Name: Faisal Bin Ashraf(fashr003)

ID: 862334529

HW-3

In [ ]:
```python
import numpy as np
np.random.seed(123)
import pickle
import matplotlib.pyplot as plt

import torch
import torchvision.models as models
import torch.nn as nn
import torch.optim as optim
from torch.utils.data import DataLoader, TensorDataset, Dataset
torch.manual_seed(42)
from torchsummary import summary
```

In [ ]:
```python
def unpickle(file):
    with open(file, 'rb') as fo:
        dict = pickle.load(fo, encoding='bytes')
    return dict

def preprocess_data(data):
    data = data.astype('float32') / 255.0
    data = data.reshape((-1, 3, 32, 32)) # pytorch dimension = (B, C, H, W)
    mean = np.mean(data, axis=(0, 1, 2))
    std = np.std(data, axis=(0, 1, 2))
    data = (data - mean) / std
    return data
```

In [ ]:
```python
def run_model(model, train_loader, test_loader, ITR=100, data_aug = 'None', alpha=0.2):
    criterion = nn.CrossEntropyLoss()
    optimizer = optim.Adam(model.parameters(), lr=0.001)
    num_epochs = ITR

    train_loss_values = []
    train_acc_values = []
    test_loss_values = []
    test_acc_values = []

    for epoch in range(num_epochs):
        model.train()
        train_loss = 0
        correct = 0
        total = 0

        for images, labels in train_loader:
            if data_aug == 'cutout':
                images = apply_cutout_minibatch(images.detach().clone(), 16)
            if data_aug == 'mixup':
                images, labels = apply_mixup_minibatch(images.detach().clone(), labels.detach().clone(), alpha)
            if data_aug == 'standard':
                images = apply_standard_minibatch(images.detach().clone(), 4)
            if data_aug == 'all':
                images = apply_standard_minibatch(images.detach().clone(), 4)
                images = apply_cutout_minibatch(images.detach().clone(), 16)
                images, labels = apply_mixup_minibatch(images.detach().clone(), labels.detach().clone(), 0.2)
            optimizer.zero_grad()
            outputs = model(images)
            loss = criterion(outputs, labels)
            loss.backward()
            optimizer.step()

            train_loss += loss.item()
            _, predicted = torch.max(outputs.data, 1)
            total += labels.size(0)
            correct += (predicted == labels).sum().item()

        train_loss /= len(train_loader)
        train_accuracy = correct / total
        train_loss_values.append(train_loss)
        train_acc_values.append(train_accuracy)

        model.eval()
        test_loss = 0
        correct = 0
        total = 0

        with torch.no_grad():
            for images, labels in test_loader:
                outputs = model(images)
                loss = criterion(outputs, labels)
                test_loss += loss.item()
                _, predicted = torch.max(outputs.data, 1)
                total += labels.size(0)
                correct += (predicted == labels).sum().item()

        test_loss /= len(test_loader)
        test_accuracy = correct / total
        test_loss_values.append(test_loss)
        test_acc_values.append(test_accuracy)

        print(f"Epoch [{epoch+1}/{num_epochs}], Train Loss: {train_loss:.4f}, Train Acc: {train_accuracy:.3f}, Test Loss: {test_loss:.4f}, Test Acc: {test_accuracy:.3f}")

    history = {}
    history['train_acc'] = train_acc_values
    history['train_loss'] = train_loss_values
    history['test_acc'] = test_acc_values
    history['test_loss'] = test_loss_values

    return history
```

In [ ]:
```python
def plot_history(history, figsize=(12,5), title = 'Training History'):
    num_epochs = len(history['train_loss'])
    plt.figure(figsize=figsize)
    a = plt.subplot(1, 2, 1)
    a.plot(range(1, num_epochs+1), history['train_loss'], label='Train Loss')
    a.plot(range(1, num_epochs+1), history['test_loss'], label='Test Loss')
    a.set_xlabel('Epoch')
    a.set_ylabel('Loss')
    a.set_title('Training and Test Loss')
    a.legend()

    b = plt.subplot(1, 2, 2)
    b.plot(range(1, num_epochs+1), history['train_acc'], label='Train Acc')
    b.plot(range(1, num_epochs+1), history['test_acc'], label='Test Acc')
    b.set_xlabel('Epoch')
    b.set_ylabel('Accuracy')
    b.set_title('Training and Test Accuracy')
    b.legend()

    plt.tight_layout()
    plt.show()

    print(f"Final Test Accuracy is {history['test_acc'][-1]}")
```

In [ ]:
```python
#model = torch.hub.load('pytorch/vision:v0.10.0', 'resnet18', pretrained=False)
from resnet20 import ResNet, BasicBlock
model = ResNet(BasicBlock, [3, 3, 3], num_classes=10)
```

In [ ]:
```python
summary(model, (3, 32, 32))
```

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1           [-1, 16, 32, 32]             432
       BatchNorm2d-2           [-1, 16, 32, 32]              32
            Conv2d-3           [-1, 16, 32, 32]           2,304
       BatchNorm2d-4           [-1, 16, 32, 32]              32
            Conv2d-5           [-1, 16, 32, 32]           2,304
       BatchNorm2d-6           [-1, 16, 32, 32]              32
        BasicBlock-7           [-1, 16, 32, 32]               0
            Conv2d-8           [-1, 16, 32, 32]           2,304
       BatchNorm2d-9           [-1, 16, 32, 32]              32
           Conv2d-10           [-1, 16, 32, 32]           2,304
      BatchNorm2d-11           [-1, 16, 32, 32]              32
       BasicBlock-12           [-1, 16, 32, 32]               0
           Conv2d-13           [-1, 16, 32, 32]           2,304
      BatchNorm2d-14           [-1, 16, 32, 32]              32
           Conv2d-15           [-1, 16, 32, 32]           2,304
      BatchNorm2d-16           [-1, 16, 32, 32]              32
       BasicBlock-17           [-1, 16, 32, 32]               0
           Conv2d-18           [-1, 32, 16, 16]           4,608
      BatchNorm2d-19           [-1, 32, 16, 16]              64
           Conv2d-20           [-1, 32, 16, 16]           9,216
      BatchNorm2d-21           [-1, 32, 16, 16]              64
           Conv2d-22           [-1, 32, 16, 16]             512
      BatchNorm2d-23           [-1, 32, 16, 16]              64
       BasicBlock-24           [-1, 32, 16, 16]               0
           Conv2d-25           [-1, 32, 16, 16]           9,216
      BatchNorm2d-26           [-1, 32, 16, 16]              64
           Conv2d-27           [-1, 32, 16, 16]           9,216
      BatchNorm2d-28           [-1, 32, 16, 16]              64
       BasicBlock-29           [-1, 32, 16, 16]               0
           Conv2d-30           [-1, 32, 16, 16]           9,216
      BatchNorm2d-31           [-1, 32, 16, 16]              64
           Conv2d-32           [-1, 32, 16, 16]           9,216
      BatchNorm2d-33           [-1, 32, 16, 16]              64
       BasicBlock-34           [-1, 32, 16, 16]               0
           Conv2d-35             [-1, 64, 8, 8]          18,432
      BatchNorm2d-36             [-1, 64, 8, 8]             128
           Conv2d-37             [-1, 64, 8, 8]          36,864
      BatchNorm2d-38             [-1, 64, 8, 8]             128
           Conv2d-39             [-1, 64, 8, 8]           2,048
      BatchNorm2d-40             [-1, 64, 8, 8]             128
       BasicBlock-41             [-1, 64, 8, 8]               0
           Conv2d-42             [-1, 64, 8, 8]          36,864
      BatchNorm2d-43             [-1, 64, 8, 8]             128
           Conv2d-44             [-1, 64, 8, 8]          36,864
      BatchNorm2d-45             [-1, 64, 8, 8]             128
       BasicBlock-46             [-1, 64, 8, 8]               0
           Conv2d-47             [-1, 64, 8, 8]          36,864
      BatchNorm2d-48             [-1, 64, 8, 8]             128
           Conv2d-49             [-1, 64, 8, 8]          36,864
      BatchNorm2d-50             [-1, 64, 8, 8]             128
       BasicBlock-51             [-1, 64, 8, 8]               0
           Linear-52                   [-1, 10]             650
================================================================
Total params: 272,474
Trainable params: 272,474
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.01
Forward/backward pass size (MB): 3.72
Params size (MB): 1.04
Estimated Total Size (MB): 4.77
----------------------------------------------------------------
```

In [ ]:
```python
def get_data(n=1000):
    data_path = 'cifar-10-batches-py/'

    train_data = np.empty((50000, 3072), dtype=np.uint8)
    train_labels = np.empty((50000,), dtype=np.int64)
    for i in range(1, 6):
        train_batch = unpickle(data_path + 'data_batch_' + str(i))
        train_data[(i - 1) * 10000: i * 10000, :] = train_batch[b'data']
        train_labels[(i - 1) * 10000: i * 10000] = train_batch[b'labels']


    test_batch = unpickle(data_path + 'test_batch')
    test_data = test_batch[b'data']
    test_labels = np.array(test_batch[b'labels'])

    # Sample n examples uniformly at random for each class from the training set
    classes = np.unique(train_labels)
    sampled_train_data = []
    sampled_train_labels = []

    for class_label in classes:
        indices = np.where(train_labels == class_label)[0]
        np.random.shuffle(indices)
        sampled_indices = indices[:n]
        sampled_train_data.extend(train_data[sampled_indices])
        sampled_train_labels.extend(train_labels[sampled_indices])

    indices = np.array(range(len(sampled_train_data)))
    np.random.shuffle(indices)
    sampled_train_data = np.array(sampled_train_data)[indices]
    sampled_train_labels = np.array(sampled_train_labels)[indices]

    # normalize features (zero mean and unit variance)
    sampled_train_data = preprocess_data(sampled_train_data)
    test_data = preprocess_data(test_data)

    return sampled_train_data, test_data, sampled_train_labels, test_labels
```

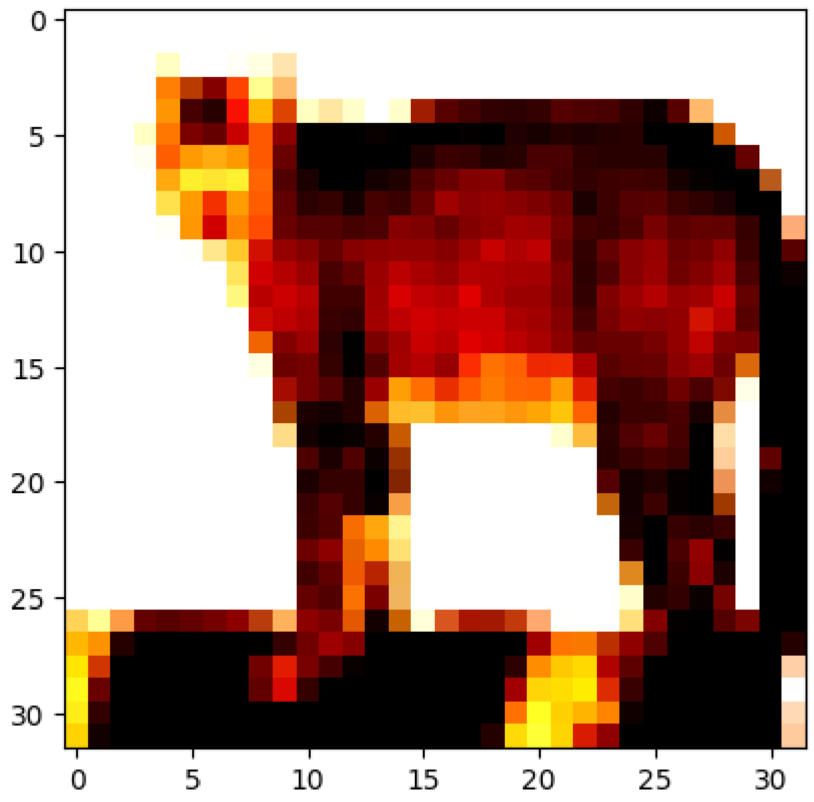
In [ ]:
```python
sampled_train_data, test_data, sampled_train_labels, test_labels = get_data()

print("Sampled Train Data Shape:", sampled_train_data.shape)
print("Sampled Train Labels Shape:", sampled_train_labels.shape)
print("Test Data Shape:", test_data.shape)
print("Test Labels Shape:", test_labels.shape)

train_dataset = TensorDataset(torch.from_numpy(sampled_train_data), torch.from_numpy(sampled_train_labels))
test_dataset = TensorDataset(torch.from_numpy(test_data), torch.from_numpy(test_labels))

batch_size = 64
train_loader = DataLoader(train_dataset, batch_size=batch_size, shuffle=True)
test_loader = DataLoader(test_dataset, batch_size=batch_size, shuffle=False)
```

```
Sampled Train Data Shape: (10000, 3, 32, 32)
Sampled Train Labels Shape: (10000,)
Test Data Shape: (10000, 3, 32, 32)
Test Labels Shape: (10000,)
```

In [ ]:
```python
plt.imshow(sampled_train_data[3].transpose( 1, 2, 0))
```

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

Out[ ]: <matplotlib.image.AxesImage at 0x7f2944490370>



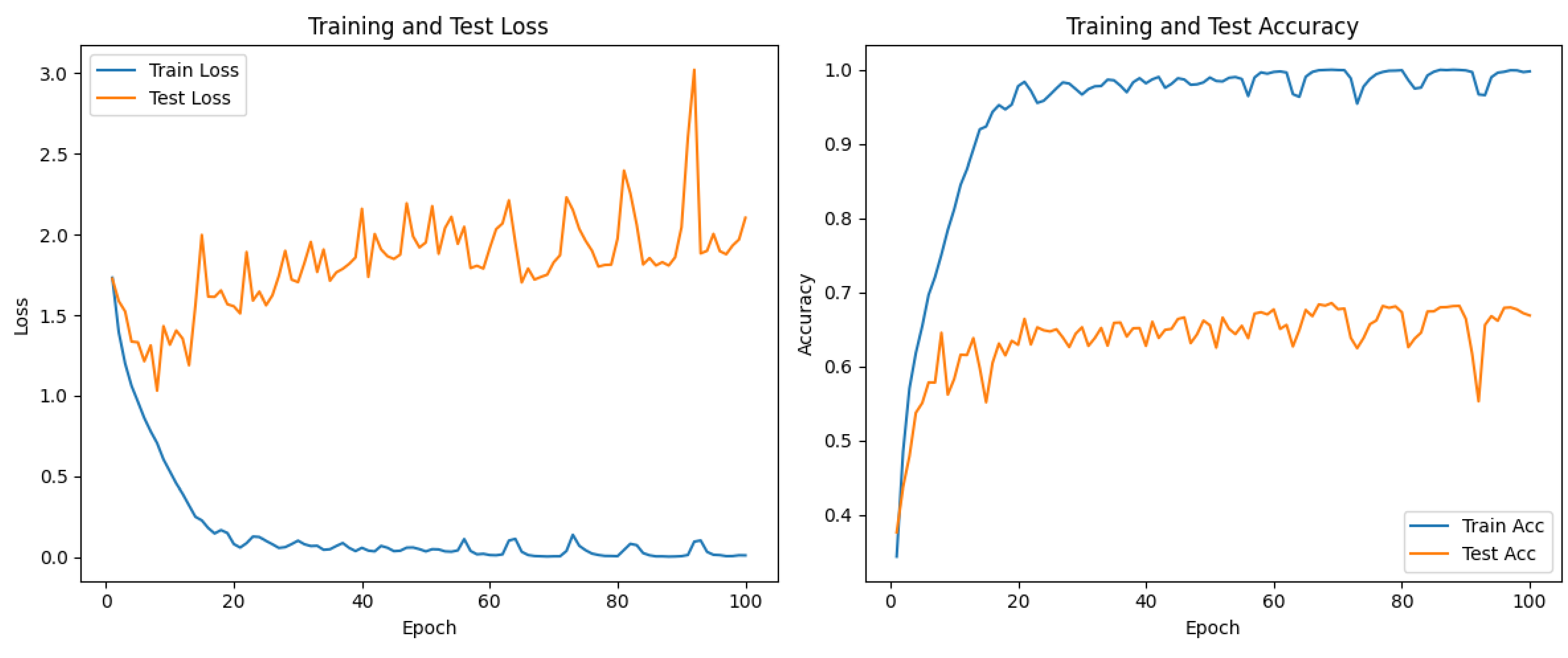
In [ ]:

1. (3 pts) Train your Resnet model without augmentation and report the results.

In [ ]:
```
history1 = run_model(model, train_loader, test_loader)
```

