Group_19_CNN

December 8, 2021

[1]:
```
import torch
import torchvision
import matplotlib.pyplot as plt
import time
```

[2]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[3]:
```
# Device configuration
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
device
```

[3]:
```
device(type='cuda')
```

[4]:
```
#Logger
import pandas as pd
import numpy as np

expLog_R = pd.DataFrame(columns=["exp_name",
                                 "Loss",
                                 "Accuracy"
                                ])
```

1 Load Training Data

[5]:
```
#Train Data Loader
mean = [0.485, 0.456, 0.406]
std = [0.229, 0.224, 0.225]

def trainload(batch_size_train):
    train_loader = torch.utils.data.DataLoader(torchvision.datasets.
 ↪ImageFolder("/content/drive/My Drive/Colab Notebooks/cat_dog/train/",
                                                                        ␣
 ↪transform = torchvision.transforms.Compose([
```

```
↪        torchvision.transforms.Resize((128, 128)),
↪        torchvision.transforms.ToTensor(),
↪        torchvision.transforms.Normalize(
↪        mean = mean, std = std)
↪, batch_size = batch_size_train, shuffle = True)
    return train_loader
```

2 Load Testing Data

[6]:
```
#Test Data Loader
mean = [0.485, 0.456, 0.406]
std = [0.229, 0.224, 0.225]

def testload(batch_size_train):
    test_loader = torch.utils.data.DataLoader(torchvision.datasets.
↪ImageFolder("/content/drive/My Drive/Colab Notebooks/cat_dog/test/",
↪      transform = torchvision.transforms.Compose([
↪      torchvision.transforms.Resize((128, 128)),
↪      torchvision.transforms.ToTensor(),
↪      torchvision.transforms.Normalize(
↪      mean = mean, std = std)
↪])), batch_size = batch_size_train, shuffle = True)
    return test_loader
```

3 Create a CNN model

[7]:
```
class CNN(torch.nn.Module):
    def __init__(self):
        super(CNN, self).__init__()
        # Define 1st layer
        self.conv1 = torch.nn.Sequential(
            torch.nn.Conv2d(
                in_channels = 3,
```

```
                out_channels = 16,
                kernel_size = 5,
                stride = 1,
                padding= 2,
            ),
            torch.nn.ReLU(),
            # torch.nn.LogSoftmax(),
            torch.nn.MaxPool2d(kernel_size = 2),
        )

        # Define 2nd layer
        self.conv2 = torch.nn.Sequential(
            torch.nn.Conv2d(
                in_channels = 16,
                out_channels = 32,
                kernel_size = 5,
                stride = 1,
                padding= 2,
            ),
            torch.nn.ReLU(),
            torch.nn.MaxPool2d(kernel_size = 2) ,
        )

        # Define a fully connected layer
        self.fc1 = torch.nn.Linear(32*32*32, 5000)  # # 512*8*8 # 32*32*32
        # print(self.fc1)
        # Define a fully connected layer
        self.fc2 = torch.nn.Linear(5000, 500)
        # print(self.fc2)
        self.fc3 = torch.nn.Linear(500, 2)
        # print(self.fc3)
        # self.fc4 = torch.nn.Linear(50, 2)
        # print(self.fc3)
        # self.fc5 = torch.nn.Linear(50, 2)


    def forward(self, x):
        x = self.conv1(x)
        x = self.conv2(x)

        # Flatten the output
        x = x.view(x.size(0), -1)
        x = torch.nn.functional.relu(self.fc1(x))
        #x = x.view(x.size(0), -1)
        output = self.fc2(x)
        output = torch.nn.functional.log_softmax(output, dim=1)
        #print(output,x)
```

```
        return output, x
```

