CS_228_DL_HW_3

May 21, 2023

0.0.1 Imports

[1]:

```
import torch
import torch.nn as nn
import torchvision
import torchvision.transforms as transforms
import torch.optim as optim

import numpy as np
import matplotlib.pyplot as plt
```

0.0.2 Get Dataset

[2]:

```
BATCH_SIZE = 16

# Transform the images from PIL to tensor and normalize them
transform = transforms.Compose([
    transforms.ToTensor(),
])

cifar_10_train = torchvision.datasets.CIFAR10(root='./', train=True,␣
 ↪download=True, transform=transform)
cifar_10_test = torchvision.datasets.CIFAR10(root='./',train=False,␣
 ↪download=True, transform=transform)
```

```
Files already downloaded and verified
Files already downloaded and verified
```

[3]:

```
# Visualize 1 sample
sample_idx_ctr = 0
num_rows = 10
num_cols = 5

fig, ax = plt.subplots(num_rows, num_cols, sharex=True, sharey=True)

for i in range(num_rows):
    for j in range(num_cols):
        img, label = cifar_10_train[sample_idx_ctr]
        ax[i,j].imshow(img.squeeze().permute(1,2,0))
        sample_idx_ctr += 1
```

0.0.3 Sample Dataset

[4]:
```
def get_torch_dataset(dataset):
    images = []
    labels = []
    for img, label in dataset:
        images.append(img.squeeze())
        labels.append(label)
    return torch.stack(images), torch.tensor(labels, dtype = torch.float32)
```

[5]:
```
X_train, y_train = get_torch_dataset(cifar_10_train)
X_test, y_test = get_torch_dataset(cifar_10_test)
```

[6]:
```
def get_sampled_dataset(X, y, n_samples=1000):
    X_sampled = []
    y_sampled = []
    for unique_label in torch.unique(y):

        # get all examples of 1 class.
        class_examples = X[y == unique_label]

        # generate indices and randomly permutate them.
        indices = torch.arange(class_examples.shape[0])
        shuffled_indices = torch.randperm(indices.shape[0])

        # select top k samples and use it as dataset.
        shuffled_indices = shuffled_indices[:n_samples]
        selected_samples = class_examples[shuffled_indices]

        selected_labels = torch.full((n_samples,), unique_label)

        X_sampled.append(selected_samples)
        y_sampled.append(selected_labels)

    return torch.cat(X_sampled, dim=0), torch.cat(y_sampled)
```

[7]:

```
X_train_sampled, y_train_sampled = get_sampled_dataset(X_train, y_train, 1000)
X_test_sampled, y_test_sampled = get_sampled_dataset(X_test, y_test, 1000)
```

[8]:

```
X_train_sampled.shape, y_train_sampled.shape
```

[8]:

```
(torch.Size([10000, 3, 32, 32]), torch.Size([10000]))
```

[9]:

```
X_test_sampled.shape, y_test_sampled.shape
```

[9]:

```
(torch.Size([10000, 3, 32, 32]), torch.Size([10000]))
```

0.0.4 Shuffle Dataset

[10]:

```
# generate indices and randomly permutate them.
indices = torch.arange(X_train_sampled.shape[0])
shuffled_indices = torch.randperm(indices.shape[0])

# select top k samples and use it as dataset.
X_train_sampled = X_train_sampled[shuffled_indices]
y_train_sampled = y_train_sampled[shuffled_indices]

# generate indices and randomly permutate them.
indices = torch.arange(X_test_sampled.shape[0])
shuffled_indices = torch.randperm(indices.shape[0])

# select top k samples and use it as dataset.
X_test_sampled = X_test_sampled[shuffled_indices]
```

```
y_test_sampled = y_test_sampled[shuffled_indices]
```

0.0.5 Normalize Dataset

[11]:

```
def normalize_images(images):
    mean = torch.mean(images, axis=(1, 2, 3), keepdims=True)
    std = torch.std(images, axis=(1, 2, 3), keepdims=True)
    normalized_images = (images - mean) / std
    return normalized_images
```

[12]:

```
x_train = normalize_images(X_train_sampled)
x_test = normalize_images(X_test_sampled)
y_train = y_train_sampled
y_test = y_test_sampled
```

[13]:

```
# function to convert y output vector of size n,1 to onhot vector of size n,10
def convert_to_one_hot(labels):
    unique = torch.unique(labels)
    onehot = torch.zeros((labels.shape[0], unique.shape[0]))
    onehot[torch.arange(labels.shape[0]), labels.int()] = 1.
    return onehot


y_train_oh = convert_to_one_hot(y_train)
y_test_oh = convert_to_one_hot(y_test)
```

0.0.6 Resnet Class and Baseline

[14]:

```
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

# Define the ResNet-18 model
model = torchvision.models.resnet18(pretrained=False)
num_classes = 10

model.fc = nn.Linear(model.fc.in_features, num_classes)
model = model.to(device)

model.eval()
```

```
/usr/local/lib/python3.10/dist-packages/torchvision/models/_utils.py:208:
UserWarning: The parameter 'pretrained' is deprecated since 0.13 and may be
removed in the future, please use 'weights' instead.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/torchvision/models/_utils.py:223:
UserWarning: Arguments other than a weight enum or `None` for 'weights' are
deprecated since 0.13 and may be removed in the future. The current behavior is
```

```
equivalent to passing `weights=None`.
  warnings.warn(msg)
```

```
[14]: ResNet(
        (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3),
      bias=False)
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
        (relu): ReLU(inplace=True)
        (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1,
      ceil_mode=False)
        (layer1): Sequential(
          (0): BasicBlock(
            (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
      bias=False)
            (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
            (relu): ReLU(inplace=True)
            (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
      bias=False)
            (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
          )
          (1): BasicBlock(
            (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
      bias=False)
            (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
            (relu): ReLU(inplace=True)
            (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
      bias=False)
            (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
          )
        )
        (layer2): Sequential(
          (0): BasicBlock(
            (conv1): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1,
      1), bias=False)
            (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
            (relu): ReLU(inplace=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)
            (downsample): Sequential(
              (0): Conv2d(64, 128, kernel_size=(1, 1), stride=(2, 2), bias=False)
              (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
            )
          )
          (1): BasicBlock(
            (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
      1), bias=False)
            (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
            (relu): ReLU(inplace=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)
          )
        )
        (layer3): Sequential(
          (0): BasicBlock(
            (conv1): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1,
      1), bias=False)
            (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
            (relu): ReLU(inplace=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)
            (downsample): Sequential(
              (0): Conv2d(128, 256, kernel_size=(1, 1), stride=(2, 2), bias=False)
              (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
            )
          )
          (1): BasicBlock(
            (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
      1), bias=False)
            (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
            (relu): ReLU(inplace=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)
          )
        )
        (layer4): Sequential(
          (0): BasicBlock(
            (conv1): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1,
      1), bias=False)
            (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
            (relu): ReLU(inplace=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)
            (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): BasicBlock(
            (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
      1), bias=False)
            (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
            (relu): ReLU(inplace=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)
          )
        )
        (avgpool): AdaptiveAvgPool2d(output_size=(1, 1))
        (fc): Linear(in_features=512, out_features=10, bias=True)
      )
```

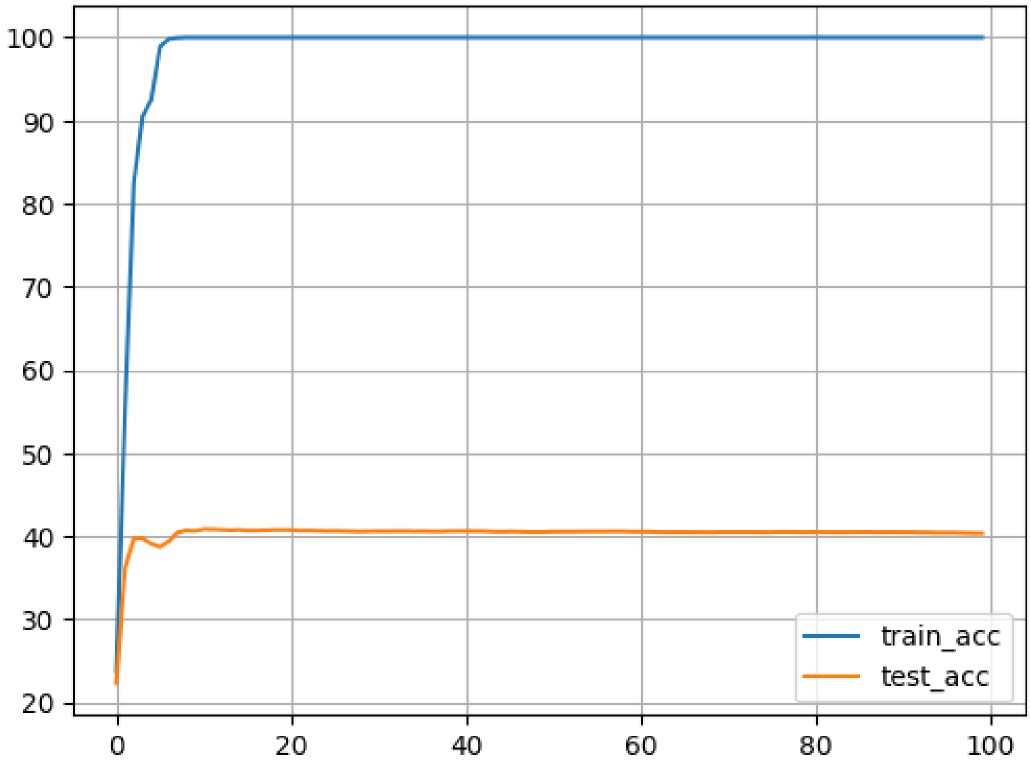
[15]:

```python
# Define loss function and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.0001)

test_acc = []
train_acc = []

# Training loop
num_epochs = 100

for epoch in range(num_epochs):

    i = 0
    iter = 0
    batch_size = 500
    while i < x_train.shape[0]:
        model.train()

        inputs = x_train[i:i+batch_size].to(device)
        labels = y_train[i:i+batch_size].type(torch.LongTensor).to(device)
        labels_oh = y_train_oh[i:i+batch_size].to(device)

        optimizer.zero_grad()

        outputs = model(inputs)
        loss = criterion(outputs, labels_oh)
        loss.backward()
        optimizer.step()

        i += batch_size
        iter += 1

        if (iter + 1) % 5 == 0:
            print('[epoch: %d, iter: %d]-loss: %.3f' % (epoch + 1, iter + 1,loss.item() / batch_size))

    # Evaluation on the test set
    model.eval()

    # Test Accuracy
    correct = 0
    total = 0
    with torch.no_grad():

        inputs = x_test.to(device)
        labels = y_test.type(torch.LongTensor).to(device)
        outputs = model(inputs)
        _, predicted = torch.max(outputs.data, 1)
        total += labels.size(0)
        correct += (predicted == labels).sum().item()

    accuracy = 100 * correct / total
    test_acc.append(accuracy)
    print('Epoch %d: Accuracy on test set: %.2f %%' % (epoch + 1, accuracy))

    # Train Accuracy
    correct = 0
    total = 0
    with torch.no_grad():

        inputs = x_train.to(device)
        labels = y_train.type(torch.LongTensor).to(device)
        outputs = model(inputs)
        _, predicted = torch.max(outputs.data, 1)
        total += labels.size(0)
        correct += (predicted == labels).sum().item()

    accuracy = 100 * correct / total
    train_acc.append(accuracy)
    print('Epoch %d: Accuracy on Train set: %.2f %%' % (epoch + 1, accuracy))

print('Training completed!')
```

```
[epoch: 1, iter: 5]-loss: 0.005
[epoch: 1, iter: 10]-loss: 0.004
[epoch: 1, iter: 15]-loss: 0.004
[epoch: 1, iter: 20]-loss: 0.004
Epoch 1: Accuracy on test set: 22.34 %
Epoch 1: Accuracy on Train set: 23.78 %
[epoch: 2, iter: 5]-loss: 0.003
[epoch: 2, iter: 10]-loss: 0.003
[epoch: 2, iter: 15]-loss: 0.003
[epoch: 2, iter: 20]-loss: 0.003
Epoch 2: Accuracy on test set: 36.08 %
Epoch 2: Accuracy on Train set: 55.31 %
[epoch: 3, iter: 5]-loss: 0.002
[epoch: 3, iter: 10]-loss: 0.002
[epoch: 3, iter: 15]-loss: 0.002
[epoch: 3, iter: 20]-loss: 0.002
Epoch 3: Accuracy on test set: 39.74 %
Epoch 3: Accuracy on Train set: 82.46 %
[epoch: 4, iter: 5]-loss: 0.001
[epoch: 4, iter: 10]-loss: 0.001
[epoch: 4, iter: 15]-loss: 0.001
[epoch: 4, iter: 20]-loss: 0.001
Epoch 4: Accuracy on test set: 39.74 %
Epoch 4: Accuracy on Train set: 90.49 %
[epoch: 5, iter: 5]-loss: 0.001
[epoch: 5, iter: 10]-loss: 0.001
[epoch: 5, iter: 15]-loss: 0.001
[epoch: 5, iter: 20]-loss: 0.001
Epoch 5: Accuracy on test set: 39.08 %
Epoch 5: Accuracy on Train set: 92.49 %
[epoch: 6, iter: 5]-loss: 0.001
[epoch: 6, iter: 10]-loss: 0.001
[epoch: 6, iter: 15]-loss: 0.001
[epoch: 6, iter: 20]-loss: 0.001
Epoch 6: Accuracy on test set: 38.71 %
Epoch 6: Accuracy on Train set: 98.85 %
[epoch: 7, iter: 5]-loss: 0.000
[epoch: 7, iter: 10]-loss: 0.000
[epoch: 7, iter: 15]-loss: 0.000
[epoch: 7, iter: 20]-loss: 0.000
Epoch 7: Accuracy on test set: 39.37 %
Epoch 7: Accuracy on Train set: 99.82 %
[epoch: 8, iter: 5]-loss: 0.000
[epoch: 8, iter: 10]-loss: 0.000
[epoch: 8, iter: 15]-loss: 0.000
[epoch: 8, iter: 20]-loss: 0.000
Epoch 8: Accuracy on test set: 40.42 %
Epoch 8: Accuracy on Train set: 99.95 %
[epoch: 9, iter: 5]-loss: 0.000
[epoch: 9, iter: 10]-loss: 0.000
[epoch: 9, iter: 15]-loss: 0.000
[epoch: 9, iter: 20]-loss: 0.000
Epoch 9: Accuracy on test set: 40.69 %
Epoch 9: Accuracy on Train set: 100.00 %
[epoch: 10, iter: 5]-loss: 0.000
[epoch: 10, iter: 10]-loss: 0.000
[epoch: 10, iter: 15]-loss: 0.000
[epoch: 10, iter: 20]-loss: 0.000
Epoch 10: Accuracy on test set: 40.62 %
Epoch 10: Accuracy on Train set: 100.00 %
[epoch: 11, iter: 5]-loss: 0.000
[epoch: 11, iter: 10]-loss: 0.000
[epoch: 11, iter: 15]-loss: 0.000
[epoch: 11, iter: 20]-loss: 0.000
Epoch 11: Accuracy on test set: 40.83 %
Epoch 11: Accuracy on Train set: 100.00 %
[epoch: 12, iter: 5]-loss: 0.000
[epoch: 12, iter: 10]-loss: 0.000
[epoch: 12, iter: 15]-loss: 0.000
[epoch: 12, iter: 20]-loss: 0.000
Epoch 12: Accuracy on test set: 40.82 %
Epoch 12: Accuracy on Train set: 100.00 %
[epoch: 13, iter: 5]-loss: 0.000
[epoch: 13, iter: 10]-loss: 0.000
[epoch: 13, iter: 15]-loss: 0.000
[epoch: 13, iter: 20]-loss: 0.000
Epoch 13: Accuracy on test set: 40.77 %
Epoch 13: Accuracy on Train set: 100.00 %
[epoch: 14, iter: 5]-loss: 0.000
[epoch: 14, iter: 10]-loss: 0.000
[epoch: 14, iter: 15]-loss: 0.000
[epoch: 14, iter: 20]-loss: 0.000
Epoch 14: Accuracy on test set: 40.71 %
Epoch 14: Accuracy on Train set: 100.00 %
[epoch: 15, iter: 5]-loss: 0.000
[epoch: 15, iter: 10]-loss: 0.000
[epoch: 15, iter: 15]-loss: 0.000
[epoch: 15, iter: 20]-loss: 0.000
Epoch 15: Accuracy on test set: 40.76 %
Epoch 15: Accuracy on Train set: 100.00 %
[epoch: 16, iter: 5]-loss: 0.000
[epoch: 16, iter: 10]-loss: 0.000
[epoch: 16, iter: 15]-loss: 0.000
[epoch: 16, iter: 20]-loss: 0.000
Epoch 16: Accuracy on test set: 40.68 %
Epoch 16: Accuracy on Train set: 100.00 %
[epoch: 17, iter: 5]-loss: 0.000
[epoch: 17, iter: 10]-loss: 0.000
[epoch: 17, iter: 15]-loss: 0.000
[epoch: 17, iter: 20]-loss: 0.000
Epoch 17: Accuracy on test set: 40.67 %
Epoch 17: Accuracy on Train set: 100.00 %
[epoch: 18, iter: 5]-loss: 0.000
[epoch: 18, iter: 10]-loss: 0.000
[epoch: 18, iter: 15]-loss: 0.000
[epoch: 18, iter: 20]-loss: 0.000
Epoch 18: Accuracy on test set: 40.71 %
Epoch 18: Accuracy on Train set: 100.00 %
[epoch: 19, iter: 5]-loss: 0.000
[epoch: 19, iter: 10]-loss: 0.000
[epoch: 19, iter: 15]-loss: 0.000
[epoch: 19, iter: 20]-loss: 0.000
Epoch 19: Accuracy on test set: 40.75 %
Epoch 19: Accuracy on Train set: 100.00 %
[epoch: 20, iter: 5]-loss: 0.000
[epoch: 20, iter: 10]-loss: 0.000
[epoch: 20, iter: 15]-loss: 0.000
[epoch: 20, iter: 20]-loss: 0.000
Epoch 20: Accuracy on test set: 40.76 %
Epoch 20: Accuracy on Train set: 100.00 %
[epoch: 21, iter: 5]-loss: 0.000
[epoch: 21, iter: 10]-loss: 0.000
[epoch: 21, iter: 15]-loss: 0.000
[epoch: 21, iter: 20]-loss: 0.000
Epoch 21: Accuracy on test set: 40.74 %
Epoch 21: Accuracy on Train set: 100.00 %
[epoch: 22, iter: 5]-loss: 0.000
[epoch: 22, iter: 10]-loss: 0.000
[epoch: 22, iter: 15]-loss: 0.000
[epoch: 22, iter: 20]-loss: 0.000
Epoch 22: Accuracy on test set: 40.66 %
Epoch 22: Accuracy on Train set: 100.00 %
[epoch: 23, iter: 5]-loss: 0.000
[epoch: 23, iter: 10]-loss: 0.000
[epoch: 23, iter: 15]-loss: 0.000
[epoch: 23, iter: 20]-loss: 0.000
Epoch 23: Accuracy on test set: 40.69 %
Epoch 23: Accuracy on Train set: 100.00 %
[epoch: 24, iter: 5]-loss: 0.000
[epoch: 24, iter: 10]-loss: 0.000
[epoch: 24, iter: 15]-loss: 0.000
[epoch: 24, iter: 20]-loss: 0.000
Epoch 24: Accuracy on test set: 40.65 %
Epoch 24: Accuracy on Train set: 100.00 %
[epoch: 25, iter: 5]-loss: 0.000
[epoch: 25, iter: 10]-loss: 0.000
[epoch: 25, iter: 15]-loss: 0.000
[epoch: 25, iter: 20]-loss: 0.000
Epoch 25: Accuracy on test set: 40.62 %
Epoch 25: Accuracy on Train set: 100.00 %
[epoch: 26, iter: 5]-loss: 0.000
[epoch: 26, iter: 10]-loss: 0.000
[epoch: 26, iter: 15]-loss: 0.000
[epoch: 26, iter: 20]-loss: 0.000
Epoch 26: Accuracy on test set: 40.63 %
Epoch 26: Accuracy on Train set: 100.00 %
[epoch: 27, iter: 5]-loss: 0.000
[epoch: 27, iter: 10]-loss: 0.000
[epoch: 27, iter: 15]-loss: 0.000
[epoch: 27, iter: 20]-loss: 0.000
Epoch 27: Accuracy on test set: 40.61 %
Epoch 27: Accuracy on Train set: 100.00 %
[epoch: 28, iter: 5]-loss: 0.000
[epoch: 28, iter: 10]-loss: 0.000
[epoch: 28, iter: 15]-loss: 0.000
[epoch: 28, iter: 20]-loss: 0.000
Epoch 28: Accuracy on test set: 40.57 %
Epoch 28: Accuracy on Train set: 100.00 %
[epoch: 29, iter: 5]-loss: 0.000
[epoch: 29, iter: 10]-loss: 0.000
[epoch: 29, iter: 15]-loss: 0.000
[epoch: 29, iter: 20]-loss: 0.000
Epoch 29: Accuracy on test set: 40.55 %
Epoch 29: Accuracy on Train set: 100.00 %
[epoch: 30, iter: 5]-loss: 0.000
[epoch: 30, iter: 10]-loss: 0.000
[epoch: 30, iter: 15]-loss: 0.000
[epoch: 30, iter: 20]-loss: 0.000
Epoch 30: Accuracy on test set: 40.57 %
Epoch 30: Accuracy on Train set: 100.00 %
[epoch: 31, iter: 5]-loss: 0.000
[epoch: 31, iter: 10]-loss: 0.000
[epoch: 31, iter: 15]-loss: 0.000
[epoch: 31, iter: 20]-loss: 0.000
Epoch 31: Accuracy on test set: 40.60 %
Epoch 31: Accuracy on Train set: 100.00 %
[epoch: 32, iter: 5]-loss: 0.000
[epoch: 32, iter: 10]-loss: 0.000
[epoch: 32, iter: 15]-loss: 0.000
[epoch: 32, iter: 20]-loss: 0.000
Epoch 32: Accuracy on test set: 40.58 %
Epoch 32: Accuracy on Train set: 100.00 %
[epoch: 33, iter: 5]-loss: 0.000
[epoch: 33, iter: 10]-loss: 0.000
[epoch: 33, iter: 15]-loss: 0.000
[epoch: 33, iter: 20]-loss: 0.000
Epoch 33: Accuracy on test set: 40.61 %
Epoch 33: Accuracy on Train set: 100.00 %
[epoch: 34, iter: 5]-loss: 0.000
[epoch: 34, iter: 10]-loss: 0.000
[epoch: 34, iter: 15]-loss: 0.000
[epoch: 34, iter: 20]-loss: 0.000
Epoch 34: Accuracy on test set: 40.61 %
Epoch 34: Accuracy on Train set: 100.00 %
[epoch: 35, iter: 5]-loss: 0.000
[epoch: 35, iter: 10]-loss: 0.000
[epoch: 35, iter: 15]-loss: 0.000
[epoch: 35, iter: 20]-loss: 0.000
Epoch 35: Accuracy on test set: 40.58 %
Epoch 35: Accuracy on Train set: 100.00 %
[epoch: 36, iter: 5]-loss: 0.000
[epoch: 36, iter: 10]-loss: 0.000
[epoch: 36, iter: 15]-loss: 0.000
[epoch: 36, iter: 20]-loss: 0.000
Epoch 36: Accuracy on test set: 40.60 %
Epoch 36: Accuracy on Train set: 100.00 %
[epoch: 37, iter: 5]-loss: 0.000
[epoch: 37, iter: 10]-loss: 0.000
[epoch: 37, iter: 15]-loss: 0.000
[epoch: 37, iter: 20]-loss: 0.000
Epoch 37: Accuracy on test set: 40.56 %
Epoch 37: Accuracy on Train set: 100.00 %
[epoch: 38, iter: 5]-loss: 0.000
[epoch: 38, iter: 10]-loss: 0.000
[epoch: 38, iter: 15]-loss: 0.000
[epoch: 38, iter: 20]-loss: 0.000
Epoch 38: Accuracy on test set: 40.56 %
Epoch 38: Accuracy on Train set: 100.00 %
[epoch: 39, iter: 5]-loss: 0.000
[epoch: 39, iter: 10]-loss: 0.000
[epoch: 39, iter: 15]-loss: 0.000
[epoch: 39, iter: 20]-loss: 0.000
Epoch 39: Accuracy on test set: 40.61 %
Epoch 39: Accuracy on Train set: 100.00 %
[epoch: 40, iter: 5]-loss: 0.000
[epoch: 40, iter: 10]-loss: 0.000
[epoch: 40, iter: 15]-loss: 0.000
[epoch: 40, iter: 20]-loss: 0.000
Epoch 40: Accuracy on test set: 40.61 %
Epoch 40: Accuracy on Train set: 100.00 %
[epoch: 41, iter: 5]-loss: 0.000
[epoch: 41, iter: 10]-loss: 0.000
[epoch: 41, iter: 15]-loss: 0.000
[epoch: 41, iter: 20]-loss: 0.000
Epoch 41: Accuracy on test set: 40.62 %
Epoch 41: Accuracy on Train set: 100.00 %
[epoch: 42, iter: 5]-loss: 0.000
[epoch: 42, iter: 10]-loss: 0.000
[epoch: 42, iter: 15]-loss: 0.000
[epoch: 42, iter: 20]-loss: 0.000
Epoch 42: Accuracy on test set: 40.61 %
Epoch 42: Accuracy on Train set: 100.00 %
[epoch: 43, iter: 5]-loss: 0.000
[epoch: 43, iter: 10]-loss: 0.000
[epoch: 43, iter: 15]-loss: 0.000
[epoch: 43, iter: 20]-loss: 0.000
Epoch 43: Accuracy on test set: 40.60 %
Epoch 43: Accuracy on Train set: 100.00 %
[epoch: 44, iter: 5]-loss: 0.000
[epoch: 44, iter: 10]-loss: 0.000
[epoch: 44, iter: 15]-loss: 0.000
[epoch: 44, iter: 20]-loss: 0.000
Epoch 44: Accuracy on test set: 40.53 %
Epoch 44: Accuracy on Train set: 100.00 %
[epoch: 45, iter: 5]-loss: 0.000
[epoch: 45, iter: 10]-loss: 0.000
[epoch: 45, iter: 15]-loss: 0.000
[epoch: 45, iter: 20]-loss: 0.000
Epoch 45: Accuracy on test set: 40.51 %
Epoch 45: Accuracy on Train set: 100.00 %
[epoch: 46, iter: 5]-loss: 0.000
[epoch: 46, iter: 10]-loss: 0.000
[epoch: 46, iter: 15]-loss: 0.000
[epoch: 46, iter: 20]-loss: 0.000
Epoch 46: Accuracy on test set: 40.54 %
Epoch 46: Accuracy on Train set: 100.00 %
[epoch: 47, iter: 5]-loss: 0.000
[epoch: 47, iter: 10]-loss: 0.000
[epoch: 47, iter: 15]-loss: 0.000
[epoch: 47, iter: 20]-loss: 0.000
Epoch 47: Accuracy on test set: 40.53 %
Epoch 47: Accuracy on Train set: 100.00 %
[epoch: 48, iter: 5]-loss: 0.000
[epoch: 48, iter: 10]-loss: 0.000
[epoch: 48, iter: 15]-loss: 0.000
[epoch: 48, iter: 20]-loss: 0.000
Epoch 48: Accuracy on test set: 40.48 %
Epoch 48: Accuracy on Train set: 100.00 %
[epoch: 49, iter: 5]-loss: 0.000
[epoch: 49, iter: 10]-loss: 0.000
[epoch: 49, iter: 15]-loss: 0.000
[epoch: 49, iter: 20]-loss: 0.000
Epoch 49: Accuracy on test set: 40.47 %
Epoch 49: Accuracy on Train set: 100.00 %
[epoch: 50, iter: 5]-loss: 0.000
[epoch: 50, iter: 10]-loss: 0.000
[epoch: 50, iter: 15]-loss: 0.000
[epoch: 50, iter: 20]-loss: 0.000
Epoch 50: Accuracy on test set: 40.48 %
Epoch 50: Accuracy on Train set: 100.00 %
[epoch: 51, iter: 5]-loss: 0.000
[epoch: 51, iter: 10]-loss: 0.000
[epoch: 51, iter: 15]-loss: 0.000
[epoch: 51, iter: 20]-loss: 0.000
Epoch 51: Accuracy on test set: 40.53 %
Epoch 51: Accuracy on Train set: 100.00 %
[epoch: 52, iter: 5]-loss: 0.000
[epoch: 52, iter: 10]-loss: 0.000
[epoch: 52, iter: 15]-loss: 0.000
[epoch: 52, iter: 20]-loss: 0.000
Epoch 52: Accuracy on test set: 40.53 %
Epoch 52: Accuracy on Train set: 100.00 %
[epoch: 53, iter: 5]-loss: 0.000
[epoch: 53, iter: 10]-loss: 0.000
[epoch: 53, iter: 15]-loss: 0.000
[epoch: 53, iter: 20]-loss: 0.000
Epoch 53: Accuracy on test set: 40.53 %
Epoch 53: Accuracy on Train set: 100.00 %
[epoch: 54, iter: 5]-loss: 0.000
[epoch: 54, iter: 10]-loss: 0.000
[epoch: 54, iter: 15]-loss: 0.000
[epoch: 54, iter: 20]-loss: 0.000
Epoch 54: Accuracy on test set: 40.54 %
Epoch 54: Accuracy on Train set: 100.00 %
[epoch: 55, iter: 5]-loss: 0.000
[epoch: 55, iter: 10]-loss: 0.000
[epoch: 55, iter: 15]-loss: 0.000
[epoch: 55, iter: 20]-loss: 0.000
Epoch 55: Accuracy on test set: 40.55 %
Epoch 55: Accuracy on Train set: 100.00 %
[epoch: 56, iter: 5]-loss: 0.000
[epoch: 56, iter: 10]-loss: 0.000
[epoch: 56, iter: 15]-loss: 0.000
[epoch: 56, iter: 20]-loss: 0.000
Epoch 56: Accuracy on test set: 40.55 %
Epoch 56: Accuracy on Train set: 100.00 %
[epoch: 57, iter: 5]-loss: 0.000
[epoch: 57, iter: 10]-loss: 0.000
[epoch: 57, iter: 15]-loss: 0.000
[epoch: 57, iter: 20]-loss: 0.000
Epoch 57: Accuracy on test set: 40.55 %
Epoch 57: Accuracy on Train set: 100.00 %
[epoch: 58, iter: 5]-loss: 0.000
[epoch: 58, iter: 10]-loss: 0.000
[epoch: 58, iter: 15]-loss: 0.000
[epoch: 58, iter: 20]-loss: 0.000
Epoch 58: Accuracy on test set: 40.57 %
Epoch 58: Accuracy on Train set: 100.00 %
[epoch: 59, iter: 5]-loss: 0.000
[epoch: 59, iter: 10]-loss: 0.000
[epoch: 59, iter: 15]-loss: 0.000
[epoch: 59, iter: 20]-loss: 0.000
Epoch 59: Accuracy on test set: 40.56 %
Epoch 59: Accuracy on Train set: 100.00 %
[epoch: 60, iter: 5]-loss: 0.000
[epoch: 60, iter: 10]-loss: 0.000
[epoch: 60, iter: 15]-loss: 0.000
[epoch: 60, iter: 20]-loss: 0.000
Epoch 60: Accuracy on test set: 40.52 %
Epoch 60: Accuracy on Train set: 100.00 %
[epoch: 61, iter: 5]-loss: 0.000
[epoch: 61, iter: 10]-loss: 0.000
[epoch: 61, iter: 15]-loss: 0.000
[epoch: 61, iter: 20]-loss: 0.000
Epoch 61: Accuracy on test set: 40.52 %
Epoch 61: Accuracy on Train set: 100.00 %
[epoch: 62, iter: 5]-loss: 0.000
[epoch: 62, iter: 10]-loss: 0.000
[epoch: 62, iter: 15]-loss: 0.000
[epoch: 62, iter: 20]-loss: 0.000
Epoch 62: Accuracy on test set: 40.51 %
Epoch 62: Accuracy on Train set: 100.00 %
[epoch: 63, iter: 5]-loss: 0.000
[epoch: 63, iter: 10]-loss: 0.000
[epoch: 63, iter: 15]-loss: 0.000
[epoch: 63, iter: 20]-loss: 0.000
Epoch 63: Accuracy on test set: 40.47 %
Epoch 63: Accuracy on Train set: 100.00 %
[epoch: 64, iter: 5]-loss: 0.000
[epoch: 64, iter: 10]-loss: 0.000
[epoch: 64, iter: 15]-loss: 0.000
[epoch: 64, iter: 20]-loss: 0.000
Epoch 64: Accuracy on test set: 40.45 %
Epoch 64: Accuracy on Train set: 100.00 %
[epoch: 65, iter: 5]-loss: 0.000
[epoch: 65, iter: 10]-loss: 0.000
[epoch: 65, iter: 15]-loss: 0.000
[epoch: 65, iter: 20]-loss: 0.000
Epoch 65: Accuracy on test set: 40.46 %
Epoch 65: Accuracy on Train set: 100.00 %
[epoch: 66, iter: 5]-loss: 0.000
[epoch: 66, iter: 10]-loss: 0.000
[epoch: 66, iter: 15]-loss: 0.000
[epoch: 66, iter: 20]-loss: 0.000
Epoch 66: Accuracy on test set: 40.46 %
Epoch 66: Accuracy on Train set: 100.00 %
[epoch: 67, iter: 5]-loss: 0.000
[epoch: 67, iter: 10]-loss: 0.000
[epoch: 67, iter: 15]-loss: 0.000
[epoch: 67, iter: 20]-loss: 0.000
Epoch 67: Accuracy on test set: 40.45 %
Epoch 67: Accuracy on Train set: 100.00 %
[epoch: 68, iter: 5]-loss: 0.000
[epoch: 68, iter: 10]-loss: 0.000
[epoch: 68, iter: 15]-loss: 0.000
[epoch: 68, iter: 20]-loss: 0.000
Epoch 68: Accuracy on test set: 40.45 %
Epoch 68: Accuracy on Train set: 100.00 %
[epoch: 69, iter: 5]-loss: 0.000
[epoch: 69, iter: 10]-loss: 0.000
[epoch: 69, iter: 15]-loss: 0.000
[epoch: 69, iter: 20]-loss: 0.000
Epoch 69: Accuracy on test set: 40.44 %
Epoch 69: Accuracy on Train set: 100.00 %
[epoch: 70, iter: 5]-loss: 0.000
[epoch: 70, iter: 10]-loss: 0.000
[epoch: 70, iter: 15]-loss: 0.000
[epoch: 70, iter: 20]-loss: 0.000
Epoch 70: Accuracy on test set: 40.46 %
Epoch 70: Accuracy on Train set: 100.00 %
[epoch: 71, iter: 5]-loss: 0.000
[epoch: 71, iter: 10]-loss: 0.000
[epoch: 71, iter: 15]-loss: 0.000
[epoch: 71, iter: 20]-loss: 0.000
Epoch 71: Accuracy on test set: 40.47 %
Epoch 71: Accuracy on Train set: 100.00 %
[epoch: 72, iter: 5]-loss: 0.000
[epoch: 72, iter: 10]-loss: 0.000
[epoch: 72, iter: 15]-loss: 0.000
[epoch: 72, iter: 20]-loss: 0.000
Epoch 72: Accuracy on test set: 40.47 %
Epoch 72: Accuracy on Train set: 100.00 %
[epoch: 73, iter: 5]-loss: 0.000
[epoch: 73, iter: 10]-loss: 0.000
[epoch: 73, iter: 15]-loss: 0.000
[epoch: 73, iter: 20]-loss: 0.000
Epoch 73: Accuracy on test set: 40.47 %
Epoch 73: Accuracy on Train set: 100.00 %
[epoch: 74, iter: 5]-loss: 0.000
[epoch: 74, iter: 10]-loss: 0.000
[epoch: 74, iter: 15]-loss: 0.000
[epoch: 74, iter: 20]-loss: 0.000
Epoch 74: Accuracy on test set: 40.46 %
Epoch 74: Accuracy on Train set: 100.00 %
[epoch: 75, iter: 5]-loss: 0.000
[epoch: 75, iter: 10]-loss: 0.000
[epoch: 75, iter: 15]-loss: 0.000
[epoch: 75, iter: 20]-loss: 0.000
Epoch 75: Accuracy on test set: 40.45 %
Epoch 75: Accuracy on Train set: 100.00 %
[epoch: 76, iter: 5]-loss: 0.000
[epoch: 76, iter: 10]-loss: 0.000
[epoch: 76, iter: 15]-loss: 0.000
[epoch: 76, iter: 20]-loss: 0.000
Epoch 76: Accuracy on test set: 40.45 %
Epoch 76: Accuracy on Train set: 100.00 %
[epoch: 77, iter: 5]-loss: 0.000
[epoch: 77, iter: 10]-loss: 0.000
[epoch: 77, iter: 15]-loss: 0.000
[epoch: 77, iter: 20]-loss: 0.000
Epoch 77: Accuracy on test set: 40.49 %
Epoch 77: Accuracy on Train set: 100.00 %
[epoch: 78, iter: 5]-loss: 0.000
[epoch: 78, iter: 10]-loss: 0.000
[epoch: 78, iter: 15]-loss: 0.000
[epoch: 78, iter: 20]-loss: 0.000
Epoch 78: Accuracy on test set: 40.48 %
Epoch 78: Accuracy on Train set: 100.00 %
[epoch: 79, iter: 5]-loss: 0.000
[epoch: 79, iter: 10]-loss: 0.000
[epoch: 79, iter: 15]-loss: 0.000
[epoch: 79, iter: 20]-loss: 0.000
Epoch 79: Accuracy on test set: 40.46 %
Epoch 79: Accuracy on Train set: 100.00 %
[epoch: 80, iter: 5]-loss: 0.000
[epoch: 80, iter: 10]-loss: 0.000
[epoch: 80, iter: 15]-loss: 0.000
[epoch: 80, iter: 20]-loss: 0.000
Epoch 80: Accuracy on test set: 40.46 %
Epoch 80: Accuracy on Train set: 100.00 %
[epoch: 81, iter: 5]-loss: 0.000
[epoch: 81, iter: 10]-loss: 0.000
[epoch: 81, iter: 15]-loss: 0.000
[epoch: 81, iter: 20]-loss: 0.000
Epoch 81: Accuracy on test set: 40.47 %
Epoch 81: Accuracy on Train set: 100.00 %
[epoch: 82, iter: 5]-loss: 0.000
[epoch: 82, iter: 10]-loss: 0.000
[epoch: 82, iter: 15]-loss: 0.000
[epoch: 82, iter: 20]-loss: 0.000
Epoch 82: Accuracy on test set: 40.46 %
Epoch 82: Accuracy on Train set: 100.00 %
[epoch: 83, iter: 5]-loss: 0.000
[epoch: 83, iter: 10]-loss: 0.000
[epoch: 83, iter: 15]-loss: 0.000
[epoch: 83, iter: 20]-loss: 0.000
Epoch 83: Accuracy on test set: 40.46 %
Epoch 83: Accuracy on Train set: 100.00 %
[epoch: 84, iter: 5]-loss: 0.000
[epoch: 84, iter: 10]-loss: 0.000
[epoch: 84, iter: 15]-loss: 0.000
[epoch: 84, iter: 20]-loss: 0.000
Epoch 84: Accuracy on test set: 40.45 %
Epoch 84: Accuracy on Train set: 100.00 %
[epoch: 85, iter: 5]-loss: 0.000
[epoch: 85, iter: 10]-loss: 0.000
[epoch: 85, iter: 15]-loss: 0.000
[epoch: 85, iter: 20]-loss: 0.000
Epoch 85: Accuracy on test set: 40.46 %
Epoch 85: Accuracy on Train set: 100.00 %
[epoch: 86, iter: 5]-loss: 0.000
[epoch: 86, iter: 10]-loss: 0.000
[epoch: 86, iter: 15]-loss: 0.000
[epoch: 86, iter: 20]-loss: 0.000
Epoch 86: Accuracy on test set: 40.47 %
Epoch 86: Accuracy on Train set: 100.00 %
[epoch: 87, iter: 5]-loss: 0.000
[epoch: 87, iter: 10]-loss: 0.000
[epoch: 87, iter: 15]-loss: 0.000
[epoch: 87, iter: 20]-loss: 0.000
Epoch 87: Accuracy on test set: 40.47 %
Epoch 87: Accuracy on Train set: 100.00 %
[epoch: 88, iter: 5]-loss: 0.000
[epoch: 88, iter: 10]-loss: 0.000
[epoch: 88, iter: 15]-loss: 0.000
[epoch: 88, iter: 20]-loss: 0.000
Epoch 88: Accuracy on test set: 40.45 %
Epoch 88: Accuracy on Train set: 100.00 %
[epoch: 89, iter: 5]-loss: 0.000
[epoch: 89, iter: 10]-loss: 0.000
[epoch: 89, iter: 15]-loss: 0.000
[epoch: 89, iter: 20]-loss: 0.000
Epoch 89: Accuracy on test set: 40.46 %
Epoch 89: Accuracy on Train set: 100.00 %
[epoch: 90, iter: 5]-loss: 0.000
[epoch: 90, iter: 10]-loss: 0.000
[epoch: 90, iter: 15]-loss: 0.000
[epoch: 90, iter: 20]-loss: 0.000
Epoch 90: Accuracy on test set: 40.46 %
Epoch 90: Accuracy on Train set: 100.00 %
[epoch: 91, iter: 5]-loss: 0.000
[epoch: 91, iter: 10]-loss: 0.000
[epoch: 91, iter: 15]-loss: 0.000
[epoch: 91, iter: 20]-loss: 0.000
Epoch 91: Accuracy on test set: 40.45 %
Epoch 91: Accuracy on Train set: 100.00 %
[epoch: 92, iter: 5]-loss: 0.000
[epoch: 92, iter: 10]-loss: 0.000
[epoch: 92, iter: 15]-loss: 0.000
[epoch: 92, iter: 20]-loss: 0.000
Epoch 92: Accuracy on test set: 40.45 %
Epoch 92: Accuracy on Train set: 100.00 %
[epoch: 93, iter: 5]-loss: 0.000
[epoch: 93, iter: 10]-loss: 0.000
[epoch: 93, iter: 15]-loss: 0.000
[epoch: 93, iter: 20]-loss: 0.000
Epoch 93: Accuracy on test set: 40.43 %
Epoch 93: Accuracy on Train set: 100.00 %
[epoch: 94, iter: 5]-loss: 0.000
[epoch: 94, iter: 10]-loss: 0.000
[epoch: 94, iter: 15]-loss: 0.000
[epoch: 94, iter: 20]-loss: 0.000
Epoch 94: Accuracy on test set: 40.42 %
Epoch 94: Accuracy on Train set: 100.00 %
[epoch: 95, iter: 5]-loss: 0.000
[epoch: 95, iter: 10]-loss: 0.000
[epoch: 95, iter: 15]-loss: 0.000
[epoch: 95, iter: 20]-loss: 0.000
Epoch 95: Accuracy on test set: 40.39 %
Epoch 95: Accuracy on Train set: 100.00 %
[epoch: 96, iter: 5]-loss: 0.000
[epoch: 96, iter: 10]-loss: 0.000
[epoch: 96, iter: 15]-loss: 0.000
[epoch: 96, iter: 20]-loss: 0.000
Epoch 96: Accuracy on test set: 40.40 %
Epoch 96: Accuracy on Train set: 100.00 %
[epoch: 97, iter: 5]-loss: 0.000
[epoch: 97, iter: 10]-loss: 0.000
[epoch: 97, iter: 15]-loss: 0.000
[epoch: 97, iter: 20]-loss: 0.000
Epoch 97: Accuracy on test set: 40.39 %
Epoch 97: Accuracy on Train set: 100.00 %
[epoch: 98, iter: 5]-loss: 0.000
[epoch: 98, iter: 10]-loss: 0.000
[epoch: 98, iter: 15]-loss: 0.000
[epoch: 98, iter: 20]-loss: 0.000
Epoch 98: Accuracy on test set: 40.36 %
Epoch 98: Accuracy on Train set: 100.00 %
[epoch: 99, iter: 5]-loss: 0.000
[epoch: 99, iter: 10]-loss: 0.000
[epoch: 99, iter: 15]-loss: 0.000
[epoch: 99, iter: 20]-loss: 0.000
Epoch 99: Accuracy on test set: 40.33 %
Epoch 99: Accuracy on Train set: 100.00 %
[epoch: 100, iter: 5]-loss: 0.000
[epoch: 100, iter: 10]-loss: 0.000
[epoch: 100, iter: 15]-loss: 0.000
[epoch: 100, iter: 20]-loss: 0.000
Epoch 100: Accuracy on test set: 40.33 %
Epoch 100: Accuracy on Train set: 100.00 %
Training completed!
```