```
Epoch [1/100], Train Loss: 1.7321, Train Acc: 0.344, Test Loss: 1.7259, Test Acc: 0.376
Epoch [2/100], Train Loss: 1.3953, Train Acc: 0.485, Test Loss: 1.5875, Test Acc: 0.437
Epoch [3/100], Train Loss: 1.1995, Train Acc: 0.570, Test Loss: 1.5227, Test Acc: 0.479
Epoch [4/100], Train Loss: 1.0610, Train Acc: 0.618, Test Loss: 1.3357, Test Acc: 0.537
Epoch [5/100], Train Loss: 0.9621, Train Acc: 0.654, Test Loss: 1.3311, Test Acc: 0.550
Epoch [6/100], Train Loss: 0.8608, Train Acc: 0.697, Test Loss: 1.2135, Test Acc: 0.578
Epoch [7/100], Train Loss: 0.7780, Train Acc: 0.721, Test Loss: 1.3121, Test Acc: 0.578
Epoch [8/100], Train Loss: 0.7053, Train Acc: 0.751, Test Loss: 1.0307, Test Acc: 0.645
Epoch [9/100], Train Loss: 0.6040, Train Acc: 0.784, Test Loss: 1.4318, Test Acc: 0.562
Epoch [10/100], Train Loss: 0.5295, Train Acc: 0.812, Test Loss: 1.3162, Test Acc: 0.583
Epoch [11/100], Train Loss: 0.4546, Train Acc: 0.845, Test Loss: 1.4046, Test Acc: 0.616
Epoch [12/100], Train Loss: 0.3920, Train Acc: 0.866, Test Loss: 1.3540, Test Acc: 0.615
Epoch [13/100], Train Loss: 0.3200, Train Acc: 0.892, Test Loss: 1.1884, Test Acc: 0.638
Epoch [14/100], Train Loss: 0.2491, Train Acc: 0.919, Test Loss: 1.5616, Test Acc: 0.598
Epoch [15/100], Train Loss: 0.2269, Train Acc: 0.923, Test Loss: 1.9984, Test Acc: 0.551
Epoch [16/100], Train Loss: 0.1796, Train Acc: 0.943, Test Loss: 1.6149, Test Acc: 0.605
Epoch [17/100], Train Loss: 0.1457, Train Acc: 0.952, Test Loss: 1.6132, Test Acc: 0.631
Epoch [18/100], Train Loss: 0.1666, Train Acc: 0.946, Test Loss: 1.6537, Test Acc: 0.615
Epoch [19/100], Train Loss: 0.1486, Train Acc: 0.953, Test Loss: 1.5676, Test Acc: 0.634
Epoch [20/100], Train Loss: 0.0801, Train Acc: 0.978, Test Loss: 1.5550, Test Acc: 0.629
Epoch [21/100], Train Loss: 0.0580, Train Acc: 0.983, Test Loss: 1.5105, Test Acc: 0.664
Epoch [22/100], Train Loss: 0.0855, Train Acc: 0.971, Test Loss: 1.8923, Test Acc: 0.629
Epoch [23/100], Train Loss: 0.1267, Train Acc: 0.955, Test Loss: 1.5902, Test Acc: 0.653
Epoch [24/100], Train Loss: 0.1236, Train Acc: 0.958, Test Loss: 1.6464, Test Acc: 0.649
Epoch [25/100], Train Loss: 0.0994, Train Acc: 0.966, Test Loss: 1.5608, Test Acc: 0.647
Epoch [26/100], Train Loss: 0.0785, Train Acc: 0.975, Test Loss: 1.6235, Test Acc: 0.650
Epoch [27/100], Train Loss: 0.0558, Train Acc: 0.983, Test Loss: 1.7435, Test Acc: 0.639
Epoch [28/100], Train Loss: 0.0607, Train Acc: 0.981, Test Loss: 1.8997, Test Acc: 0.626
Epoch [29/100], Train Loss: 0.0797, Train Acc: 0.974, Test Loss: 1.7205, Test Acc: 0.644
Epoch [30/100], Train Loss: 0.1010, Train Acc: 0.966, Test Loss: 1.7049, Test Acc: 0.653
Epoch [31/100], Train Loss: 0.0788, Train Acc: 0.974, Test Loss: 1.8243, Test Acc: 0.627
Epoch [32/100], Train Loss: 0.0676, Train Acc: 0.977, Test Loss: 1.9552, Test Acc: 0.638
Epoch [33/100], Train Loss: 0.0692, Train Acc: 0.978, Test Loss: 1.7679, Test Acc: 0.652
Epoch [34/100], Train Loss: 0.0443, Train Acc: 0.986, Test Loss: 1.9080, Test Acc: 0.628
Epoch [35/100], Train Loss: 0.0476, Train Acc: 0.985, Test Loss: 1.7135, Test Acc: 0.659
Epoch [36/100], Train Loss: 0.0680, Train Acc: 0.978, Test Loss: 1.7658, Test Acc: 0.659
Epoch [37/100], Train Loss: 0.0863, Train Acc: 0.969, Test Loss: 1.7874, Test Acc: 0.640
Epoch [38/100], Train Loss: 0.0562, Train Acc: 0.983, Test Loss: 1.8183, Test Acc: 0.651
Epoch [39/100], Train Loss: 0.0361, Train Acc: 0.988, Test Loss: 1.8582, Test Acc: 0.652
Epoch [40/100], Train Loss: 0.0569, Train Acc: 0.981, Test Loss: 2.1607, Test Acc: 0.627
Epoch [41/100], Train Loss: 0.0391, Train Acc: 0.987, Test Loss: 1.7373, Test Acc: 0.660
Epoch [42/100], Train Loss: 0.0344, Train Acc: 0.990, Test Loss: 2.0042, Test Acc: 0.638
Epoch [43/100], Train Loss: 0.0683, Train Acc: 0.975, Test Loss: 1.9081, Test Acc: 0.649
Epoch [44/100], Train Loss: 0.0569, Train Acc: 0.981, Test Loss: 1.8658, Test Acc: 0.650
Epoch [45/100], Train Loss: 0.0358, Train Acc: 0.988, Test Loss: 1.8492, Test Acc: 0.664
Epoch [46/100], Train Loss: 0.0383, Train Acc: 0.987, Test Loss: 1.8760, Test Acc: 0.666
Epoch [47/100], Train Loss: 0.0576, Train Acc: 0.979, Test Loss: 2.1947, Test Acc: 0.631
Epoch [48/100], Train Loss: 0.0586, Train Acc: 0.980, Test Loss: 1.9895, Test Acc: 0.643
Epoch [49/100], Train Loss: 0.0482, Train Acc: 0.983, Test Loss: 1.9203, Test Acc: 0.662
Epoch [50/100], Train Loss: 0.0341, Train Acc: 0.989, Test Loss: 1.9516, Test Acc: 0.655
Epoch [51/100], Train Loss: 0.0479, Train Acc: 0.985, Test Loss: 2.1771, Test Acc: 0.625
Epoch [52/100], Train Loss: 0.0465, Train Acc: 0.984, Test Loss: 1.8807, Test Acc: 0.666
Epoch [53/100], Train Loss: 0.0345, Train Acc: 0.989, Test Loss: 2.0410, Test Acc: 0.650
Epoch [54/100], Train Loss: 0.0325, Train Acc: 0.990, Test Loss: 2.1112, Test Acc: 0.643
Epoch [55/100], Train Loss: 0.0404, Train Acc: 0.987, Test Loss: 1.9424, Test Acc: 0.655
Epoch [56/100], Train Loss: 0.1110, Train Acc: 0.964, Test Loss: 2.0490, Test Acc: 0.638
Epoch [57/100], Train Loss: 0.0369, Train Acc: 0.989, Test Loss: 1.7920, Test Acc: 0.671
Epoch [58/100], Train Loss: 0.0161, Train Acc: 0.996, Test Loss: 1.8070, Test Acc: 0.673
Epoch [59/100], Train Loss: 0.0192, Train Acc: 0.994, Test Loss: 1.7890, Test Acc: 0.670
Epoch [60/100], Train Loss: 0.0115, Train Acc: 0.997, Test Loss: 1.9164, Test Acc: 0.677
Epoch [61/100], Train Loss: 0.0101, Train Acc: 0.997, Test Loss: 2.0341, Test Acc: 0.650
Epoch [62/100], Train Loss: 0.0159, Train Acc: 0.996, Test Loss: 2.0697, Test Acc: 0.656
Epoch [63/100], Train Loss: 0.1018, Train Acc: 0.967, Test Loss: 2.2134, Test Acc: 0.627
Epoch [64/100], Train Loss: 0.1122, Train Acc: 0.963, Test Loss: 1.9515, Test Acc: 0.649
Epoch [65/100], Train Loss: 0.0321, Train Acc: 0.990, Test Loss: 1.7030, Test Acc: 0.676
Epoch [66/100], Train Loss: 0.0116, Train Acc: 0.997, Test Loss: 1.7894, Test Acc: 0.668
Epoch [67/100], Train Loss: 0.0056, Train Acc: 0.999, Test Loss: 1.7209, Test Acc: 0.683
Epoch [68/100], Train Loss: 0.0041, Train Acc: 0.999, Test Loss: 1.7371, Test Acc: 0.682
Epoch [69/100], Train Loss: 0.0026, Train Acc: 1.000, Test Loss: 1.7516, Test Acc: 0.685
Epoch [70/100], Train Loss: 0.0039, Train Acc: 0.999, Test Loss: 1.8278, Test Acc: 0.677
Epoch [71/100], Train Loss: 0.0040, Train Acc: 0.999, Test Loss: 1.8722, Test Acc: 0.678
Epoch [72/100], Train Loss: 0.0368, Train Acc: 0.988, Test Loss: 2.2317, Test Acc: 0.638
Epoch [73/100], Train Loss: 0.1375, Train Acc: 0.954, Test Loss: 2.1538, Test Acc: 0.624
Epoch [74/100], Train Loss: 0.0694, Train Acc: 0.977, Test Loss: 2.0358, Test Acc: 0.638
Epoch [75/100], Train Loss: 0.0413, Train Acc: 0.987, Test Loss: 1.9615, Test Acc: 0.657
Epoch [76/100], Train Loss: 0.0206, Train Acc: 0.994, Test Loss: 1.8999, Test Acc: 0.662
Epoch [77/100], Train Loss: 0.0119, Train Acc: 0.997, Test Loss: 1.8016, Test Acc: 0.681
Epoch [78/100], Train Loss: 0.0063, Train Acc: 0.998, Test Loss: 1.8117, Test Acc: 0.679
Epoch [79/100], Train Loss: 0.0058, Train Acc: 0.999, Test Loss: 1.8136, Test Acc: 0.681
Epoch [80/100], Train Loss: 0.0047, Train Acc: 0.999, Test Loss: 1.9762, Test Acc: 0.673
Epoch [81/100], Train Loss: 0.0431, Train Acc: 0.986, Test Loss: 2.3984, Test Acc: 0.626
Epoch [82/100], Train Loss: 0.0815, Train Acc: 0.974, Test Loss: 2.2565, Test Acc: 0.637
Epoch [83/100], Train Loss: 0.0730, Train Acc: 0.976, Test Loss: 2.0598, Test Acc: 0.645
Epoch [84/100], Train Loss: 0.0237, Train Acc: 0.992, Test Loss: 1.8148, Test Acc: 0.674
Epoch [85/100], Train Loss: 0.0100, Train Acc: 0.997, Test Loss: 1.8545, Test Acc: 0.674
Epoch [86/100], Train Loss: 0.0035, Train Acc: 1.000, Test Loss: 1.8084, Test Acc: 0.680
Epoch [87/100], Train Loss: 0.0036, Train Acc: 0.999, Test Loss: 1.8289, Test Acc: 0.680
Epoch [88/100], Train Loss: 0.0019, Train Acc: 1.000, Test Loss: 1.8080, Test Acc: 0.681
Epoch [89/100], Train Loss: 0.0028, Train Acc: 0.999, Test Loss: 1.8592, Test Acc: 0.681
Epoch [90/100], Train Loss: 0.0047, Train Acc: 0.999, Test Loss: 2.0458, Test Acc: 0.664
Epoch [91/100], Train Loss: 0.0119, Train Acc: 0.997, Test Loss: 2.6002, Test Acc: 0.616
Epoch [92/100], Train Loss: 0.0949, Train Acc: 0.966, Test Loss: 3.0225, Test Acc: 0.553
Epoch [93/100], Train Loss: 0.1024, Train Acc: 0.965, Test Loss: 1.8832, Test Acc: 0.656
Epoch [94/100], Train Loss: 0.0315, Train Acc: 0.990, Test Loss: 1.8996, Test Acc: 0.668
Epoch [95/100], Train Loss: 0.0131, Train Acc: 0.996, Test Loss: 2.0050, Test Acc: 0.661
Epoch [96/100], Train Loss: 0.0112, Train Acc: 0.997, Test Loss: 1.8973, Test Acc: 0.679
Epoch [97/100], Train Loss: 0.0048, Train Acc: 0.999, Test Loss: 1.8782, Test Acc: 0.679
Epoch [98/100], Train Loss: 0.0054, Train Acc: 0.999, Test Loss: 1.9325, Test Acc: 0.677
Epoch [99/100], Train Loss: 0.0103, Train Acc: 0.997, Test Loss: 1.9698, Test Acc: 0.672
Epoch [100/100], Train Loss: 0.0096, Train Acc: 0.998, Test Loss: 2.1048, Test Acc: 0.669
```

In [ ]:
```
plot_history(history1)
```



```
Final Test Accuracy is 0.6686
```

2. (4 pts) Implement mixup and report the results for α = 0.2 and α = 0.4

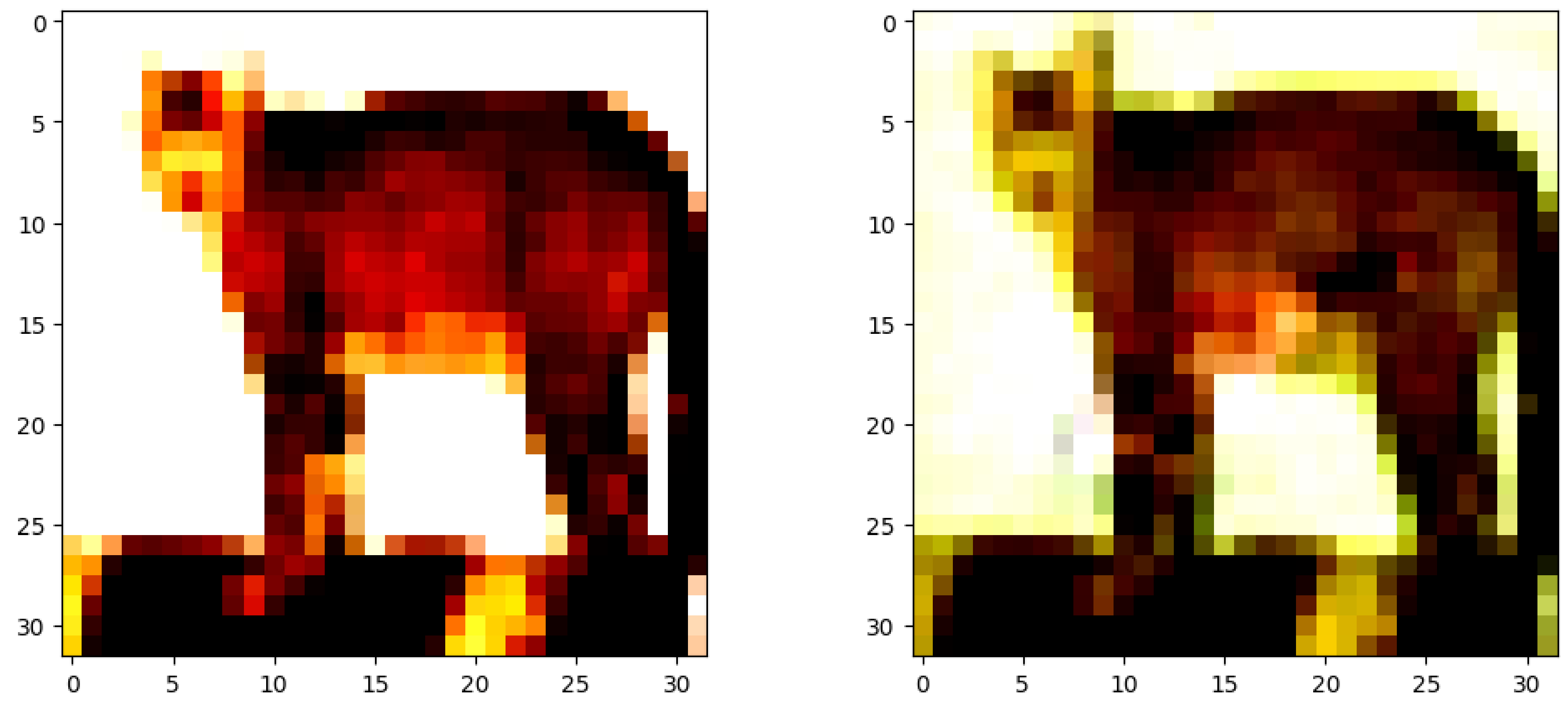
In [ ]:
```python
def mixup(img1, img2, lb1, lb2, alpha_value):
    lam = np.random.beta(alpha_value, alpha_value)
    mixed_image = lam * img1 + (1 - lam) * img2
    mixed_label = lam * lb1 + (1 - lam) * lb2
    mixed_image = np.array(mixed_image)
    mixed_label = np.array(mixed_label)

    return mixed_image, mixed_label
```

In [ ]:
```python
def apply_mixup_minibatch(minibatch_images, minibatch_labels, mask_size):
    for i in range(minibatch_images.shape[0]):
        idx = np.random.randint(minibatch_images.shape[0])
        img1 = minibatch_images[i]
        img2 = minibatch_images[idx]
        lb1 = minibatch_labels[i]
        lb2 = minibatch_labels[idx]
        new_im, new_lb = mixup(img1, img2, lb1, lb2, mask_size)
        minibatch_images[i] = torch.from_numpy(new_im).float()
        minibatch_labels[i] = torch.from_numpy(new_lb).float()

    return minibatch_images, minibatch_labels
```

In [ ]:
```python
sampled_train_data, test_data, sampled_train_labels, test_labels = get_data()
```

In [ ]:
```python
i=3
plt.figure(figsize=(12,5))
plt.subplot(121)
plt.imshow(sampled_train_data[i].transpose( 1, 2, 0))

plt.subplot(122)
im, lb = mixup(sampled_train_data[i].copy(), sampled_train_data[i*2].copy(), sampled_train_labels[i].copy(), sampled_train_labels[i*2].copy(), 0.2)
plt.imshow(im.transpose( 1, 2, 0))

plt.show()
```

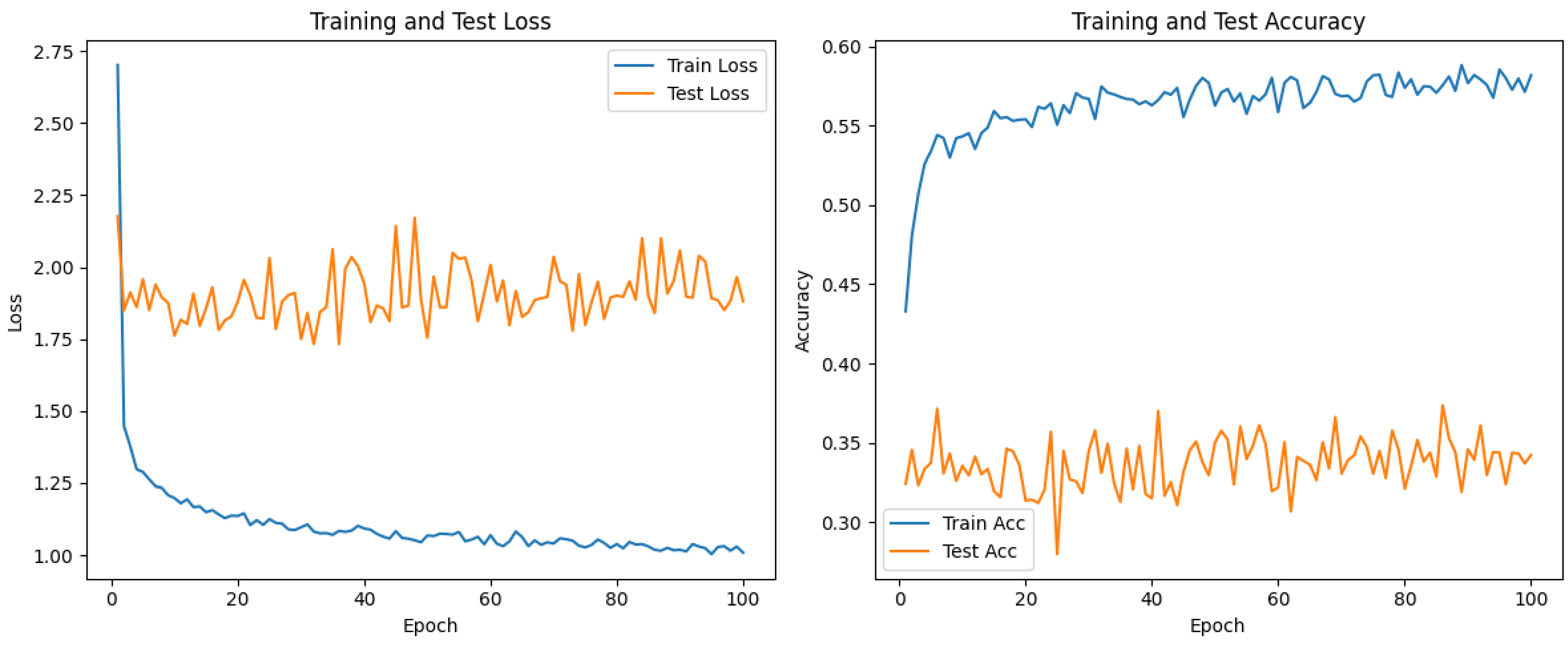
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



In [ ]:
```python
history2 = run_model(model, train_loader, test_loader, data_aug='mixup', alpha=0.2)
```