4 Train the CNN model

[8]:
```
# Train your model
def train(num_epochs, cnn, train_loader, test_loader, mini_batch):
    total_train_time = 0
    total_step = len(train_loader)
    train_loss, test_accuracy = [], []
    count = 0
    for epoch in range(num_epochs):
        start = time.time()
        cnn.train()
        total_train_loss = 0
        for i, (images, labels) in enumerate(train_loader):
            images = torch.autograd.Variable(images)
            labels = torch.autograd.Variable(labels)
            count += images.size(0)
            if device.type == "cuda":
                images, labels = images.to("cuda"), labels.to("cuda")
            optimizer.zero_grad()
            pred = cnn(images)[0]

            loss = loss_func(pred, labels)

            # Do not optimize for epoch 0
            if epoch > 0:
                loss.backward()
                optimizer.step()
            # Track the loss
            total_train_loss += loss.item()
            if count % 2000 != 0:
                print("Epoch [{}/{}], Step [{}/{}], Loss: {:.4f}".
 ↪format(epoch+1, num_epochs, i+1, total_step, loss.item()))
                train_loss.append(total_train_loss/(i+1))
        epoch_train_time = time.time() - start
        total_train_time += epoch_train_time
        print("Epoch Train Time: {:.2f} secs".format(epoch_train_time))
        test_accuracy = test(cnn, test_loader, test_accuracy)
    plot_model(train_loss, test_accuracy)
    print("Total Train Time: {:.2f} mins".format(total_train_time/60))
    return train_loss, test_accuracy

# Track the test accuracy
def test(cnn, test_loader, test_accuracy):
    cnn.eval()
```

```
    accuracy = 0
    count = 0
    with torch.no_grad():
        for i, (images, labels) in enumerate(test_loader):
            count += images.size(0)
            if device.type == "cuda":
                images, labels = images.to("cuda"), labels.to("cuda")
            test_output, last_layer = cnn(images)
            pred_y = torch.max(test_output, 1)[1].data.squeeze()
            accuracy = (pred_y == labels).sum().item() / float(labels.size(0))
            if count % 2000 != 0:
                test_accuracy.append(accuracy * 100.0)
    print("test_accuracy= ",test_accuracy[-1])
    return test_accuracy

# Plot the traning loss and testing accuracy
def plot_model(train_loss, test_accuracy):
    plt.figure(1, figsize=(15, 8))
    plt.plot(train_loss, label = "Training Loss", linewidth = 2.5)
    plt.ylabel("Cross Entropy Loss")
    plt.grid()
    plt.legend()
    plt.figure(2)
    plt.figure(figsize=(15,8))
    plt.plot(test_accuracy, label = "Testing Accuracy", linewidth = 2.5)
    plt.ylabel("Testing Accuracy in %")
    plt.grid()
    plt.legend()
```

5 Model 1

[10]:
```
# Download the training and testing data
mini_batch = 30
train_loader = trainload(mini_batch)
test_loader = testload(mini_batch)

# Initialize model
cnn1 = CNN()
if device.type == "cuda":
    cnn1.to('cuda')

# Define loss function
loss_func = torch.nn.CrossEntropyLoss()

# Define optimizer
optimizer = torch.optim.SGD(cnn1.parameters(), lr = 0.01)
```

```
# Train the model
num_epochs = 30
train_loss, test_accuracy = train(num_epochs, cnn1, train_loader, test_loader,␣
 ↪mini_batch)


#Logging the experiments
exp_name = f"loss: CXE, optimizer: SGD, lr: 0.001, epoch: 30, mini_batch=30"
expLog_R.loc[0,:3] = [f"{exp_name}"] + [f"{np.round(train_loss[-1],3)}"] +␣
 ↪[f"{np.round(test_accuracy[-1],3)}"]
expLog_R
```

```
Epoch [1/30], Step [1/3], Loss: 6.2015
Epoch [1/30], Step [2/3], Loss: 6.1896
Epoch [1/30], Step [3/3], Loss: 6.2129
Epoch Train Time: 26.26 secs
test_accuracy=  0.0
Epoch [2/30], Step [1/3], Loss: 6.1998
Epoch [2/30], Step [2/3], Loss: 5.5873
Epoch [2/30], Step [3/3], Loss: 4.4518
Epoch Train Time: 0.57 secs
test_accuracy=  43.47826086956522
Epoch [3/30], Step [1/3], Loss: 1.4402
Epoch [3/30], Step [2/3], Loss: 0.7330
Epoch [3/30], Step [3/3], Loss: 3.0893
Epoch Train Time: 0.53 secs
test_accuracy=  56.52173913043478
Epoch [4/30], Step [1/3], Loss: 5.2821
Epoch [4/30], Step [2/3], Loss: 4.7617
Epoch [4/30], Step [3/3], Loss: 2.3485
Epoch Train Time: 0.51 secs
test_accuracy=  65.21739130434783
Epoch [5/30], Step [1/3], Loss: 0.6525
Epoch [5/30], Step [2/3], Loss: 0.9387
Epoch [5/30], Step [3/3], Loss: 2.8728
Epoch Train Time: 0.48 secs
test_accuracy=  43.47826086956522
Epoch [6/30], Step [1/3], Loss: 1.6646
Epoch [6/30], Step [2/3], Loss: 0.7794
Epoch [6/30], Step [3/3], Loss: 0.9253
Epoch Train Time: 0.48 secs
test_accuracy=  56.52173913043478
Epoch [7/30], Step [1/3], Loss: 1.7697
Epoch [7/30], Step [2/3], Loss: 0.9329
Epoch [7/30], Step [3/3], Loss: 0.8472
Epoch Train Time: 0.49 secs
```