[16]:

```
plt.figure()
plt.plot(torch.arange(len(train_acc)), train_acc, label='train_acc')
plt.plot(torch.arange(len(test_acc)), test_acc, label='test_acc')
plt.grid()
plt.legend()
plt.show()
```



We can achieve upto 100% accuracy on Train set and 40% accuracy on Test set

0.0.7 Mixup Augmentation

[17]:
```python
def mixup_data(inputs, labels, alpha):

    inputs = inputs.clone()
    labels = labels.clone()

    batch_size = inputs.shape[0]

    beta_distribution = np.random.beta(alpha,alpha,batch_size)
    lam = torch.Tensor(beta_distribution)

    # Permute the inputs and labels
    index = torch.randperm(batch_size)
    mixed_inputs = lam.view(batch_size, 1, 1, 1) * inputs + (1 - lam).
 ↪view(batch_size, 1, 1, 1) * inputs[index, :]
    mixed_labels = lam.view(batch_size, 1) * labels + (1 - lam).
 ↪view(batch_size, 1) * labels[index]

    return mixed_inputs, mixed_labels
```

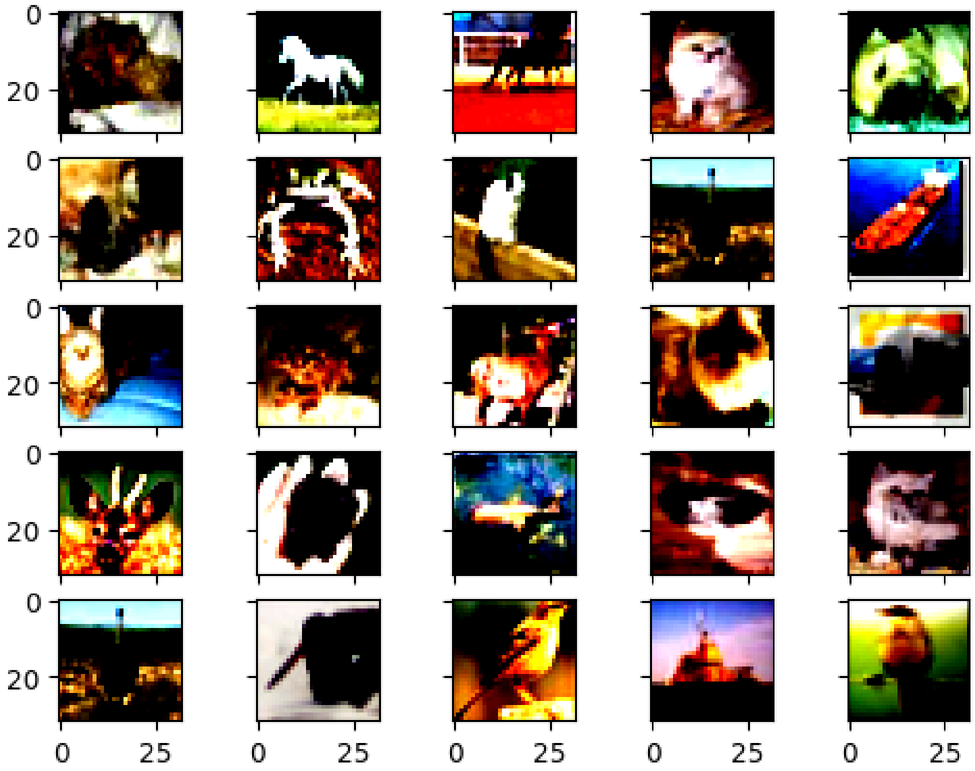
```
mixed_up_images, mixed_up_labels = mixup_data(x_train[:100], y_train_oh[:100],␣
↪0.2)
```

[18]:

```
# Visualize 1 sample
sample_idx_ctr = 0
num_rows = 5
num_cols = 5

fig, ax = plt.subplots(num_rows, num_cols, sharex=True, sharey=True)

for i in range(num_rows):
    for j in range(num_cols):

        img = mixed_up_images[sample_idx_ctr]
        ax[i,j].imshow(img.squeeze().permute(1,2,0))
        sample_idx_ctr += 1
```