```
Epoch [1/100], Train Loss: 2.7010, Train Acc: 0.433, Test Loss: 2.1768, Test Acc: 0.324
Epoch [2/100], Train Loss: 1.4474, Train Acc: 0.481, Test Loss: 1.8494, Test Acc: 0.346
Epoch [3/100], Train Loss: 1.3768, Train Acc: 0.507, Test Loss: 1.9118, Test Acc: 0.323
Epoch [4/100], Train Loss: 1.2973, Train Acc: 0.526, Test Loss: 1.8607, Test Acc: 0.334
Epoch [5/100], Train Loss: 1.2880, Train Acc: 0.534, Test Loss: 1.9578, Test Acc: 0.337
Epoch [6/100], Train Loss: 1.2622, Train Acc: 0.544, Test Loss: 1.8506, Test Acc: 0.371
Epoch [7/100], Train Loss: 1.2387, Train Acc: 0.542, Test Loss: 1.9392, Test Acc: 0.331
Epoch [8/100], Train Loss: 1.2334, Train Acc: 0.530, Test Loss: 1.8943, Test Acc: 0.343
Epoch [9/100], Train Loss: 1.2080, Train Acc: 0.542, Test Loss: 1.8753, Test Acc: 0.326
Epoch [10/100], Train Loss: 1.1984, Train Acc: 0.543, Test Loss: 1.7618, Test Acc: 0.336
Epoch [11/100], Train Loss: 1.1800, Train Acc: 0.545, Test Loss: 1.8171, Test Acc: 0.330
Epoch [12/100], Train Loss: 1.1939, Train Acc: 0.535, Test Loss: 1.8023, Test Acc: 0.341
Epoch [13/100], Train Loss: 1.1671, Train Acc: 0.545, Test Loss: 1.9079, Test Acc: 0.330
Epoch [14/100], Train Loss: 1.1692, Train Acc: 0.549, Test Loss: 1.7954, Test Acc: 0.334
Epoch [15/100], Train Loss: 1.1494, Train Acc: 0.559, Test Loss: 1.8574, Test Acc: 0.320
Epoch [16/100], Train Loss: 1.1562, Train Acc: 0.555, Test Loss: 1.9299, Test Acc: 0.316
Epoch [17/100], Train Loss: 1.1413, Train Acc: 0.555, Test Loss: 1.7816, Test Acc: 0.346
Epoch [18/100], Train Loss: 1.1285, Train Acc: 0.553, Test Loss: 1.8152, Test Acc: 0.345
Epoch [19/100], Train Loss: 1.1372, Train Acc: 0.554, Test Loss: 1.8281, Test Acc: 0.336
Epoch [20/100], Train Loss: 1.1360, Train Acc: 0.554, Test Loss: 1.8782, Test Acc: 0.314
Epoch [21/100], Train Loss: 1.1452, Train Acc: 0.549, Test Loss: 1.9556, Test Acc: 0.314
Epoch [22/100], Train Loss: 1.1044, Train Acc: 0.562, Test Loss: 1.9028, Test Acc: 0.312
Epoch [23/100], Train Loss: 1.1213, Train Acc: 0.560, Test Loss: 1.8233, Test Acc: 0.321
Epoch [24/100], Train Loss: 1.1051, Train Acc: 0.564, Test Loss: 1.8215, Test Acc: 0.357
Epoch [25/100], Train Loss: 1.1252, Train Acc: 0.550, Test Loss: 2.0315, Test Acc: 0.280
Epoch [26/100], Train Loss: 1.1122, Train Acc: 0.563, Test Loss: 1.7846, Test Acc: 0.345
Epoch [27/100], Train Loss: 1.1092, Train Acc: 0.558, Test Loss: 1.8806, Test Acc: 0.327
Epoch [28/100], Train Loss: 1.0892, Train Acc: 0.570, Test Loss: 1.9026, Test Acc: 0.326
Epoch [29/100], Train Loss: 1.0872, Train Acc: 0.568, Test Loss: 1.9105, Test Acc: 0.319
Epoch [30/100], Train Loss: 1.0967, Train Acc: 0.567, Test Loss: 1.7494, Test Acc: 0.345
Epoch [31/100], Train Loss: 1.1069, Train Acc: 0.554, Test Loss: 1.8407, Test Acc: 0.358
Epoch [32/100], Train Loss: 1.0818, Train Acc: 0.575, Test Loss: 1.7325, Test Acc: 0.331
Epoch [33/100], Train Loss: 1.0759, Train Acc: 0.571, Test Loss: 1.8445, Test Acc: 0.349
Epoch [34/100], Train Loss: 1.0765, Train Acc: 0.570, Test Loss: 1.8611, Test Acc: 0.325
Epoch [35/100], Train Loss: 1.0705, Train Acc: 0.568, Test Loss: 2.0622, Test Acc: 0.313
Epoch [36/100], Train Loss: 1.0842, Train Acc: 0.567, Test Loss: 1.7325, Test Acc: 0.346
Epoch [37/100], Train Loss: 1.0808, Train Acc: 0.566, Test Loss: 1.9944, Test Acc: 0.321
Epoch [38/100], Train Loss: 1.0853, Train Acc: 0.563, Test Loss: 2.0348, Test Acc: 0.348
Epoch [39/100], Train Loss: 1.1018, Train Acc: 0.565, Test Loss: 2.0030, Test Acc: 0.318
Epoch [40/100], Train Loss: 1.0925, Train Acc: 0.563, Test Loss: 1.9431, Test Acc: 0.315
Epoch [41/100], Train Loss: 1.0883, Train Acc: 0.566, Test Loss: 1.8084, Test Acc: 0.370
Epoch [42/100], Train Loss: 1.0738, Train Acc: 0.571, Test Loss: 1.8666, Test Acc: 0.317
Epoch [43/100], Train Loss: 1.0642, Train Acc: 0.569, Test Loss: 1.8562, Test Acc: 0.325
Epoch [44/100], Train Loss: 1.0574, Train Acc: 0.574, Test Loss: 1.8124, Test Acc: 0.311
Epoch [45/100], Train Loss: 1.0832, Train Acc: 0.555, Test Loss: 2.1429, Test Acc: 0.332
Epoch [46/100], Train Loss: 1.0599, Train Acc: 0.566, Test Loss: 1.8595, Test Acc: 0.345
Epoch [47/100], Train Loss: 1.0569, Train Acc: 0.575, Test Loss: 1.8662, Test Acc: 0.351
Epoch [48/100], Train Loss: 1.0517, Train Acc: 0.580, Test Loss: 2.1714, Test Acc: 0.338
Epoch [49/100], Train Loss: 1.0448, Train Acc: 0.577, Test Loss: 1.8854, Test Acc: 0.330
Epoch [50/100], Train Loss: 1.0684, Train Acc: 0.562, Test Loss: 1.7544, Test Acc: 0.350
Epoch [51/100], Train Loss: 1.0662, Train Acc: 0.571, Test Loss: 1.9669, Test Acc: 0.358
Epoch [52/100], Train Loss: 1.0744, Train Acc: 0.573, Test Loss: 1.8601, Test Acc: 0.352
Epoch [53/100], Train Loss: 1.0735, Train Acc: 0.565, Test Loss: 1.8604, Test Acc: 0.324
Epoch [54/100], Train Loss: 1.0713, Train Acc: 0.570, Test Loss: 2.0491, Test Acc: 0.360
Epoch [55/100], Train Loss: 1.0804, Train Acc: 0.557, Test Loss: 2.0283, Test Acc: 0.340
Epoch [56/100], Train Loss: 1.0482, Train Acc: 0.569, Test Loss: 2.0333, Test Acc: 0.348
Epoch [57/100], Train Loss: 1.0541, Train Acc: 0.566, Test Loss: 1.9537, Test Acc: 0.361
Epoch [58/100], Train Loss: 1.0638, Train Acc: 0.570, Test Loss: 1.8118, Test Acc: 0.349
Epoch [59/100], Train Loss: 1.0376, Train Acc: 0.580, Test Loss: 1.9078, Test Acc: 0.320
Epoch [60/100], Train Loss: 1.0693, Train Acc: 0.558, Test Loss: 2.0076, Test Acc: 0.322
Epoch [61/100], Train Loss: 1.0396, Train Acc: 0.577, Test Loss: 1.8802, Test Acc: 0.350
Epoch [62/100], Train Loss: 1.0311, Train Acc: 0.581, Test Loss: 1.9536, Test Acc: 0.307
Epoch [63/100], Train Loss: 1.0479, Train Acc: 0.578, Test Loss: 1.7981, Test Acc: 0.341
Epoch [64/100], Train Loss: 1.0823, Train Acc: 0.561, Test Loss: 1.9168, Test Acc: 0.339
Epoch [65/100], Train Loss: 1.0629, Train Acc: 0.564, Test Loss: 1.8268, Test Acc: 0.336
Epoch [66/100], Train Loss: 1.0312, Train Acc: 0.571, Test Loss: 1.8435, Test Acc: 0.327
Epoch [67/100], Train Loss: 1.0513, Train Acc: 0.581, Test Loss: 1.8855, Test Acc: 0.350
Epoch [68/100], Train Loss: 1.0360, Train Acc: 0.579, Test Loss: 1.8914, Test Acc: 0.334
Epoch [69/100], Train Loss: 1.0446, Train Acc: 0.570, Test Loss: 1.8964, Test Acc: 0.366
Epoch [70/100], Train Loss: 1.0403, Train Acc: 0.568, Test Loss: 2.0361, Test Acc: 0.331
Epoch [71/100], Train Loss: 1.0585, Train Acc: 0.569, Test Loss: 1.9509, Test Acc: 0.339
Epoch [72/100], Train Loss: 1.0553, Train Acc: 0.565, Test Loss: 1.9384, Test Acc: 0.342
Epoch [73/100], Train Loss: 1.0500, Train Acc: 0.567, Test Loss: 1.7787, Test Acc: 0.354
Epoch [74/100], Train Loss: 1.0329, Train Acc: 0.578, Test Loss: 1.9753, Test Acc: 0.347
Epoch [75/100], Train Loss: 1.0268, Train Acc: 0.582, Test Loss: 1.7987, Test Acc: 0.331
Epoch [76/100], Train Loss: 1.0354, Train Acc: 0.582, Test Loss: 1.8777, Test Acc: 0.345
Epoch [77/100], Train Loss: 1.0544, Train Acc: 0.569, Test Loss: 1.9491, Test Acc: 0.328
Epoch [78/100], Train Loss: 1.0422, Train Acc: 0.568, Test Loss: 1.8195, Test Acc: 0.358
Epoch [79/100], Train Loss: 1.0252, Train Acc: 0.583, Test Loss: 1.8942, Test Acc: 0.346
Epoch [80/100], Train Loss: 1.0385, Train Acc: 0.574, Test Loss: 1.9009, Test Acc: 0.321
Epoch [81/100], Train Loss: 1.0236, Train Acc: 0.579, Test Loss: 1.8964, Test Acc: 0.336
Epoch [82/100], Train Loss: 1.0459, Train Acc: 0.569, Test Loss: 1.9502, Test Acc: 0.352
Epoch [83/100], Train Loss: 1.0367, Train Acc: 0.575, Test Loss: 1.8866, Test Acc: 0.338
Epoch [84/100], Train Loss: 1.0383, Train Acc: 0.575, Test Loss: 2.1009, Test Acc: 0.344
Epoch [85/100], Train Loss: 1.0303, Train Acc: 0.571, Test Loss: 1.8972, Test Acc: 0.329
Epoch [86/100], Train Loss: 1.0186, Train Acc: 0.575, Test Loss: 1.8397, Test Acc: 0.374
Epoch [87/100], Train Loss: 1.0152, Train Acc: 0.581, Test Loss: 2.0996, Test Acc: 0.353
Epoch [88/100], Train Loss: 1.0254, Train Acc: 0.572, Test Loss: 1.9078, Test Acc: 0.344
Epoch [89/100], Train Loss: 1.0168, Train Acc: 0.588, Test Loss: 1.9516, Test Acc: 0.319
Epoch [90/100], Train Loss: 1.0186, Train Acc: 0.577, Test Loss: 2.0580, Test Acc: 0.346
Epoch [91/100], Train Loss: 1.0130, Train Acc: 0.582, Test Loss: 1.8970, Test Acc: 0.339
Epoch [92/100], Train Loss: 1.0385, Train Acc: 0.579, Test Loss: 1.8936, Test Acc: 0.361
Epoch [93/100], Train Loss: 1.0299, Train Acc: 0.576, Test Loss: 2.0385, Test Acc: 0.330
Epoch [94/100], Train Loss: 1.0243, Train Acc: 0.568, Test Loss: 2.0197, Test Acc: 0.344
Epoch [95/100], Train Loss: 1.0033, Train Acc: 0.585, Test Loss: 1.8918, Test Acc: 0.344
Epoch [96/100], Train Loss: 1.0281, Train Acc: 0.580, Test Loss: 1.8845, Test Acc: 0.324
Epoch [97/100], Train Loss: 1.0311, Train Acc: 0.573, Test Loss: 1.8507, Test Acc: 0.344
Epoch [98/100], Train Loss: 1.0156, Train Acc: 0.580, Test Loss: 1.8823, Test Acc: 0.343
Epoch [99/100], Train Loss: 1.0296, Train Acc: 0.571, Test Loss: 1.9652, Test Acc: 0.337
Epoch [100/100], Train Loss: 1.0087, Train Acc: 0.582, Test Loss: 1.8812, Test Acc: 0.342
```

In [ ]:
```
plot_history(history2)
```

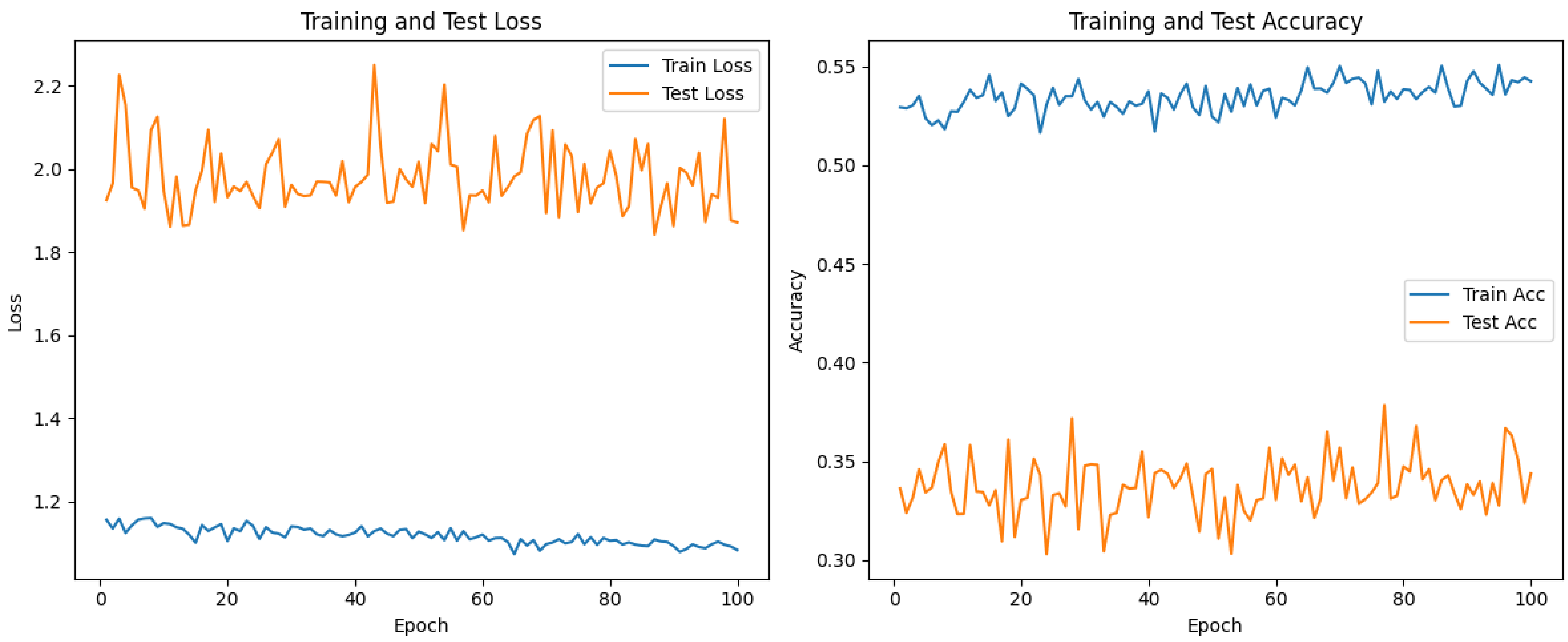