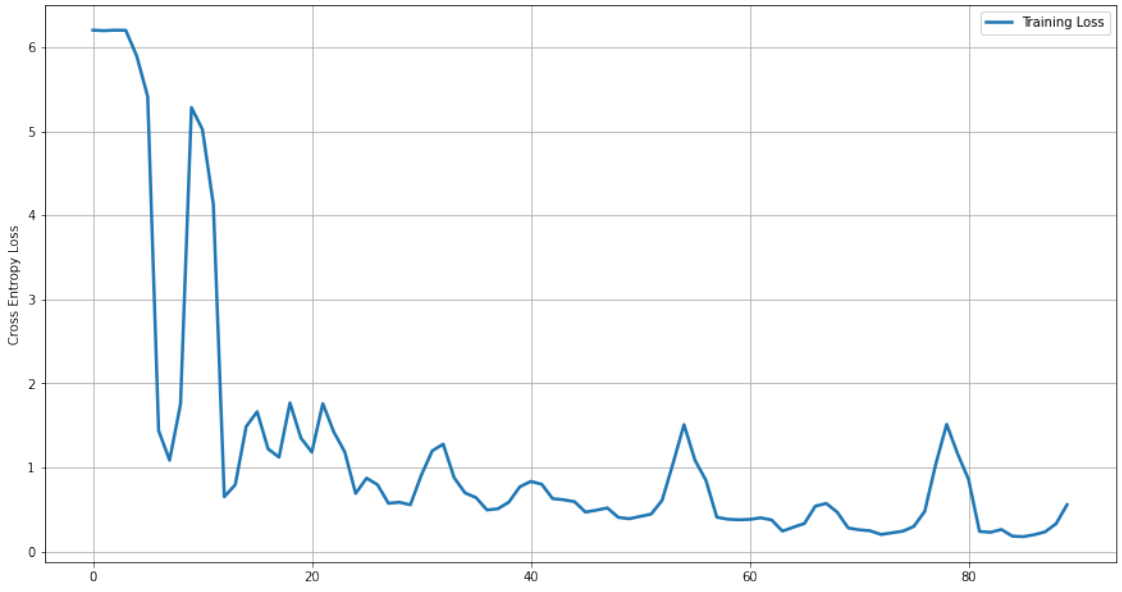
```
test_accuracy=  56.52173913043478
Epoch [8/30], Step [1/3], Loss: 1.7610
Epoch [8/30], Step [2/3], Loss: 1.0882
Epoch [8/30], Step [3/3], Loss: 0.7152
Epoch Train Time: 0.48 secs
test_accuracy=  56.52173913043478
Epoch [9/30], Step [1/3], Loss: 0.6922
Epoch [9/30], Step [2/3], Loss: 1.0558
Epoch [9/30], Step [3/3], Loss: 0.6399
Epoch Train Time: 0.47 secs
test_accuracy=  65.21739130434783
Epoch [10/30], Step [1/3], Loss: 0.5747
Epoch [10/30], Step [2/3], Loss: 0.5995
Epoch [10/30], Step [3/3], Loss: 0.4962
Epoch Train Time: 0.48 secs
test_accuracy=  43.47826086956522
Epoch [11/30], Step [1/3], Loss: 0.9101
Epoch [11/30], Step [2/3], Loss: 1.4883
Epoch [11/30], Step [3/3], Loss: 1.4366
Epoch Train Time: 0.48 secs
test_accuracy=  56.52173913043478
Epoch [12/30], Step [1/3], Loss: 0.8791
Epoch [12/30], Step [2/3], Loss: 0.5168
Epoch [12/30], Step [3/3], Loss: 0.5318
Epoch Train Time: 0.48 secs
test_accuracy=  56.52173913043478
Epoch [13/30], Step [1/3], Loss: 0.4957
Epoch [13/30], Step [2/3], Loss: 0.5239
Epoch [13/30], Step [3/3], Loss: 0.7461
Epoch Train Time: 0.48 secs
test_accuracy=  56.52173913043478
Epoch [14/30], Step [1/3], Loss: 0.7711
Epoch [14/30], Step [2/3], Loss: 0.8994
Epoch [14/30], Step [3/3], Loss: 0.7346
Epoch Train Time: 0.49 secs
test_accuracy=  47.82608695652174
Epoch [15/30], Step [1/3], Loss: 0.6301
Epoch [15/30], Step [2/3], Loss: 0.6039
Epoch [15/30], Step [3/3], Loss: 0.5499
Epoch Train Time: 0.49 secs
test_accuracy=  56.52173913043478
Epoch [16/30], Step [1/3], Loss: 0.4702
Epoch [16/30], Step [2/3], Loss: 0.5161
Epoch [16/30], Step [3/3], Loss: 0.5725
Epoch Train Time: 0.49 secs
test_accuracy=  43.47826086956522
Epoch [17/30], Step [1/3], Loss: 0.4071
Epoch [17/30], Step [2/3], Loss: 0.3754
