class: 163.Cape_May_Warbler predicted: 183.Northern_Waterthrush

class: 046.Gadwall predicted: 052.Pied_billed_Grebe

class: 161.Blue_winged_Warbler predicted: 161.Blue_winged_Warbler