```
Final Test Accuracy is 0.3424
```

In [ ]:
```
history22 = run_model(model, train_loader, test_loader, data_aug='mixup', alpha=0.4)
```

```
Epoch [1/100], Train Loss: 1.1550, Train Acc: 0.529, Test Loss: 1.9249, Test Acc: 0.336
Epoch [2/100], Train Loss: 1.1342, Train Acc: 0.529, Test Loss: 1.9654, Test Acc: 0.324
Epoch [3/100], Train Loss: 1.1579, Train Acc: 0.530, Test Loss: 2.2257, Test Acc: 0.332
Epoch [4/100], Train Loss: 1.1234, Train Acc: 0.535, Test Loss: 2.1534, Test Acc: 0.346
Epoch [5/100], Train Loss: 1.1418, Train Acc: 0.524, Test Loss: 1.9551, Test Acc: 0.334
Epoch [6/100], Train Loss: 1.1555, Train Acc: 0.520, Test Loss: 1.9481, Test Acc: 0.337
Epoch [7/100], Train Loss: 1.1586, Train Acc: 0.523, Test Loss: 1.9041, Test Acc: 0.350
Epoch [8/100], Train Loss: 1.1597, Train Acc: 0.518, Test Loss: 2.0926, Test Acc: 0.359
Epoch [9/100], Train Loss: 1.1380, Train Acc: 0.527, Test Loss: 2.1252, Test Acc: 0.335
Epoch [10/100], Train Loss: 1.1472, Train Acc: 0.527, Test Loss: 1.9465, Test Acc: 0.323
Epoch [11/100], Train Loss: 1.1450, Train Acc: 0.532, Test Loss: 1.8610, Test Acc: 0.324
Epoch [12/100], Train Loss: 1.1370, Train Acc: 0.538, Test Loss: 1.9813, Test Acc: 0.358
Epoch [13/100], Train Loss: 1.1334, Train Acc: 0.534, Test Loss: 1.8632, Test Acc: 0.335
Epoch [14/100], Train Loss: 1.1190, Train Acc: 0.535, Test Loss: 1.8654, Test Acc: 0.335
Epoch [15/100], Train Loss: 1.0999, Train Acc: 0.546, Test Loss: 1.9484, Test Acc: 0.328
Epoch [16/100], Train Loss: 1.1425, Train Acc: 0.532, Test Loss: 1.9963, Test Acc: 0.336
Epoch [17/100], Train Loss: 1.1282, Train Acc: 0.537, Test Loss: 2.0937, Test Acc: 0.310
Epoch [18/100], Train Loss: 1.1368, Train Acc: 0.525, Test Loss: 1.9206, Test Acc: 0.361
Epoch [19/100], Train Loss: 1.1445, Train Acc: 0.529, Test Loss: 2.0363, Test Acc: 0.312
Epoch [20/100], Train Loss: 1.1040, Train Acc: 0.541, Test Loss: 1.9315, Test Acc: 0.331
Epoch [21/100], Train Loss: 1.1344, Train Acc: 0.539, Test Loss: 1.9575, Test Acc: 0.332
Epoch [22/100], Train Loss: 1.1279, Train Acc: 0.535, Test Loss: 1.9466, Test Acc: 0.351
Epoch [23/100], Train Loss: 1.1526, Train Acc: 0.516, Test Loss: 1.9687, Test Acc: 0.343
Epoch [24/100], Train Loss: 1.1405, Train Acc: 0.531, Test Loss: 1.9349, Test Acc: 0.303
Epoch [25/100], Train Loss: 1.1092, Train Acc: 0.539, Test Loss: 1.9053, Test Acc: 0.333
Epoch [26/100], Train Loss: 1.1374, Train Acc: 0.530, Test Loss: 2.0112, Test Acc: 0.334
Epoch [27/100], Train Loss: 1.1245, Train Acc: 0.535, Test Loss: 2.0377, Test Acc: 0.327
Epoch [28/100], Train Loss: 1.1218, Train Acc: 0.535, Test Loss: 2.0708, Test Acc: 0.372
Epoch [29/100], Train Loss: 1.1128, Train Acc: 0.544, Test Loss: 1.9087, Test Acc: 0.316
Epoch [30/100], Train Loss: 1.1393, Train Acc: 0.533, Test Loss: 1.9612, Test Acc: 0.348
Epoch [31/100], Train Loss: 1.1378, Train Acc: 0.528, Test Loss: 1.9398, Test Acc: 0.348
Epoch [32/100], Train Loss: 1.1312, Train Acc: 0.532, Test Loss: 1.9345, Test Acc: 0.348
Epoch [33/100], Train Loss: 1.1340, Train Acc: 0.524, Test Loss: 1.9365, Test Acc: 0.304
Epoch [34/100], Train Loss: 1.1202, Train Acc: 0.532, Test Loss: 1.9694, Test Acc: 0.323
Epoch [35/100], Train Loss: 1.1157, Train Acc: 0.529, Test Loss: 1.9689, Test Acc: 0.324
Epoch [36/100], Train Loss: 1.1307, Train Acc: 0.526, Test Loss: 1.9674, Test Acc: 0.338
Epoch [37/100], Train Loss: 1.1200, Train Acc: 0.532, Test Loss: 1.9360, Test Acc: 0.336
Epoch [38/100], Train Loss: 1.1155, Train Acc: 0.530, Test Loss: 2.0185, Test Acc: 0.337
Epoch [39/100], Train Loss: 1.1188, Train Acc: 0.531, Test Loss: 1.9199, Test Acc: 0.355
Epoch [40/100], Train Loss: 1.1246, Train Acc: 0.537, Test Loss: 1.9565, Test Acc: 0.322
Epoch [41/100], Train Loss: 1.1397, Train Acc: 0.517, Test Loss: 1.9689, Test Acc: 0.344
Epoch [42/100], Train Loss: 1.1151, Train Acc: 0.536, Test Loss: 1.9863, Test Acc: 0.346
Epoch [43/100], Train Loss: 1.1278, Train Acc: 0.534, Test Loss: 2.2495, Test Acc: 0.344
Epoch [44/100], Train Loss: 1.1339, Train Acc: 0.528, Test Loss: 2.0526, Test Acc: 0.337
Epoch [45/100], Train Loss: 1.1219, Train Acc: 0.536, Test Loss: 1.9185, Test Acc: 0.341
Epoch [46/100], Train Loss: 1.1156, Train Acc: 0.541, Test Loss: 1.9213, Test Acc: 0.349
Epoch [47/100], Train Loss: 1.1309, Train Acc: 0.529, Test Loss: 1.9995, Test Acc: 0.332
Epoch [48/100], Train Loss: 1.1326, Train Acc: 0.525, Test Loss: 1.9746, Test Acc: 0.315
Epoch [49/100], Train Loss: 1.1109, Train Acc: 0.540, Test Loss: 1.9568, Test Acc: 0.344
Epoch [50/100], Train Loss: 1.1267, Train Acc: 0.524, Test Loss: 2.0171, Test Acc: 0.346
Epoch [51/100], Train Loss: 1.1201, Train Acc: 0.522, Test Loss: 1.9178, Test Acc: 0.311
Epoch [52/100], Train Loss: 1.1113, Train Acc: 0.536, Test Loss: 2.0598, Test Acc: 0.332
Epoch [53/100], Train Loss: 1.1254, Train Acc: 0.527, Test Loss: 2.0422, Test Acc: 0.303
Epoch [54/100], Train Loss: 1.1063, Train Acc: 0.539, Test Loss: 2.2022, Test Acc: 0.338
Epoch [55/100], Train Loss: 1.1348, Train Acc: 0.530, Test Loss: 2.0103, Test Acc: 0.325
Epoch [56/100], Train Loss: 1.1051, Train Acc: 0.541, Test Loss: 2.0050, Test Acc: 0.320
Epoch [57/100], Train Loss: 1.1278, Train Acc: 0.530, Test Loss: 1.8523, Test Acc: 0.330
Epoch [58/100], Train Loss: 1.1082, Train Acc: 0.538, Test Loss: 1.9363, Test Acc: 0.331
Epoch [59/100], Train Loss: 1.1129, Train Acc: 0.539, Test Loss: 1.9356, Test Acc: 0.357
Epoch [60/100], Train Loss: 1.1195, Train Acc: 0.524, Test Loss: 1.9477, Test Acc: 0.331
Epoch [61/100], Train Loss: 1.1050, Train Acc: 0.534, Test Loss: 1.9196, Test Acc: 0.351
Epoch [62/100], Train Loss: 1.1110, Train Acc: 0.533, Test Loss: 2.0789, Test Acc: 0.343
Epoch [63/100], Train Loss: 1.1117, Train Acc: 0.530, Test Loss: 1.9352, Test Acc: 0.348
Epoch [64/100], Train Loss: 1.1014, Train Acc: 0.538, Test Loss: 1.9561, Test Acc: 0.330
Epoch [65/100], Train Loss: 1.0725, Train Acc: 0.550, Test Loss: 1.9818, Test Acc: 0.342
Epoch [66/100], Train Loss: 1.1085, Train Acc: 0.539, Test Loss: 1.9920, Test Acc: 0.321
Epoch [67/100], Train Loss: 1.0932, Train Acc: 0.539, Test Loss: 2.0839, Test Acc: 0.331
Epoch [68/100], Train Loss: 1.1062, Train Acc: 0.537, Test Loss: 2.1172, Test Acc: 0.365
Epoch [69/100], Train Loss: 1.0804, Train Acc: 0.542, Test Loss: 2.1266, Test Acc: 0.340
Epoch [70/100], Train Loss: 1.0963, Train Acc: 0.550, Test Loss: 1.8935, Test Acc: 0.357
Epoch [71/100], Train Loss: 1.1006, Train Acc: 0.542, Test Loss: 2.0922, Test Acc: 0.331
Epoch [72/100], Train Loss: 1.1084, Train Acc: 0.544, Test Loss: 1.8831, Test Acc: 0.347
Epoch [73/100], Train Loss: 1.0984, Train Acc: 0.544, Test Loss: 2.0584, Test Acc: 0.329
Epoch [74/100], Train Loss: 1.1019, Train Acc: 0.541, Test Loss: 2.0306, Test Acc: 0.331
Epoch [75/100], Train Loss: 1.1209, Train Acc: 0.531, Test Loss: 1.8958, Test Acc: 0.334
Epoch [76/100], Train Loss: 1.0963, Train Acc: 0.548, Test Loss: 2.0124, Test Acc: 0.339
Epoch [77/100], Train Loss: 1.1131, Train Acc: 0.532, Test Loss: 1.9168, Test Acc: 0.378
Epoch [78/100], Train Loss: 1.0948, Train Acc: 0.537, Test Loss: 1.9553, Test Acc: 0.331
Epoch [79/100], Train Loss: 1.1114, Train Acc: 0.533, Test Loss: 1.9654, Test Acc: 0.333
Epoch [80/100], Train Loss: 1.1048, Train Acc: 0.538, Test Loss: 2.0426, Test Acc: 0.347
Epoch [81/100], Train Loss: 1.1061, Train Acc: 0.538, Test Loss: 1.9839, Test Acc: 0.345
Epoch [82/100], Train Loss: 1.0957, Train Acc: 0.533, Test Loss: 1.8859, Test Acc: 0.368
Epoch [83/100], Train Loss: 1.1008, Train Acc: 0.537, Test Loss: 1.9098, Test Acc: 0.341
Epoch [84/100], Train Loss: 1.0954, Train Acc: 0.540, Test Loss: 2.0712, Test Acc: 0.346
Epoch [85/100], Train Loss: 1.0929, Train Acc: 0.537, Test Loss: 1.9967, Test Acc: 0.330
Epoch [86/100], Train Loss: 1.0919, Train Acc: 0.550, Test Loss: 2.0599, Test Acc: 0.340
Epoch [87/100], Train Loss: 1.1077, Train Acc: 0.539, Test Loss: 1.8422, Test Acc: 0.343
Epoch [88/100], Train Loss: 1.1031, Train Acc: 0.530, Test Loss: 1.9093, Test Acc: 0.334
Epoch [89/100], Train Loss: 1.1019, Train Acc: 0.530, Test Loss: 1.9656, Test Acc: 0.326
Epoch [90/100], Train Loss: 1.0919, Train Acc: 0.543, Test Loss: 1.8622, Test Acc: 0.338
Epoch [91/100], Train Loss: 1.0779, Train Acc: 0.548, Test Loss: 2.0023, Test Acc: 0.333
Epoch [92/100], Train Loss: 1.0846, Train Acc: 0.542, Test Loss: 1.9913, Test Acc: 0.340
Epoch [93/100], Train Loss: 1.0958, Train Acc: 0.539, Test Loss: 1.9602, Test Acc: 0.323
Epoch [94/100], Train Loss: 1.0896, Train Acc: 0.536, Test Loss: 2.0386, Test Acc: 0.339
Epoch [95/100], Train Loss: 1.0866, Train Acc: 0.551, Test Loss: 1.8724, Test Acc: 0.328
Epoch [96/100], Train Loss: 1.0961, Train Acc: 0.536, Test Loss: 1.9388, Test Acc: 0.367
Epoch [97/100], Train Loss: 1.1028, Train Acc: 0.543, Test Loss: 1.9308, Test Acc: 0.363
Epoch [98/100], Train Loss: 1.0950, Train Acc: 0.542, Test Loss: 2.1197, Test Acc: 0.351
Epoch [99/100], Train Loss: 1.0910, Train Acc: 0.544, Test Loss: 1.8762, Test Acc: 0.329
Epoch [100/100], Train Loss: 1.0824, Train Acc: 0.542, Test Loss: 1.8714, Test Acc: 0.344
```

In [ ]:
```
plot_history(history22)
```




```
Final Test Accuracy is 0.3438
```

3. (4 pts) Cutout augmentation (K = 16)

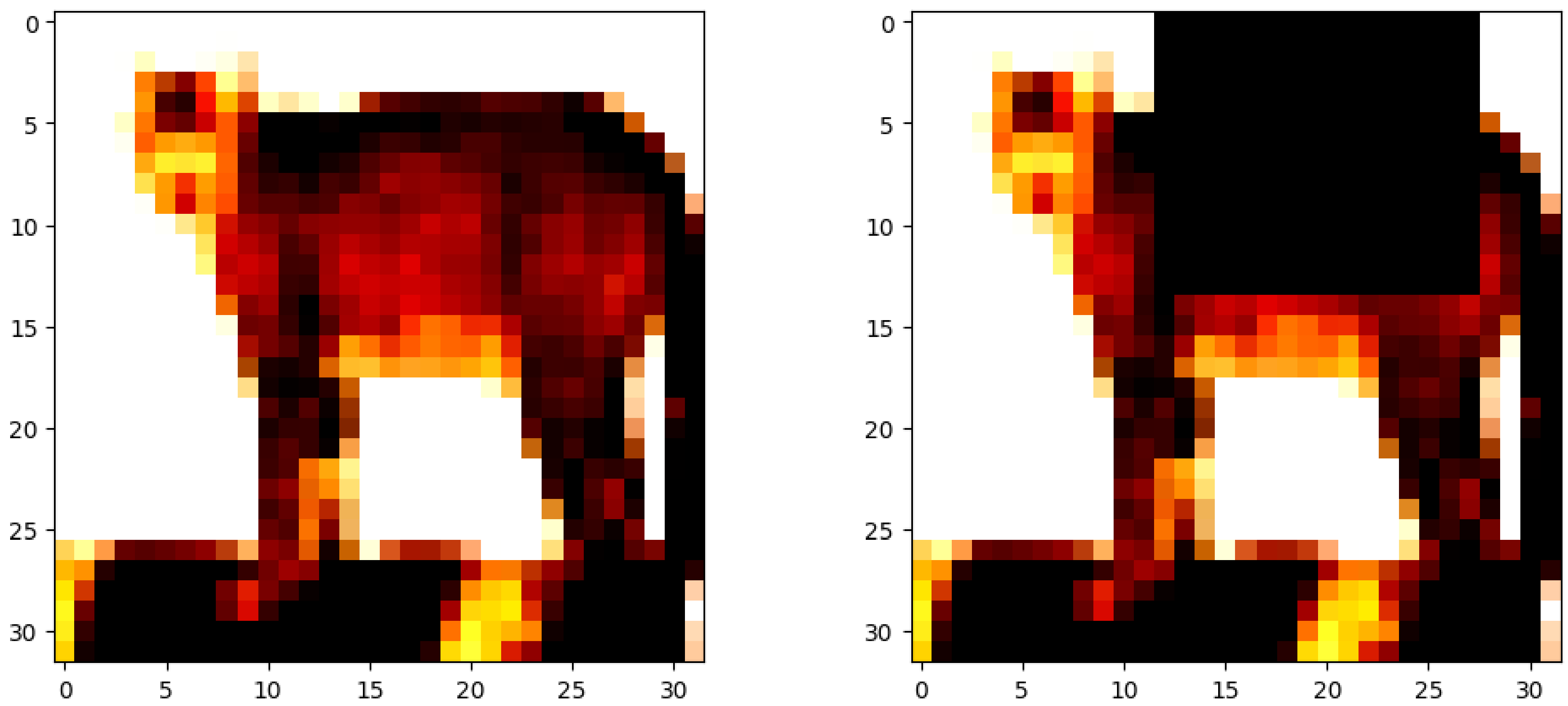
In [ ]:
```
def cutout(image, mask_size):
    if np.random.rand() < 0.5:
        return image