```

```
Epoch [17/30], Step [3/3], Loss: 0.4744
Epoch Train Time: 0.49 secs
test_accuracy=  43.47826086956522
Epoch [18/30], Step [1/3], Loss: 0.4464
Epoch [18/30], Step [2/3], Loss: 0.7725
Epoch [18/30], Step [3/3], Loss: 1.9283
Epoch Train Time: 0.47 secs
test_accuracy=  56.52173913043478
Epoch [19/30], Step [1/3], Loss: 1.5108
Epoch [19/30], Step [2/3], Loss: 0.6678
Epoch [19/30], Step [3/3], Loss: 0.3710
Epoch Train Time: 0.49 secs
test_accuracy=  56.52173913043478
Epoch [20/30], Step [1/3], Loss: 0.4094
Epoch [20/30], Step [2/3], Loss: 0.3618
Epoch [20/30], Step [3/3], Loss: 0.3652
Epoch Train Time: 0.48 secs
test_accuracy=  43.47826086956522
Epoch [21/30], Step [1/3], Loss: 0.3835
Epoch [21/30], Step [2/3], Loss: 0.4213
Epoch [21/30], Step [3/3], Loss: 0.3282
Epoch Train Time: 0.49 secs
test_accuracy=  47.82608695652174
Epoch [22/30], Step [1/3], Loss: 0.2440
Epoch [22/30], Step [2/3], Loss: 0.3378
Epoch [22/30], Step [3/3], Loss: 0.4207
Epoch Train Time: 0.47 secs
test_accuracy=  56.52173913043478
Epoch [23/30], Step [1/3], Loss: 0.5412
Epoch [23/30], Step [2/3], Loss: 0.6063
Epoch [23/30], Step [3/3], Loss: 0.2614
Epoch Train Time: 0.48 secs
test_accuracy=  47.82608695652174
Epoch [24/30], Step [1/3], Loss: 0.2821
Epoch [24/30], Step [2/3], Loss: 0.2372
Epoch [24/30], Step [3/3], Loss: 0.2265
Epoch Train Time: 0.48 secs
test_accuracy=  52.17391304347826
Epoch [25/30], Step [1/3], Loss: 0.2042
Epoch [25/30], Step [2/3], Loss: 0.2437
Epoch [25/30], Step [3/3], Loss: 0.2819
Epoch Train Time: 0.48 secs
test_accuracy=  52.17391304347826
Epoch [26/30], Step [1/3], Loss: 0.3000
Epoch [26/30], Step [2/3], Loss: 0.6606
Epoch [26/30], Step [3/3], Loss: 2.1585
Epoch Train Time: 0.49 secs
test_accuracy=  43.47826086956522
```