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

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

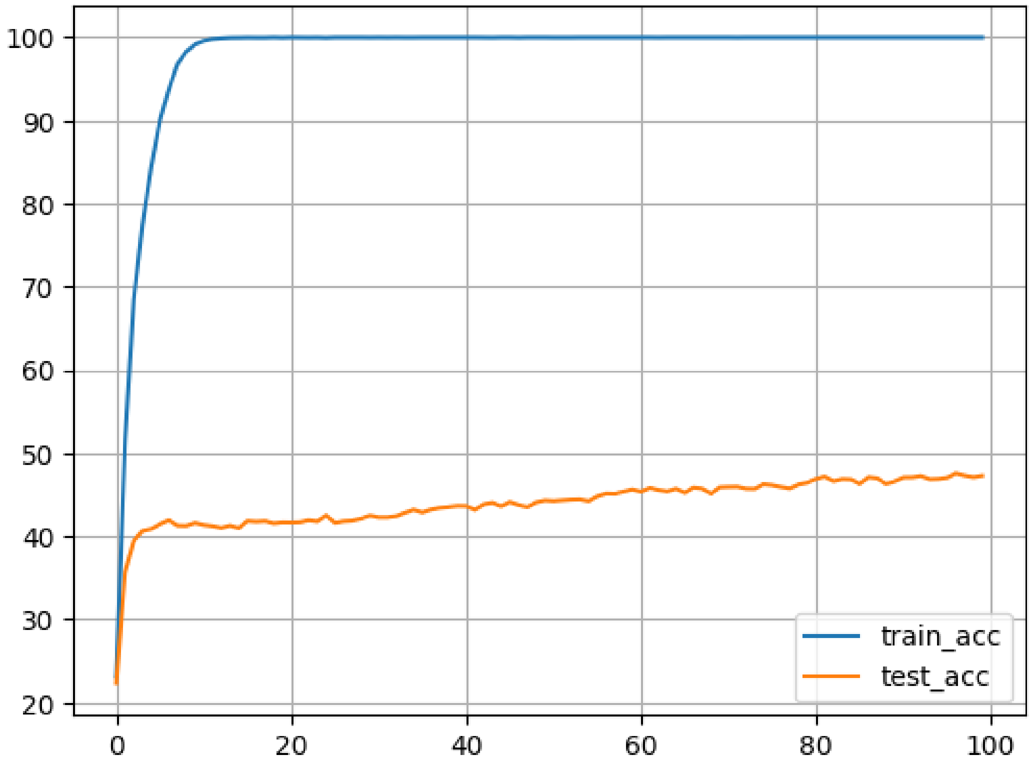


[19]:

```
# Define loss function and optimizer


for alpha in [0.2, 0.4]:
    # Training loop
    num_epochs = 100

    device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

    # Define the ResNet-18 model
    model = torchvision.models.resnet18(pretrained=False)
    num_classes = 10

    model.fc = nn.Linear(model.fc.in_features, num_classes)
    model = model.to(device)

    model.eval()

    criterion = nn.CrossEntropyLoss()
    optimizer = optim.Adam(model.parameters(), lr=0.0001)

    test_acc = []
    train_acc = []

    for epoch in range(num_epochs):

        i = 0
        iter = 0
        batch_size = 500

        while i < x_train.shape[0]:
            model.train()

            inputs = x_train[i:i+batch_size]
            labels = y_train[i:i+batch_size].type(torch.LongTensor)
            labels_oh = y_train_oh[i:i+batch_size]

            inputs, labels_oh = mixup_data(inputs, labels_oh, alpha)
            inputs = inputs.to(device)
            labels = labels.to(device)
            labels_oh = labels_oh.to(device)

            optimizer.zero_grad()

            outputs = model(inputs)
            loss = criterion(outputs, labels_oh)
            loss.backward()
            optimizer.step()

            i += batch_size
            iter += 1

        if epoch % 5 == 0:
            print('[epoch: %d]-loss: %.3f' % (epoch + 1,loss.item() /␣
↪batch_size))

        # Evaluation on the test set
        model.eval()

        # Test Accuracy
        correct = 0
        total = 0
        with torch.no_grad():

            inputs = x_test.to(device)
            labels = y_test.type(torch.LongTensor).to(device)
            outputs = model(inputs)
            _, predicted = torch.max(outputs.data, 1)
            total += labels.size(0)
            correct += (predicted == labels).sum().item()

        accuracy = 100 * correct / total
        test_acc.append(accuracy)
        print('Epoch %d: Accuracy on test set: %.2f %%' % (epoch + 1, accuracy))

        # Train Accuracy
        correct = 0
        total = 0
        with torch.no_grad():

            inputs = x_train.to(device)
            labels = y_train.type(torch.LongTensor).to(device)
            outputs = model(inputs)
            _, predicted = torch.max(outputs.data, 1)
            total += labels.size(0)
            correct += (predicted == labels).sum().item()

        accuracy = 100 * correct / total
        train_acc.append(accuracy)
        print('Epoch %d: Accuracy on Train set: %.2f %%' % (epoch + 1,␣
↪accuracy))

    print('Training completed!')
    plt.figure()
    plt.plot(torch.arange(len(train_acc)), train_acc, label='train_acc')
    plt.plot(torch.arange(len(test_acc)), test_acc, label='test_acc')
    plt.grid()
    plt.legend()
    plt.show()
```

```
[epoch: 1]-loss: 0.004
Epoch 1: Accuracy on test set: 22.37 %
Epoch 1: Accuracy on Train set: 23.12 %
Epoch 2: Accuracy on test set: 35.66 %
Epoch 2: Accuracy on Train set: 51.43 %
Epoch 3: Accuracy on test set: 39.49 %
Epoch 3: Accuracy on Train set: 68.52 %
Epoch 4: Accuracy on test set: 40.65 %
Epoch 4: Accuracy on Train set: 77.51 %
Epoch 5: Accuracy on test set: 40.87 %
Epoch 5: Accuracy on Train set: 84.42 %
[epoch: 6]-loss: 0.002
Epoch 6: Accuracy on test set: 41.46 %
Epoch 6: Accuracy on Train set: 90.06 %
Epoch 7: Accuracy on test set: 41.93 %
Epoch 7: Accuracy on Train set: 93.73 %
Epoch 8: Accuracy on test set: 41.28 %
Epoch 8: Accuracy on Train set: 96.83 %
Epoch 9: Accuracy on test set: 41.25 %
Epoch 9: Accuracy on Train set: 98.27 %
Epoch 10: Accuracy on test set: 41.61 %
Epoch 10: Accuracy on Train set: 99.15 %
[epoch: 11]-loss: 0.001
Epoch 11: Accuracy on test set: 41.35 %
Epoch 11: Accuracy on Train set: 99.59 %
Epoch 12: Accuracy on test set: 41.22 %
Epoch 12: Accuracy on Train set: 99.80 %
Epoch 13: Accuracy on test set: 41.00 %
Epoch 13: Accuracy on Train set: 99.89 %
Epoch 14: Accuracy on test set: 41.28 %
Epoch 14: Accuracy on Train set: 99.95 %
Epoch 15: Accuracy on test set: 40.98 %
Epoch 15: Accuracy on Train set: 99.96 %
[epoch: 16]-loss: 0.001
Epoch 16: Accuracy on test set: 41.84 %
Epoch 16: Accuracy on Train set: 99.98 %
Epoch 17: Accuracy on test set: 41.76 %
Epoch 17: Accuracy on Train set: 99.97 %
Epoch 18: Accuracy on test set: 41.83 %
Epoch 18: Accuracy on Train set: 99.97 %
Epoch 19: Accuracy on test set: 41.56 %
Epoch 19: Accuracy on Train set: 100.00 %
Epoch 20: Accuracy on test set: 41.67 %
Epoch 20: Accuracy on Train set: 99.97 %
[epoch: 21]-loss: 0.001
Epoch 21: Accuracy on test set: 41.65 %
Epoch 21: Accuracy on Train set: 100.00 %
Epoch 22: Accuracy on test set: 41.69 %
Epoch 22: Accuracy on Train set: 99.99 %
Epoch 23: Accuracy on test set: 41.93 %
Epoch 23: Accuracy on Train set: 99.98 %
Epoch 24: Accuracy on test set: 41.79 %
Epoch 24: Accuracy on Train set: 99.99 %
Epoch 25: Accuracy on test set: 42.53 %
Epoch 25: Accuracy on Train set: 99.96 %
[epoch: 26]-loss: 0.001
Epoch 26: Accuracy on test set: 41.61 %
Epoch 26: Accuracy on Train set: 100.00 %
Epoch 27: Accuracy on test set: 41.81 %
Epoch 27: Accuracy on Train set: 100.00 %
Epoch 28: Accuracy on test set: 41.88 %
Epoch 28: Accuracy on Train set: 100.00 %
Epoch 29: Accuracy on test set: 42.13 %
Epoch 29: Accuracy on Train set: 100.00 %
Epoch 30: Accuracy on test set: 42.49 %
Epoch 30: Accuracy on Train set: 99.99 %
[epoch: 31]-loss: 0.001
Epoch 31: Accuracy on test set: 42.28 %
Epoch 31: Accuracy on Train set: 100.00 %
Epoch 32: Accuracy on test set: 42.28 %
Epoch 32: Accuracy on Train set: 99.99 %
Epoch 33: Accuracy on test set: 42.42 %
Epoch 33: Accuracy on Train set: 100.00 %
Epoch 34: Accuracy on test set: 42.83 %
Epoch 34: Accuracy on Train set: 99.99 %
Epoch 35: Accuracy on test set: 43.20 %
Epoch 35: Accuracy on Train set: 99.99 %
[epoch: 36]-loss: 0.001
Epoch 36: Accuracy on test set: 42.87 %
Epoch 36: Accuracy on Train set: 100.00 %
Epoch 37: Accuracy on test set: 43.23 %
Epoch 37: Accuracy on Train set: 100.00 %
Epoch 38: Accuracy on test set: 43.43 %
Epoch 38: Accuracy on Train set: 100.00 %
Epoch 39: Accuracy on test set: 43.53 %
Epoch 39: Accuracy on Train set: 100.00 %
Epoch 40: Accuracy on test set: 43.66 %
Epoch 40: Accuracy on Train set: 100.00 %
```

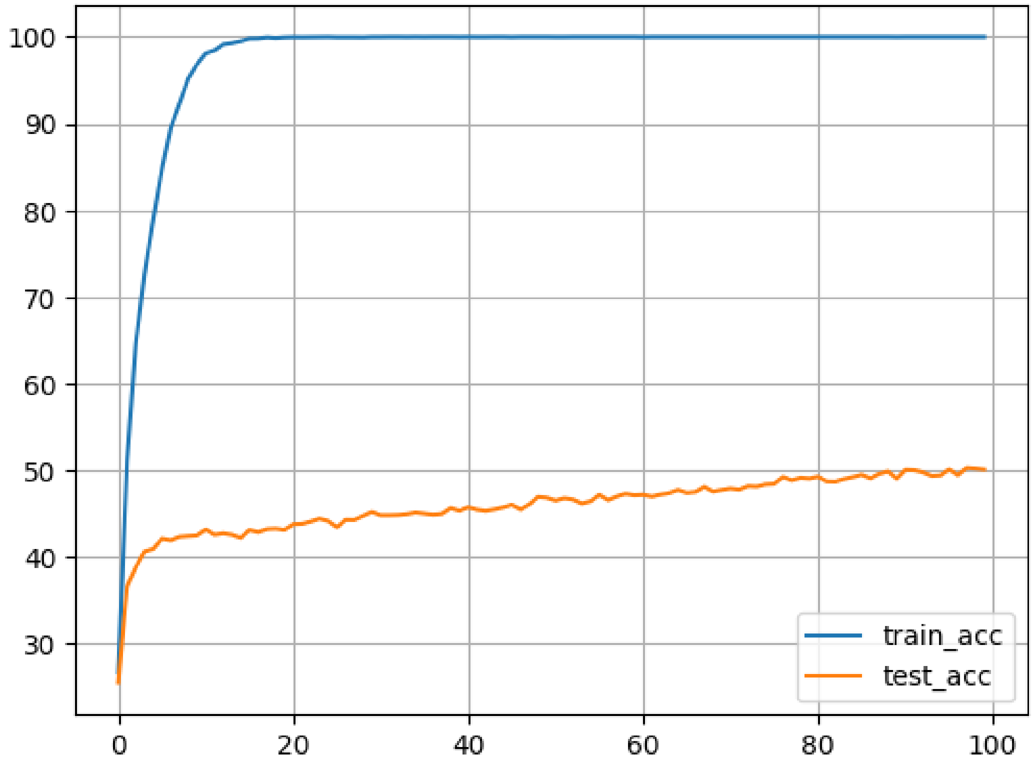
```
[epoch: 41]-loss: 0.001
Epoch 41: Accuracy on test set: 43.64 %
Epoch 41: Accuracy on Train set: 100.00 %
Epoch 42: Accuracy on test set: 43.19 %
Epoch 42: Accuracy on Train set: 100.00 %
Epoch 43: Accuracy on test set: 43.85 %
Epoch 43: Accuracy on Train set: 99.99 %
Epoch 44: Accuracy on test set: 44.04 %
Epoch 44: Accuracy on Train set: 99.98 %
Epoch 45: Accuracy on test set: 43.61 %
Epoch 45: Accuracy on Train set: 100.00 %
[epoch: 46]-loss: 0.001
Epoch 46: Accuracy on test set: 44.13 %
Epoch 46: Accuracy on Train set: 100.00 %
Epoch 47: Accuracy on test set: 43.76 %
Epoch 47: Accuracy on Train set: 99.98 %
Epoch 48: Accuracy on test set: 43.52 %
Epoch 48: Accuracy on Train set: 100.00 %
Epoch 49: Accuracy on test set: 44.08 %
Epoch 49: Accuracy on Train set: 100.00 %
Epoch 50: Accuracy on test set: 44.31 %
Epoch 50: Accuracy on Train set: 100.00 %
[epoch: 51]-loss: 0.001
Epoch 51: Accuracy on test set: 44.24 %
Epoch 51: Accuracy on Train set: 99.99 %
Epoch 52: Accuracy on test set: 44.33 %
Epoch 52: Accuracy on Train set: 99.99 %
Epoch 53: Accuracy on test set: 44.41 %
Epoch 53: Accuracy on Train set: 99.99 %
Epoch 54: Accuracy on test set: 44.45 %
Epoch 54: Accuracy on Train set: 100.00 %
Epoch 55: Accuracy on test set: 44.23 %
Epoch 55: Accuracy on Train set: 100.00 %
[epoch: 56]-loss: 0.001
Epoch 56: Accuracy on test set: 44.80 %
Epoch 56: Accuracy on Train set: 100.00 %
Epoch 57: Accuracy on test set: 45.15 %
Epoch 57: Accuracy on Train set: 100.00 %
Epoch 58: Accuracy on test set: 45.11 %
Epoch 58: Accuracy on Train set: 100.00 %
Epoch 59: Accuracy on test set: 45.37 %
Epoch 59: Accuracy on Train set: 100.00 %
Epoch 60: Accuracy on test set: 45.63 %
Epoch 60: Accuracy on Train set: 100.00 %
[epoch: 61]-loss: 0.001
Epoch 61: Accuracy on test set: 45.35 %
Epoch 61: Accuracy on Train set: 100.00 %
Epoch 62: Accuracy on test set: 45.82 %
Epoch 62: Accuracy on Train set: 100.00 %
Epoch 63: Accuracy on test set: 45.56 %
Epoch 63: Accuracy on Train set: 99.99 %
Epoch 64: Accuracy on test set: 45.39 %
Epoch 64: Accuracy on Train set: 100.00 %
Epoch 65: Accuracy on test set: 45.73 %
Epoch 65: Accuracy on Train set: 100.00 %
[epoch: 66]-loss: 0.001
Epoch 66: Accuracy on test set: 45.23 %
Epoch 66: Accuracy on Train set: 100.00 %
Epoch 67: Accuracy on test set: 45.86 %
Epoch 67: Accuracy on Train set: 100.00 %
Epoch 68: Accuracy on test set: 45.73 %
Epoch 68: Accuracy on Train set: 100.00 %
Epoch 69: Accuracy on test set: 45.12 %
Epoch 69: Accuracy on Train set: 100.00 %
Epoch 70: Accuracy on test set: 45.90 %
Epoch 70: Accuracy on Train set: 100.00 %
[epoch: 71]-loss: 0.001
Epoch 71: Accuracy on test set: 45.94 %
Epoch 71: Accuracy on Train set: 100.00 %
Epoch 72: Accuracy on test set: 45.96 %
Epoch 72: Accuracy on Train set: 99.99 %
Epoch 73: Accuracy on test set: 45.73 %
Epoch 73: Accuracy on Train set: 100.00 %
Epoch 74: Accuracy on test set: 45.71 %
Epoch 74: Accuracy on Train set: 100.00 %
Epoch 75: Accuracy on test set: 46.27 %
Epoch 75: Accuracy on Train set: 100.00 %
[epoch: 76]-loss: 0.001
Epoch 76: Accuracy on test set: 46.15 %
Epoch 76: Accuracy on Train set: 100.00 %
Epoch 77: Accuracy on test set: 45.93 %
Epoch 77: Accuracy on Train set: 100.00 %
Epoch 78: Accuracy on test set: 45.75 %
Epoch 78: Accuracy on Train set: 100.00 %
Epoch 79: Accuracy on test set: 46.23 %
Epoch 79: Accuracy on Train set: 100.00 %
Epoch 80: Accuracy on test set: 46.45 %
Epoch 80: Accuracy on Train set: 100.00 %
[epoch: 81]-loss: 0.001
Epoch 81: Accuracy on test set: 46.90 %
Epoch 81: Accuracy on Train set: 100.00 %
Epoch 82: Accuracy on test set: 47.18 %
Epoch 82: Accuracy on Train set: 100.00 %
Epoch 83: Accuracy on test set: 46.62 %
Epoch 83: Accuracy on Train set: 100.00 %
Epoch 84: Accuracy on test set: 46.88 %
Epoch 84: Accuracy on Train set: 100.00 %
Epoch 85: Accuracy on test set: 46.84 %
Epoch 85: Accuracy on Train set: 100.00 %
[epoch: 86]-loss: 0.001
Epoch 86: Accuracy on test set: 46.31 %
Epoch 86: Accuracy on Train set: 100.00 %
Epoch 87: Accuracy on test set: 47.11 %
Epoch 87: Accuracy on Train set: 100.00 %
Epoch 88: Accuracy on test set: 46.98 %
Epoch 88: Accuracy on Train set: 100.00 %
Epoch 89: Accuracy on test set: 46.30 %
Epoch 89: Accuracy on Train set: 100.00 %
Epoch 90: Accuracy on test set: 46.60 %
Epoch 90: Accuracy on Train set: 100.00 %
[epoch: 91]-loss: 0.001
Epoch 91: Accuracy on test set: 47.08 %
Epoch 91: Accuracy on Train set: 100.00 %
Epoch 92: Accuracy on test set: 47.09 %
Epoch 92: Accuracy on Train set: 100.00 %
Epoch 93: Accuracy on test set: 47.25 %
Epoch 93: Accuracy on Train set: 100.00 %
Epoch 94: Accuracy on test set: 46.86 %
Epoch 94: Accuracy on Train set: 100.00 %
Epoch 95: Accuracy on test set: 46.90 %
Epoch 95: Accuracy on Train set: 100.00 %
[epoch: 96]-loss: 0.001
Epoch 96: Accuracy on test set: 47.04 %
Epoch 96: Accuracy on Train set: 100.00 %
Epoch 97: Accuracy on test set: 47.57 %
Epoch 97: Accuracy on Train set: 100.00 %
Epoch 98: Accuracy on test set: 47.29 %
Epoch 98: Accuracy on Train set: 100.00 %
Epoch 99: Accuracy on test set: 47.11 %
Epoch 99: Accuracy on Train set: 100.00 %
Epoch 100: Accuracy on test set: 47.27 %
Epoch 100: Accuracy on Train set: 100.00 %
Training completed!
```



```
[epoch: 1]-loss: 0.004
Epoch 1: Accuracy on test set: 25.57 %
Epoch 1: Accuracy on Train set: 26.75 %
Epoch 2: Accuracy on test set: 36.64 %
Epoch 2: Accuracy on Train set: 51.10 %
Epoch 3: Accuracy on test set: 38.86 %
Epoch 3: Accuracy on Train set: 64.66 %
Epoch 4: Accuracy on test set: 40.67 %
Epoch 4: Accuracy on Train set: 72.67 %
Epoch 5: Accuracy on test set: 40.96 %
Epoch 5: Accuracy on Train set: 78.94 %
[epoch: 6]-loss: 0.003
Epoch 6: Accuracy on test set: 42.16 %
Epoch 6: Accuracy on Train set: 84.79 %
Epoch 7: Accuracy on test set: 41.97 %
Epoch 7: Accuracy on Train set: 89.52 %
Epoch 8: Accuracy on test set: 42.37 %
Epoch 8: Accuracy on Train set: 92.34 %
Epoch 9: Accuracy on test set: 42.45 %
Epoch 9: Accuracy on Train set: 95.20 %
Epoch 10: Accuracy on test set: 42.52 %
Epoch 10: Accuracy on Train set: 96.84 %
```

```
[epoch: 11]-loss: 0.002
Epoch 11: Accuracy on test set: 43.20 %
Epoch 11: Accuracy on Train set: 98.11 %
Epoch 12: Accuracy on test set: 42.61 %
Epoch 12: Accuracy on Train set: 98.44 %
Epoch 13: Accuracy on test set: 42.77 %
Epoch 13: Accuracy on Train set: 99.13 %
Epoch 14: Accuracy on test set: 42.60 %
Epoch 14: Accuracy on Train set: 99.28 %
Epoch 15: Accuracy on test set: 42.25 %
Epoch 15: Accuracy on Train set: 99.50 %
[epoch: 16]-loss: 0.002
Epoch 16: Accuracy on test set: 43.14 %
Epoch 16: Accuracy on Train set: 99.80 %
Epoch 17: Accuracy on test set: 42.89 %
Epoch 17: Accuracy on Train set: 99.82 %
Epoch 18: Accuracy on test set: 43.25 %
Epoch 18: Accuracy on Train set: 99.94 %
Epoch 19: Accuracy on test set: 43.30 %
Epoch 19: Accuracy on Train set: 99.87 %
Epoch 20: Accuracy on test set: 43.15 %
Epoch 20: Accuracy on Train set: 99.94 %
[epoch: 21]-loss: 0.002
Epoch 21: Accuracy on test set: 43.80 %
Epoch 21: Accuracy on Train set: 99.97 %
Epoch 22: Accuracy on test set: 43.85 %
Epoch 22: Accuracy on Train set: 99.96 %
Epoch 23: Accuracy on test set: 44.11 %
Epoch 23: Accuracy on Train set: 99.97 %
Epoch 24: Accuracy on test set: 44.46 %
Epoch 24: Accuracy on Train set: 99.98 %
Epoch 25: Accuracy on test set: 44.17 %
Epoch 25: Accuracy on Train set: 99.99 %
[epoch: 26]-loss: 0.002
Epoch 26: Accuracy on test set: 43.47 %
Epoch 26: Accuracy on Train set: 99.96 %
Epoch 27: Accuracy on test set: 44.32 %
Epoch 27: Accuracy on Train set: 99.96 %
Epoch 28: Accuracy on test set: 44.30 %
Epoch 28: Accuracy on Train set: 99.96 %
Epoch 29: Accuracy on test set: 44.79 %
Epoch 29: Accuracy on Train set: 99.95 %
Epoch 30: Accuracy on test set: 45.23 %
Epoch 30: Accuracy on Train set: 99.99 %
[epoch: 31]-loss: 0.002
Epoch 31: Accuracy on test set: 44.85 %
Epoch 31: Accuracy on Train set: 99.99 %
Epoch 32: Accuracy on test set: 44.85 %
Epoch 32: Accuracy on Train set: 100.00 %
Epoch 33: Accuracy on test set: 44.88 %
Epoch 33: Accuracy on Train set: 100.00 %
Epoch 34: Accuracy on test set: 44.98 %
Epoch 34: Accuracy on Train set: 99.99 %
Epoch 35: Accuracy on test set: 45.17 %
Epoch 35: Accuracy on Train set: 100.00 %
[epoch: 36]-loss: 0.002
Epoch 36: Accuracy on test set: 45.04 %
Epoch 36: Accuracy on Train set: 99.99 %
Epoch 37: Accuracy on test set: 44.92 %
Epoch 37: Accuracy on Train set: 100.00 %
Epoch 38: Accuracy on test set: 45.01 %
Epoch 38: Accuracy on Train set: 100.00 %
Epoch 39: Accuracy on test set: 45.69 %
Epoch 39: Accuracy on Train set: 100.00 %
Epoch 40: Accuracy on test set: 45.36 %
Epoch 40: Accuracy on Train set: 99.99 %
[epoch: 41]-loss: 0.002
Epoch 41: Accuracy on test set: 45.77 %
Epoch 41: Accuracy on Train set: 100.00 %
Epoch 42: Accuracy on test set: 45.50 %
Epoch 42: Accuracy on Train set: 99.99 %
Epoch 43: Accuracy on test set: 45.35 %
Epoch 43: Accuracy on Train set: 99.99 %
Epoch 44: Accuracy on test set: 45.53 %
Epoch 44: Accuracy on Train set: 100.00 %
Epoch 45: Accuracy on test set: 45.75 %
Epoch 45: Accuracy on Train set: 100.00 %
[epoch: 46]-loss: 0.002
Epoch 46: Accuracy on test set: 46.06 %
Epoch 46: Accuracy on Train set: 99.98 %
Epoch 47: Accuracy on test set: 45.51 %
Epoch 47: Accuracy on Train set: 100.00 %
Epoch 48: Accuracy on test set: 46.11 %
Epoch 48: Accuracy on Train set: 100.00 %
Epoch 49: Accuracy on test set: 46.97 %
Epoch 49: Accuracy on Train set: 100.00 %
Epoch 50: Accuracy on test set: 46.87 %
Epoch 50: Accuracy on Train set: 100.00 %
[epoch: 51]-loss: 0.002
Epoch 51: Accuracy on test set: 46.54 %
Epoch 51: Accuracy on Train set: 99.99 %
Epoch 52: Accuracy on test set: 46.81 %
Epoch 52: Accuracy on Train set: 99.99 %
Epoch 53: Accuracy on test set: 46.66 %
Epoch 53: Accuracy on Train set: 100.00 %
Epoch 54: Accuracy on test set: 46.19 %
Epoch 54: Accuracy on Train set: 100.00 %
Epoch 55: Accuracy on test set: 46.46 %
Epoch 55: Accuracy on Train set: 100.00 %
[epoch: 56]-loss: 0.001
Epoch 56: Accuracy on test set: 47.19 %
Epoch 56: Accuracy on Train set: 100.00 %
Epoch 57: Accuracy on test set: 46.59 %
Epoch 57: Accuracy on Train set: 100.00 %
Epoch 58: Accuracy on test set: 47.02 %
Epoch 58: Accuracy on Train set: 100.00 %
Epoch 59: Accuracy on test set: 47.32 %
Epoch 59: Accuracy on Train set: 100.00 %
Epoch 60: Accuracy on test set: 47.15 %
Epoch 60: Accuracy on Train set: 100.00 %
[epoch: 61]-loss: 0.002
Epoch 61: Accuracy on test set: 47.21 %
Epoch 61: Accuracy on Train set: 99.98 %
Epoch 62: Accuracy on test set: 47.00 %
Epoch 62: Accuracy on Train set: 100.00 %
Epoch 63: Accuracy on test set: 47.23 %
Epoch 63: Accuracy on Train set: 100.00 %
Epoch 64: Accuracy on test set: 47.40 %
Epoch 64: Accuracy on Train set: 100.00 %
Epoch 65: Accuracy on test set: 47.77 %
Epoch 65: Accuracy on Train set: 100.00 %
[epoch: 66]-loss: 0.001
Epoch 66: Accuracy on test set: 47.42 %
Epoch 66: Accuracy on Train set: 100.00 %
Epoch 67: Accuracy on test set: 47.53 %
Epoch 67: Accuracy on Train set: 100.00 %
Epoch 68: Accuracy on test set: 48.14 %
Epoch 68: Accuracy on Train set: 100.00 %
Epoch 69: Accuracy on test set: 47.56 %
Epoch 69: Accuracy on Train set: 100.00 %
Epoch 70: Accuracy on test set: 47.77 %
Epoch 70: Accuracy on Train set: 100.00 %
[epoch: 71]-loss: 0.001
Epoch 71: Accuracy on test set: 47.93 %
Epoch 71: Accuracy on Train set: 100.00 %
Epoch 72: Accuracy on test set: 47.80 %
Epoch 72: Accuracy on Train set: 100.00 %
Epoch 73: Accuracy on test set: 48.21 %
Epoch 73: Accuracy on Train set: 100.00 %
Epoch 74: Accuracy on test set: 48.16 %
Epoch 74: Accuracy on Train set: 100.00 %
Epoch 75: Accuracy on test set: 48.41 %
Epoch 75: Accuracy on Train set: 100.00 %
[epoch: 76]-loss: 0.002
```

```
Epoch 76: Accuracy on test set: 48.47 %
Epoch 76: Accuracy on Train set: 100.00 %
Epoch 77: Accuracy on test set: 49.26 %
Epoch 77: Accuracy on Train set: 100.00 %
Epoch 78: Accuracy on test set: 48.87 %
Epoch 78: Accuracy on Train set: 100.00 %
Epoch 79: Accuracy on test set: 49.17 %
Epoch 79: Accuracy on Train set: 100.00 %
Epoch 80: Accuracy on test set: 49.08 %
Epoch 80: Accuracy on Train set: 100.00 %
[epoch: 81]-loss: 0.002
Epoch 81: Accuracy on test set: 49.29 %
Epoch 81: Accuracy on Train set: 100.00 %
Epoch 82: Accuracy on test set: 48.75 %
Epoch 82: Accuracy on Train set: 100.00 %
Epoch 83: Accuracy on test set: 48.72 %
Epoch 83: Accuracy on Train set: 100.00 %
Epoch 84: Accuracy on test set: 49.03 %
Epoch 84: Accuracy on Train set: 100.00 %
Epoch 85: Accuracy on test set: 49.25 %
Epoch 85: Accuracy on Train set: 100.00 %
[epoch: 86]-loss: 0.001
Epoch 86: Accuracy on test set: 49.51 %
Epoch 86: Accuracy on Train set: 100.00 %
Epoch 87: Accuracy on test set: 49.10 %
Epoch 87: Accuracy on Train set: 100.00 %
Epoch 88: Accuracy on test set: 49.62 %
Epoch 88: Accuracy on Train set: 100.00 %
Epoch 89: Accuracy on test set: 49.90 %
Epoch 89: Accuracy on Train set: 100.00 %
Epoch 90: Accuracy on test set: 49.06 %
Epoch 90: Accuracy on Train set: 99.99 %
[epoch: 91]-loss: 0.001
Epoch 91: Accuracy on test set: 50.09 %
Epoch 91: Accuracy on Train set: 100.00 %
Epoch 92: Accuracy on test set: 50.04 %
Epoch 92: Accuracy on Train set: 100.00 %
Epoch 93: Accuracy on test set: 49.80 %
Epoch 93: Accuracy on Train set: 100.00 %
Epoch 94: Accuracy on test set: 49.37 %
Epoch 94: Accuracy on Train set: 100.00 %
Epoch 95: Accuracy on test set: 49.44 %
Epoch 95: Accuracy on Train set: 100.00 %
[epoch: 96]-loss: 0.002
Epoch 96: Accuracy on test set: 50.12 %
Epoch 96: Accuracy on Train set: 100.00 %
Epoch 97: Accuracy on test set: 49.47 %
Epoch 97: Accuracy on Train set: 100.00 %
Epoch 98: Accuracy on test set: 50.32 %
Epoch 98: Accuracy on Train set: 100.00 %
Epoch 99: Accuracy on test set: 50.24 %
Epoch 99: Accuracy on Train set: 100.00 %
Epoch 100: Accuracy on test set: 50.12 %
Epoch 100: Accuracy on Train set: 100.00 %
Training completed!
```