    channels, height, width = image.shape

    center_y = np.random.randint(0, height)
    center_x = np.random.randint(0, width)

    half_size = mask_size // 2
    top = max(0, center_y - half_size)
    bottom = min(height, center_y + half_size)
    left = max(0, center_x - half_size)
    right = min(width, center_x + half_size)

    image[:, top:bottom, left:right] = 0

    return image
```

In [ ]:
```
def apply_cutout_minibatch(minibatch_images, mask_size):
    for i in range(minibatch_images.shape[0]):
        minibatch_images[i] = cutout(minibatch_images[i], mask_size)
    return minibatch_images
```

In [ ]:
```
sampled_train_data, test_data, sampled_train_labels, test_labels = get_data()
```

In [ ]:
```
i=3
plt.figure(figsize=(12,5))
plt.subplot(121)
plt.imshow(sampled_train_data[i].transpose( 1, 2, 0))

plt.subplot(122)
plt.imshow(cutout(sampled_train_data[i].copy(), 16).transpose( 1, 2, 0))

plt.show()
```

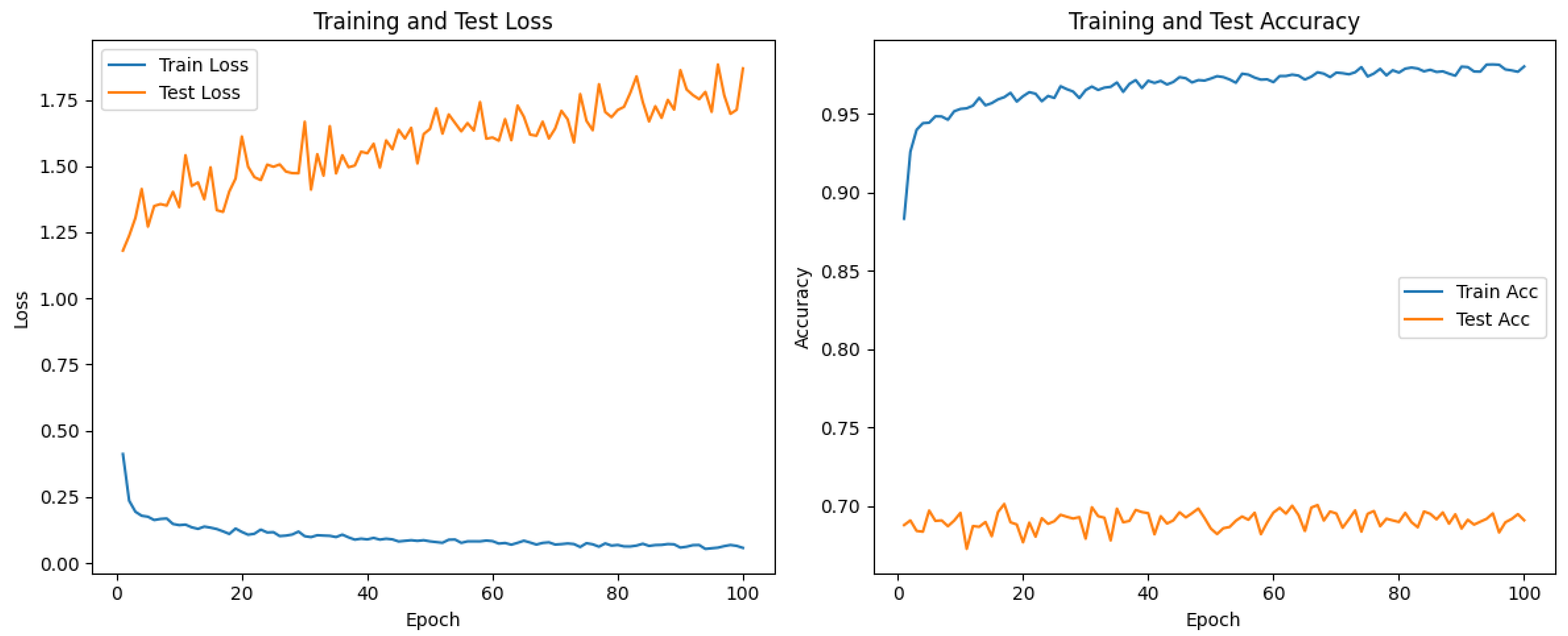
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```



In [ ]:
```
history3 = run_model(model, train_loader, test_loader, data_aug='cutout')
```

```
Epoch [1/100], Train Loss: 0.4110, Train Acc: 0.883, Test Loss: 1.1804, Test Acc: 0.688
Epoch [2/100], Train Loss: 0.2357, Train Acc: 0.926, Test Loss: 1.2368, Test Acc: 0.691
Epoch [3/100], Train Loss: 0.1939, Train Acc: 0.940, Test Loss: 1.3044, Test Acc: 0.684
Epoch [4/100], Train Loss: 0.1787, Train Acc: 0.944, Test Loss: 1.4138, Test Acc: 0.684
Epoch [5/100], Train Loss: 0.1750, Train Acc: 0.944, Test Loss: 1.2710, Test Acc: 0.697
Epoch [6/100], Train Loss: 0.1626, Train Acc: 0.948, Test Loss: 1.3491, Test Acc: 0.691
Epoch [7/100], Train Loss: 0.1667, Train Acc: 0.948, Test Loss: 1.3565, Test Acc: 0.691
Epoch [8/100], Train Loss: 0.1684, Train Acc: 0.946, Test Loss: 1.3507, Test Acc: 0.687
Epoch [9/100], Train Loss: 0.1476, Train Acc: 0.952, Test Loss: 1.4028, Test Acc: 0.691
Epoch [10/100], Train Loss: 0.1432, Train Acc: 0.953, Test Loss: 1.3446, Test Acc: 0.696
Epoch [11/100], Train Loss: 0.1452, Train Acc: 0.953, Test Loss: 1.5402, Test Acc: 0.673
Epoch [12/100], Train Loss: 0.1350, Train Acc: 0.955, Test Loss: 1.4250, Test Acc: 0.687
Epoch [13/100], Train Loss: 0.1291, Train Acc: 0.960, Test Loss: 1.4384, Test Acc: 0.687
Epoch [14/100], Train Loss: 0.1378, Train Acc: 0.955, Test Loss: 1.3747, Test Acc: 0.690
Epoch [15/100], Train Loss: 0.1337, Train Acc: 0.957, Test Loss: 1.4956, Test Acc: 0.681
Epoch [16/100], Train Loss: 0.1284, Train Acc: 0.959, Test Loss: 1.3333, Test Acc: 0.696
Epoch [17/100], Train Loss: 0.1197, Train Acc: 0.961, Test Loss: 1.3269, Test Acc: 0.701
Epoch [18/100], Train Loss: 0.1099, Train Acc: 0.963, Test Loss: 1.4044, Test Acc: 0.690
Epoch [19/100], Train Loss: 0.1304, Train Acc: 0.958, Test Loss: 1.4531, Test Acc: 0.688
Epoch [20/100], Train Loss: 0.1176, Train Acc: 0.961, Test Loss: 1.6112, Test Acc: 0.677
Epoch [21/100], Train Loss: 0.1070, Train Acc: 0.964, Test Loss: 1.4979, Test Acc: 0.689
Epoch [22/100], Train Loss: 0.1106, Train Acc: 0.963, Test Loss: 1.4586, Test Acc: 0.680
Epoch [23/100], Train Loss: 0.1262, Train Acc: 0.958, Test Loss: 1.4471, Test Acc: 0.692
Epoch [24/100], Train Loss: 0.1156, Train Acc: 0.961, Test Loss: 1.5059, Test Acc: 0.689
Epoch [25/100], Train Loss: 0.1166, Train Acc: 0.960, Test Loss: 1.4975, Test Acc: 0.690
Epoch [26/100], Train Loss: 0.1018, Train Acc: 0.968, Test Loss: 1.5065, Test Acc: 0.694
Epoch [27/100], Train Loss: 0.1037, Train Acc: 0.966, Test Loss: 1.4794, Test Acc: 0.693
Epoch [28/100], Train Loss: 0.1078, Train Acc: 0.964, Test Loss: 1.4735, Test Acc: 0.692
Epoch [29/100], Train Loss: 0.1190, Train Acc: 0.960, Test Loss: 1.4730, Test Acc: 0.693
Epoch [30/100], Train Loss: 0.1013, Train Acc: 0.965, Test Loss: 1.6668, Test Acc: 0.679
Epoch [31/100], Train Loss: 0.0979, Train Acc: 0.967, Test Loss: 1.4111, Test Acc: 0.699
Epoch [32/100], Train Loss: 0.1051, Train Acc: 0.965, Test Loss: 1.5440, Test Acc: 0.694
Epoch [33/100], Train Loss: 0.1038, Train Acc: 0.967, Test Loss: 1.4637, Test Acc: 0.693
Epoch [34/100], Train Loss: 0.1028, Train Acc: 0.967, Test Loss: 1.6501, Test Acc: 0.678
Epoch [35/100], Train Loss: 0.0980, Train Acc: 0.970, Test Loss: 1.4722, Test Acc: 0.698
Epoch [36/100], Train Loss: 0.1074, Train Acc: 0.964, Test Loss: 1.5401, Test Acc: 0.690
Epoch [37/100], Train Loss: 0.0969, Train Acc: 0.969, Test Loss: 1.4956, Test Acc: 0.691
Epoch [38/100], Train Loss: 0.0884, Train Acc: 0.971, Test Loss: 1.5024, Test Acc: 0.698
Epoch [39/100], Train Loss: 0.0921, Train Acc: 0.966, Test Loss: 1.5534, Test Acc: 0.696
Epoch [40/100], Train Loss: 0.0893, Train Acc: 0.971, Test Loss: 1.5474, Test Acc: 0.695
Epoch [41/100], Train Loss: 0.0951, Train Acc: 0.970, Test Loss: 1.5835, Test Acc: 0.682
Epoch [42/100], Train Loss: 0.0885, Train Acc: 0.971, Test Loss: 1.4946, Test Acc: 0.694
Epoch [43/100], Train Loss: 0.0920, Train Acc: 0.969, Test Loss: 1.5962, Test Acc: 0.689
Epoch [44/100], Train Loss: 0.0894, Train Acc: 0.970, Test Loss: 1.5628, Test Acc: 0.691
Epoch [45/100], Train Loss: 0.0814, Train Acc: 0.973, Test Loss: 1.6367, Test Acc: 0.696
Epoch [46/100], Train Loss: 0.0839, Train Acc: 0.973, Test Loss: 1.6032, Test Acc: 0.693
Epoch [47/100], Train Loss: 0.0862, Train Acc: 0.970, Test Loss: 1.6437, Test Acc: 0.695
Epoch [48/100], Train Loss: 0.0837, Train Acc: 0.971, Test Loss: 1.5100, Test Acc: 0.698
Epoch [49/100], Train Loss: 0.0862, Train Acc: 0.971, Test Loss: 1.6201, Test Acc: 0.692
Epoch [50/100], Train Loss: 0.0819, Train Acc: 0.972, Test Loss: 1.6393, Test Acc: 0.686
Epoch [51/100], Train Loss: 0.0791, Train Acc: 0.974, Test Loss: 1.7172, Test Acc: 0.682
Epoch [52/100], Train Loss: 0.0765, Train Acc: 0.973, Test Loss: 1.6215, Test Acc: 0.686
Epoch [53/100], Train Loss: 0.0881, Train Acc: 0.972, Test Loss: 1.6944, Test Acc: 0.687
Epoch [54/100], Train Loss: 0.0889, Train Acc: 0.970, Test Loss: 1.6623, Test Acc: 0.691
Epoch [55/100], Train Loss: 0.0762, Train Acc: 0.976, Test Loss: 1.6301, Test Acc: 0.693
Epoch [56/100], Train Loss: 0.0819, Train Acc: 0.975, Test Loss: 1.6617, Test Acc: 0.691
Epoch [57/100], Train Loss: 0.0820, Train Acc: 0.973, Test Loss: 1.6332, Test Acc: 0.696
Epoch [58/100], Train Loss: 0.0818, Train Acc: 0.972, Test Loss: 1.7414, Test Acc: 0.682
Epoch [59/100], Train Loss: 0.0849, Train Acc: 0.972, Test Loss: 1.6025, Test Acc: 0.690
Epoch [60/100], Train Loss: 0.0827, Train Acc: 0.970, Test Loss: 1.6071, Test Acc: 0.696
Epoch [61/100], Train Loss: 0.0735, Train Acc: 0.974, Test Loss: 1.5949, Test Acc: 0.699
Epoch [62/100], Train Loss: 0.0758, Train Acc: 0.974, Test Loss: 1.6767, Test Acc: 0.695
Epoch [63/100], Train Loss: 0.0693, Train Acc: 0.975, Test Loss: 1.5966, Test Acc: 0.700
Epoch [64/100], Train Loss: 0.0764, Train Acc: 0.974, Test Loss: 1.7278, Test Acc: 0.694
Epoch [65/100], Train Loss: 0.0843, Train Acc: 0.972, Test Loss: 1.6854, Test Acc: 0.684
Epoch [66/100], Train Loss: 0.0775, Train Acc: 0.974, Test Loss: 1.6181, Test Acc: 0.699
Epoch [67/100], Train Loss: 0.0698, Train Acc: 0.976, Test Loss: 1.6137, Test Acc: 0.701
Epoch [68/100], Train Loss: 0.0760, Train Acc: 0.976, Test Loss: 1.6670, Test Acc: 0.691
Epoch [69/100], Train Loss: 0.0783, Train Acc: 0.973, Test Loss: 1.6033, Test Acc: 0.697
Epoch [70/100], Train Loss: 0.0699, Train Acc: 0.976, Test Loss: 1.6410, Test Acc: 0.695
Epoch [71/100], Train Loss: 0.0715, Train Acc: 0.976, Test Loss: 1.7080, Test Acc: 0.686
Epoch [72/100], Train Loss: 0.0739, Train Acc: 0.975, Test Loss: 1.6757, Test Acc: 0.692
Epoch [73/100], Train Loss: 0.0717, Train Acc: 0.976, Test Loss: 1.5883, Test Acc: 0.697
Epoch [74/100], Train Loss: 0.0605, Train Acc: 0.980, Test Loss: 1.7716, Test Acc: 0.684
Epoch [75/100], Train Loss: 0.0750, Train Acc: 0.974, Test Loss: 1.6702, Test Acc: 0.695
Epoch [76/100], Train Loss: 0.0707, Train Acc: 0.976, Test Loss: 1.6345, Test Acc: 0.697
Epoch [77/100], Train Loss: 0.0616, Train Acc: 0.979, Test Loss: 1.8090, Test Acc: 0.687
Epoch [78/100], Train Loss: 0.0742, Train Acc: 0.974, Test Loss: 1.7034, Test Acc: 0.692
Epoch [79/100], Train Loss: 0.0658, Train Acc: 0.978, Test Loss: 1.6840, Test Acc: 0.691
Epoch [80/100], Train Loss: 0.0682, Train Acc: 0.976, Test Loss: 1.7111, Test Acc: 0.690
Epoch [81/100], Train Loss: 0.0628, Train Acc: 0.979, Test Loss: 1.7231, Test Acc: 0.696
Epoch [82/100], Train Loss: 0.0626, Train Acc: 0.979, Test Loss: 1.7758, Test Acc: 0.690
Epoch [83/100], Train Loss: 0.0659, Train Acc: 0.979, Test Loss: 1.8386, Test Acc: 0.686
Epoch [84/100], Train Loss: 0.0729, Train Acc: 0.977, Test Loss: 1.7412, Test Acc: 0.697
Epoch [85/100], Train Loss: 0.0647, Train Acc: 0.978, Test Loss: 1.6678, Test Acc: 0.695
Epoch [86/100], Train Loss: 0.0678, Train Acc: 0.977, Test Loss: 1.7254, Test Acc: 0.692
Epoch [87/100], Train Loss: 0.0685, Train Acc: 0.977, Test Loss: 1.6812, Test Acc: 0.696
Epoch [88/100], Train Loss: 0.0717, Train Acc: 0.976, Test Loss: 1.7496, Test Acc: 0.689
Epoch [89/100], Train Loss: 0.0706, Train Acc: 0.974, Test Loss: 1.7125, Test Acc: 0.695
Epoch [90/100], Train Loss: 0.0584, Train Acc: 0.980, Test Loss: 1.8621, Test Acc: 0.686
Epoch [91/100], Train Loss: 0.0617, Train Acc: 0.980, Test Loss: 1.7886, Test Acc: 0.691
Epoch [92/100], Train Loss: 0.0675, Train Acc: 0.977, Test Loss: 1.7676, Test Acc: 0.688
Epoch [93/100], Train Loss: 0.0681, Train Acc: 0.977, Test Loss: 1.7519, Test Acc: 0.690
Epoch [94/100], Train Loss: 0.0533, Train Acc: 0.981, Test Loss: 1.7796, Test Acc: 0.692
Epoch [95/100], Train Loss: 0.0558, Train Acc: 0.981, Test Loss: 1.7039, Test Acc: 0.695
Epoch [96/100], Train Loss: 0.0581, Train Acc: 0.981, Test Loss: 1.8834, Test Acc: 0.683
Epoch [97/100], Train Loss: 0.0639, Train Acc: 0.978, Test Loss: 1.7673, Test Acc: 0.690
Epoch [98/100], Train Loss: 0.0682, Train Acc: 0.978, Test Loss: 1.6964, Test Acc: 0.692
Epoch [99/100], Train Loss: 0.0648, Train Acc: 0.977, Test Loss: 1.7122, Test Acc: 0.695
Epoch [100/100], Train Loss: 0.0572, Train Acc: 0.980, Test Loss: 1.8683, Test Acc: 0.691
```

In [ ]:
```
plot_history(history3)
```