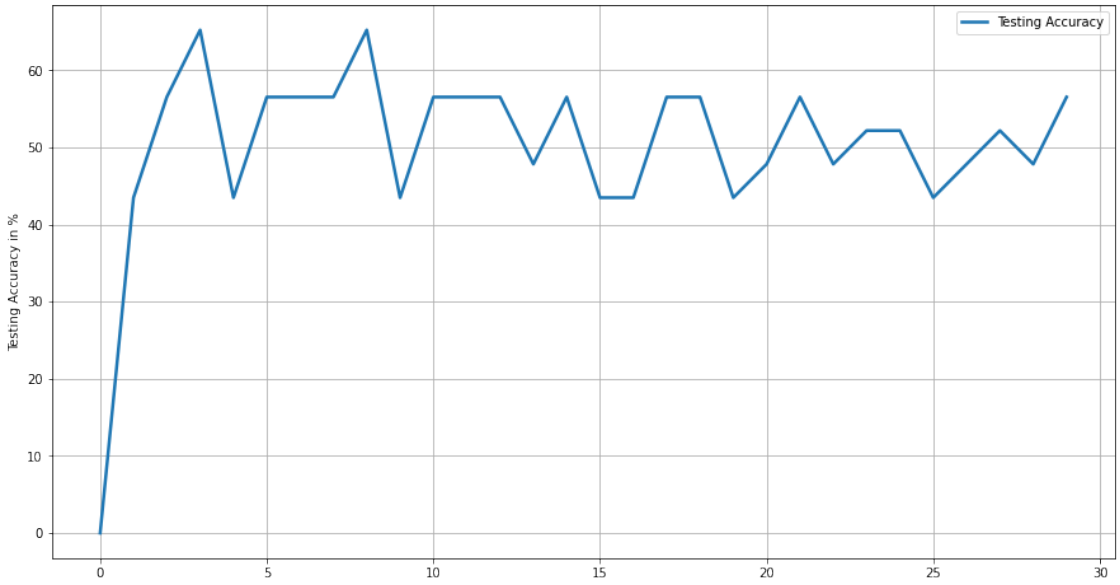
```
Epoch [27/30], Step [1/3], Loss: 1.5144
Epoch [27/30], Step [2/3], Loss: 0.8141
Epoch [27/30], Step [3/3], Loss: 0.2460
Epoch Train Time: 0.48 secs
test_accuracy=  47.82608695652174
Epoch [28/30], Step [1/3], Loss: 0.2401
Epoch [28/30], Step [2/3], Loss: 0.2199
Epoch [28/30], Step [3/3], Loss: 0.3301
Epoch Train Time: 0.48 secs
test_accuracy=  52.17391304347826
Epoch [29/30], Step [1/3], Loss: 0.1840
Epoch [29/30], Step [2/3], Loss: 0.1720
Epoch [29/30], Step [3/3], Loss: 0.2484
Epoch Train Time: 0.48 secs
test_accuracy=  47.82608695652174
Epoch [30/30], Step [1/3], Loss: 0.2356
Epoch [30/30], Step [2/3], Loss: 0.4293
Epoch [30/30], Step [3/3], Loss: 1.0111
Epoch Train Time: 0.50 secs
test_accuracy=  56.52173913043478
Total Train Time: 0.67 mins

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:24: FutureWarning:
Slicing a positional slice with .loc is not supported, and will raise TypeError
in a future version.  Use .loc with labels or .iloc with positions instead.
```