For Alpha = 0.2. We can achieve upto 100% accuracy on Train set and 47% accuracy on Test set

For Alpha = 0.4. We can achieve upto 100% accuracy on Train set and 50% accuracy on Test set

0.0.8 Cutout Augmentation

[20]:

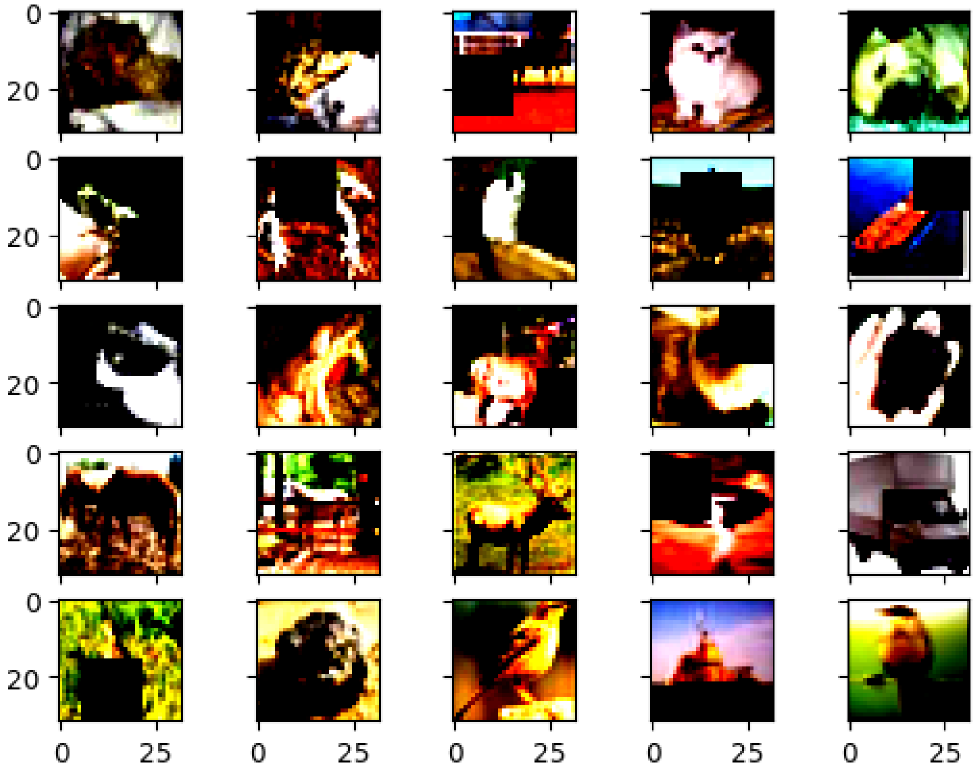
```python
def cutout(inputs, length=16, p=0.5):

    inputs = inputs.clone()

    batch_size, channels, height, width = inputs.size()

    for img in inputs:

        if np.random.random() > p:
            continue

        # Generate random center coordinates for cutout position
        center_x = np.random.randint(0, width)
        center_y = np.random.randint(0, height)

        # Calculate cutout region boundaries
        left = max(0, center_x - length // 2)
        upper = max(0, center_y - length // 2)
        right = min(width, center_x + length // 2)
        lower = min(height, center_y + length // 2)

        # Apply cutout to each image in the batch
        for i in range(batch_size):
            img[:, upper:lower, left:right] = 0.

    return inputs

cutout_images = cutout(x_train[:100], 16,0.5)
```

[21]:

```
# Visualize 1 sample
sample_idx_ctr = 0
num_rows = 5
num_cols = 5

fig, ax = plt.subplots(num_rows, num_cols, sharex=True, sharey=True)

for i in range(num_rows):
    for j in range(num_cols):

        img = cutout_images[sample_idx_ctr]
        ax[i,j].imshow(img.squeeze().permute(1,2,0))
        sample_idx_ctr += 1
```

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

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



```
[22]: device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

      # Define the ResNet-18 model
      model = torchvision.models.resnet18(pretrained=False)
      num_classes = 10

      model.fc = nn.Linear(model.fc.in_features, num_classes)
      model = model.to(device)

      model.eval()
```

```
[22]: ResNet(
        (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3),
      bias=False)
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
        (relu): ReLU(inplace=True)
        (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1,
      ceil_mode=False)
        (layer1): Sequential(
          (0): BasicBlock(
            (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
      bias=False)
            (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
            (relu): ReLU(inplace=True)
            (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
      bias=False)
            (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
          )
          (1): BasicBlock(
            (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
      bias=False)
            (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
            (relu): ReLU(inplace=True)
            (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
      bias=False)
            (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
          )
        )
        (layer2): Sequential(
          (0): BasicBlock(
            (conv1): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1,
      1), bias=False)
            (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
            (relu): ReLU(inplace=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)
            (downsample): Sequential(
              (0): Conv2d(64, 128, kernel_size=(1, 1), stride=(2, 2), bias=False)
              (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
            )
          )
          (1): BasicBlock(
            (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
      1), bias=False)
            (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
            (relu): ReLU(inplace=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)
          )
        )
        (layer3): Sequential(
          (0): BasicBlock(
            (conv1): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1,
      1), bias=False)
            (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
            (relu): ReLU(inplace=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)
            (downsample): Sequential(
              (0): Conv2d(128, 256, kernel_size=(1, 1), stride=(2, 2), bias=False)
              (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
            )
          )
          (1): BasicBlock(
            (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
      1), bias=False)
            (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
            (relu): ReLU(inplace=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)
          )
        )
        (layer4): Sequential(
          (0): BasicBlock(
            (conv1): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1,
      1), bias=False)
            (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
      track_running_stats=True)
            (relu): ReLU(inplace=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)
            (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): BasicBlock(
      (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=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)
    )
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(1, 1))
  (fc): Linear(in_features=512, out_features=10, bias=True)
)
```

[23]:

```
# Define loss function and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.0001)

test_acc = []
train_acc = []

# Training loop
num_epochs = 100

for epoch in range(num_epochs):

    i = 0
    iter = 0
    batch_size = 500

    while i < x_train.shape[0]:
        model.train()

        inputs = x_train[i:i+batch_size]
        labels = y_train[i:i+batch_size].type(torch.LongTensor).to(device)
        labels_oh = y_train_oh[i:i+batch_size].to(device)
        inputs = cutout(inputs, 16, 0.5)
        inputs = inputs.to(device)

        optimizer.zero_grad()

        outputs = model(inputs)
        loss = criterion(outputs, labels_oh)
        loss.backward()
        optimizer.step()

        i += batch_size
        iter += 1

        if (iter + 1) % 5 == 0:
            print('[epoch: %d, iter: %d]-loss: %.3f' % (epoch + 1, iter + 1,loss.item() / batch_size))

    # Evaluation on the test set
    model.eval()

    # Test Accuracy
    correct = 0
    total = 0
    with torch.no_grad():

        inputs = x_test.to(device)
        labels = y_test.type(torch.LongTensor).to(device)
        outputs = model(inputs)
        _, predicted = torch.max(outputs.data, 1)
        total += labels.size(0)
        correct += (predicted == labels).sum().item()

    accuracy = 100 * correct / total
    test_acc.append(accuracy)
    print('Epoch %d: Accuracy on test set: %.2f %%' % (epoch + 1, accuracy))

    # Train Accuracy
    correct = 0
    total = 0
    with torch.no_grad():

        inputs = x_train.to(device)
        labels = y_train.type(torch.LongTensor).to(device)
        outputs = model(inputs)
        _, predicted = torch.max(outputs.data, 1)
        total += labels.size(0)
        correct += (predicted == labels).sum().item()

    accuracy = 100 * correct / total
    train_acc.append(accuracy)
    print('Epoch %d: Accuracy on Train set: %.2f %%' % (epoch + 1, accuracy))

print('Training completed!')
```

```
[epoch: 1, iter: 5]-loss: 0.005
[epoch: 1, iter: 10]-loss: 0.004
[epoch: 1, iter: 15]-loss: 0.004
[epoch: 1, iter: 20]-loss: 0.004
Epoch 1: Accuracy on test set: 21.77 %
Epoch 1: Accuracy on Train set: 22.60 %
[epoch: 2, iter: 5]-loss: 0.003
[epoch: 2, iter: 10]-loss: 0.003
[epoch: 2, iter: 15]-loss: 0.003
[epoch: 2, iter: 20]-loss: 0.003
Epoch 2: Accuracy on test set: 36.73 %
Epoch 2: Accuracy on Train set: 50.83 %
[epoch: 3, iter: 5]-loss: 0.003
[epoch: 3, iter: 10]-loss: 0.003
[epoch: 3, iter: 15]-loss: 0.003
[epoch: 3, iter: 20]-loss: 0.003
Epoch 3: Accuracy on test set: 39.82 %
Epoch 3: Accuracy on Train set: 66.86 %
[epoch: 4, iter: 5]-loss: 0.003
[epoch: 4, iter: 10]-loss: 0.003
[epoch: 4, iter: 15]-loss: 0.003
[epoch: 4, iter: 20]-loss: 0.002
Epoch 4: Accuracy on test set: 41.58 %
Epoch 4: Accuracy on Train set: 76.59 %
[epoch: 5, iter: 5]-loss: 0.002
[epoch: 5, iter: 10]-loss: 0.002
[epoch: 5, iter: 15]-loss: 0.002
[epoch: 5, iter: 20]-loss: 0.002
Epoch 5: Accuracy on test set: 42.53 %
Epoch 5: Accuracy on Train set: 83.77 %
[epoch: 6, iter: 5]-loss: 0.002
[epoch: 6, iter: 10]-loss: 0.002
[epoch: 6, iter: 15]-loss: 0.002
[epoch: 6, iter: 20]-loss: 0.002
Epoch 6: Accuracy on test set: 43.46 %
Epoch 6: Accuracy on Train set: 89.71 %
[epoch: 7, iter: 5]-loss: 0.002
[epoch: 7, iter: 10]-loss: 0.002
[epoch: 7, iter: 15]-loss: 0.002
[epoch: 7, iter: 20]-loss: 0.002
Epoch 7: Accuracy on test set: 43.65 %
Epoch 7: Accuracy on Train set: 93.67 %
[epoch: 8, iter: 5]-loss: 0.002
[epoch: 8, iter: 10]-loss: 0.002
[epoch: 8, iter: 15]-loss: 0.002
[epoch: 8, iter: 20]-loss: 0.001
Epoch 8: Accuracy on test set: 43.57 %
Epoch 8: Accuracy on Train set: 96.36 %
```

```
[epoch: 9, iter: 5]-loss: 0.001
[epoch: 9, iter: 10]-loss: 0.001
[epoch: 9, iter: 15]-loss: 0.001
[epoch: 9, iter: 20]-loss: 0.001
Epoch 9: Accuracy on test set: 44.28 %
Epoch 9: Accuracy on Train set: 98.08 %
[epoch: 10, iter: 5]-loss: 0.001
[epoch: 10, iter: 10]-loss: 0.001
[epoch: 10, iter: 15]-loss: 0.001
[epoch: 10, iter: 20]-loss: 0.001
Epoch 10: Accuracy on test set: 43.56 %
Epoch 10: Accuracy on Train set: 98.61 %
[epoch: 11, iter: 5]-loss: 0.001
[epoch: 11, iter: 10]-loss: 0.001
[epoch: 11, iter: 15]-loss: 0.001
[epoch: 11, iter: 20]-loss: 0.001
Epoch 11: Accuracy on test set: 44.32 %
Epoch 11: Accuracy on Train set: 99.24 %
[epoch: 12, iter: 5]-loss: 0.001
[epoch: 12, iter: 10]-loss: 0.001
[epoch: 12, iter: 15]-loss: 0.001
[epoch: 12, iter: 20]-loss: 0.001
Epoch 12: Accuracy on test set: 44.15 %
Epoch 12: Accuracy on Train set: 99.54 %
[epoch: 13, iter: 5]-loss: 0.001
[epoch: 13, iter: 10]-loss: 0.001
[epoch: 13, iter: 15]-loss: 0.001
[epoch: 13, iter: 20]-loss: 0.001
Epoch 13: Accuracy on test set: 45.07 %
Epoch 13: Accuracy on Train set: 99.77 %
[epoch: 14, iter: 5]-loss: 0.001
[epoch: 14, iter: 10]-loss: 0.001
[epoch: 14, iter: 15]-loss: 0.001
[epoch: 14, iter: 20]-loss: 0.001
Epoch 14: Accuracy on test set: 45.42 %
Epoch 14: Accuracy on Train set: 99.82 %
[epoch: 15, iter: 5]-loss: 0.001
[epoch: 15, iter: 10]-loss: 0.001
[epoch: 15, iter: 15]-loss: 0.001
[epoch: 15, iter: 20]-loss: 0.001
Epoch 15: Accuracy on test set: 45.04 %
Epoch 15: Accuracy on Train set: 99.85 %
[epoch: 16, iter: 5]-loss: 0.001
[epoch: 16, iter: 10]-loss: 0.001
[epoch: 16, iter: 15]-loss: 0.001
[epoch: 16, iter: 20]-loss: 0.001
Epoch 16: Accuracy on test set: 45.38 %
Epoch 16: Accuracy on Train set: 99.90 %
```

```
[epoch: 17, iter: 5]-loss: 0.001
[epoch: 17, iter: 10]-loss: 0.001
[epoch: 17, iter: 15]-loss: 0.001
[epoch: 17, iter: 20]-loss: 0.001
Epoch 17: Accuracy on test set: 45.61 %
Epoch 17: Accuracy on Train set: 99.77 %
[epoch: 18, iter: 5]-loss: 0.001
[epoch: 18, iter: 10]-loss: 0.001
[epoch: 18, iter: 15]-loss: 0.001
[epoch: 18, iter: 20]-loss: 0.001
Epoch 18: Accuracy on test set: 45.51 %
Epoch 18: Accuracy on Train set: 99.90 %
[epoch: 19, iter: 5]-loss: 0.001
[epoch: 19, iter: 10]-loss: 0.001
[epoch: 19, iter: 15]-loss: 0.001
[epoch: 19, iter: 20]-loss: 0.001
Epoch 19: Accuracy on test set: 45.86 %
Epoch 19: Accuracy on Train set: 99.91 %
[epoch: 20, iter: 5]-loss: 0.001
[epoch: 20, iter: 10]-loss: 0.001
[epoch: 20, iter: 15]-loss: 0.001
[epoch: 20, iter: 20]-loss: 0.001
Epoch 20: Accuracy on test set: 45.59 %
Epoch 20: Accuracy on Train set: 99.87 %
[epoch: 21, iter: 5]-loss: 0.001
[epoch: 21, iter: 10]-loss: 0.001
[epoch: 21, iter: 15]-loss: 0.001
[epoch: 21, iter: 20]-loss: 0.001
Epoch 21: Accuracy on test set: 45.53 %
Epoch 21: Accuracy on Train set: 99.90 %
[epoch: 22, iter: 5]-loss: 0.001
[epoch: 22, iter: 10]-loss: 0.001
[epoch: 22, iter: 15]-loss: 0.001
[epoch: 22, iter: 20]-loss: 0.001
Epoch 22: Accuracy on test set: 45.99 %
Epoch 22: Accuracy on Train set: 99.94 %
[epoch: 23, iter: 5]-loss: 0.001
[epoch: 23, iter: 10]-loss: 0.001
[epoch: 23, iter: 15]-loss: 0.001
[epoch: 23, iter: 20]-loss: 0.001
Epoch 23: Accuracy on test set: 46.50 %
Epoch 23: Accuracy on Train set: 99.93 %
[epoch: 24, iter: 5]-loss: 0.001
[epoch: 24, iter: 10]-loss: 0.001
[epoch: 24, iter: 15]-loss: 0.001
[epoch: 24, iter: 20]-loss: 0.001
Epoch 24: Accuracy on test set: 46.44 %
Epoch 24: Accuracy on Train set: 99.97 %
```

```
[epoch: 25, iter: 5]-loss: 0.001
[epoch: 25, iter: 10]-loss: 0.001
[epoch: 25, iter: 15]-loss: 0.001
[epoch: 25, iter: 20]-loss: 0.001
Epoch 25: Accuracy on test set: 46.28 %
Epoch 25: Accuracy on Train set: 99.95 %
[epoch: 26, iter: 5]-loss: 0.001
[epoch: 26, iter: 10]-loss: 0.001
[epoch: 26, iter: 15]-loss: 0.001
[epoch: 26, iter: 20]-loss: 0.001
Epoch 26: Accuracy on test set: 46.52 %
Epoch 26: Accuracy on Train set: 99.96 %
[epoch: 27, iter: 5]-loss: 0.001
[epoch: 27, iter: 10]-loss: 0.001
[epoch: 27, iter: 15]-loss: 0.001
[epoch: 27, iter: 20]-loss: 0.001
Epoch 27: Accuracy on test set: 46.41 %
Epoch 27: Accuracy on Train set: 99.88 %
[epoch: 28, iter: 5]-loss: 0.001
[epoch: 28, iter: 10]-loss: 0.001
[epoch: 28, iter: 15]-loss: 0.001
[epoch: 28, iter: 20]-loss: 0.001
Epoch 28: Accuracy on test set: 46.90 %
Epoch 28: Accuracy on Train set: 99.95 %
[epoch: 29, iter: 5]-loss: 0.001
[epoch: 29, iter: 10]-loss: 0.001
[epoch: 29, iter: 15]-loss: 0.001
[epoch: 29, iter: 20]-loss: 0.001
Epoch 29: Accuracy on test set: 46.84 %
Epoch 29: Accuracy on Train set: 99.91 %
[epoch: 30, iter: 5]-loss: 0.001
[epoch: 30, iter: 10]-loss: 0.001
[epoch: 30, iter: 15]-loss: 0.001
[epoch: 30, iter: 20]-loss: 0.001
Epoch 30: Accuracy on test set: 47.12 %
Epoch 30: Accuracy on Train set: 99.97 %
[epoch: 31, iter: 5]-loss: 0.000
[epoch: 31, iter: 10]-loss: 0.001
[epoch: 31, iter: 15]-loss: 0.001
[epoch: 31, iter: 20]-loss: 0.000
Epoch 31: Accuracy on test set: 46.59 %
Epoch 31: Accuracy on Train set: 99.93 %
[epoch: 32, iter: 5]-loss: 0.001
[epoch: 32, iter: 10]-loss: 0.000
[epoch: 32, iter: 15]-loss: 0.001
[epoch: 32, iter: 20]-loss: 0.000
Epoch 32: Accuracy on test set: 46.93 %
Epoch 32: Accuracy on Train set: 99.97 %
```

```
[epoch: 33, iter: 5]-loss: 0.001
[epoch: 33, iter: 10]-loss: 0.000
[epoch: 33, iter: 15]-loss: 0.000
[epoch: 33, iter: 20]-loss: 0.001
Epoch 33: Accuracy on test set: 47.35 %
Epoch 33: Accuracy on Train set: 100.00 %
[epoch: 34, iter: 5]-loss: 0.000
[epoch: 34, iter: 10]-loss: 0.001
[epoch: 34, iter: 15]-loss: 0.000
[epoch: 34, iter: 20]-loss: 0.000
Epoch 34: Accuracy on test set: 47.81 %
Epoch 34: Accuracy on Train set: 99.99 %
[epoch: 35, iter: 5]-loss: 0.000
[epoch: 35, iter: 10]-loss: 0.000
[epoch: 35, iter: 15]-loss: 0.001
[epoch: 35, iter: 20]-loss: 0.000
Epoch 35: Accuracy on test set: 47.62 %
Epoch 35: Accuracy on Train set: 99.95 %
[epoch: 36, iter: 5]-loss: 0.001
[epoch: 36, iter: 10]-loss: 0.001
[epoch: 36, iter: 15]-loss: 0.000
[epoch: 36, iter: 20]-loss: 0.001
Epoch 36: Accuracy on test set: 47.60 %
Epoch 36: Accuracy on Train set: 99.97 %
[epoch: 37, iter: 5]-loss: 0.001
[epoch: 37, iter: 10]-loss: 0.001
[epoch: 37, iter: 15]-loss: 0.000
[epoch: 37, iter: 20]-loss: 0.001
Epoch 37: Accuracy on test set: 46.53 %
Epoch 37: Accuracy on Train set: 99.94 %
[epoch: 38, iter: 5]-loss: 0.000
[epoch: 38, iter: 10]-loss: 0.001
[epoch: 38, iter: 15]-loss: 0.000
[epoch: 38, iter: 20]-loss: 0.000
Epoch 38: Accuracy on test set: 47.81 %
Epoch 38: Accuracy on Train set: 99.97 %
[epoch: 39, iter: 5]-loss: 0.000
[epoch: 39, iter: 10]-loss: 0.000
[epoch: 39, iter: 15]-loss: 0.000
[epoch: 39, iter: 20]-loss: 0.000
Epoch 39: Accuracy on test set: 48.01 %
Epoch 39: Accuracy on Train set: 99.95 %
[epoch: 40, iter: 5]-loss: 0.001
[epoch: 40, iter: 10]-loss: 0.001
[epoch: 40, iter: 15]-loss: 0.001
[epoch: 40, iter: 20]-loss: 0.001
Epoch 40: Accuracy on test set: 47.88 %
Epoch 40: Accuracy on Train set: 99.99 %
```

```
[epoch: 41, iter: 5]-loss: 0.000
[epoch: 41, iter: 10]-loss: 0.000
[epoch: 41, iter: 15]-loss: 0.000
[epoch: 41, iter: 20]-loss: 0.001
Epoch 41: Accuracy on test set: 47.80 %
Epoch 41: Accuracy on Train set: 99.97 %
[epoch: 42, iter: 5]-loss: 0.000
[epoch: 42, iter: 10]-loss: 0.000
[epoch: 42, iter: 15]-loss: 0.000
[epoch: 42, iter: 20]-loss: 0.000
Epoch 42: Accuracy on test set: 48.43 %
Epoch 42: Accuracy on Train set: 99.98 %
[epoch: 43, iter: 5]-loss: 0.000
[epoch: 43, iter: 10]-loss: 0.000
[epoch: 43, iter: 15]-loss: 0.000
[epoch: 43, iter: 20]-loss: 0.000
Epoch 43: Accuracy on test set: 48.29 %
Epoch 43: Accuracy on Train set: 99.99 %
[epoch: 44, iter: 5]-loss: 0.000
[epoch: 44, iter: 10]-loss: 0.000
[epoch: 44, iter: 15]-loss: 0.000
[epoch: 44, iter: 20]-loss: 0.000
Epoch 44: Accuracy on test set: 47.86 %
Epoch 44: Accuracy on Train set: 99.97 %
[epoch: 45, iter: 5]-loss: 0.000
[epoch: 45, iter: 10]-loss: 0.000
[epoch: 45, iter: 15]-loss: 0.000
[epoch: 45, iter: 20]-loss: 0.000
Epoch 45: Accuracy on test set: 47.88 %
Epoch 45: Accuracy on Train set: 99.99 %
[epoch: 46, iter: 5]-loss: 0.000
[epoch: 46, iter: 10]-loss: 0.001
[epoch: 46, iter: 15]-loss: 0.000
[epoch: 46, iter: 20]-loss: 0.000
Epoch 46: Accuracy on test set: 48.55 %
Epoch 46: Accuracy on Train set: 99.97 %
[epoch: 47, iter: 5]-loss: 0.000
[epoch: 47, iter: 10]-loss: 0.000
[epoch: 47, iter: 15]-loss: 0.000
[epoch: 47, iter: 20]-loss: 0.000
Epoch 47: Accuracy on test set: 47.39 %
Epoch 47: Accuracy on Train set: 99.96 %
[epoch: 48, iter: 5]-loss: 0.000
[epoch: 48, iter: 10]-loss: 0.000
[epoch: 48, iter: 15]-loss: 0.000
[epoch: 48, iter: 20]-loss: 0.000
Epoch 48: Accuracy on test set: 48.29 %
Epoch 48: Accuracy on Train set: 100.00 %
```

```
[epoch: 49, iter: 5]-loss: 0.000
[epoch: 49, iter: 10]-loss: 0.000
[epoch: 49, iter: 15]-loss: 0.000
[epoch: 49, iter: 20]-loss: 0.000
Epoch 49: Accuracy on test set: 48.20 %
Epoch 49: Accuracy on Train set: 100.00 %
[epoch: 50, iter: 5]-loss: 0.000
[epoch: 50, iter: 10]-loss: 0.000
[epoch: 50, iter: 15]-loss: 0.000
[epoch: 50, iter: 20]-loss: 0.000
Epoch 50: Accuracy on test set: 48.29 %
Epoch 50: Accuracy on Train set: 100.00 %
[epoch: 51, iter: 5]-loss: 0.000
[epoch: 51, iter: 10]-loss: 0.000
[epoch: 51, iter: 15]-loss: 0.000
[epoch: 51, iter: 20]-loss: 0.000
Epoch 51: Accuracy on test set: 48.35 %
Epoch 51: Accuracy on Train set: 99.98 %
[epoch: 52, iter: 5]-loss: 0.000
[epoch: 52, iter: 10]-loss: 0.000
[epoch: 52, iter: 15]-loss: 0.000
[epoch: 52, iter: 20]-loss: 0.000
Epoch 52: Accuracy on test set: 48.29 %
Epoch 52: Accuracy on Train set: 99.99 %
[epoch: 53, iter: 5]-loss: 0.000
[epoch: 53, iter: 10]-loss: 0.000
[epoch: 53, iter: 15]-loss: 0.000
[epoch: 53, iter: 20]-loss: 0.000
Epoch 53: Accuracy on test set: 47.59 %
Epoch 53: Accuracy on Train set: 99.98 %
[epoch: 54, iter: 5]-loss: 0.000
[epoch: 54, iter: 10]-loss: 0.000
[epoch: 54, iter: 15]-loss: 0.000
[epoch: 54, iter: 20]-loss: 0.000
Epoch 54: Accuracy on test set: 48.42 %
Epoch 54: Accuracy on Train set: 99.99 %
[epoch: 55, iter: 5]-loss: 0.000
[epoch: 55, iter: 10]-loss: 0.000
[epoch: 55, iter: 15]-loss: 0.000
[epoch: 55, iter: 20]-loss: 0.000
Epoch 55: Accuracy on test set: 48.58 %
Epoch 55: Accuracy on Train set: 99.97 %
[epoch: 56, iter: 5]-loss: 0.000
[epoch: 56, iter: 10]-loss: 0.000
[epoch: 56, iter: 15]-loss: 0.000
[epoch: 56, iter: 20]-loss: 0.000
Epoch 56: Accuracy on test set: 47.55 %
Epoch 56: Accuracy on Train set: 99.93 %
```

```
[epoch: 57, iter: 5]-loss: 0.000
[epoch: 57, iter: 10]-loss: 0.000
[epoch: 57, iter: 15]-loss: 0.000
[epoch: 57, iter: 20]-loss: 0.000
Epoch 57: Accuracy on test set: 48.69 %
Epoch 57: Accuracy on Train set: 100.00 %
[epoch: 58, iter: 5]-loss: 0.000
[epoch: 58, iter: 10]-loss: 0.000
[epoch: 58, iter: 15]-loss: 0.000
[epoch: 58, iter: 20]-loss: 0.000
Epoch 58: Accuracy on test set: 49.16 %
Epoch 58: Accuracy on Train set: 99.99 %
[epoch: 59, iter: 5]-loss: 0.000
[epoch: 59, iter: 10]-loss: 0.000
[epoch: 59, iter: 15]-loss: 0.000
[epoch: 59, iter: 20]-loss: 0.000
Epoch 59: Accuracy on test set: 49.29 %
Epoch 59: Accuracy on Train set: 99.99 %
[epoch: 60, iter: 5]-loss: 0.000
[epoch: 60, iter: 10]-loss: 0.000
[epoch: 60, iter: 15]-loss: 0.000
[epoch: 60, iter: 20]-loss: 0.000
Epoch 60: Accuracy on test set: 49.19 %
Epoch 60: Accuracy on Train set: 99.98 %
[epoch: 61, iter: 5]-loss: 0.000
[epoch: 61, iter: 10]-loss: 0.000
[epoch: 61, iter: 15]-loss: 0.000
[epoch: 61, iter: 20]-loss: 0.000
Epoch 61: Accuracy on test set: 49.30 %
Epoch 61: Accuracy on Train set: 99.98 %
[epoch: 62, iter: 5]-loss: 0.000
[epoch: 62, iter: 10]-loss: 0.000
[epoch: 62, iter: 15]-loss: 0.000
[epoch: 62, iter: 20]-loss: 0.000
Epoch 62: Accuracy on test set: 48.41 %
Epoch 62: Accuracy on Train set: 99.99 %
[epoch: 63, iter: 5]-loss: 0.000
[epoch: 63, iter: 10]-loss: 0.000
[epoch: 63, iter: 15]-loss: 0.000
[epoch: 63, iter: 20]-loss: 0.000
Epoch 63: Accuracy on test set: 49.21 %
Epoch 63: Accuracy on Train set: 100.00 %
[epoch: 64, iter: 5]-loss: 0.000
[epoch: 64, iter: 10]-loss: 0.000
[epoch: 64, iter: 15]-loss: 0.000
[epoch: 64, iter: 20]-loss: 0.000
Epoch 64: Accuracy on test set: 49.06 %
Epoch 64: Accuracy on Train set: 99.99 %
```

```
[epoch: 65, iter: 5]-loss: 0.000
[epoch: 65, iter: 10]-loss: 0.000
[epoch: 65, iter: 15]-loss: 0.000
[epoch: 65, iter: 20]-loss: 0.000
Epoch 65: Accuracy on test set: 48.78 %
Epoch 65: Accuracy on Train set: 99.98 %
[epoch: 66, iter: 5]-loss: 0.000
[epoch: 66, iter: 10]-loss: 0.000
[epoch: 66, iter: 15]-loss: 0.000
[epoch: 66, iter: 20]-loss: 0.000
Epoch 66: Accuracy on test set: 48.37 %
Epoch 66: Accuracy on Train set: 99.98 %
[epoch: 67, iter: 5]-loss: 0.000
[epoch: 67, iter: 10]-loss: 0.000
[epoch: 67, iter: 15]-loss: 0.000
[epoch: 67, iter: 20]-loss: 0.000
Epoch 67: Accuracy on test set: 48.64 %
Epoch 67: Accuracy on Train set: 99.99 %
[epoch: 68, iter: 5]-loss: 0.000
[epoch: 68, iter: 10]-loss: 0.000
[epoch: 68, iter: 15]-loss: 0.000
[epoch: 68, iter: 20]-loss: 0.000
Epoch 68: Accuracy on test set: 48.87 %
Epoch 68: Accuracy on Train set: 100.00 %
[epoch: 69, iter: 5]-loss: 0.000
[epoch: 69, iter: 10]-loss: 0.000
[epoch: 69, iter: 15]-loss: 0.000
[epoch: 69, iter: 20]-loss: 0.000
Epoch 69: Accuracy on test set: 48.80 %
Epoch 69: Accuracy on Train set: 100.00 %
[epoch: 70, iter: 5]-loss: 0.000
[epoch: 70, iter: 10]-loss: 0.000
[epoch: 70, iter: 15]-loss: 0.000
[epoch: 70, iter: 20]-loss: 0.000
Epoch 70: Accuracy on test set: 48.95 %
Epoch 70: Accuracy on Train set: 99.99 %
[epoch: 71, iter: 5]-loss: 0.000
[epoch: 71, iter: 10]-loss: 0.000
[epoch: 71, iter: 15]-loss: 0.000
[epoch: 71, iter: 20]-loss: 0.000
Epoch 71: Accuracy on test set: 48.68 %
Epoch 71: Accuracy on Train set: 100.00 %
[epoch: 72, iter: 5]-loss: 0.000
[epoch: 72, iter: 10]-loss: 0.000
[epoch: 72, iter: 15]-loss: 0.000
[epoch: 72, iter: 20]-loss: 0.000
Epoch 72: Accuracy on test set: 49.45 %
Epoch 72: Accuracy on Train set: 100.00 %
```

```
[epoch: 73, iter: 5]-loss: 0.000
[epoch: 73, iter: 10]-loss: 0.000
[epoch: 73, iter: 15]-loss: 0.000
[epoch: 73, iter: 20]-loss: 0.000
Epoch 73: Accuracy on test set: 48.85 %
Epoch 73: Accuracy on Train set: 100.00 %
[epoch: 74, iter: 5]-loss: 0.000
[epoch: 74, iter: 10]-loss: 0.000
[epoch: 74, iter: 15]-loss: 0.000
[epoch: 74, iter: 20]-loss: 0.000
Epoch 74: Accuracy on test set: 48.98 %
Epoch 74: Accuracy on Train set: 99.98 %
[epoch: 75, iter: 5]-loss: 0.000
[epoch: 75, iter: 10]-loss: 0.000
[epoch: 75, iter: 15]-loss: 0.000
[epoch: 75, iter: 20]-loss: 0.000
Epoch 75: Accuracy on test set: 48.85 %
Epoch 75: Accuracy on Train set: 99.99 %
[epoch: 76, iter: 5]-loss: 0.000
[epoch: 76, iter: 10]-loss: 0.000
[epoch: 76, iter: 15]-loss: 0.000
[epoch: 76, iter: 20]-loss: 0.000
Epoch 76: Accuracy on test set: 48.72 %
Epoch 76: Accuracy on Train set: 99.98 %
[epoch: 77, iter: 5]-loss: 0.000
[epoch: 77, iter: 10]-loss: 0.000
[epoch: 77, iter: 15]-loss: 0.000
[epoch: 77, iter: 20]-loss: 0.000
Epoch 77: Accuracy on test set: 49.39 %
Epoch 77: Accuracy on Train set: 99.99 %
[epoch: 78, iter: 5]-loss: 0.000
[epoch: 78, iter: 10]-loss: 0.000
[epoch: 78, iter: 15]-loss: 0.000
[epoch: 78, iter: 20]-loss: 0.000
Epoch 78: Accuracy on test set: 49.47 %
Epoch 78: Accuracy on Train set: 100.00 %
[epoch: 79, iter: 5]-loss: 0.000
[epoch: 79, iter: 10]-loss: 0.000
[epoch: 79, iter: 15]-loss: 0.000
[epoch: 79, iter: 20]-loss: 0.000
Epoch 79: Accuracy on test set: 49.71 %
Epoch 79: Accuracy on Train set: 99.99 %
[epoch: 80, iter: 5]-loss: 0.000
[epoch: 80, iter: 10]-loss: 0.000
[epoch: 80, iter: 15]-loss: 0.000
[epoch: 80, iter: 20]-loss: 0.000
Epoch 80: Accuracy on test set: 49.52 %
Epoch 80: Accuracy on Train set: 99.99 %
```