```
Final Test Accuracy is 0.6909
```

In [ ]:

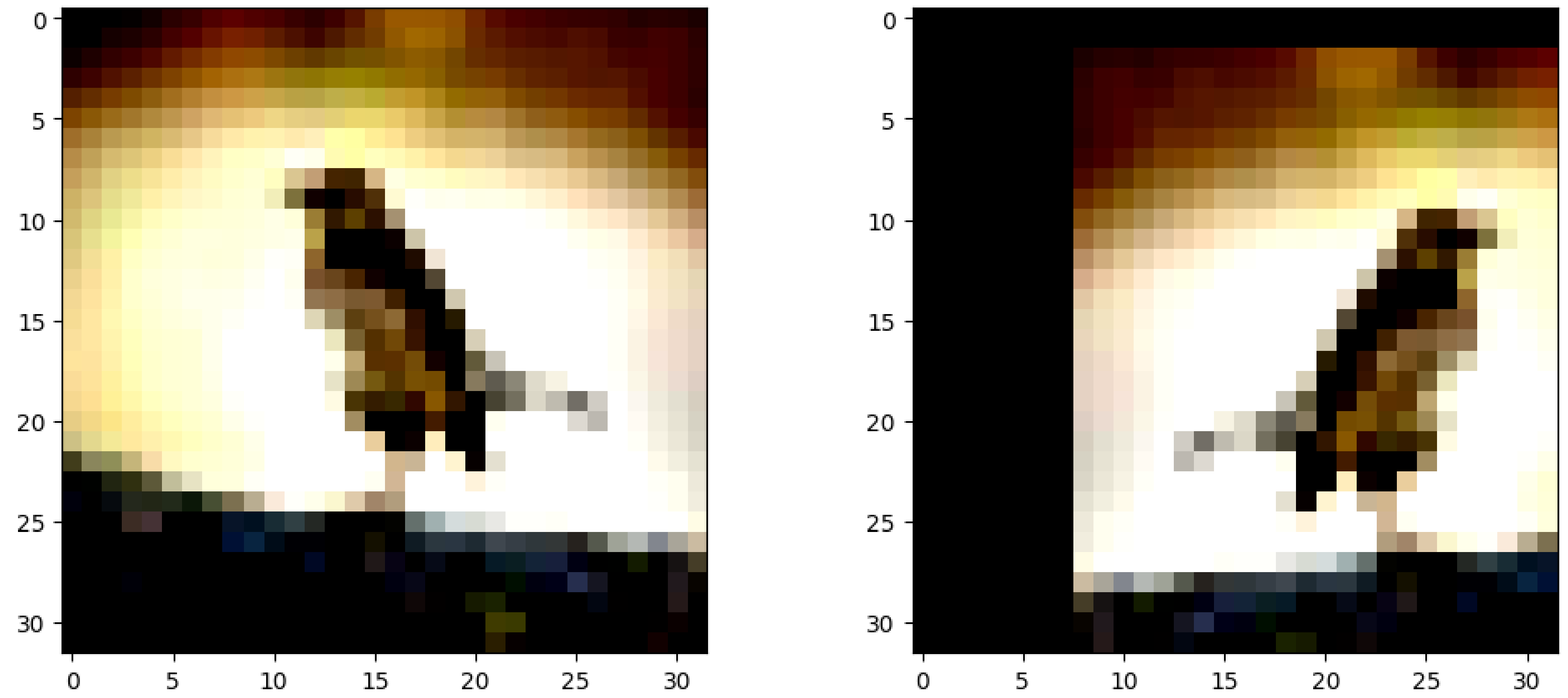
4. (4 pts) Standard augmentation

In [ ]:
```python
def standard(image, K):
    k1 = np.random.randint(-K, K+1)
    k2 = np.random.randint(-K, K+1)
    shifted_image = np.zeros_like(image)
    if k1 >= 0 and k2 >= 0:
        shifted_image[:, :image.shape[1]-k1, :image.shape[2]-k2] = image[:, k1:, k2:]
    elif k1 >= 0 and k2 < 0:
        shifted_image[:, :image.shape[1]-k1, -k2:] = image[:, k1:, :image.shape[2]+k2]
    elif k1 < 0 and k2 >= 0:
        shifted_image[:, -k1:, :image.shape[2]-k2] = image[:, :image.shape[1]+k1, k2:]
    else:
        shifted_image[:, -k1:, -k2:] = image[:, :image.shape[1]+k1, :image.shape[2]+k2]

    if np.random.rand() < 0.5:
        flipped_image = np.flip(shifted_image, axis=2)
    else:
        flipped_image = shifted_image

    return flipped_image
```

In [ ]:
```python
sampled_train_data, test_data, sampled_train_labels, test_labels = get_data()
```

In [ ]:
```python
i=0
plt.figure(figsize=(12,5))
plt.subplot(121)
plt.imshow(sampled_train_data[i].transpose( 1, 2, 0))

plt.subplot(122)
plt.imshow(standard(sampled_train_data[i].copy(), 10).transpose( 1, 2, 0))

plt.show()
```

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

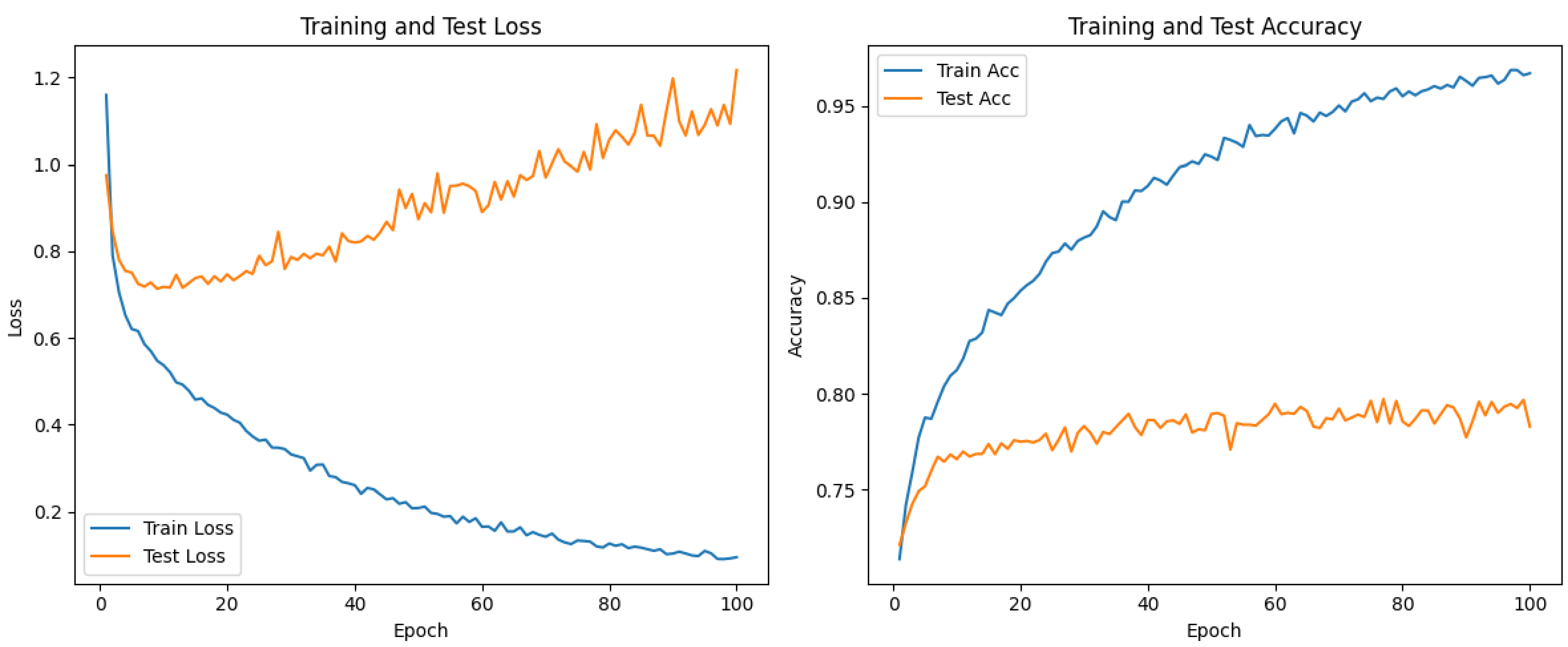


In [ ]:
```python
def apply_standard_minibatch(minibatch_images, K):
    for i in range(minibatch_images.shape[0]):
        std_img = standard(minibatch_images[i], K)
        minibatch_images[i] = torch.from_numpy(std_img.copy()).float()
    return minibatch_images
```

In [ ]:
```python
history4 = run_model(model, train_loader, test_loader, data_aug='standard')
```

```
Epoch [1/100], Train Loss: 1.1595, Train Acc: 0.714, Test Loss: 0.9749, Test Acc: 0.721
Epoch [2/100], Train Loss: 0.7905, Train Acc: 0.742, Test Loss: 0.8455, Test Acc: 0.733
Epoch [3/100], Train Loss: 0.7057, Train Acc: 0.759, Test Loss: 0.7802, Test Acc: 0.743
Epoch [4/100], Train Loss: 0.6528, Train Acc: 0.777, Test Loss: 0.7548, Test Acc: 0.749
Epoch [5/100], Train Loss: 0.6207, Train Acc: 0.787, Test Loss: 0.7508, Test Acc: 0.752
Epoch [6/100], Train Loss: 0.6161, Train Acc: 0.787, Test Loss: 0.7249, Test Acc: 0.760
Epoch [7/100], Train Loss: 0.5859, Train Acc: 0.796, Test Loss: 0.7185, Test Acc: 0.767
Epoch [8/100], Train Loss: 0.5700, Train Acc: 0.804, Test Loss: 0.7280, Test Acc: 0.764
Epoch [9/100], Train Loss: 0.5476, Train Acc: 0.809, Test Loss: 0.7135, Test Acc: 0.768
Epoch [10/100], Train Loss: 0.5373, Train Acc: 0.812, Test Loss: 0.7180, Test Acc: 0.766
Epoch [11/100], Train Loss: 0.5216, Train Acc: 0.818, Test Loss: 0.7161, Test Acc: 0.770
Epoch [12/100], Train Loss: 0.4977, Train Acc: 0.828, Test Loss: 0.7457, Test Acc: 0.767
Epoch [13/100], Train Loss: 0.4925, Train Acc: 0.829, Test Loss: 0.7158, Test Acc: 0.768
Epoch [14/100], Train Loss: 0.4782, Train Acc: 0.832, Test Loss: 0.7268, Test Acc: 0.769
Epoch [15/100], Train Loss: 0.4577, Train Acc: 0.844, Test Loss: 0.7382, Test Acc: 0.774
Epoch [16/100], Train Loss: 0.4607, Train Acc: 0.842, Test Loss: 0.7419, Test Acc: 0.768
Epoch [17/100], Train Loss: 0.4458, Train Acc: 0.841, Test Loss: 0.7248, Test Acc: 0.774
Epoch [18/100], Train Loss: 0.4385, Train Acc: 0.847, Test Loss: 0.7424, Test Acc: 0.771
Epoch [19/100], Train Loss: 0.4279, Train Acc: 0.850, Test Loss: 0.7302, Test Acc: 0.776
Epoch [20/100], Train Loss: 0.4232, Train Acc: 0.854, Test Loss: 0.7467, Test Acc: 0.775
Epoch [21/100], Train Loss: 0.4112, Train Acc: 0.857, Test Loss: 0.7332, Test Acc: 0.775
Epoch [22/100], Train Loss: 0.4046, Train Acc: 0.859, Test Loss: 0.7426, Test Acc: 0.774
Epoch [23/100], Train Loss: 0.3858, Train Acc: 0.863, Test Loss: 0.7541, Test Acc: 0.776
Epoch [24/100], Train Loss: 0.3731, Train Acc: 0.869, Test Loss: 0.7477, Test Acc: 0.779
Epoch [25/100], Train Loss: 0.3631, Train Acc: 0.873, Test Loss: 0.7896, Test Acc: 0.770
Epoch [26/100], Train Loss: 0.3656, Train Acc: 0.874, Test Loss: 0.7678, Test Acc: 0.776
Epoch [27/100], Train Loss: 0.3473, Train Acc: 0.878, Test Loss: 0.7767, Test Acc: 0.782
Epoch [28/100], Train Loss: 0.3469, Train Acc: 0.875, Test Loss: 0.8449, Test Acc: 0.770
Epoch [29/100], Train Loss: 0.3439, Train Acc: 0.880, Test Loss: 0.7592, Test Acc: 0.779
Epoch [30/100], Train Loss: 0.3315, Train Acc: 0.881, Test Loss: 0.7868, Test Acc: 0.783
Epoch [31/100], Train Loss: 0.3272, Train Acc: 0.883, Test Loss: 0.7799, Test Acc: 0.780
Epoch [32/100], Train Loss: 0.3229, Train Acc: 0.887, Test Loss: 0.7940, Test Acc: 0.774
Epoch [33/100], Train Loss: 0.2942, Train Acc: 0.895, Test Loss: 0.7838, Test Acc: 0.780
Epoch [34/100], Train Loss: 0.3075, Train Acc: 0.892, Test Loss: 0.7943, Test Acc: 0.779
Epoch [35/100], Train Loss: 0.3082, Train Acc: 0.891, Test Loss: 0.7907, Test Acc: 0.782
Epoch [36/100], Train Loss: 0.2817, Train Acc: 0.900, Test Loss: 0.8106, Test Acc: 0.786
Epoch [37/100], Train Loss: 0.2792, Train Acc: 0.900, Test Loss: 0.7763, Test Acc: 0.789
Epoch [38/100], Train Loss: 0.2681, Train Acc: 0.906, Test Loss: 0.8412, Test Acc: 0.782
Epoch [39/100], Train Loss: 0.2649, Train Acc: 0.906, Test Loss: 0.8236, Test Acc: 0.778
Epoch [40/100], Train Loss: 0.2607, Train Acc: 0.908, Test Loss: 0.8201, Test Acc: 0.786
Epoch [41/100], Train Loss: 0.2414, Train Acc: 0.913, Test Loss: 0.8227, Test Acc: 0.786
Epoch [42/100], Train Loss: 0.2542, Train Acc: 0.911, Test Loss: 0.8353, Test Acc: 0.782
Epoch [43/100], Train Loss: 0.2510, Train Acc: 0.909, Test Loss: 0.8266, Test Acc: 0.785
Epoch [44/100], Train Loss: 0.2399, Train Acc: 0.914, Test Loss: 0.8436, Test Acc: 0.786
Epoch [45/100], Train Loss: 0.2286, Train Acc: 0.918, Test Loss: 0.8678, Test Acc: 0.784
Epoch [46/100], Train Loss: 0.2315, Train Acc: 0.919, Test Loss: 0.8486, Test Acc: 0.789
Epoch [47/100], Train Loss: 0.2184, Train Acc: 0.921, Test Loss: 0.9422, Test Acc: 0.780
Epoch [48/100], Train Loss: 0.2222, Train Acc: 0.920, Test Loss: 0.8994, Test Acc: 0.781
Epoch [49/100], Train Loss: 0.2081, Train Acc: 0.925, Test Loss: 0.9323, Test Acc: 0.781
Epoch [50/100], Train Loss: 0.2087, Train Acc: 0.924, Test Loss: 0.8738, Test Acc: 0.789
Epoch [51/100], Train Loss: 0.2119, Train Acc: 0.922, Test Loss: 0.9108, Test Acc: 0.790
Epoch [52/100], Train Loss: 0.1975, Train Acc: 0.933, Test Loss: 0.8897, Test Acc: 0.788
Epoch [53/100], Train Loss: 0.1950, Train Acc: 0.932, Test Loss: 0.9794, Test Acc: 0.771
Epoch [54/100], Train Loss: 0.1887, Train Acc: 0.931, Test Loss: 0.8887, Test Acc: 0.785
Epoch [55/100], Train Loss: 0.1901, Train Acc: 0.929, Test Loss: 0.9504, Test Acc: 0.784
Epoch [56/100], Train Loss: 0.1733, Train Acc: 0.940, Test Loss: 0.9513, Test Acc: 0.784
Epoch [57/100], Train Loss: 0.1886, Train Acc: 0.934, Test Loss: 0.9559, Test Acc: 0.783
Epoch [58/100], Train Loss: 0.1766, Train Acc: 0.935, Test Loss: 0.9497, Test Acc: 0.786
Epoch [59/100], Train Loss: 0.1851, Train Acc: 0.935, Test Loss: 0.9390, Test Acc: 0.789
Epoch [60/100], Train Loss: 0.1654, Train Acc: 0.938, Test Loss: 0.8905, Test Acc: 0.795
Epoch [61/100], Train Loss: 0.1661, Train Acc: 0.942, Test Loss: 0.9055, Test Acc: 0.789
Epoch [62/100], Train Loss: 0.1563, Train Acc: 0.944, Test Loss: 0.9599, Test Acc: 0.790
Epoch [63/100], Train Loss: 0.1756, Train Acc: 0.936, Test Loss: 0.9192, Test Acc: 0.789
Epoch [64/100], Train Loss: 0.1546, Train Acc: 0.946, Test Loss: 0.9615, Test Acc: 0.793
Epoch [65/100], Train Loss: 0.1547, Train Acc: 0.945, Test Loss: 0.9259, Test Acc: 0.791
Epoch [66/100], Train Loss: 0.1642, Train Acc: 0.942, Test Loss: 0.9754, Test Acc: 0.783
Epoch [67/100], Train Loss: 0.1458, Train Acc: 0.947, Test Loss: 0.9642, Test Acc: 0.782
Epoch [68/100], Train Loss: 0.1531, Train Acc: 0.945, Test Loss: 0.9735, Test Acc: 0.787
Epoch [69/100], Train Loss: 0.1469, Train Acc: 0.947, Test Loss: 1.0305, Test Acc: 0.787
Epoch [70/100], Train Loss: 0.1425, Train Acc: 0.950, Test Loss: 0.9699, Test Acc: 0.792
Epoch [71/100], Train Loss: 0.1498, Train Acc: 0.947, Test Loss: 1.0017, Test Acc: 0.786
Epoch [72/100], Train Loss: 0.1358, Train Acc: 0.952, Test Loss: 1.0346, Test Acc: 0.787
Epoch [73/100], Train Loss: 0.1295, Train Acc: 0.954, Test Loss: 1.0059, Test Acc: 0.789
Epoch [74/100], Train Loss: 0.1256, Train Acc: 0.957, Test Loss: 0.9957, Test Acc: 0.788
Epoch [75/100], Train Loss: 0.1338, Train Acc: 0.953, Test Loss: 0.9835, Test Acc: 0.796
Epoch [76/100], Train Loss: 0.1326, Train Acc: 0.954, Test Loss: 1.0284, Test Acc: 0.785
Epoch [77/100], Train Loss: 0.1312, Train Acc: 0.954, Test Loss: 0.9881, Test Acc: 0.797
Epoch [78/100], Train Loss: 0.1203, Train Acc: 0.958, Test Loss: 1.0922, Test Acc: 0.784
Epoch [79/100], Train Loss: 0.1177, Train Acc: 0.959, Test Loss: 1.0143, Test Acc: 0.796
Epoch [80/100], Train Loss: 0.1269, Train Acc: 0.955, Test Loss: 1.0564, Test Acc: 0.785
Epoch [81/100], Train Loss: 0.1217, Train Acc: 0.958, Test Loss: 1.0781, Test Acc: 0.783
Epoch [82/100], Train Loss: 0.1252, Train Acc: 0.956, Test Loss: 1.0632, Test Acc: 0.787
Epoch [83/100], Train Loss: 0.1163, Train Acc: 0.958, Test Loss: 1.0452, Test Acc: 0.791
Epoch [84/100], Train Loss: 0.1196, Train Acc: 0.959, Test Loss: 1.0721, Test Acc: 0.791
Epoch [85/100], Train Loss: 0.1174, Train Acc: 0.960, Test Loss: 1.1370, Test Acc: 0.784
Epoch [86/100], Train Loss: 0.1137, Train Acc: 0.959, Test Loss: 1.0657, Test Acc: 0.789
Epoch [87/100], Train Loss: 0.1102, Train Acc: 0.961, Test Loss: 1.0657, Test Acc: 0.794
Epoch [88/100], Train Loss: 0.1135, Train Acc: 0.960, Test Loss: 1.0425, Test Acc: 0.793
Epoch [89/100], Train Loss: 0.1022, Train Acc: 0.965, Test Loss: 1.1240, Test Acc: 0.787
Epoch [90/100], Train Loss: 0.1037, Train Acc: 0.963, Test Loss: 1.1979, Test Acc: 0.777
Epoch [91/100], Train Loss: 0.1080, Train Acc: 0.961, Test Loss: 1.0986, Test Acc: 0.786
Epoch [92/100], Train Loss: 0.1039, Train Acc: 0.965, Test Loss: 1.0661, Test Acc: 0.796
Epoch [93/100], Train Loss: 0.0993, Train Acc: 0.965, Test Loss: 1.1217, Test Acc: 0.789
Epoch [94/100], Train Loss: 0.0979, Train Acc: 0.966, Test Loss: 1.0675, Test Acc: 0.796
Epoch [95/100], Train Loss: 0.1098, Train Acc: 0.962, Test Loss: 1.0904, Test Acc: 0.790
Epoch [96/100], Train Loss: 0.1042, Train Acc: 0.964, Test Loss: 1.1268, Test Acc: 0.793
Epoch [97/100], Train Loss: 0.0913, Train Acc: 0.969, Test Loss: 1.0891, Test Acc: 0.795
Epoch [98/100], Train Loss: 0.0909, Train Acc: 0.969, Test Loss: 1.1367, Test Acc: 0.792
Epoch [99/100], Train Loss: 0.0925, Train Acc: 0.966, Test Loss: 1.0927, Test Acc: 0.797
Epoch [100/100], Train Loss: 0.0953, Train Acc: 0.967, Test Loss: 1.2170, Test Acc: 0.783
```

In [ ]:
```
plot_history(history4)
```