[10]:

	exp_name	Loss	Accuracy
0	loss: CXE, optimizer: SGD, lr: 0.001, epoch: 3...	0.559	56.522



```
<Figure size 432x288 with 0 Axes>
```



6 Model 2

[18]:
```
# Download the training and testing data
mini_batch = 20
train_loader = trainload(mini_batch)
test_loader = testload(mini_batch)

# Initialize model
cnn2 = CNN()
if device.type == "cuda":
    cnn2.to('cuda')

# Define loss function
loss_func = torch.nn.CrossEntropyLoss()

# Define optimizer
optimizer = torch.optim.SGD(cnn2.parameters(), lr = 0.01)

# Train the model
num_epochs = 30

train_loss, test_accuracy = train(num_epochs, cnn2, train_loader, test_loader,␣
 ↪mini_batch)
```

```
#Logging the experiments
exp_name = f"loss: CXE, optimizer: SGD, lr: 0.01, epoch: 30, mini_batch=20"
expLog_R.loc[1,:3] = [f"{exp_name}"] + [f"{np.round(train_loss[-1],3)}"] +␣
 ↪[f"{np.round(test_accuracy[-1],3)}"]
expLog_R
```

```
Epoch [1/30], Step [1/4], Loss: 6.1968
Epoch [1/30], Step [2/4], Loss: 6.1972
Epoch [1/30], Step [3/4], Loss: 6.2051
Epoch [1/30], Step [4/4], Loss: 6.1895
Epoch Train Time: 0.33 secs
test_accuracy=  0.0
Epoch [2/30], Step [1/4], Loss: 6.2113
Epoch [2/30], Step [2/4], Loss: 5.4449
Epoch [2/30], Step [3/4], Loss: 4.0893
Epoch [2/30], Step [4/4], Loss: 1.1411
Epoch Train Time: 0.63 secs
test_accuracy=  33.33333333333333
Epoch [3/30], Step [1/4], Loss: 0.6393
Epoch [3/30], Step [2/4], Loss: 2.2571
Epoch [3/30], Step [3/4], Loss: 6.9173
Epoch [3/30], Step [4/4], Loss: 5.2226
Epoch Train Time: 0.64 secs
test_accuracy=  66.66666666666666
Epoch [4/30], Step [1/4], Loss: 3.7392
Epoch [4/30], Step [2/4], Loss: 2.4260
Epoch [4/30], Step [3/4], Loss: 2.3619
Epoch [4/30], Step [4/4], Loss: 0.9046
Epoch Train Time: 0.57 secs
test_accuracy=  66.66666666666666
Epoch [5/30], Step [1/4], Loss: 0.9665
Epoch [5/30], Step [2/4], Loss: 1.1883
Epoch [5/30], Step [3/4], Loss: 0.8127
Epoch [5/30], Step [4/4], Loss: 0.6914
Epoch Train Time: 0.55 secs
test_accuracy=  33.33333333333333
Epoch [6/30], Step [1/4], Loss: 0.6583
Epoch [6/30], Step [2/4], Loss: 0.6986