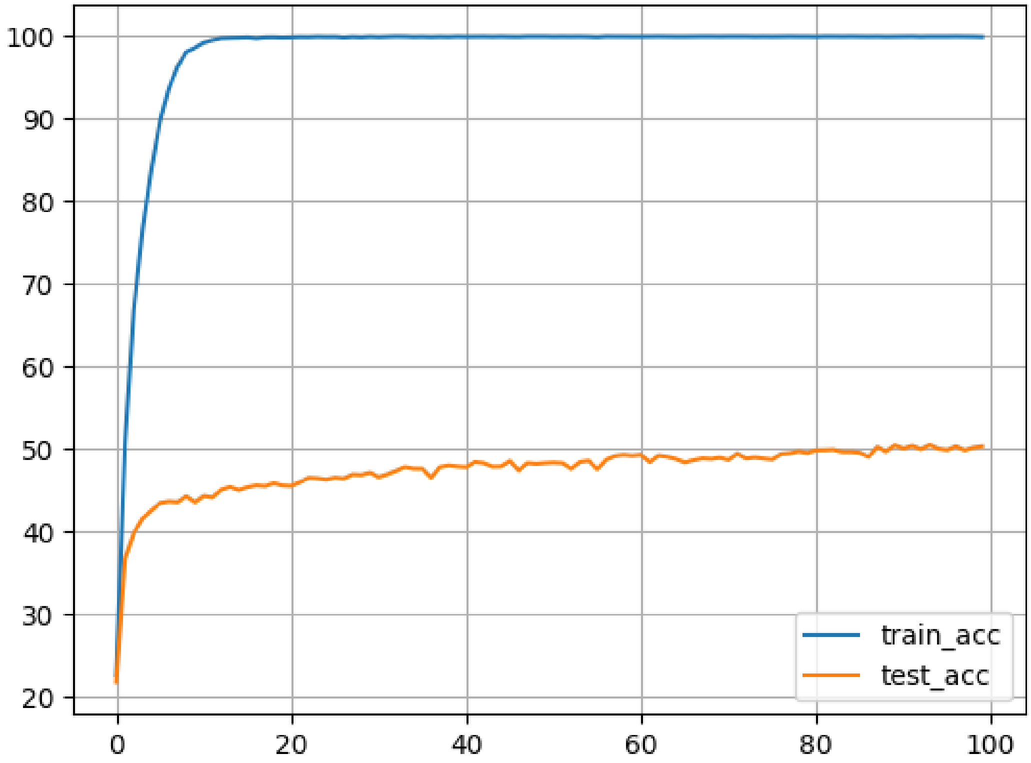
```
[epoch: 81, iter: 5]-loss: 0.000
[epoch: 81, iter: 10]-loss: 0.000
[epoch: 81, iter: 15]-loss: 0.000
[epoch: 81, iter: 20]-loss: 0.000
Epoch 81: Accuracy on test set: 49.84 %
Epoch 81: Accuracy on Train set: 99.97 %
[epoch: 82, iter: 5]-loss: 0.000
[epoch: 82, iter: 10]-loss: 0.000
[epoch: 82, iter: 15]-loss: 0.000
[epoch: 82, iter: 20]-loss: 0.000
Epoch 82: Accuracy on test set: 49.86 %
Epoch 82: Accuracy on Train set: 100.00 %
[epoch: 83, iter: 5]-loss: 0.000
[epoch: 83, iter: 10]-loss: 0.000
[epoch: 83, iter: 15]-loss: 0.000
[epoch: 83, iter: 20]-loss: 0.000
Epoch 83: Accuracy on test set: 49.91 %
Epoch 83: Accuracy on Train set: 100.00 %
[epoch: 84, iter: 5]-loss: 0.000
[epoch: 84, iter: 10]-loss: 0.000
[epoch: 84, iter: 15]-loss: 0.000
[epoch: 84, iter: 20]-loss: 0.000
Epoch 84: Accuracy on test set: 49.65 %
Epoch 84: Accuracy on Train set: 99.99 %
[epoch: 85, iter: 5]-loss: 0.000
[epoch: 85, iter: 10]-loss: 0.000
[epoch: 85, iter: 15]-loss: 0.000
[epoch: 85, iter: 20]-loss: 0.000
Epoch 85: Accuracy on test set: 49.65 %
Epoch 85: Accuracy on Train set: 100.00 %
[epoch: 86, iter: 5]-loss: 0.000
[epoch: 86, iter: 10]-loss: 0.000
[epoch: 86, iter: 15]-loss: 0.000
[epoch: 86, iter: 20]-loss: 0.000
Epoch 86: Accuracy on test set: 49.56 %
Epoch 86: Accuracy on Train set: 99.99 %
[epoch: 87, iter: 5]-loss: 0.000
[epoch: 87, iter: 10]-loss: 0.000
[epoch: 87, iter: 15]-loss: 0.000
[epoch: 87, iter: 20]-loss: 0.000
Epoch 87: Accuracy on test set: 49.04 %
Epoch 87: Accuracy on Train set: 99.98 %
[epoch: 88, iter: 5]-loss: 0.000
[epoch: 88, iter: 10]-loss: 0.000
[epoch: 88, iter: 15]-loss: 0.000
[epoch: 88, iter: 20]-loss: 0.000
Epoch 88: Accuracy on test set: 50.25 %
Epoch 88: Accuracy on Train set: 99.99 %
```

```
[epoch: 89, iter: 5]-loss: 0.000
[epoch: 89, iter: 10]-loss: 0.000
[epoch: 89, iter: 15]-loss: 0.000
[epoch: 89, iter: 20]-loss: 0.000
Epoch 89: Accuracy on test set: 49.70 %
Epoch 89: Accuracy on Train set: 99.97 %
[epoch: 90, iter: 5]-loss: 0.000
[epoch: 90, iter: 10]-loss: 0.000
[epoch: 90, iter: 15]-loss: 0.000
[epoch: 90, iter: 20]-loss: 0.000
Epoch 90: Accuracy on test set: 50.47 %
Epoch 90: Accuracy on Train set: 99.98 %
[epoch: 91, iter: 5]-loss: 0.000
[epoch: 91, iter: 10]-loss: 0.000
[epoch: 91, iter: 15]-loss: 0.000
[epoch: 91, iter: 20]-loss: 0.000
Epoch 91: Accuracy on test set: 50.06 %
Epoch 91: Accuracy on Train set: 99.99 %
[epoch: 92, iter: 5]-loss: 0.000
[epoch: 92, iter: 10]-loss: 0.000
[epoch: 92, iter: 15]-loss: 0.000
[epoch: 92, iter: 20]-loss: 0.000
Epoch 92: Accuracy on test set: 50.40 %
Epoch 92: Accuracy on Train set: 100.00 %
[epoch: 93, iter: 5]-loss: 0.000
[epoch: 93, iter: 10]-loss: 0.000
[epoch: 93, iter: 15]-loss: 0.000
[epoch: 93, iter: 20]-loss: 0.000
Epoch 93: Accuracy on test set: 49.98 %
Epoch 93: Accuracy on Train set: 99.97 %
[epoch: 94, iter: 5]-loss: 0.000
[epoch: 94, iter: 10]-loss: 0.000
[epoch: 94, iter: 15]-loss: 0.000
[epoch: 94, iter: 20]-loss: 0.000
Epoch 94: Accuracy on test set: 50.53 %
Epoch 94: Accuracy on Train set: 99.99 %
[epoch: 95, iter: 5]-loss: 0.000
[epoch: 95, iter: 10]-loss: 0.000
[epoch: 95, iter: 15]-loss: 0.000
[epoch: 95, iter: 20]-loss: 0.000
Epoch 95: Accuracy on test set: 50.04 %
Epoch 95: Accuracy on Train set: 99.99 %
[epoch: 96, iter: 5]-loss: 0.000
[epoch: 96, iter: 10]-loss: 0.000
[epoch: 96, iter: 15]-loss: 0.000
[epoch: 96, iter: 20]-loss: 0.000
Epoch 96: Accuracy on test set: 49.88 %
Epoch 96: Accuracy on Train set: 99.99 %
```

```
[epoch: 97, iter: 5]-loss: 0.000
[epoch: 97, iter: 10]-loss: 0.000
[epoch: 97, iter: 15]-loss: 0.000
[epoch: 97, iter: 20]-loss: 0.000
Epoch 97: Accuracy on test set: 50.35 %
Epoch 97: Accuracy on Train set: 100.00 %
[epoch: 98, iter: 5]-loss: 0.000
[epoch: 98, iter: 10]-loss: 0.000
[epoch: 98, iter: 15]-loss: 0.000
[epoch: 98, iter: 20]-loss: 0.000
Epoch 98: Accuracy on test set: 49.84 %
Epoch 98: Accuracy on Train set: 99.99 %
[epoch: 99, iter: 5]-loss: 0.000
[epoch: 99, iter: 10]-loss: 0.000
[epoch: 99, iter: 15]-loss: 0.000
[epoch: 99, iter: 20]-loss: 0.000
Epoch 99: Accuracy on test set: 50.19 %
Epoch 99: Accuracy on Train set: 99.98 %
[epoch: 100, iter: 5]-loss: 0.000
[epoch: 100, iter: 10]-loss: 0.000
[epoch: 100, iter: 15]-loss: 0.000
[epoch: 100, iter: 20]-loss: 0.000
Epoch 100: Accuracy on test set: 50.33 %
Epoch 100: Accuracy on Train set: 99.95 %
Training completed!
```

[24]:

```
plt.figure()
plt.plot(torch.arange(len(train_acc)), train_acc, label='train_acc')
plt.plot(torch.arange(len(test_acc)), test_acc, label='test_acc')
plt.grid()
plt.legend()
plt.show()
```



We can achieve upto 100% accuracy on Train set and 50% accuracy on Test set

0.0.9 Standard Augmentation

[25]:

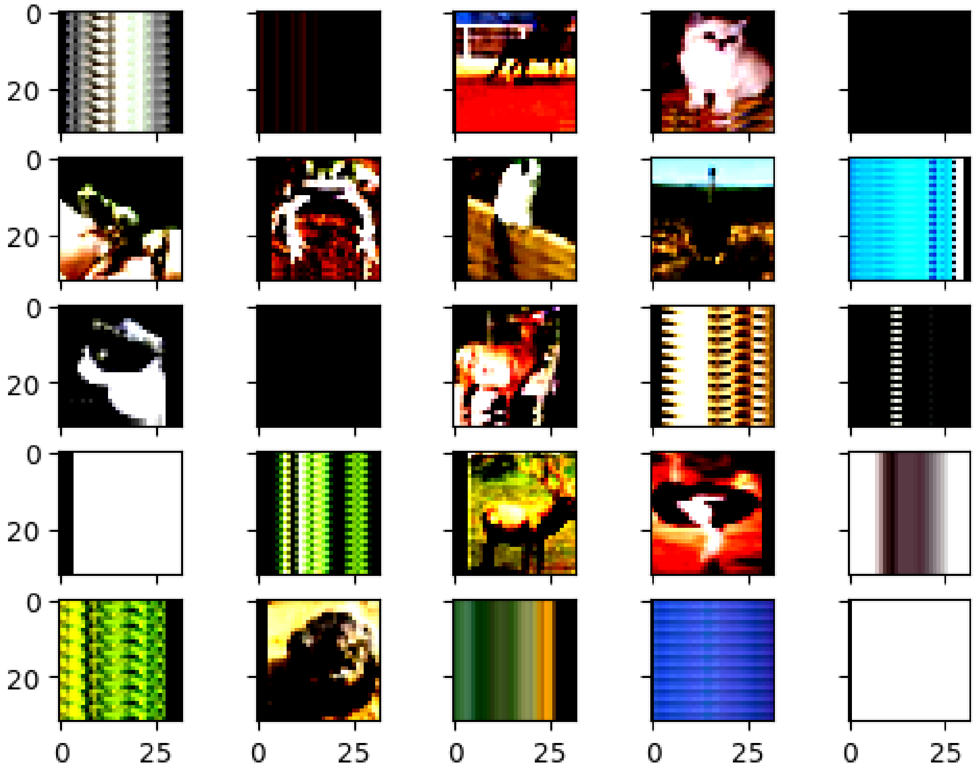
```
import torch
import random

def random_shift_and_flip(images, K):

    images = images.clone()

    batch_size, channels, height, width = images.size()
    shifted_images = torch.zeros_like(images)

    for i in range(batch_size):
        image = images[i]
        k1 = random.randint(-K, K)
        k2 = random.randint(-K, K)

        # Apply horizontal shift
        if k1 > 0:
            shifted_images[i, :, :, k1:] = image[:, :, :-k1]
        elif k1 < 0:
            shifted_images[i, :, :, :k1] = image[:, :, -k1:]
        else:
            shifted_images[i] = image

        # Apply vertical shift
        if k2 > 0:
            shifted_images[i, :, k2:, :] = shifted_images[i, :, :-k2, :]
        elif k2 < 0:
            shifted_images[i, :, :k2, :] = shifted_images[i, :, -k2:, :]

        # Apply horizontal flip with 50% probability
        for image in shifted_images:
            if random.random() < 0.5:
                image = torch.flip(image, [2])

    return shifted_images

K = 4  # Define the value of K

shifted_and_flipped_images = random_shift_and_flip(x_train[:100], K)
```

[26]:

```
# Visualize 1 sample
sample_idx_ctr = 0
num_rows = 5
num_cols = 5

fig, ax = plt.subplots(num_rows, num_cols, sharex=True, sharey=True)

for i in range(num_rows):
    for j in range(num_cols):

        img = shifted_and_flipped_images[sample_idx_ctr]
        ax[i,j].imshow(img.squeeze().permute(1,2,0))
        sample_idx_ctr += 1
```

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

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



[27]:

```python
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

# Define the ResNet-18 model
model = torchvision.models.resnet18(pretrained=False)
num_classes = 10

model.fc = nn.Linear(model.fc.in_features, num_classes)
model = model.to(device)

model.eval()
```

[27]:

```
ResNet(
  (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3),
bias=False)
  (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  (relu): ReLU(inplace=True)
  (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1,
ceil_mode=False)
  (layer1): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    )
    (1): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    )
  )
  (layer2): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=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)
      (downsample): Sequential(
        (0): Conv2d(64, 128, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
    )
    (1): BasicBlock(
      (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=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)
    )
  )
  (layer3): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=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)
      (downsample): Sequential(
        (0): Conv2d(128, 256, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
    )
    (1): BasicBlock(
      (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=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)
    )
  )
  (layer4): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=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)
      (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): BasicBlock(
        (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
        (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (relu): ReLU(inplace=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)
      )
    )
  (avgpool): AdaptiveAvgPool2d(output_size=(1, 1))
  (fc): Linear(in_features=512, out_features=10, bias=True)
)
```