```
Final Test Accuracy is 0.7828
```

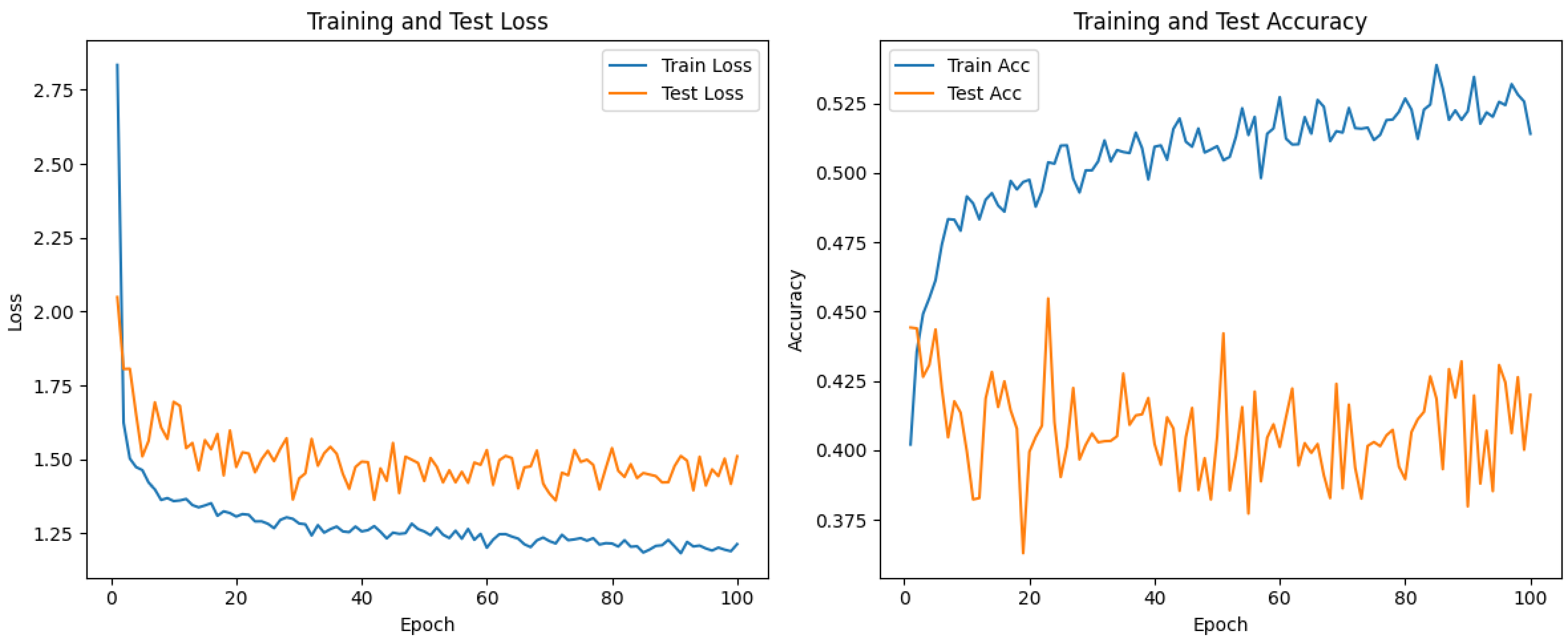
In [ ]:

5. (3 pts) Combine all augmentations together.

In [ ]:
```python
history5 = run_model(model, train_loader, test_loader, data_aug='all')
```

```
Epoch [1/100], Train Loss: 2.8334, Train Acc: 0.402, Test Loss: 2.0489, Test Acc: 0.444
Epoch [2/100], Train Loss: 1.6240, Train Acc: 0.436, Test Loss: 1.8058, Test Acc: 0.444
Epoch [3/100], Train Loss: 1.5026, Train Acc: 0.449, Test Loss: 1.8065, Test Acc: 0.426
Epoch [4/100], Train Loss: 1.4737, Train Acc: 0.455, Test Loss: 1.6529, Test Acc: 0.431
Epoch [5/100], Train Loss: 1.4635, Train Acc: 0.461, Test Loss: 1.5099, Test Acc: 0.444
Epoch [6/100], Train Loss: 1.4214, Train Acc: 0.474, Test Loss: 1.5623, Test Acc: 0.422
Epoch [7/100], Train Loss: 1.3994, Train Acc: 0.483, Test Loss: 1.6930, Test Acc: 0.405
Epoch [8/100], Train Loss: 1.3635, Train Acc: 0.483, Test Loss: 1.6082, Test Acc: 0.418
Epoch [9/100], Train Loss: 1.3701, Train Acc: 0.479, Test Loss: 1.5688, Test Acc: 0.413
Epoch [10/100], Train Loss: 1.3597, Train Acc: 0.491, Test Loss: 1.6947, Test Acc: 0.400
Epoch [11/100], Train Loss: 1.3619, Train Acc: 0.489, Test Loss: 1.6811, Test Acc: 0.382
Epoch [12/100], Train Loss: 1.3670, Train Acc: 0.483, Test Loss: 1.5378, Test Acc: 0.383
Epoch [13/100], Train Loss: 1.3463, Train Acc: 0.490, Test Loss: 1.5555, Test Acc: 0.419
Epoch [14/100], Train Loss: 1.3386, Train Acc: 0.493, Test Loss: 1.4626, Test Acc: 0.428
Epoch [15/100], Train Loss: 1.3450, Train Acc: 0.488, Test Loss: 1.5650, Test Acc: 0.415
Epoch [16/100], Train Loss: 1.3527, Train Acc: 0.486, Test Loss: 1.5340, Test Acc: 0.425
Epoch [17/100], Train Loss: 1.3100, Train Acc: 0.497, Test Loss: 1.5863, Test Acc: 0.414
Epoch [18/100], Train Loss: 1.3252, Train Acc: 0.494, Test Loss: 1.4456, Test Acc: 0.408
Epoch [19/100], Train Loss: 1.3193, Train Acc: 0.497, Test Loss: 1.5976, Test Acc: 0.363
Epoch [20/100], Train Loss: 1.3074, Train Acc: 0.497, Test Loss: 1.4739, Test Acc: 0.399
Epoch [21/100], Train Loss: 1.3158, Train Acc: 0.488, Test Loss: 1.5238, Test Acc: 0.405
Epoch [22/100], Train Loss: 1.3136, Train Acc: 0.493, Test Loss: 1.5196, Test Acc: 0.409
Epoch [23/100], Train Loss: 1.2910, Train Acc: 0.504, Test Loss: 1.4560, Test Acc: 0.455
Epoch [24/100], Train Loss: 1.2914, Train Acc: 0.503, Test Loss: 1.5000, Test Acc: 0.410
Epoch [25/100], Train Loss: 1.2831, Train Acc: 0.510, Test Loss: 1.5289, Test Acc: 0.390
Epoch [26/100], Train Loss: 1.2680, Train Acc: 0.510, Test Loss: 1.4938, Test Acc: 0.401
Epoch [27/100], Train Loss: 1.2961, Train Acc: 0.498, Test Loss: 1.5363, Test Acc: 0.422
Epoch [28/100], Train Loss: 1.3049, Train Acc: 0.493, Test Loss: 1.5716, Test Acc: 0.397
Epoch [29/100], Train Loss: 1.2997, Train Acc: 0.501, Test Loss: 1.3649, Test Acc: 0.402
Epoch [30/100], Train Loss: 1.2839, Train Acc: 0.501, Test Loss: 1.4353, Test Acc: 0.406
Epoch [31/100], Train Loss: 1.2813, Train Acc: 0.504, Test Loss: 1.4527, Test Acc: 0.403
Epoch [32/100], Train Loss: 1.2435, Train Acc: 0.512, Test Loss: 1.5694, Test Acc: 0.403
Epoch [33/100], Train Loss: 1.2785, Train Acc: 0.504, Test Loss: 1.4782, Test Acc: 0.403
Epoch [34/100], Train Loss: 1.2530, Train Acc: 0.508, Test Loss: 1.5205, Test Acc: 0.405
Epoch [35/100], Train Loss: 1.2649, Train Acc: 0.507, Test Loss: 1.5425, Test Acc: 0.428
Epoch [36/100], Train Loss: 1.2736, Train Acc: 0.507, Test Loss: 1.5188, Test Acc: 0.409
Epoch [37/100], Train Loss: 1.2571, Train Acc: 0.514, Test Loss: 1.4503, Test Acc: 0.412
Epoch [38/100], Train Loss: 1.2546, Train Acc: 0.509, Test Loss: 1.4011, Test Acc: 0.413
Epoch [39/100], Train Loss: 1.2737, Train Acc: 0.498, Test Loss: 1.4740, Test Acc: 0.419
Epoch [40/100], Train Loss: 1.2572, Train Acc: 0.509, Test Loss: 1.4922, Test Acc: 0.402
Epoch [41/100], Train Loss: 1.2615, Train Acc: 0.510, Test Loss: 1.4904, Test Acc: 0.395
Epoch [42/100], Train Loss: 1.2751, Train Acc: 0.505, Test Loss: 1.3639, Test Acc: 0.412
Epoch [43/100], Train Loss: 1.2560, Train Acc: 0.516, Test Loss: 1.4694, Test Acc: 0.408
Epoch [44/100], Train Loss: 1.2334, Train Acc: 0.519, Test Loss: 1.4272, Test Acc: 0.385
Epoch [45/100], Train Loss: 1.2530, Train Acc: 0.511, Test Loss: 1.5555, Test Acc: 0.405
Epoch [46/100], Train Loss: 1.2489, Train Acc: 0.509, Test Loss: 1.3868, Test Acc: 0.415
Epoch [47/100], Train Loss: 1.2512, Train Acc: 0.516, Test Loss: 1.5092, Test Acc: 0.386
Epoch [48/100], Train Loss: 1.2835, Train Acc: 0.507, Test Loss: 1.4986, Test Acc: 0.397
Epoch [49/100], Train Loss: 1.2653, Train Acc: 0.508, Test Loss: 1.4878, Test Acc: 0.382
Epoch [50/100], Train Loss: 1.2570, Train Acc: 0.509, Test Loss: 1.4265, Test Acc: 0.405
Epoch [51/100], Train Loss: 1.2445, Train Acc: 0.504, Test Loss: 1.5047, Test Acc: 0.442
Epoch [52/100], Train Loss: 1.2698, Train Acc: 0.506, Test Loss: 1.4747, Test Acc: 0.386
Epoch [53/100], Train Loss: 1.2460, Train Acc: 0.513, Test Loss: 1.4218, Test Acc: 0.398
Epoch [54/100], Train Loss: 1.2347, Train Acc: 0.523, Test Loss: 1.4629, Test Acc: 0.415
Epoch [55/100], Train Loss: 1.2596, Train Acc: 0.513, Test Loss: 1.4220, Test Acc: 0.377
Epoch [56/100], Train Loss: 1.2325, Train Acc: 0.520, Test Loss: 1.4578, Test Acc: 0.421
Epoch [57/100], Train Loss: 1.2656, Train Acc: 0.498, Test Loss: 1.4203, Test Acc: 0.389
Epoch [58/100], Train Loss: 1.2286, Train Acc: 0.514, Test Loss: 1.4892, Test Acc: 0.405
Epoch [59/100], Train Loss: 1.2490, Train Acc: 0.516, Test Loss: 1.4813, Test Acc: 0.409
Epoch [60/100], Train Loss: 1.2019, Train Acc: 0.527, Test Loss: 1.5314, Test Acc: 0.401
Epoch [61/100], Train Loss: 1.2292, Train Acc: 0.512, Test Loss: 1.4142, Test Acc: 0.412
Epoch [62/100], Train Loss: 1.2478, Train Acc: 0.510, Test Loss: 1.4970, Test Acc: 0.422
Epoch [63/100], Train Loss: 1.2481, Train Acc: 0.510, Test Loss: 1.5116, Test Acc: 0.394
Epoch [64/100], Train Loss: 1.2396, Train Acc: 0.520, Test Loss: 1.5044, Test Acc: 0.403
Epoch [65/100], Train Loss: 1.2329, Train Acc: 0.514, Test Loss: 1.4024, Test Acc: 0.399
Epoch [66/100], Train Loss: 1.2136, Train Acc: 0.526, Test Loss: 1.4728, Test Acc: 0.402
Epoch [67/100], Train Loss: 1.2041, Train Acc: 0.524, Test Loss: 1.4760, Test Acc: 0.391
Epoch [68/100], Train Loss: 1.2270, Train Acc: 0.511, Test Loss: 1.5300, Test Acc: 0.383
Epoch [69/100], Train Loss: 1.2364, Train Acc: 0.515, Test Loss: 1.4179, Test Acc: 0.424
Epoch [70/100], Train Loss: 1.2244, Train Acc: 0.514, Test Loss: 1.3856, Test Acc: 0.386
Epoch [71/100], Train Loss: 1.2164, Train Acc: 0.523, Test Loss: 1.3617, Test Acc: 0.416
Epoch [72/100], Train Loss: 1.2459, Train Acc: 0.516, Test Loss: 1.4549, Test Acc: 0.394
Epoch [73/100], Train Loss: 1.2274, Train Acc: 0.516, Test Loss: 1.4461, Test Acc: 0.383
Epoch [74/100], Train Loss: 1.2302, Train Acc: 0.516, Test Loss: 1.5319, Test Acc: 0.402
Epoch [75/100], Train Loss: 1.2345, Train Acc: 0.512, Test Loss: 1.4913, Test Acc: 0.403
Epoch [76/100], Train Loss: 1.2257, Train Acc: 0.514, Test Loss: 1.4988, Test Acc: 0.401
Epoch [77/100], Train Loss: 1.2343, Train Acc: 0.519, Test Loss: 1.4813, Test Acc: 0.405
Epoch [78/100], Train Loss: 1.2127, Train Acc: 0.519, Test Loss: 1.3990, Test Acc: 0.407
Epoch [79/100], Train Loss: 1.2174, Train Acc: 0.522, Test Loss: 1.4698, Test Acc: 0.394
Epoch [80/100], Train Loss: 1.2163, Train Acc: 0.527, Test Loss: 1.5381, Test Acc: 0.390
Epoch [81/100], Train Loss: 1.2060, Train Acc: 0.523, Test Loss: 1.4612, Test Acc: 0.406
Epoch [82/100], Train Loss: 1.2274, Train Acc: 0.512, Test Loss: 1.4404, Test Acc: 0.411
Epoch [83/100], Train Loss: 1.2053, Train Acc: 0.523, Test Loss: 1.4838, Test Acc: 0.414
Epoch [84/100], Train Loss: 1.2072, Train Acc: 0.524, Test Loss: 1.4365, Test Acc: 0.427
Epoch [85/100], Train Loss: 1.1857, Train Acc: 0.539, Test Loss: 1.4537, Test Acc: 0.418
Epoch [86/100], Train Loss: 1.1956, Train Acc: 0.530, Test Loss: 1.4485, Test Acc: 0.393
Epoch [87/100], Train Loss: 1.2081, Train Acc: 0.519, Test Loss: 1.4433, Test Acc: 0.429
Epoch [88/100], Train Loss: 1.2104, Train Acc: 0.522, Test Loss: 1.4220, Test Acc: 0.419
Epoch [89/100], Train Loss: 1.2285, Train Acc: 0.519, Test Loss: 1.4225, Test Acc: 0.432
Epoch [90/100], Train Loss: 1.2065, Train Acc: 0.522, Test Loss: 1.4772, Test Acc: 0.380
Epoch [91/100], Train Loss: 1.1833, Train Acc: 0.534, Test Loss: 1.5114, Test Acc: 0.420
Epoch [92/100], Train Loss: 1.2219, Train Acc: 0.518, Test Loss: 1.4954, Test Acc: 0.388
Epoch [93/100], Train Loss: 1.2063, Train Acc: 0.522, Test Loss: 1.3962, Test Acc: 0.407
Epoch [94/100], Train Loss: 1.2093, Train Acc: 0.520, Test Loss: 1.5090, Test Acc: 0.385
Epoch [95/100], Train Loss: 1.1994, Train Acc: 0.525, Test Loss: 1.4123, Test Acc: 0.431
Epoch [96/100], Train Loss: 1.1925, Train Acc: 0.524, Test Loss: 1.4661, Test Acc: 0.424
Epoch [97/100], Train Loss: 1.2025, Train Acc: 0.532, Test Loss: 1.4432, Test Acc: 0.406
Epoch [98/100], Train Loss: 1.1954, Train Acc: 0.528, Test Loss: 1.5025, Test Acc: 0.426
Epoch [99/100], Train Loss: 1.1902, Train Acc: 0.526, Test Loss: 1.4165, Test Acc: 0.400
Epoch [100/100], Train Loss: 1.2141, Train Acc: 0.514, Test Loss: 1.5102, Test Acc: 0.420
```

In [ ]:
```python
plot_history(history5)
```