Epoch [6/30], Step [3/4], Loss: 1.0120
Epoch [6/30], Step [4/4], Loss: 1.8792
Epoch Train Time: 0.58 secs
test_accuracy=  100.0
Epoch [7/30], Step [1/4], Loss: 0.9917
Epoch [7/30], Step [2/4], Loss: 0.8016
Epoch [7/30], Step [3/4], Loss: 0.6023
Epoch [7/30], Step [4/4], Loss: 0.6381
```

```
Epoch Train Time: 0.57 secs
test_accuracy=  33.33333333333333
Epoch [8/30], Step [1/4], Loss: 0.6658
Epoch [8/30], Step [2/4], Loss: 0.9039
Epoch [8/30], Step [3/4], Loss: 1.3293
Epoch [8/30], Step [4/4], Loss: 0.7970
Epoch Train Time: 0.56 secs
test_accuracy=  33.33333333333333
Epoch [9/30], Step [1/4], Loss: 0.6587
Epoch [9/30], Step [2/4], Loss: 0.6482
Epoch [9/30], Step [3/4], Loss: 0.9863
Epoch [9/30], Step [4/4], Loss: 1.5293
Epoch Train Time: 0.60 secs
test_accuracy=  100.0
Epoch [10/30], Step [1/4], Loss: 1.1779
Epoch [10/30], Step [2/4], Loss: 0.7171
Epoch [10/30], Step [3/4], Loss: 0.6147
Epoch [10/30], Step [4/4], Loss: 0.7536
Epoch Train Time: 0.59 secs
test_accuracy=  100.0
Epoch [11/30], Step [1/4], Loss: 0.6846
Epoch [11/30], Step [2/4], Loss: 0.6169
Epoch [11/30], Step [3/4], Loss: 0.6506
Epoch [11/30], Step [4/4], Loss: 0.5441
Epoch Train Time: 0.58 secs
test_accuracy=  33.33333333333333
Epoch [12/30], Step [1/4], Loss: 0.7274
Epoch [12/30], Step [2/4], Loss: 1.0719
Epoch [12/30], Step [3/4], Loss: 1.2163
Epoch [12/30], Step [4/4], Loss: 0.7169
Epoch Train Time: 0.58 secs
test_accuracy=  33.33333333333333
Epoch [13/30], Step [1/4], Loss: 0.5477
Epoch [13/30], Step [2/4], Loss: 0.5401
Epoch [13/30], Step [3/4], Loss: 0.6113
Epoch [13/30], Step [4/4], Loss: 0.5588
Epoch Train Time: 0.58 secs
test_accuracy=  100.0
Epoch [14/30], Step [1/4], Loss: 0.5218
Epoch [14/30], Step [2/4], Loss: 0.6147
Epoch [14/30], Step [3/4], Loss: 0.8128
Epoch [14/30], Step [4/4], Loss: 0.6154
Epoch Train Time: 0.57 secs
test_accuracy=  0.0
Epoch [15/30], Step [1/4], Loss: 0.5360
Epoch [15/30], Step [2/4], Loss: 0.5005
Epoch [15/30], Step [3/4], Loss: 0.5595
Epoch [15/30], Step [4/4], Loss: 0.4659
```