[28]:

```python
# Define loss function and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.0001)

test_acc = []
train_acc = []

# Training loop
num_epochs = 100

for epoch in range(num_epochs):

    i = 0
    iter = 0
    batch_size = 500

    while i < x_train.shape[0]:
        model.train()

        inputs = x_train[i:i+batch_size].to(device)
        labels = y_train[i:i+batch_size].type(torch.LongTensor).to(device)
        labels_oh = y_train_oh[i:i+batch_size].to(device)

        inputs = random_shift_and_flip(inputs, 4)

        optimizer.zero_grad()

        outputs = model(inputs)
        loss = criterion(outputs, labels_oh)
        loss.backward()
        optimizer.step()

        i += batch_size
        iter += 1

        if (iter + 1) % 5 == 0:
            print('[epoch: %d, iter: %d]-loss: %.3f' % (epoch + 1, iter + 1,loss.item() / batch_size))

    # Evaluation on the test set
    model.eval()

    # Test Accuracy
    correct = 0
    total = 0
    with torch.no_grad():

        inputs = x_test.to(device)
        labels = y_test.type(torch.LongTensor).to(device)
        outputs = model(inputs)
        _, predicted = torch.max(outputs.data, 1)
        total += labels.size(0)
        correct += (predicted == labels).sum().item()

    accuracy = 100 * correct / total
    test_acc.append(accuracy)
    print('Epoch %d: Accuracy on test set: %.2f %%' % (epoch + 1, accuracy))

    # Train Accuracy
    correct = 0
    total = 0
    with torch.no_grad():

        inputs = x_train.to(device)
        labels = y_train.type(torch.LongTensor).to(device)
        outputs = model(inputs)
        _, predicted = torch.max(outputs.data, 1)
        total += labels.size(0)
        correct += (predicted == labels).sum().item()

    accuracy = 100 * correct / total
    train_acc.append(accuracy)
    print('Epoch %d: Accuracy on Train set: %.2f %%' % (epoch + 1, accuracy))

print('Training completed!')
```

```
[epoch: 1, iter: 5]-loss: 0.005
[epoch: 1, iter: 10]-loss: 0.004
[epoch: 1, iter: 15]-loss: 0.004
[epoch: 1, iter: 20]-loss: 0.004
Epoch 1: Accuracy on test set: 22.58 %
Epoch 1: Accuracy on Train set: 23.73 %
[epoch: 2, iter: 5]-loss: 0.004
[epoch: 2, iter: 10]-loss: 0.004
[epoch: 2, iter: 15]-loss: 0.004
[epoch: 2, iter: 20]-loss: 0.004
Epoch 2: Accuracy on test set: 35.52 %
Epoch 2: Accuracy on Train set: 37.23 %
[epoch: 3, iter: 5]-loss: 0.004
[epoch: 3, iter: 10]-loss: 0.004
[epoch: 3, iter: 15]-loss: 0.004
[epoch: 3, iter: 20]-loss: 0.003
Epoch 3: Accuracy on test set: 38.89 %
Epoch 3: Accuracy on Train set: 41.81 %
[epoch: 4, iter: 5]-loss: 0.003
[epoch: 4, iter: 10]-loss: 0.004
[epoch: 4, iter: 15]-loss: 0.003
[epoch: 4, iter: 20]-loss: 0.003
Epoch 4: Accuracy on test set: 41.83 %
Epoch 4: Accuracy on Train set: 44.89 %
[epoch: 5, iter: 5]-loss: 0.003
[epoch: 5, iter: 10]-loss: 0.003
[epoch: 5, iter: 15]-loss: 0.003
[epoch: 5, iter: 20]-loss: 0.003
Epoch 5: Accuracy on test set: 42.90 %
Epoch 5: Accuracy on Train set: 46.95 %
[epoch: 6, iter: 5]-loss: 0.003
[epoch: 6, iter: 10]-loss: 0.003
[epoch: 6, iter: 15]-loss: 0.003
[epoch: 6, iter: 20]-loss: 0.003
Epoch 6: Accuracy on test set: 44.30 %
Epoch 6: Accuracy on Train set: 48.93 %
[epoch: 7, iter: 5]-loss: 0.003
[epoch: 7, iter: 10]-loss: 0.003
[epoch: 7, iter: 15]-loss: 0.003
[epoch: 7, iter: 20]-loss: 0.003
Epoch 7: Accuracy on test set: 45.05 %
Epoch 7: Accuracy on Train set: 50.78 %
[epoch: 8, iter: 5]-loss: 0.003
[epoch: 8, iter: 10]-loss: 0.003
[epoch: 8, iter: 15]-loss: 0.003
[epoch: 8, iter: 20]-loss: 0.003
Epoch 8: Accuracy on test set: 46.29 %
Epoch 8: Accuracy on Train set: 52.44 %
```

```
[epoch: 9, iter: 5]-loss: 0.003
[epoch: 9, iter: 10]-loss: 0.003
[epoch: 9, iter: 15]-loss: 0.003
[epoch: 9, iter: 20]-loss: 0.003
Epoch 9: Accuracy on test set: 47.02 %
Epoch 9: Accuracy on Train set: 54.14 %
[epoch: 10, iter: 5]-loss: 0.003
[epoch: 10, iter: 10]-loss: 0.003
[epoch: 10, iter: 15]-loss: 0.003
[epoch: 10, iter: 20]-loss: 0.003
Epoch 10: Accuracy on test set: 47.44 %
Epoch 10: Accuracy on Train set: 55.99 %
[epoch: 11, iter: 5]-loss: 0.003
[epoch: 11, iter: 10]-loss: 0.003
[epoch: 11, iter: 15]-loss: 0.003
[epoch: 11, iter: 20]-loss: 0.003
Epoch 11: Accuracy on test set: 48.18 %
Epoch 11: Accuracy on Train set: 56.53 %
[epoch: 12, iter: 5]-loss: 0.003
[epoch: 12, iter: 10]-loss: 0.003
[epoch: 12, iter: 15]-loss: 0.003
[epoch: 12, iter: 20]-loss: 0.003
Epoch 12: Accuracy on test set: 49.49 %
Epoch 12: Accuracy on Train set: 59.29 %
[epoch: 13, iter: 5]-loss: 0.003
[epoch: 13, iter: 10]-loss: 0.003
[epoch: 13, iter: 15]-loss: 0.003
[epoch: 13, iter: 20]-loss: 0.003
Epoch 13: Accuracy on test set: 49.78 %
Epoch 13: Accuracy on Train set: 59.72 %
[epoch: 14, iter: 5]-loss: 0.003
[epoch: 14, iter: 10]-loss: 0.003
[epoch: 14, iter: 15]-loss: 0.003
[epoch: 14, iter: 20]-loss: 0.003
Epoch 14: Accuracy on test set: 50.23 %
Epoch 14: Accuracy on Train set: 61.43 %
[epoch: 15, iter: 5]-loss: 0.003
[epoch: 15, iter: 10]-loss: 0.003
[epoch: 15, iter: 15]-loss: 0.003
[epoch: 15, iter: 20]-loss: 0.002
Epoch 15: Accuracy on test set: 49.74 %
Epoch 15: Accuracy on Train set: 61.87 %
[epoch: 16, iter: 5]-loss: 0.002
[epoch: 16, iter: 10]-loss: 0.003
[epoch: 16, iter: 15]-loss: 0.003
[epoch: 16, iter: 20]-loss: 0.003
Epoch 16: Accuracy on test set: 50.38 %
Epoch 16: Accuracy on Train set: 63.61 %
```

```
[epoch: 17, iter: 5]-loss: 0.003
[epoch: 17, iter: 10]-loss: 0.003
[epoch: 17, iter: 15]-loss: 0.003
[epoch: 17, iter: 20]-loss: 0.002
Epoch 17: Accuracy on test set: 51.16 %
Epoch 17: Accuracy on Train set: 64.76 %
[epoch: 18, iter: 5]-loss: 0.002
[epoch: 18, iter: 10]-loss: 0.002
[epoch: 18, iter: 15]-loss: 0.003
[epoch: 18, iter: 20]-loss: 0.002
Epoch 18: Accuracy on test set: 51.50 %
Epoch 18: Accuracy on Train set: 65.71 %
[epoch: 19, iter: 5]-loss: 0.003
[epoch: 19, iter: 10]-loss: 0.003
[epoch: 19, iter: 15]-loss: 0.003
[epoch: 19, iter: 20]-loss: 0.002
Epoch 19: Accuracy on test set: 51.74 %
Epoch 19: Accuracy on Train set: 67.04 %
[epoch: 20, iter: 5]-loss: 0.002
[epoch: 20, iter: 10]-loss: 0.003
[epoch: 20, iter: 15]-loss: 0.002
[epoch: 20, iter: 20]-loss: 0.002
Epoch 20: Accuracy on test set: 51.56 %
Epoch 20: Accuracy on Train set: 68.10 %
[epoch: 21, iter: 5]-loss: 0.002
[epoch: 21, iter: 10]-loss: 0.002
[epoch: 21, iter: 15]-loss: 0.002
[epoch: 21, iter: 20]-loss: 0.002
Epoch 21: Accuracy on test set: 52.06 %
Epoch 21: Accuracy on Train set: 68.84 %
[epoch: 22, iter: 5]-loss: 0.002
[epoch: 22, iter: 10]-loss: 0.002
[epoch: 22, iter: 15]-loss: 0.002
[epoch: 22, iter: 20]-loss: 0.002
Epoch 22: Accuracy on test set: 52.29 %
Epoch 22: Accuracy on Train set: 69.85 %
[epoch: 23, iter: 5]-loss: 0.002
[epoch: 23, iter: 10]-loss: 0.002
[epoch: 23, iter: 15]-loss: 0.002
[epoch: 23, iter: 20]-loss: 0.002
Epoch 23: Accuracy on test set: 52.92 %
Epoch 23: Accuracy on Train set: 71.33 %
[epoch: 24, iter: 5]-loss: 0.002
[epoch: 24, iter: 10]-loss: 0.002
[epoch: 24, iter: 15]-loss: 0.002
[epoch: 24, iter: 20]-loss: 0.002
Epoch 24: Accuracy on test set: 52.70 %
Epoch 24: Accuracy on Train set: 72.48 %
```

```
[epoch: 25, iter: 5]-loss: 0.002
[epoch: 25, iter: 10]-loss: 0.002
[epoch: 25, iter: 15]-loss: 0.002
[epoch: 25, iter: 20]-loss: 0.002
Epoch 25: Accuracy on test set: 53.03 %
Epoch 25: Accuracy on Train set: 73.06 %
[epoch: 26, iter: 5]-loss: 0.002
[epoch: 26, iter: 10]-loss: 0.002
[epoch: 26, iter: 15]-loss: 0.002
[epoch: 26, iter: 20]-loss: 0.002
Epoch 26: Accuracy on test set: 53.12 %
Epoch 26: Accuracy on Train set: 74.30 %
[epoch: 27, iter: 5]-loss: 0.002
[epoch: 27, iter: 10]-loss: 0.002
[epoch: 27, iter: 15]-loss: 0.002
[epoch: 27, iter: 20]-loss: 0.002
Epoch 27: Accuracy on test set: 53.67 %
Epoch 27: Accuracy on Train set: 75.29 %
[epoch: 28, iter: 5]-loss: 0.002
[epoch: 28, iter: 10]-loss: 0.002
[epoch: 28, iter: 15]-loss: 0.002
[epoch: 28, iter: 20]-loss: 0.002
Epoch 28: Accuracy on test set: 52.98 %
Epoch 28: Accuracy on Train set: 75.98 %
[epoch: 29, iter: 5]-loss: 0.002
[epoch: 29, iter: 10]-loss: 0.002
[epoch: 29, iter: 15]-loss: 0.002
[epoch: 29, iter: 20]-loss: 0.002
Epoch 29: Accuracy on test set: 53.32 %
Epoch 29: Accuracy on Train set: 77.11 %
[epoch: 30, iter: 5]-loss: 0.002
[epoch: 30, iter: 10]-loss: 0.002
[epoch: 30, iter: 15]-loss: 0.002
[epoch: 30, iter: 20]-loss: 0.002
Epoch 30: Accuracy on test set: 53.90 %
Epoch 30: Accuracy on Train set: 79.02 %
[epoch: 31, iter: 5]-loss: 0.002
[epoch: 31, iter: 10]-loss: 0.002
[epoch: 31, iter: 15]-loss: 0.002
[epoch: 31, iter: 20]-loss: 0.002
Epoch 31: Accuracy on test set: 54.41 %
Epoch 31: Accuracy on Train set: 78.69 %
[epoch: 32, iter: 5]-loss: 0.002
[epoch: 32, iter: 10]-loss: 0.002
[epoch: 32, iter: 15]-loss: 0.002
[epoch: 32, iter: 20]-loss: 0.002
Epoch 32: Accuracy on test set: 53.75 %
Epoch 32: Accuracy on Train set: 80.22 %
```

```
[epoch: 33, iter: 5]-loss: 0.002
[epoch: 33, iter: 10]-loss: 0.002
[epoch: 33, iter: 15]-loss: 0.002
[epoch: 33, iter: 20]-loss: 0.002
Epoch 33: Accuracy on test set: 54.25 %
Epoch 33: Accuracy on Train set: 81.05 %
[epoch: 34, iter: 5]-loss: 0.002
[epoch: 34, iter: 10]-loss: 0.002
[epoch: 34, iter: 15]-loss: 0.002
[epoch: 34, iter: 20]-loss: 0.002
Epoch 34: Accuracy on test set: 53.78 %
Epoch 34: Accuracy on Train set: 81.28 %
[epoch: 35, iter: 5]-loss: 0.002
[epoch: 35, iter: 10]-loss: 0.002
[epoch: 35, iter: 15]-loss: 0.002
[epoch: 35, iter: 20]-loss: 0.002
Epoch 35: Accuracy on test set: 54.61 %
Epoch 35: Accuracy on Train set: 82.23 %
[epoch: 36, iter: 5]-loss: 0.002
[epoch: 36, iter: 10]-loss: 0.002
[epoch: 36, iter: 15]-loss: 0.002
[epoch: 36, iter: 20]-loss: 0.002
Epoch 36: Accuracy on test set: 54.41 %
Epoch 36: Accuracy on Train set: 82.34 %
[epoch: 37, iter: 5]-loss: 0.002
[epoch: 37, iter: 10]-loss: 0.002
[epoch: 37, iter: 15]-loss: 0.002
[epoch: 37, iter: 20]-loss: 0.001
Epoch 37: Accuracy on test set: 54.58 %
Epoch 37: Accuracy on Train set: 84.17 %
[epoch: 38, iter: 5]-loss: 0.002
[epoch: 38, iter: 10]-loss: 0.002
[epoch: 38, iter: 15]-loss: 0.002
[epoch: 38, iter: 20]-loss: 0.002
Epoch 38: Accuracy on test set: 54.50 %
Epoch 38: Accuracy on Train set: 84.26 %
[epoch: 39, iter: 5]-loss: 0.002
[epoch: 39, iter: 10]-loss: 0.002
[epoch: 39, iter: 15]-loss: 0.002
[epoch: 39, iter: 20]-loss: 0.002
Epoch 39: Accuracy on test set: 54.17 %
Epoch 39: Accuracy on Train set: 84.80 %
[epoch: 40, iter: 5]-loss: 0.002
[epoch: 40, iter: 10]-loss: 0.002
[epoch: 40, iter: 15]-loss: 0.001
[epoch: 40, iter: 20]-loss: 0.002
Epoch 40: Accuracy on test set: 54.20 %
Epoch 40: Accuracy on Train set: 85.67 %
```

```
[epoch: 41, iter: 5]-loss: 0.002
[epoch: 41, iter: 10]-loss: 0.002
[epoch: 41, iter: 15]-loss: 0.002
[epoch: 41, iter: 20]-loss: 0.002
Epoch 41: Accuracy on test set: 54.60 %
Epoch 41: Accuracy on Train set: 87.03 %
[epoch: 42, iter: 5]-loss: 0.001
[epoch: 42, iter: 10]-loss: 0.002
[epoch: 42, iter: 15]-loss: 0.002
[epoch: 42, iter: 20]-loss: 0.001
Epoch 42: Accuracy on test set: 54.77 %
Epoch 42: Accuracy on Train set: 87.57 %
[epoch: 43, iter: 5]-loss: 0.001
[epoch: 43, iter: 10]-loss: 0.002
[epoch: 43, iter: 15]-loss: 0.002
[epoch: 43, iter: 20]-loss: 0.002
Epoch 43: Accuracy on test set: 54.17 %
Epoch 43: Accuracy on Train set: 88.08 %
[epoch: 44, iter: 5]-loss: 0.001
[epoch: 44, iter: 10]-loss: 0.001
[epoch: 44, iter: 15]-loss: 0.002
[epoch: 44, iter: 20]-loss: 0.001
Epoch 44: Accuracy on test set: 54.32 %
Epoch 44: Accuracy on Train set: 89.13 %
[epoch: 45, iter: 5]-loss: 0.001
[epoch: 45, iter: 10]-loss: 0.002
[epoch: 45, iter: 15]-loss: 0.001
[epoch: 45, iter: 20]-loss: 0.001
Epoch 45: Accuracy on test set: 54.39 %
Epoch 45: Accuracy on Train set: 89.08 %
[epoch: 46, iter: 5]-loss: 0.001
[epoch: 46, iter: 10]-loss: 0.002
[epoch: 46, iter: 15]-loss: 0.002
[epoch: 46, iter: 20]-loss: 0.001
Epoch 46: Accuracy on test set: 53.85 %
Epoch 46: Accuracy on Train set: 89.72 %
[epoch: 47, iter: 5]-loss: 0.001
[epoch: 47, iter: 10]-loss: 0.001
[epoch: 47, iter: 15]-loss: 0.001
[epoch: 47, iter: 20]-loss: 0.001
Epoch 47: Accuracy on test set: 54.59 %
Epoch 47: Accuracy on Train set: 89.55 %
[epoch: 48, iter: 5]-loss: 0.001
[epoch: 48, iter: 10]-loss: 0.001
[epoch: 48, iter: 15]-loss: 0.001
[epoch: 48, iter: 20]-loss: 0.001
Epoch 48: Accuracy on test set: 54.78 %
Epoch 48: Accuracy on Train set: 90.54 %
```

```
[epoch: 49, iter: 5]-loss: 0.001
[epoch: 49, iter: 10]-loss: 0.001
[epoch: 49, iter: 15]-loss: 0.001
[epoch: 49, iter: 20]-loss: 0.001
Epoch 49: Accuracy on test set: 54.36 %
Epoch 49: Accuracy on Train set: 90.99 %
[epoch: 50, iter: 5]-loss: 0.001
[epoch: 50, iter: 10]-loss: 0.001
[epoch: 50, iter: 15]-loss: 0.001
[epoch: 50, iter: 20]-loss: 0.001
Epoch 50: Accuracy on test set: 55.21 %
Epoch 50: Accuracy on Train set: 91.82 %
[epoch: 51, iter: 5]-loss: 0.001
[epoch: 51, iter: 10]-loss: 0.001
[epoch: 51, iter: 15]-loss: 0.001
[epoch: 51, iter: 20]-loss: 0.001
Epoch 51: Accuracy on test set: 54.75 %
Epoch 51: Accuracy on Train set: 92.04 %
[epoch: 52, iter: 5]-loss: 0.001
[epoch: 52, iter: 10]-loss: 0.001
[epoch: 52, iter: 15]-loss: 0.001
[epoch: 52, iter: 20]-loss: 0.001
Epoch 52: Accuracy on test set: 54.77 %
Epoch 52: Accuracy on Train set: 92.69 %
[epoch: 53, iter: 5]-loss: 0.001
[epoch: 53, iter: 10]-loss: 0.001
[epoch: 53, iter: 15]-loss: 0.001
[epoch: 53, iter: 20]-loss: 0.001
Epoch 53: Accuracy on test set: 54.50 %
Epoch 53: Accuracy on Train set: 92.51 %
[epoch: 54, iter: 5]-loss: 0.001
[epoch: 54, iter: 10]-loss: 0.001
[epoch: 54, iter: 15]-loss: 0.001
[epoch: 54, iter: 20]-loss: 0.001
Epoch 54: Accuracy on test set: 54.56 %
Epoch 54: Accuracy on Train set: 93.34 %
[epoch: 55, iter: 5]-loss: 0.001
[epoch: 55, iter: 10]-loss: 0.001
[epoch: 55, iter: 15]-loss: 0.001
[epoch: 55, iter: 20]-loss: 0.001
Epoch 55: Accuracy on test set: 55.03 %
Epoch 55: Accuracy on Train set: 93.42 %
[epoch: 56, iter: 5]-loss: 0.001
[epoch: 56, iter: 10]-loss: 0.001
[epoch: 56, iter: 15]-loss: 0.001
[epoch: 56, iter: 20]-loss: 0.001
Epoch 56: Accuracy on test set: 54.92 %
Epoch 56: Accuracy on Train set: 93.68 %
```

```
[epoch: 57, iter: 5]-loss: 0.001
[epoch: 57, iter: 10]-loss: 0.001
[epoch: 57, iter: 15]-loss: 0.001
[epoch: 57, iter: 20]-loss: 0.001
Epoch 57: Accuracy on test set: 55.00 %
Epoch 57: Accuracy on Train set: 93.91 %
[epoch: 58, iter: 5]-loss: 0.001
[epoch: 58, iter: 10]-loss: 0.001
[epoch: 58, iter: 15]-loss: 0.001
[epoch: 58, iter: 20]-loss: 0.001
Epoch 58: Accuracy on test set: 54.95 %
Epoch 58: Accuracy on Train set: 94.27 %
[epoch: 59, iter: 5]-loss: 0.001
[epoch: 59, iter: 10]-loss: 0.001
[epoch: 59, iter: 15]-loss: 0.001
[epoch: 59, iter: 20]-loss: 0.001
Epoch 59: Accuracy on test set: 54.91 %
Epoch 59: Accuracy on Train set: 94.52 %
[epoch: 60, iter: 5]-loss: 0.001
[epoch: 60, iter: 10]-loss: 0.001
[epoch: 60, iter: 15]-loss: 0.001
[epoch: 60, iter: 20]-loss: 0.001
Epoch 60: Accuracy on test set: 54.66 %
Epoch 60: Accuracy on Train set: 95.17 %
[epoch: 61, iter: 5]-loss: 0.001
[epoch: 61, iter: 10]-loss: 0.001
[epoch: 61, iter: 15]-loss: 0.001
[epoch: 61, iter: 20]-loss: 0.001
Epoch 61: Accuracy on test set: 55.07 %
Epoch 61: Accuracy on Train set: 95.15 %
[epoch: 62, iter: 5]-loss: 0.001
[epoch: 62, iter: 10]-loss: 0.001
[epoch: 62, iter: 15]-loss: 0.001
[epoch: 62, iter: 20]-loss: 0.001
Epoch 62: Accuracy on test set: 54.33 %
Epoch 62: Accuracy on Train set: 95.58 %
[epoch: 63, iter: 5]-loss: 0.001
[epoch: 63, iter: 10]-loss: 0.001
[epoch: 63, iter: 15]-loss: 0.001
[epoch: 63, iter: 20]-loss: 0.001
Epoch 63: Accuracy on test set: 55.01 %
Epoch 63: Accuracy on Train set: 95.30 %
[epoch: 64, iter: 5]-loss: 0.001
[epoch: 64, iter: 10]-loss: 0.001
[epoch: 64, iter: 15]-loss: 0.001
[epoch: 64, iter: 20]-loss: 0.001
Epoch 64: Accuracy on test set: 55.04 %
Epoch 64: Accuracy on Train set: 95.62 %
```

```
[epoch: 65, iter: 5]-loss: 0.001
[epoch: 65, iter: 10]-loss: 0.001
[epoch: 65, iter: 15]-loss: 0.001
[epoch: 65, iter: 20]-loss: 0.001
Epoch 65: Accuracy on test set: 54.85 %
Epoch 65: Accuracy on Train set: 95.66 %
[epoch: 66, iter: 5]-loss: 0.001
[epoch: 66, iter: 10]-loss: 0.001
[epoch: 66, iter: 15]-loss: 0.001
[epoch: 66, iter: 20]-loss: 0.001
Epoch 66: Accuracy on test set: 55.10 %
Epoch 66: Accuracy on Train set: 95.86 %
[epoch: 67, iter: 5]-loss: 0.001
[epoch: 67, iter: 10]-loss: 0.001
[epoch: 67, iter: 15]-loss: 0.001
[epoch: 67, iter: 20]-loss: 0.001
Epoch 67: Accuracy on test set: 55.32 %
Epoch 67: Accuracy on Train set: 96.92 %
[epoch: 68, iter: 5]-loss: 0.001
[epoch: 68, iter: 10]-loss: 0.001
[epoch: 68, iter: 15]-loss: 0.001
[epoch: 68, iter: 20]-loss: 0.001
Epoch 68: Accuracy on test set: 54.75 %
Epoch 68: Accuracy on Train set: 96.23 %
[epoch: 69, iter: 5]-loss: 0.001
[epoch: 69, iter: 10]-loss: 0.001
[epoch: 69, iter: 15]-loss: 0.001
[epoch: 69, iter: 20]-loss: 0.001
Epoch 69: Accuracy on test set: 55.25 %
Epoch 69: Accuracy on Train set: 96.48 %
[epoch: 70, iter: 5]-loss: 0.001
[epoch: 70, iter: 10]-loss: 0.001
[epoch: 70, iter: 15]-loss: 0.001
[epoch: 70, iter: 20]-loss: 0.001
Epoch 70: Accuracy on test set: 55.54 %
Epoch 70: Accuracy on Train set: 96.22 %
[epoch: 71, iter: 5]-loss: 0.001
[epoch: 71, iter: 10]-loss: 0.001
[epoch: 71, iter: 15]-loss: 0.001
[epoch: 71, iter: 20]-loss: 0.001
Epoch 71: Accuracy on test set: 54.81 %
Epoch 71: Accuracy on Train set: 97.04 %
[epoch: 72, iter: 5]-loss: 0.001
[epoch: 72, iter: 10]-loss: 0.001
[epoch: 72, iter: 15]-loss: 0.001
[epoch: 72, iter: 20]-loss: 0.001
Epoch 72: Accuracy on test set: 55.57 %
Epoch 72: Accuracy on Train set: 96.86 %
```

```
[epoch: 73, iter: 5]-loss: 0.001
[epoch: 73, iter: 10]-loss: 0.001
[epoch: 73, iter: 15]-loss: 0.001
[epoch: 73, iter: 20]-loss: 0.001
Epoch 73: Accuracy on test set: 55.57 %
Epoch 73: Accuracy on Train set: 97.64 %
[epoch: 74, iter: 5]-loss: 0.001
[epoch: 74, iter: 10]-loss: 0.001
[epoch: 74, iter: 15]-loss: 0.001
[epoch: 74, iter: 20]-loss: 0.001
Epoch 74: Accuracy on test set: 55.49 %
Epoch 74: Accuracy on Train set: 97.84 %
[epoch: 75, iter: 5]-loss: 0.001
[epoch: 75, iter: 10]-loss: 0.001
[epoch: 75, iter: 15]-loss: 0.001
[epoch: 75, iter: 20]-loss: 0.001
Epoch 75: Accuracy on test set: 55.28 %
Epoch 75: Accuracy on Train set: 97.49 %
[epoch: 76, iter: 5]-loss: 0.001
[epoch: 76, iter: 10]-loss: 0.001
[epoch: 76, iter: 15]-loss: 0.001
[epoch: 76, iter: 20]-loss: 0.001
Epoch 76: Accuracy on test set: 55.73 %
Epoch 76: Accuracy on Train set: 97.22 %
[epoch: 77, iter: 5]-loss: 0.001
[epoch: 77, iter: 10]-loss: 0.001
[epoch: 77, iter: 15]-loss: 0.001
[epoch: 77, iter: 20]-loss: 0.001
Epoch 77: Accuracy on test set: 55.40 %
Epoch 77: Accuracy on Train set: 97.07 %
[epoch: 78, iter: 5]-loss: 0.001
[epoch: 78, iter: 10]-loss: 0.001
[epoch: 78, iter: 15]-loss: 0.001
[epoch: 78, iter: 20]-loss: 0.001
Epoch 78: Accuracy on test set: 55.65 %
Epoch 78: Accuracy on Train set: 97.91 %
[epoch: 79, iter: 5]-loss: 0.000
[epoch: 79, iter: 10]-loss: 0.001
[epoch: 79, iter: 15]-loss: 0.001
[epoch: 79, iter: 20]-loss: 0.001
Epoch 79: Accuracy on test set: 55.39 %
Epoch 79: Accuracy on Train set: 97.91 %
[epoch: 80, iter: 5]-loss: 0.001
[epoch: 80, iter: 10]-loss: 0.001
[epoch: 80, iter: 15]-loss: 0.001
[epoch: 80, iter: 20]-loss: 0.001
Epoch 80: Accuracy on test set: 53.93 %
Epoch 80: Accuracy on Train set: 96.27 %
```