```
Final Test Accuracy is 0.4199
```

Does combining improve things further?

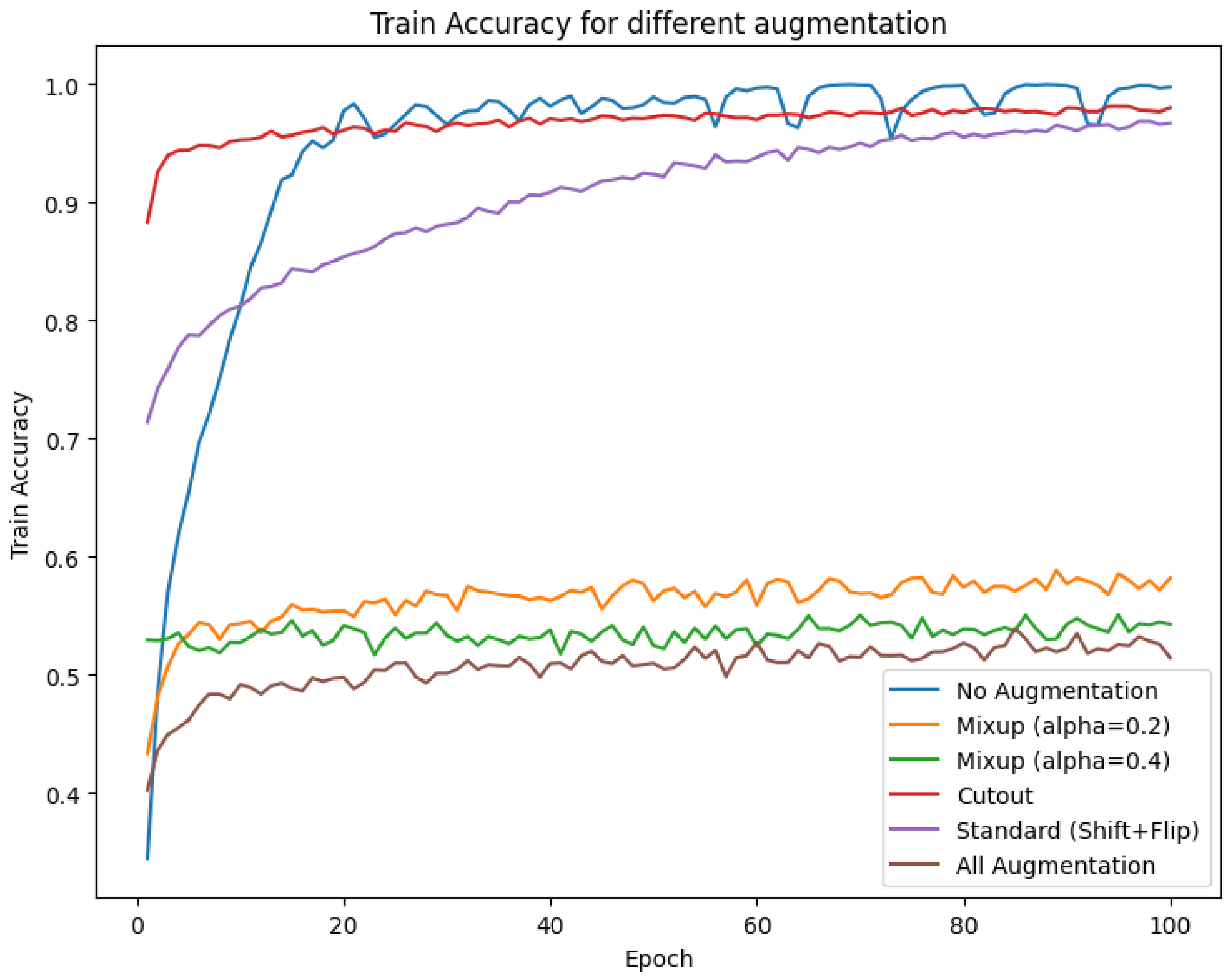
-> No. Combining all three augmentation does not improve further. Becasue, we have included augmentation which brings too much randomness to the dataset. For that reason, combining the augementation is not the best way to train model in this case.

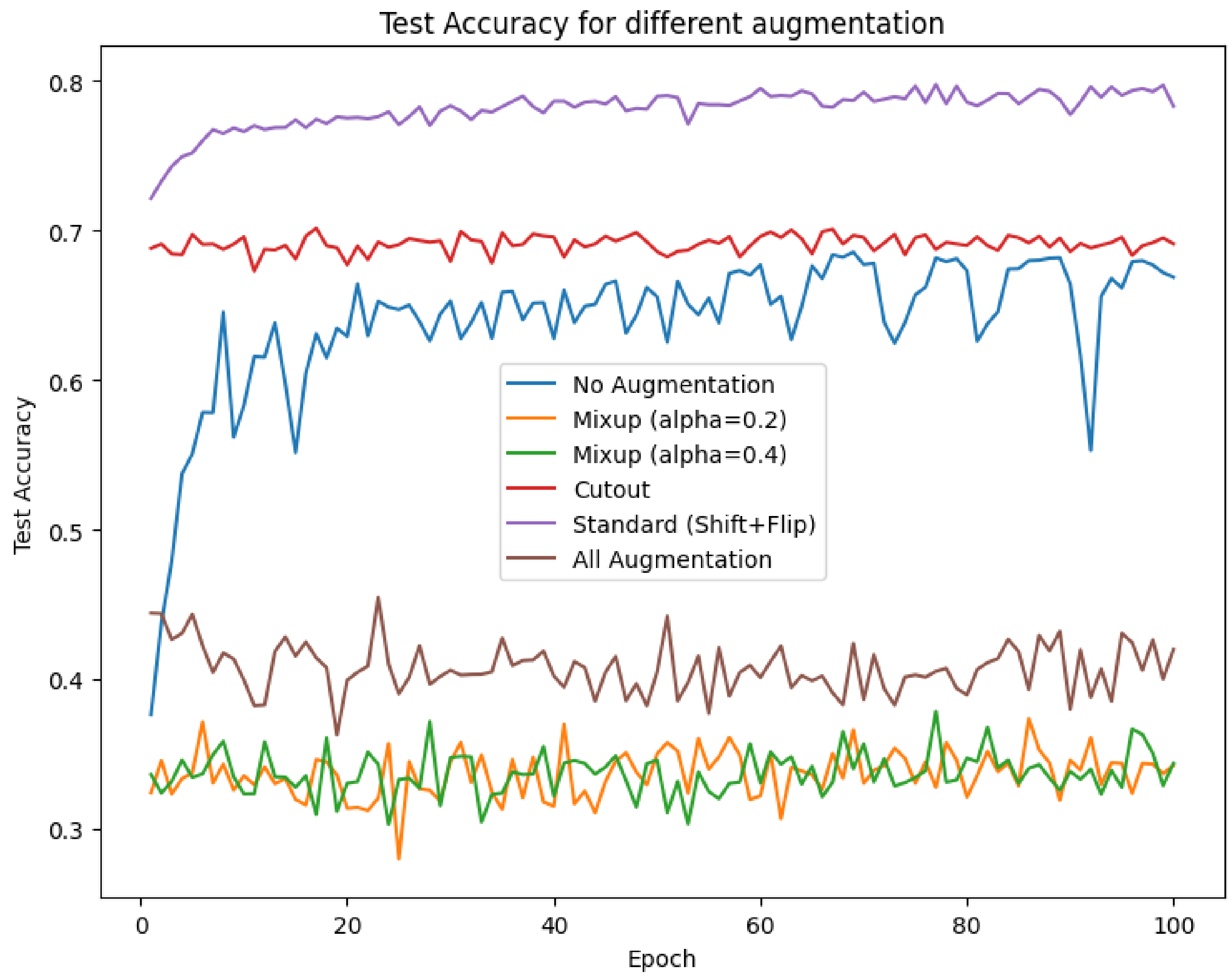
In [ ]:

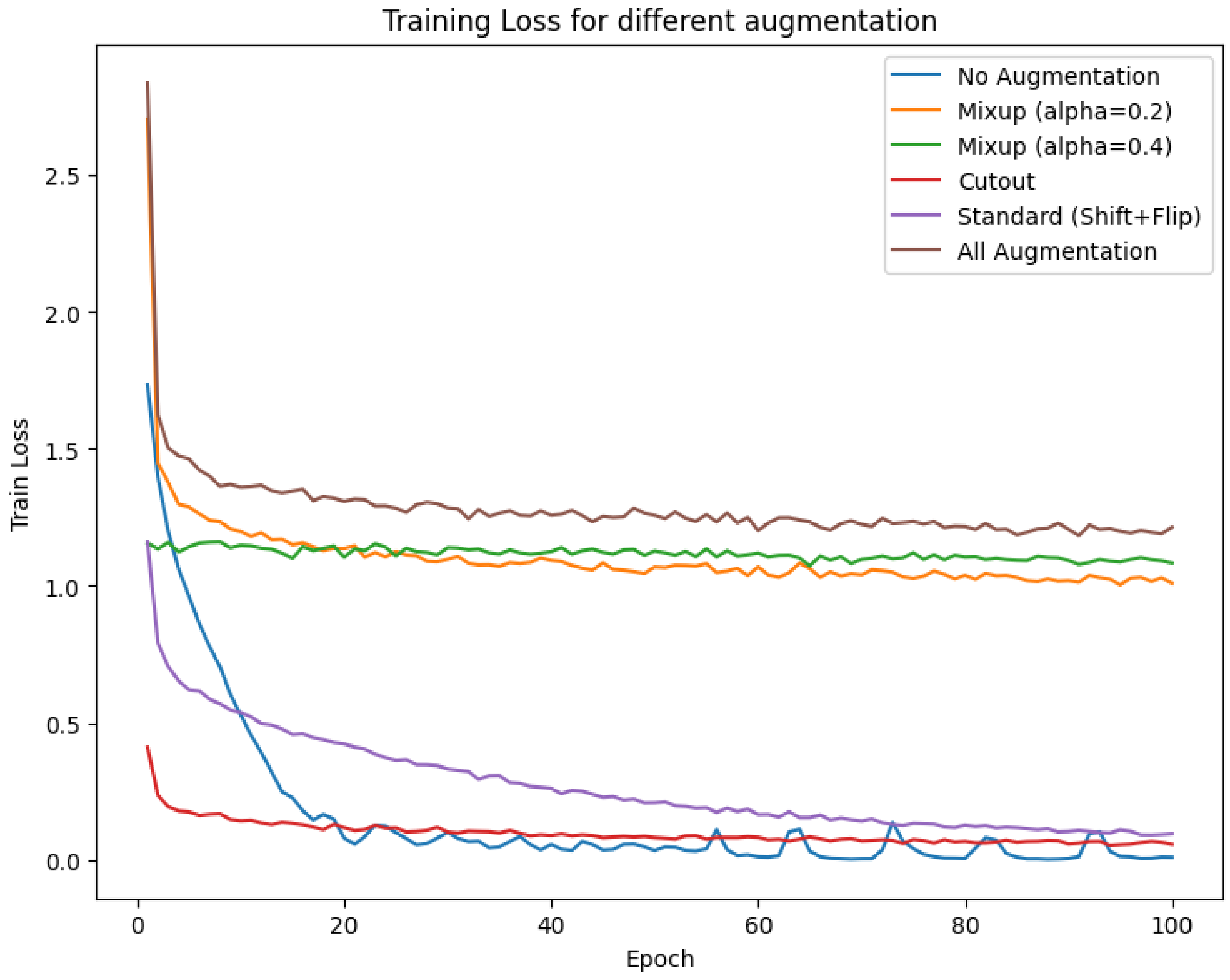
```

```

In [ ]:

```
plt.figure(figsize=(15,12))

c = plt.subplot(2, 2, 1)
c.plot(range(1, 101), history1['train_acc'], label='No Augmentation')
c.plot(range(1, 101), history2['train_acc'], label='Mixup (alpha=0.2)')
c.plot(range(1, 101), history22['train_acc'], label='Mixup (alpha=0.4)')
c.plot(range(1, 101), history3['train_acc'], label='Cutout')
c.plot(range(1, 101), history4['train_acc'], label='Standard (Shift+Flip)')
c.plot(range(1, 101), history5['train_acc'], label='All Augmentation')
c.set_xlabel('Epoch')
c.set_ylabel('Train Accuracy')
c.set_title('Train Accuracy for different augmentation')
c.legend()

b = plt.subplot(2, 2, 2)
b.plot(range(1, 101), history1['test_acc'], label='No Augmentation')
b.plot(range(1, 101), history2['test_acc'], label='Mixup (alpha=0.2)')
b.plot(range(1, 101), history22['test_acc'], label='Mixup (alpha=0.4)')
b.plot(range(1, 101), history3['test_acc'], label='Cutout')
b.plot(range(1, 101), history4['test_acc'], label='Standard (Shift+Flip)')
b.plot(range(1, 101), history5['test_acc'], label='All Augmentation')
b.set_xlabel('Epoch')
b.set_ylabel('Test Accuracy')
b.set_title('Test Accuracy for different augmentation')
b.legend()

a = plt.subplot(2, 2, 3)
a.plot(range(1, 101), history1['train_loss'], label='No Augmentation')
a.plot(range(1, 101), history2['train_loss'], label='Mixup (alpha=0.2)')
a.plot(range(1, 101), history22['train_loss'], label='Mixup (alpha=0.4)')
a.plot(range(1, 101), history3['train_loss'], label='Cutout')
a.plot(range(1, 101), history4['train_loss'], label='Standard (Shift+Flip)')
a.plot(range(1, 101), history5['train_loss'], label='All Augmentation')
a.set_xlabel('Epoch')
a.set_ylabel('Train Loss')
a.set_title('Training Loss for different augmentation')
a.legend()

plt.tight_layout()
#plt.title(title)
plt.show()
```







6. (2 pts) Comment on the role of data augmentation.

How does it affect test accuracy, train accuracy and the convergence of optimization? Is test accuracy higher? Does training loss converge faster?

Based on the observed plots, it can be concluded that:

- Without augmentation, the model tends to overfit the training data.
- Mixup augmentation leads to lower train and test accuracies. This augmentation introduces excessive randomness, making it difficult for the model to learn meaningful features. However, when comparing different mixup values, alpha = 0.2 performs relatively better than alpha = 0.4. Both values reach a plateau in terms of optimization, indicating that this augmentation technique is not suitable for this dataset.
- Cutout augmentation shows improvements compared to the base case. The difference between train and test accuracies is smaller, indicating reduced overfitting. The training loss initially decreases quickly, and then the performance plateaus.
- Standard augmentation performs the best among the techniques evaluated. It exhibits the least overfitting, with the highest test accuracy. Both train and test accuracies improve over time, and the training loss continuously decreases.
- Augmenting with a combination of techniques, including mixup, does not yield good performance. Mixup introduces excessive randomness, hindering the model's ability to find patterns. Although the accuracies and losses are better than with mixup alone, this technique does not compare favorably to the others.

Overall, the standard augmentation technique performs the best. However, it is worth noting that if the model were trained for additional epochs, it could potentially achieve even better performance.

```
In [ ]:
```