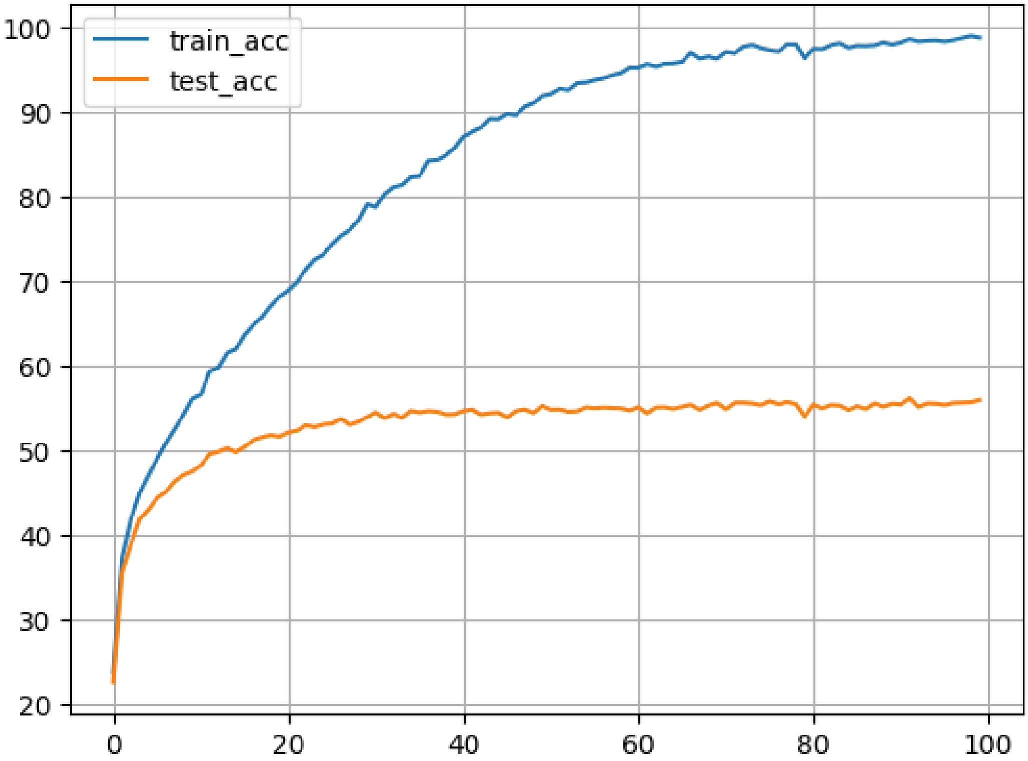
```
[epoch: 81, iter: 5]-loss: 0.001
[epoch: 81, iter: 10]-loss: 0.001
[epoch: 81, iter: 15]-loss: 0.001
[epoch: 81, iter: 20]-loss: 0.001
Epoch 81: Accuracy on test set: 55.35 %
Epoch 81: Accuracy on Train set: 97.38 %
[epoch: 82, iter: 5]-loss: 0.001
[epoch: 82, iter: 10]-loss: 0.001
[epoch: 82, iter: 15]-loss: 0.001
[epoch: 82, iter: 20]-loss: 0.001
Epoch 82: Accuracy on test set: 54.91 %
Epoch 82: Accuracy on Train set: 97.34 %
[epoch: 83, iter: 5]-loss: 0.001
[epoch: 83, iter: 10]-loss: 0.001
[epoch: 83, iter: 15]-loss: 0.001
[epoch: 83, iter: 20]-loss: 0.001
Epoch 83: Accuracy on test set: 55.30 %
Epoch 83: Accuracy on Train set: 97.82 %
[epoch: 84, iter: 5]-loss: 0.001
[epoch: 84, iter: 10]-loss: 0.001
[epoch: 84, iter: 15]-loss: 0.001
[epoch: 84, iter: 20]-loss: 0.001
Epoch 84: Accuracy on test set: 55.23 %
Epoch 84: Accuracy on Train set: 98.04 %
[epoch: 85, iter: 5]-loss: 0.001
[epoch: 85, iter: 10]-loss: 0.001
[epoch: 85, iter: 15]-loss: 0.001
[epoch: 85, iter: 20]-loss: 0.000
Epoch 85: Accuracy on test set: 54.72 %
Epoch 85: Accuracy on Train set: 97.51 %
[epoch: 86, iter: 5]-loss: 0.001
[epoch: 86, iter: 10]-loss: 0.001
[epoch: 86, iter: 15]-loss: 0.000
[epoch: 86, iter: 20]-loss: 0.000
Epoch 86: Accuracy on test set: 55.16 %
Epoch 86: Accuracy on Train set: 97.75 %
[epoch: 87, iter: 5]-loss: 0.001
[epoch: 87, iter: 10]-loss: 0.001
[epoch: 87, iter: 15]-loss: 0.001
[epoch: 87, iter: 20]-loss: 0.001
Epoch 87: Accuracy on test set: 54.84 %
Epoch 87: Accuracy on Train set: 97.72 %
[epoch: 88, iter: 5]-loss: 0.000
[epoch: 88, iter: 10]-loss: 0.001
[epoch: 88, iter: 15]-loss: 0.001
[epoch: 88, iter: 20]-loss: 0.001
Epoch 88: Accuracy on test set: 55.49 %
Epoch 88: Accuracy on Train set: 97.80 %
```

```
[epoch: 89, iter: 5]-loss: 0.000
[epoch: 89, iter: 10]-loss: 0.000
[epoch: 89, iter: 15]-loss: 0.001
[epoch: 89, iter: 20]-loss: 0.000
Epoch 89: Accuracy on test set: 55.10 %
Epoch 89: Accuracy on Train set: 98.14 %
[epoch: 90, iter: 5]-loss: 0.000
[epoch: 90, iter: 10]-loss: 0.000
[epoch: 90, iter: 15]-loss: 0.001
[epoch: 90, iter: 20]-loss: 0.000
Epoch 90: Accuracy on test set: 55.45 %
Epoch 90: Accuracy on Train set: 97.88 %
[epoch: 91, iter: 5]-loss: 0.000
[epoch: 91, iter: 10]-loss: 0.001
[epoch: 91, iter: 15]-loss: 0.001
[epoch: 91, iter: 20]-loss: 0.000
Epoch 91: Accuracy on test set: 55.37 %
Epoch 91: Accuracy on Train set: 98.13 %
[epoch: 92, iter: 5]-loss: 0.001
[epoch: 92, iter: 10]-loss: 0.000
[epoch: 92, iter: 15]-loss: 0.000
[epoch: 92, iter: 20]-loss: 0.001
Epoch 92: Accuracy on test set: 56.08 %
Epoch 92: Accuracy on Train set: 98.55 %
[epoch: 93, iter: 5]-loss: 0.001
[epoch: 93, iter: 10]-loss: 0.000
[epoch: 93, iter: 15]-loss: 0.001
[epoch: 93, iter: 20]-loss: 0.000
Epoch 93: Accuracy on test set: 55.07 %
Epoch 93: Accuracy on Train set: 98.26 %
[epoch: 94, iter: 5]-loss: 0.000
[epoch: 94, iter: 10]-loss: 0.000
[epoch: 94, iter: 15]-loss: 0.001
[epoch: 94, iter: 20]-loss: 0.000
Epoch 94: Accuracy on test set: 55.49 %
Epoch 94: Accuracy on Train set: 98.36 %
[epoch: 95, iter: 5]-loss: 0.000
[epoch: 95, iter: 10]-loss: 0.000
[epoch: 95, iter: 15]-loss: 0.000
[epoch: 95, iter: 20]-loss: 0.000
Epoch 95: Accuracy on test set: 55.45 %
Epoch 95: Accuracy on Train set: 98.39 %
[epoch: 96, iter: 5]-loss: 0.000
[epoch: 96, iter: 10]-loss: 0.000
[epoch: 96, iter: 15]-loss: 0.000
[epoch: 96, iter: 20]-loss: 0.000
Epoch 96: Accuracy on test set: 55.31 %
Epoch 96: Accuracy on Train set: 98.26 %
```

```
[epoch: 97, iter: 5]-loss: 0.000
[epoch: 97, iter: 10]-loss: 0.000
[epoch: 97, iter: 15]-loss: 0.000
[epoch: 97, iter: 20]-loss: 0.000
Epoch 97: Accuracy on test set: 55.53 %
Epoch 97: Accuracy on Train set: 98.42 %
[epoch: 98, iter: 5]-loss: 0.000
[epoch: 98, iter: 10]-loss: 0.000
[epoch: 98, iter: 15]-loss: 0.000
[epoch: 98, iter: 20]-loss: 0.000
Epoch 98: Accuracy on test set: 55.57 %
Epoch 98: Accuracy on Train set: 98.68 %
[epoch: 99, iter: 5]-loss: 0.000
[epoch: 99, iter: 10]-loss: 0.000
[epoch: 99, iter: 15]-loss: 0.000
[epoch: 99, iter: 20]-loss: 0.000
Epoch 99: Accuracy on test set: 55.62 %
Epoch 99: Accuracy on Train set: 98.92 %
[epoch: 100, iter: 5]-loss: 0.000
[epoch: 100, iter: 10]-loss: 0.000
[epoch: 100, iter: 15]-loss: 0.000
[epoch: 100, iter: 20]-loss: 0.000
Epoch 100: Accuracy on test set: 55.89 %
Epoch 100: Accuracy on Train set: 98.72 %
Training completed!
```

[29]:

```
plt.figure()
plt.plot(torch.arange(len(train_acc)), train_acc, label='train_acc')
plt.plot(torch.arange(len(test_acc)), test_acc, label='test_acc')
plt.grid()
plt.legend()
plt.show()
```



We can achieve upto 100% accuracy on Train set and 56% accuracy on Test set

0.0.10 Combined

[30]:
```
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

# Define the ResNet-18 model
model = torchvision.models.resnet18(pretrained=False)
num_classes = 10

model.fc = nn.Linear(model.fc.in_features, num_classes)
model = model.to(device)

model.eval()
```

[30]:
```
ResNet(
  (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3),
bias=False)
  (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  (relu): ReLU(inplace=True)
  (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1,
ceil_mode=False)
  (layer1): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    )
    (1): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    )
  )
  (layer2): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=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)
      (downsample): Sequential(
        (0): Conv2d(64, 128, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
    )
    (1): BasicBlock(
      (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
```

```
      (relu): ReLU(inplace=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)
    )
  )
  (layer3): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=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)
      (downsample): Sequential(
        (0): Conv2d(128, 256, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
    )
    (1): BasicBlock(
      (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=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)
    )
  )
  (layer4): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=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)
```

```
      (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): BasicBlock(
      (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=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)
    )
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(1, 1))
  (fc): Linear(in_features=512, out_features=10, bias=True)
)
```

[31]:

```
# Define loss function and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.0001)

test_acc = []
train_acc = []

# Training loop
num_epochs = 100

for epoch in range(num_epochs):

    i = 0
    iter = 0
    batch_size = 500

    while i < x_train.shape[0]:
        model.train()

        inputs = x_train[i:i+batch_size]
        labels = y_train[i:i+batch_size].type(torch.LongTensor)
        labels_oh = y_train_oh[i:i+batch_size]

        inputs = cutout(inputs, 16, 0.5)
        inputs = random_shift_and_flip(inputs, 4)
        inputs, labels_oh = mixup_data(inputs, labels_oh, 0.4)


        inputs = inputs.to(device)
        labels = labels.to(device)
        labels_oh = labels_oh.to(device)

        optimizer.zero_grad()

        outputs = model(inputs)
        loss = criterion(outputs, labels_oh)
        loss.backward()
        optimizer.step()

        i += batch_size
        iter += 1

        if (iter + 1) % 5 == 0:
            print('[epoch: %d, iter: %d]-loss: %.3f' % (epoch + 1, iter +␣
↪1,loss.item() / batch_size))

    # Evaluation on the test set
    model.eval()

    # Test Accuracy
    correct = 0
    total = 0
    with torch.no_grad():

        inputs = x_test.to(device)
        labels = y_test.type(torch.LongTensor).to(device)
        outputs = model(inputs)
        _, predicted = torch.max(outputs.data, 1)
        total += labels.size(0)
        correct += (predicted == labels).sum().item()

    accuracy = 100 * correct / total
    test_acc.append(accuracy)
    print('Epoch %d: Accuracy on test set: %.2f %%' % (epoch + 1, accuracy))

    # Train Accuracy
    correct = 0
    total = 0
    with torch.no_grad():

        inputs = x_train.to(device)
        labels = y_train.type(torch.LongTensor).to(device)
        outputs = model(inputs)
        _, predicted = torch.max(outputs.data, 1)
        total += labels.size(0)
        correct += (predicted == labels).sum().item()

    accuracy = 100 * correct / total
    train_acc.append(accuracy)
    print('Epoch %d: Accuracy on Train set: %.2f %%' % (epoch + 1, accuracy))

print('Training completed!')
```

```
[epoch: 1, iter: 5]-loss: 0.005
[epoch: 1, iter: 10]-loss: 0.005
[epoch: 1, iter: 15]-loss: 0.004
[epoch: 1, iter: 20]-loss: 0.004
Epoch 1: Accuracy on test set: 21.77 %
Epoch 1: Accuracy on Train set: 21.64 %
[epoch: 2, iter: 5]-loss: 0.004
[epoch: 2, iter: 10]-loss: 0.004
[epoch: 2, iter: 15]-loss: 0.004
[epoch: 2, iter: 20]-loss: 0.004
Epoch 2: Accuracy on test set: 32.00 %
Epoch 2: Accuracy on Train set: 32.53 %
[epoch: 3, iter: 5]-loss: 0.004
[epoch: 3, iter: 10]-loss: 0.004
[epoch: 3, iter: 15]-loss: 0.004
[epoch: 3, iter: 20]-loss: 0.004
Epoch 3: Accuracy on test set: 32.63 %
Epoch 3: Accuracy on Train set: 34.23 %
[epoch: 4, iter: 5]-loss: 0.004
[epoch: 4, iter: 10]-loss: 0.004
[epoch: 4, iter: 15]-loss: 0.004
[epoch: 4, iter: 20]-loss: 0.004
Epoch 4: Accuracy on test set: 34.24 %
Epoch 4: Accuracy on Train set: 35.82 %
[epoch: 5, iter: 5]-loss: 0.004
[epoch: 5, iter: 10]-loss: 0.004
[epoch: 5, iter: 15]-loss: 0.004
[epoch: 5, iter: 20]-loss: 0.004
Epoch 5: Accuracy on test set: 36.17 %
Epoch 5: Accuracy on Train set: 37.99 %
[epoch: 6, iter: 5]-loss: 0.004
[epoch: 6, iter: 10]-loss: 0.004
[epoch: 6, iter: 15]-loss: 0.004
[epoch: 6, iter: 20]-loss: 0.004
Epoch 6: Accuracy on test set: 37.64 %
Epoch 6: Accuracy on Train set: 39.00 %
```

```
[epoch: 7, iter: 5]-loss: 0.004
[epoch: 7, iter: 10]-loss: 0.004
[epoch: 7, iter: 15]-loss: 0.004
[epoch: 7, iter: 20]-loss: 0.004
Epoch 7: Accuracy on test set: 36.11 %
Epoch 7: Accuracy on Train set: 38.08 %
[epoch: 8, iter: 5]-loss: 0.004
[epoch: 8, iter: 10]-loss: 0.004
[epoch: 8, iter: 15]-loss: 0.004
[epoch: 8, iter: 20]-loss: 0.004
Epoch 8: Accuracy on test set: 38.66 %
Epoch 8: Accuracy on Train set: 40.74 %
[epoch: 9, iter: 5]-loss: 0.004
[epoch: 9, iter: 10]-loss: 0.004
[epoch: 9, iter: 15]-loss: 0.004
[epoch: 9, iter: 20]-loss: 0.004
Epoch 9: Accuracy on test set: 39.56 %
Epoch 9: Accuracy on Train set: 41.22 %
[epoch: 10, iter: 5]-loss: 0.004
[epoch: 10, iter: 10]-loss: 0.004
[epoch: 10, iter: 15]-loss: 0.004
[epoch: 10, iter: 20]-loss: 0.004
Epoch 10: Accuracy on test set: 38.31 %
Epoch 10: Accuracy on Train set: 41.52 %
[epoch: 11, iter: 5]-loss: 0.004
[epoch: 11, iter: 10]-loss: 0.004
[epoch: 11, iter: 15]-loss: 0.004
[epoch: 11, iter: 20]-loss: 0.004
Epoch 11: Accuracy on test set: 38.27 %
Epoch 11: Accuracy on Train set: 41.40 %
[epoch: 12, iter: 5]-loss: 0.004
[epoch: 12, iter: 10]-loss: 0.004
[epoch: 12, iter: 15]-loss: 0.004
[epoch: 12, iter: 20]-loss: 0.004
Epoch 12: Accuracy on test set: 41.77 %
Epoch 12: Accuracy on Train set: 44.79 %
[epoch: 13, iter: 5]-loss: 0.004
[epoch: 13, iter: 10]-loss: 0.004
[epoch: 13, iter: 15]-loss: 0.004
[epoch: 13, iter: 20]-loss: 0.004
Epoch 13: Accuracy on test set: 40.18 %
Epoch 13: Accuracy on Train set: 43.95 %
[epoch: 14, iter: 5]-loss: 0.004
[epoch: 14, iter: 10]-loss: 0.004
[epoch: 14, iter: 15]-loss: 0.004
[epoch: 14, iter: 20]-loss: 0.004
Epoch 14: Accuracy on test set: 40.25 %
Epoch 14: Accuracy on Train set: 44.56 %
```

```
[epoch: 15, iter: 5]-loss: 0.004
[epoch: 15, iter: 10]-loss: 0.004
[epoch: 15, iter: 15]-loss: 0.004
[epoch: 15, iter: 20]-loss: 0.004
Epoch 15: Accuracy on test set: 43.32 %
Epoch 15: Accuracy on Train set: 47.24 %
[epoch: 16, iter: 5]-loss: 0.004
[epoch: 16, iter: 10]-loss: 0.004
[epoch: 16, iter: 15]-loss: 0.004
[epoch: 16, iter: 20]-loss: 0.004
Epoch 16: Accuracy on test set: 44.41 %
Epoch 16: Accuracy on Train set: 49.33 %
[epoch: 17, iter: 5]-loss: 0.004
[epoch: 17, iter: 10]-loss: 0.004
[epoch: 17, iter: 15]-loss: 0.004
[epoch: 17, iter: 20]-loss: 0.004
Epoch 17: Accuracy on test set: 41.60 %
Epoch 17: Accuracy on Train set: 46.21 %
[epoch: 18, iter: 5]-loss: 0.004
[epoch: 18, iter: 10]-loss: 0.004
[epoch: 18, iter: 15]-loss: 0.004
[epoch: 18, iter: 20]-loss: 0.004
Epoch 18: Accuracy on test set: 45.17 %
Epoch 18: Accuracy on Train set: 50.26 %
[epoch: 19, iter: 5]-loss: 0.004
[epoch: 19, iter: 10]-loss: 0.004
[epoch: 19, iter: 15]-loss: 0.004
[epoch: 19, iter: 20]-loss: 0.004
Epoch 19: Accuracy on test set: 44.73 %
Epoch 19: Accuracy on Train set: 50.51 %
[epoch: 20, iter: 5]-loss: 0.004
[epoch: 20, iter: 10]-loss: 0.004
[epoch: 20, iter: 15]-loss: 0.004
[epoch: 20, iter: 20]-loss: 0.004
Epoch 20: Accuracy on test set: 43.73 %
Epoch 20: Accuracy on Train set: 49.25 %
[epoch: 21, iter: 5]-loss: 0.004
[epoch: 21, iter: 10]-loss: 0.004
[epoch: 21, iter: 15]-loss: 0.004
[epoch: 21, iter: 20]-loss: 0.004
Epoch 21: Accuracy on test set: 43.94 %
Epoch 21: Accuracy on Train set: 49.65 %
[epoch: 22, iter: 5]-loss: 0.004
[epoch: 22, iter: 10]-loss: 0.004
[epoch: 22, iter: 15]-loss: 0.004
[epoch: 22, iter: 20]-loss: 0.004
Epoch 22: Accuracy on test set: 45.51 %
Epoch 22: Accuracy on Train set: 52.05 %
```

```
[epoch: 23, iter: 5]-loss: 0.004
[epoch: 23, iter: 10]-loss: 0.004
[epoch: 23, iter: 15]-loss: 0.004
[epoch: 23, iter: 20]-loss: 0.004
Epoch 23: Accuracy on test set: 46.75 %
Epoch 23: Accuracy on Train set: 52.52 %
[epoch: 24, iter: 5]-loss: 0.004
[epoch: 24, iter: 10]-loss: 0.004
[epoch: 24, iter: 15]-loss: 0.004
[epoch: 24, iter: 20]-loss: 0.004
Epoch 24: Accuracy on test set: 47.44 %
Epoch 24: Accuracy on Train set: 53.19 %
[epoch: 25, iter: 5]-loss: 0.004
[epoch: 25, iter: 10]-loss: 0.004
[epoch: 25, iter: 15]-loss: 0.004
[epoch: 25, iter: 20]-loss: 0.004
Epoch 25: Accuracy on test set: 47.30 %
Epoch 25: Accuracy on Train set: 53.61 %
[epoch: 26, iter: 5]-loss: 0.004
[epoch: 26, iter: 10]-loss: 0.004
[epoch: 26, iter: 15]-loss: 0.004
[epoch: 26, iter: 20]-loss: 0.004
Epoch 26: Accuracy on test set: 47.09 %
Epoch 26: Accuracy on Train set: 54.22 %
[epoch: 27, iter: 5]-loss: 0.004
[epoch: 27, iter: 10]-loss: 0.004
[epoch: 27, iter: 15]-loss: 0.004
[epoch: 27, iter: 20]-loss: 0.004
Epoch 27: Accuracy on test set: 47.27 %
Epoch 27: Accuracy on Train set: 53.48 %
[epoch: 28, iter: 5]-loss: 0.004
[epoch: 28, iter: 10]-loss: 0.004
[epoch: 28, iter: 15]-loss: 0.004
[epoch: 28, iter: 20]-loss: 0.004
Epoch 28: Accuracy on test set: 46.82 %
Epoch 28: Accuracy on Train set: 54.26 %
[epoch: 29, iter: 5]-loss: 0.004
[epoch: 29, iter: 10]-loss: 0.004
[epoch: 29, iter: 15]-loss: 0.004
[epoch: 29, iter: 20]-loss: 0.004
Epoch 29: Accuracy on test set: 48.57 %
Epoch 29: Accuracy on Train set: 55.98 %
[epoch: 30, iter: 5]-loss: 0.004
[epoch: 30, iter: 10]-loss: 0.004
[epoch: 30, iter: 15]-loss: 0.004
[epoch: 30, iter: 20]-loss: 0.003
Epoch 30: Accuracy on test set: 49.61 %
Epoch 30: Accuracy on Train set: 57.30 %
```

```
[epoch: 31, iter: 5]-loss: 0.004
[epoch: 31, iter: 10]-loss: 0.004
[epoch: 31, iter: 15]-loss: 0.004
[epoch: 31, iter: 20]-loss: 0.004
Epoch 31: Accuracy on test set: 49.54 %
Epoch 31: Accuracy on Train set: 58.35 %
[epoch: 32, iter: 5]-loss: 0.004
[epoch: 32, iter: 10]-loss: 0.004
[epoch: 32, iter: 15]-loss: 0.004
[epoch: 32, iter: 20]-loss: 0.004
Epoch 32: Accuracy on test set: 49.70 %
Epoch 32: Accuracy on Train set: 58.47 %
[epoch: 33, iter: 5]-loss: 0.004
[epoch: 33, iter: 10]-loss: 0.004
[epoch: 33, iter: 15]-loss: 0.004
[epoch: 33, iter: 20]-loss: 0.004
Epoch 33: Accuracy on test set: 47.88 %
Epoch 33: Accuracy on Train set: 56.78 %
[epoch: 34, iter: 5]-loss: 0.004
[epoch: 34, iter: 10]-loss: 0.004
[epoch: 34, iter: 15]-loss: 0.004
[epoch: 34, iter: 20]-loss: 0.004
Epoch 34: Accuracy on test set: 49.44 %
Epoch 34: Accuracy on Train set: 58.57 %
[epoch: 35, iter: 5]-loss: 0.004
[epoch: 35, iter: 10]-loss: 0.004
[epoch: 35, iter: 15]-loss: 0.004
[epoch: 35, iter: 20]-loss: 0.004
Epoch 35: Accuracy on test set: 50.83 %
Epoch 35: Accuracy on Train set: 59.65 %
[epoch: 36, iter: 5]-loss: 0.004
[epoch: 36, iter: 10]-loss: 0.004
[epoch: 36, iter: 15]-loss: 0.003
[epoch: 36, iter: 20]-loss: 0.003
Epoch 36: Accuracy on test set: 50.94 %
Epoch 36: Accuracy on Train set: 59.76 %
[epoch: 37, iter: 5]-loss: 0.004
[epoch: 37, iter: 10]-loss: 0.004
[epoch: 37, iter: 15]-loss: 0.004
[epoch: 37, iter: 20]-loss: 0.003
Epoch 37: Accuracy on test set: 50.57 %
Epoch 37: Accuracy on Train set: 60.34 %
[epoch: 38, iter: 5]-loss: 0.004
[epoch: 38, iter: 10]-loss: 0.004
[epoch: 38, iter: 15]-loss: 0.004
[epoch: 38, iter: 20]-loss: 0.003
Epoch 38: Accuracy on test set: 48.99 %
Epoch 38: Accuracy on Train set: 59.23 %
```

```
[epoch: 39, iter: 5]-loss: 0.004
[epoch: 39, iter: 10]-loss: 0.004
[epoch: 39, iter: 15]-loss: 0.004
[epoch: 39, iter: 20]-loss: 0.004
Epoch 39: Accuracy on test set: 49.41 %
Epoch 39: Accuracy on Train set: 60.08 %
[epoch: 40, iter: 5]-loss: 0.004
[epoch: 40, iter: 10]-loss: 0.004
[epoch: 40, iter: 15]-loss: 0.004
[epoch: 40, iter: 20]-loss: 0.003
Epoch 40: Accuracy on test set: 50.72 %
Epoch 40: Accuracy on Train set: 61.82 %
[epoch: 41, iter: 5]-loss: 0.004
[epoch: 41, iter: 10]-loss: 0.004
[epoch: 41, iter: 15]-loss: 0.004
[epoch: 41, iter: 20]-loss: 0.004
Epoch 41: Accuracy on test set: 51.58 %
Epoch 41: Accuracy on Train set: 62.00 %
[epoch: 42, iter: 5]-loss: 0.004
[epoch: 42, iter: 10]-loss: 0.004
[epoch: 42, iter: 15]-loss: 0.004
[epoch: 42, iter: 20]-loss: 0.003
Epoch 42: Accuracy on test set: 51.47 %
Epoch 42: Accuracy on Train set: 61.87 %
[epoch: 43, iter: 5]-loss: 0.003
[epoch: 43, iter: 10]-loss: 0.004
[epoch: 43, iter: 15]-loss: 0.004
[epoch: 43, iter: 20]-loss: 0.003
Epoch 43: Accuracy on test set: 51.50 %
Epoch 43: Accuracy on Train set: 62.07 %
[epoch: 44, iter: 5]-loss: 0.003
[epoch: 44, iter: 10]-loss: 0.003
[epoch: 44, iter: 15]-loss: 0.004
[epoch: 44, iter: 20]-loss: 0.003
Epoch 44: Accuracy on test set: 51.02 %
Epoch 44: Accuracy on Train set: 62.27 %
[epoch: 45, iter: 5]-loss: 0.004
[epoch: 45, iter: 10]-loss: 0.003
[epoch: 45, iter: 15]-loss: 0.004
[epoch: 45, iter: 20]-loss: 0.003
Epoch 45: Accuracy on test set: 52.03 %
Epoch 45: Accuracy on Train set: 63.12 %
[epoch: 46, iter: 5]-loss: 0.004
[epoch: 46, iter: 10]-loss: 0.004
[epoch: 46, iter: 15]-loss: 0.004
[epoch: 46, iter: 20]-loss: 0.003
Epoch 46: Accuracy on test set: 52.63 %
Epoch 46: Accuracy on Train set: 64.91 %
```

```
[epoch: 47, iter: 5]-loss: 0.003
[epoch: 47, iter: 10]-loss: 0.004
[epoch: 47, iter: 15]-loss: 0.003
[epoch: 47, iter: 20]-loss: 0.003
Epoch 47: Accuracy on test set: 51.15 %
Epoch 47: Accuracy on Train set: 62.32 %
[epoch: 48, iter: 5]-loss: 0.004
[epoch: 48, iter: 10]-loss: 0.003
[epoch: 48, iter: 15]-loss: 0.003
[epoch: 48, iter: 20]-loss: 0.003
Epoch 48: Accuracy on test set: 52.42 %
Epoch 48: Accuracy on Train set: 64.74 %
[epoch: 49, iter: 5]-loss: 0.004
[epoch: 49, iter: 10]-loss: 0.003
[epoch: 49, iter: 15]-loss: 0.003
[epoch: 49, iter: 20]-loss: 0.004
Epoch 49: Accuracy on test set: 51.89 %
Epoch 49: Accuracy on Train set: 64.31 %
[epoch: 50, iter: 5]-loss: 0.003
[epoch: 50, iter: 10]-loss: 0.003
[epoch: 50, iter: 15]-loss: 0.004
[epoch: 50, iter: 20]-loss: 0.003
Epoch 50: Accuracy on test set: 52.82 %
Epoch 50: Accuracy on Train set: 65.90 %
[epoch: 51, iter: 5]-loss: 0.004
[epoch: 51, iter: 10]-loss: 0.003
[epoch: 51, iter: 15]-loss: 0.003
[epoch: 51, iter: 20]-loss: 0.003
Epoch 51: Accuracy on test set: 52.24 %
Epoch 51: Accuracy on Train set: 64.29 %
[epoch: 52, iter: 5]-loss: 0.003
[epoch: 52, iter: 10]-loss: 0.003
[epoch: 52, iter: 15]-loss: 0.003
[epoch: 52, iter: 20]-loss: 0.003
Epoch 52: Accuracy on test set: 51.87 %
Epoch 52: Accuracy on Train set: 65.90 %
[epoch: 53, iter: 5]-loss: 0.003
[epoch: 53, iter: 10]-loss: 0.003
[epoch: 53, iter: 15]-loss: 0.003
[epoch: 53, iter: 20]-loss: 0.003
Epoch 53: Accuracy on test set: 52.60 %
Epoch 53: Accuracy on Train set: 66.00 %
[epoch: 54, iter: 5]-loss: 0.003
[epoch: 54, iter: 10]-loss: 0.004
[epoch: 54, iter: 15]-loss: 0.003
[epoch: 54, iter: 20]-loss: 0.003
Epoch 54: Accuracy on test set: 53.05 %
Epoch 54: Accuracy on Train set: 66.44 %
```

```
[epoch: 55, iter: 5]-loss: 0.003
[epoch: 55, iter: 10]-loss: 0.003
[epoch: 55, iter: 15]-loss: 0.003
[epoch: 55, iter: 20]-loss: 0.003
Epoch 55: Accuracy on test set: 52.97 %
Epoch 55: Accuracy on Train set: 67.11 %
[epoch: 56, iter: 5]-loss: 0.003
[epoch: 56, iter: 10]-loss: 0.003
[epoch: 56, iter: 15]-loss: 0.003
[epoch: 56, iter: 20]-loss: 0.003
Epoch 56: Accuracy on test set: 53.65 %
Epoch 56: Accuracy on Train set: 68.07 %
[epoch: 57, iter: 5]-loss: 0.003
[epoch: 57, iter: 10]-loss: 0.003
[epoch: 57, iter: 15]-loss: 0.004
[epoch: 57, iter: 20]-loss: 0.003
Epoch 57: Accuracy on test set: 53.62 %
Epoch 57: Accuracy on Train set: 68.26 %
[epoch: 58, iter: 5]-loss: 0.003
[epoch: 58, iter: 10]-loss: 0.003
[epoch: 58, iter: 15]-loss: 0.003
[epoch: 58, iter: 20]-loss: 0.003
Epoch 58: Accuracy on test set: 53.49 %
Epoch 58: Accuracy on Train set: 67.94 %
[epoch: 59, iter: 5]-loss: 0.003
[epoch: 59, iter: 10]-loss: 0.003
[epoch: 59, iter: 15]-loss: 0.003
[epoch: 59, iter: 20]-loss: 0.003
Epoch 59: Accuracy on test set: 53.61 %
Epoch 59: Accuracy on Train set: 68.61 %
[epoch: 60, iter: 5]-loss: 0.003
[epoch: 60, iter: 10]-loss: 0.003
[epoch: 60, iter: 15]-loss: 0.004
[epoch: 60, iter: 20]-loss: 0.003
Epoch 60: Accuracy on test set: 54.08 %
Epoch 60: Accuracy on Train set: 69.76 %
[epoch: 61, iter: 5]-loss: 0.003
[epoch: 61, iter: 10]-loss: 0.003
[epoch: 61, iter: 15]-loss: 0.003
[epoch: 61, iter: 20]-loss: 0.003
Epoch 61: Accuracy on test set: 53.49 %
Epoch 61: Accuracy on Train set: 68.73 %
[epoch: 62, iter: 5]-loss: 0.003
[epoch: 62, iter: 10]-loss: 0.003
[epoch: 62, iter: 15]-loss: 0.003
[epoch: 62, iter: 20]-loss: 0.003
Epoch 62: Accuracy on test set: 54.48 %
Epoch 62: Accuracy on Train set: 70.15 %
```

```
[epoch: 63, iter: 5]-loss: 0.003
[epoch: 63, iter: 10]-loss: 0.003
[epoch: 63, iter: 15]-loss: 0.003
[epoch: 63, iter: 20]-loss: 0.003
Epoch 63: Accuracy on test set: 54.10 %
Epoch 63: Accuracy on Train set: 69.55 %
[epoch: 64, iter: 5]-loss: 0.003
[epoch: 64, iter: 10]-loss: 0.003
[epoch: 64, iter: 15]-loss: 0.003
[epoch: 64, iter: 20]-loss: 0.003
Epoch 64: Accuracy on test set: 52.14 %
Epoch 64: Accuracy on Train set: 67.14 %
[epoch: 65, iter: 5]-loss: 0.003
[epoch: 65, iter: 10]-loss: 0.003
[epoch: 65, iter: 15]-loss: 0.004
[epoch: 65, iter: 20]-loss: 0.003
Epoch 65: Accuracy on test set: 54.69 %
Epoch 65: Accuracy on Train set: 70.98 %
[epoch: 66, iter: 5]-loss: 0.003
[epoch: 66, iter: 10]-loss: 0.003
[epoch: 66, iter: 15]-loss: 0.003
[epoch: 66, iter: 20]-loss: 0.003
Epoch 66: Accuracy on test set: 54.81 %
Epoch 66: Accuracy on Train set: 71.52 %
[epoch: 67, iter: 5]-loss: 0.003
[epoch: 67, iter: 10]-loss: 0.003
[epoch: 67, iter: 15]-loss: 0.003
[epoch: 67, iter: 20]-loss: 0.003
Epoch 67: Accuracy on test set: 55.14 %
Epoch 67: Accuracy on Train set: 72.03 %
[epoch: 68, iter: 5]-loss: 0.003
[epoch: 68, iter: 10]-loss: 0.003
[epoch: 68, iter: 15]-loss: 0.003
[epoch: 68, iter: 20]-loss: 0.003
Epoch 68: Accuracy on test set: 54.63 %
Epoch 68: Accuracy on Train set: 73.64 %
[epoch: 69, iter: 5]-loss: 0.003
[epoch: 69, iter: 10]-loss: 0.003
[epoch: 69, iter: 15]-loss: 0.003
[epoch: 69, iter: 20]-loss: 0.003
Epoch 69: Accuracy on test set: 55.05 %
Epoch 69: Accuracy on Train set: 71.95 %
[epoch: 70, iter: 5]-loss: 0.003
[epoch: 70, iter: 10]-loss: 0.003
[epoch: 70, iter: 15]-loss: 0.003
[epoch: 70, iter: 20]-loss: 0.003
Epoch 70: Accuracy on test set: 54.97 %
Epoch 70: Accuracy on Train set: 73.38 %
```

```
[epoch: 71, iter: 5]-loss: 0.003
[epoch: 71, iter: 10]-loss: 0.003
[epoch: 71, iter: 15]-loss: 0.003
[epoch: 71, iter: 20]-loss: 0.003
Epoch 71: Accuracy on test set: 54.90 %
Epoch 71: Accuracy on Train set: 72.53 %
[epoch: 72, iter: 5]-loss: 0.003
[epoch: 72, iter: 10]-loss: 0.003
[epoch: 72, iter: 15]-loss: 0.003
[epoch: 72, iter: 20]-loss: 0.003
Epoch 72: Accuracy on test set: 55.55 %
Epoch 72: Accuracy on Train set: 73.96 %
[epoch: 73, iter: 5]-loss: 0.003
[epoch: 73, iter: 10]-loss: 0.003
[epoch: 73, iter: 15]-loss: 0.003
[epoch: 73, iter: 20]-loss: 0.003
Epoch 73: Accuracy on test set: 55.42 %
Epoch 73: Accuracy on Train set: 72.42 %
[epoch: 74, iter: 5]-loss: 0.003
[epoch: 74, iter: 10]-loss: 0.003
[epoch: 74, iter: 15]-loss: 0.003
[epoch: 74, iter: 20]-loss: 0.003
Epoch 74: Accuracy on test set: 55.31 %
Epoch 74: Accuracy on Train set: 74.39 %
[epoch: 75, iter: 5]-loss: 0.003
[epoch: 75, iter: 10]-loss: 0.003
[epoch: 75, iter: 15]-loss: 0.003
[epoch: 75, iter: 20]-loss: 0.003
Epoch 75: Accuracy on test set: 55.14 %
Epoch 75: Accuracy on Train set: 74.49 %
[epoch: 76, iter: 5]-loss: 0.003
[epoch: 76, iter: 10]-loss: 0.003
[epoch: 76, iter: 15]-loss: 0.003
[epoch: 76, iter: 20]-loss: 0.003
Epoch 76: Accuracy on test set: 55.45 %
Epoch 76: Accuracy on Train set: 73.39 %
[epoch: 77, iter: 5]-loss: 0.003
[epoch: 77, iter: 10]-loss: 0.003
[epoch: 77, iter: 15]-loss: 0.003
[epoch: 77, iter: 20]-loss: 0.003
Epoch 77: Accuracy on test set: 55.91 %
Epoch 77: Accuracy on Train set: 75.22 %
[epoch: 78, iter: 5]-loss: 0.003
[epoch: 78, iter: 10]-loss: 0.003
[epoch: 78, iter: 15]-loss: 0.003
[epoch: 78, iter: 20]-loss: 0.003
Epoch 78: Accuracy on test set: 56.48 %
Epoch 78: Accuracy on Train set: 75.71 %
```

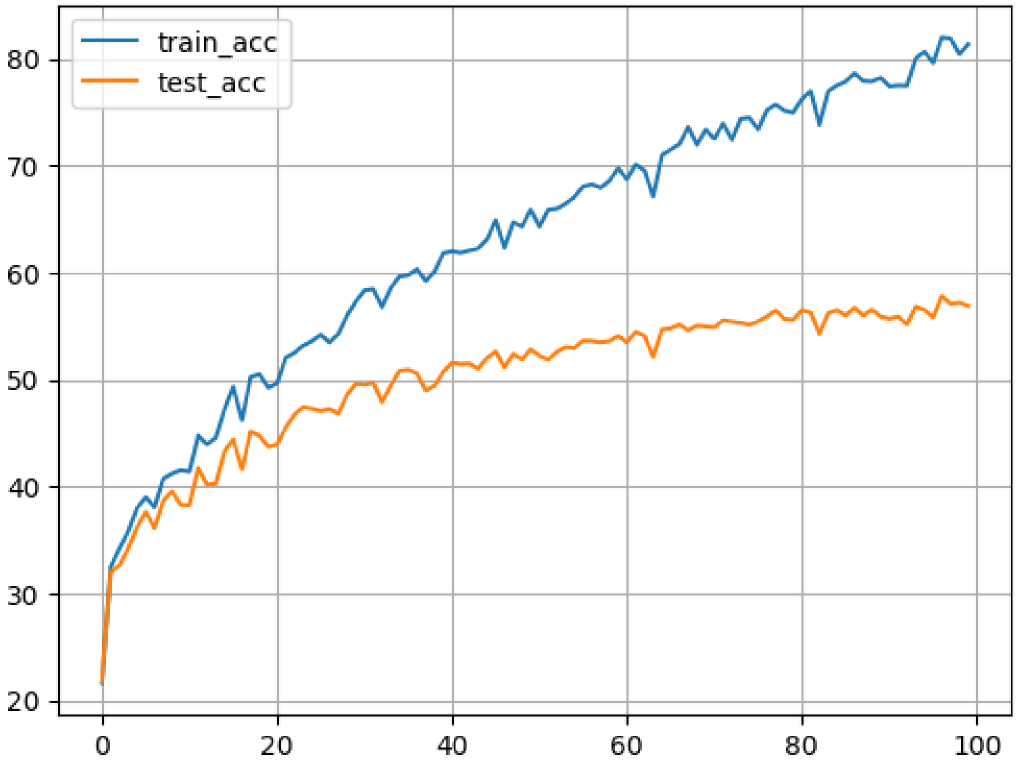
```
[epoch: 79, iter: 5]-loss: 0.003
[epoch: 79, iter: 10]-loss: 0.003
[epoch: 79, iter: 15]-loss: 0.003
[epoch: 79, iter: 20]-loss: 0.003
Epoch 79: Accuracy on test set: 55.68 %
Epoch 79: Accuracy on Train set: 75.17 %
[epoch: 80, iter: 5]-loss: 0.003
[epoch: 80, iter: 10]-loss: 0.003
[epoch: 80, iter: 15]-loss: 0.003
[epoch: 80, iter: 20]-loss: 0.003
Epoch 80: Accuracy on test set: 55.58 %
Epoch 80: Accuracy on Train set: 74.97 %
[epoch: 81, iter: 5]-loss: 0.003
[epoch: 81, iter: 10]-loss: 0.003
[epoch: 81, iter: 15]-loss: 0.003
[epoch: 81, iter: 20]-loss: 0.003
Epoch 81: Accuracy on test set: 56.51 %
Epoch 81: Accuracy on Train set: 76.25 %
[epoch: 82, iter: 5]-loss: 0.003
[epoch: 82, iter: 10]-loss: 0.003
[epoch: 82, iter: 15]-loss: 0.003
[epoch: 82, iter: 20]-loss: 0.003
Epoch 82: Accuracy on test set: 56.28 %
Epoch 82: Accuracy on Train set: 76.99 %
[epoch: 83, iter: 5]-loss: 0.003
[epoch: 83, iter: 10]-loss: 0.003
[epoch: 83, iter: 15]-loss: 0.003
[epoch: 83, iter: 20]-loss: 0.003
Epoch 83: Accuracy on test set: 54.27 %
Epoch 83: Accuracy on Train set: 73.83 %
[epoch: 84, iter: 5]-loss: 0.003
[epoch: 84, iter: 10]-loss: 0.003
[epoch: 84, iter: 15]-loss: 0.003
[epoch: 84, iter: 20]-loss: 0.003
Epoch 84: Accuracy on test set: 56.22 %
Epoch 84: Accuracy on Train set: 76.98 %
[epoch: 85, iter: 5]-loss: 0.003
[epoch: 85, iter: 10]-loss: 0.003
[epoch: 85, iter: 15]-loss: 0.003
[epoch: 85, iter: 20]-loss: 0.003
Epoch 85: Accuracy on test set: 56.51 %
Epoch 85: Accuracy on Train set: 77.52 %
[epoch: 86, iter: 5]-loss: 0.003
[epoch: 86, iter: 10]-loss: 0.003
[epoch: 86, iter: 15]-loss: 0.003
[epoch: 86, iter: 20]-loss: 0.003
Epoch 86: Accuracy on test set: 56.01 %
Epoch 86: Accuracy on Train set: 77.87 %
```

```
[epoch: 87, iter: 5]-loss: 0.003
[epoch: 87, iter: 10]-loss: 0.003
[epoch: 87, iter: 15]-loss: 0.003
[epoch: 87, iter: 20]-loss: 0.003
Epoch 87: Accuracy on test set: 56.74 %
Epoch 87: Accuracy on Train set: 78.67 %
[epoch: 88, iter: 5]-loss: 0.003
[epoch: 88, iter: 10]-loss: 0.003
[epoch: 88, iter: 15]-loss: 0.003
[epoch: 88, iter: 20]-loss: 0.003
Epoch 88: Accuracy on test set: 56.00 %
Epoch 88: Accuracy on Train set: 77.93 %
[epoch: 89, iter: 5]-loss: 0.003
[epoch: 89, iter: 10]-loss: 0.003
[epoch: 89, iter: 15]-loss: 0.003
[epoch: 89, iter: 20]-loss: 0.003
Epoch 89: Accuracy on test set: 56.55 %
Epoch 89: Accuracy on Train set: 77.89 %
[epoch: 90, iter: 5]-loss: 0.003
[epoch: 90, iter: 10]-loss: 0.003
[epoch: 90, iter: 15]-loss: 0.003
[epoch: 90, iter: 20]-loss: 0.003
Epoch 90: Accuracy on test set: 55.91 %
Epoch 90: Accuracy on Train set: 78.24 %
[epoch: 91, iter: 5]-loss: 0.003
[epoch: 91, iter: 10]-loss: 0.003
[epoch: 91, iter: 15]-loss: 0.003
[epoch: 91, iter: 20]-loss: 0.003
Epoch 91: Accuracy on test set: 55.70 %
Epoch 91: Accuracy on Train set: 77.43 %
[epoch: 92, iter: 5]-loss: 0.003
[epoch: 92, iter: 10]-loss: 0.003
[epoch: 92, iter: 15]-loss: 0.003
[epoch: 92, iter: 20]-loss: 0.003
Epoch 92: Accuracy on test set: 55.90 %
Epoch 92: Accuracy on Train set: 77.52 %
[epoch: 93, iter: 5]-loss: 0.003
[epoch: 93, iter: 10]-loss: 0.003
[epoch: 93, iter: 15]-loss: 0.003
[epoch: 93, iter: 20]-loss: 0.003
Epoch 93: Accuracy on test set: 55.18 %
Epoch 93: Accuracy on Train set: 77.49 %
[epoch: 94, iter: 5]-loss: 0.003
[epoch: 94, iter: 10]-loss: 0.003
[epoch: 94, iter: 15]-loss: 0.003
[epoch: 94, iter: 20]-loss: 0.003
Epoch 94: Accuracy on test set: 56.83 %
Epoch 94: Accuracy on Train set: 80.03 %
```

```
[epoch: 95, iter: 5]-loss: 0.003
[epoch: 95, iter: 10]-loss: 0.003
[epoch: 95, iter: 15]-loss: 0.003
[epoch: 95, iter: 20]-loss: 0.003
Epoch 95: Accuracy on test set: 56.53 %
Epoch 95: Accuracy on Train set: 80.67 %
[epoch: 96, iter: 5]-loss: 0.003
[epoch: 96, iter: 10]-loss: 0.003
[epoch: 96, iter: 15]-loss: 0.003
[epoch: 96, iter: 20]-loss: 0.003
Epoch 96: Accuracy on test set: 55.85 %
Epoch 96: Accuracy on Train set: 79.64 %
[epoch: 97, iter: 5]-loss: 0.003
[epoch: 97, iter: 10]-loss: 0.003
[epoch: 97, iter: 15]-loss: 0.003
[epoch: 97, iter: 20]-loss: 0.003
Epoch 97: Accuracy on test set: 57.82 %
Epoch 97: Accuracy on Train set: 82.01 %
[epoch: 98, iter: 5]-loss: 0.003
[epoch: 98, iter: 10]-loss: 0.003
[epoch: 98, iter: 15]-loss: 0.003
[epoch: 98, iter: 20]-loss: 0.003
Epoch 98: Accuracy on test set: 57.08 %
Epoch 98: Accuracy on Train set: 81.91 %
[epoch: 99, iter: 5]-loss: 0.003
[epoch: 99, iter: 10]-loss: 0.003
[epoch: 99, iter: 15]-loss: 0.003
[epoch: 99, iter: 20]-loss: 0.003
Epoch 99: Accuracy on test set: 57.20 %
Epoch 99: Accuracy on Train set: 80.44 %
[epoch: 100, iter: 5]-loss: 0.003
[epoch: 100, iter: 10]-loss: 0.003
[epoch: 100, iter: 15]-loss: 0.003
[epoch: 100, iter: 20]-loss: 0.003
Epoch 100: Accuracy on test set: 56.90 %
Epoch 100: Accuracy on Train set: 81.35 %
Training completed!
```

[32]:

```
plt.figure()
plt.plot(torch.arange(len(train_acc)), train_acc, label='train_acc')
plt.plot(torch.arange(len(test_acc)), test_acc, label='test_acc')
plt.grid()
plt.legend()
plt.show()
```



For Alpha = 0.4 set for combined augmentation We can achieve upto 100% accuracy on Train set and 58% accuracy on Test set

We can see a slight improvement in the classification with all the three augmentations combined.

0.1 Role of Data Augmentation

Data augmentation is a good regularization option and its helps model generalize better to have improved results. It also helps the model to converge faster.

It increase the test accuracy by atleast 10% in some cases with upto 18% when combined together.

However, for train accuracy, we see a reduction in train accuracy, which means that the model is generalizing better and is not prone to overfitting.

The model converges to solution faster and gets a better solution without being stuck in a local minima.

The test accuracy is higher and loss converges faster.

```
[ ]:
```