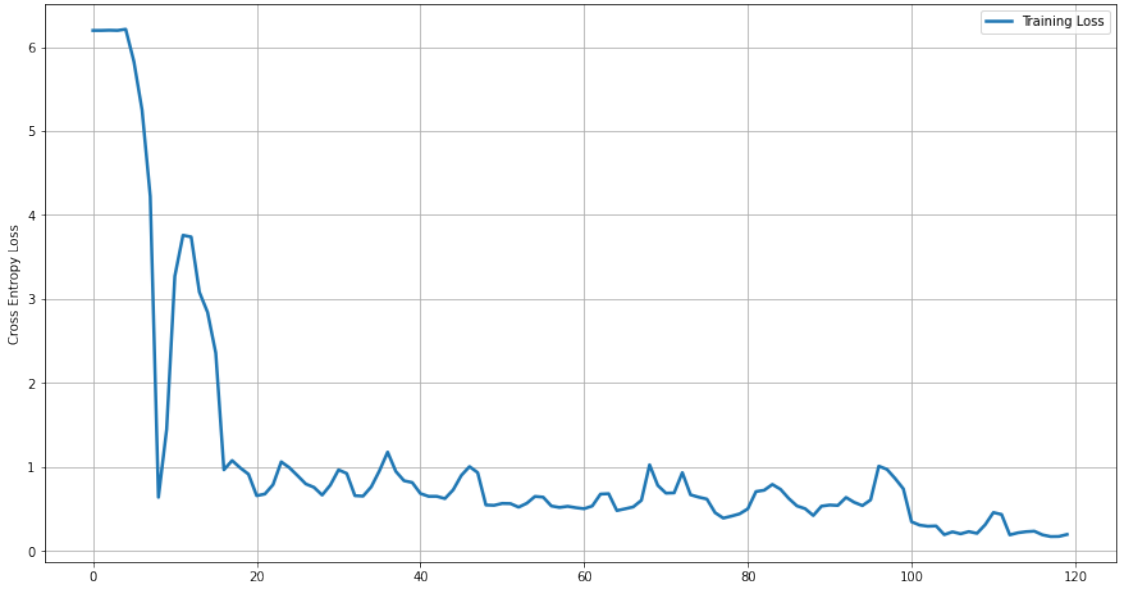
```
Epoch Train Time: 0.55 secs
test_accuracy=  66.66666666666666
Epoch [16/30], Step [1/4], Loss: 0.5036
Epoch [16/30], Step [2/4], Loss: 0.5679
Epoch [16/30], Step [3/4], Loss: 0.9611
Epoch [16/30], Step [4/4], Loss: 0.6972
Epoch Train Time: 0.57 secs
test_accuracy=  0.0
Epoch [17/30], Step [1/4], Loss: 0.4815
Epoch [17/30], Step [2/4], Loss: 0.5227
Epoch [17/30], Step [3/4], Loss: 0.5697
Epoch [17/30], Step [4/4], Loss: 0.8397
Epoch Train Time: 0.57 secs
test_accuracy=  66.66666666666666
Epoch [18/30], Step [1/4], Loss: 1.0269
Epoch [18/30], Step [2/4], Loss: 0.5349
Epoch [18/30], Step [3/4], Loss: 0.5042
Epoch [18/30], Step [4/4], Loss: 0.7036
Epoch Train Time: 0.58 secs
test_accuracy=  0.0
Epoch [19/30], Step [1/4], Loss: 0.9337
Epoch [19/30], Step [2/4], Loss: 0.4051
Epoch [19/30], Step [3/4], Loss: 0.5843
Epoch [19/30], Step [4/4], Loss: 0.5514
Epoch Train Time: 0.57 secs
test_accuracy=  66.66666666666666
Epoch [20/30], Step [1/4], Loss: 0.4565
Epoch [20/30], Step [2/4], Loss: 0.3268
Epoch [20/30], Step [3/4], Loss: 0.4619
Epoch [20/30], Step [4/4], Loss: 0.5243
Epoch Train Time: 0.55 secs
test_accuracy=  33.33333333333333
Epoch [21/30], Step [1/4], Loss: 0.5022
Epoch [21/30], Step [2/4], Loss: 0.9137
Epoch [21/30], Step [3/4], Loss: 0.7546
Epoch [21/30], Step [4/4], Loss: 1.0017
Epoch Train Time: 0.58 secs
test_accuracy=  100.0
Epoch [22/30], Step [1/4], Loss: 0.7345
Epoch [22/30], Step [2/4], Loss: 0.5172
Epoch [22/30], Step [3/4], Loss: 0.3588
Epoch [22/30], Step [4/4], Loss: 0.4060
Epoch Train Time: 0.56 secs
test_accuracy=  33.33333333333333
Epoch [23/30], Step [1/4], Loss: 0.4224
Epoch [23/30], Step [2/4], Loss: 0.6451
Epoch [23/30], Step [3/4], Loss: 0.5752
Epoch [23/30], Step [4/4], Loss: 0.5261
```

```
Epoch Train Time: 0.57 secs
test_accuracy=  33.33333333333333
Epoch [24/30], Step [1/4], Loss: 0.6378
Epoch [24/30], Step [2/4], Loss: 0.5211
Epoch [24/30], Step [3/4], Loss: 0.4627
Epoch [24/30], Step [4/4], Loss: 0.8115
Epoch Train Time: 0.57 secs
test_accuracy=  66.66666666666666
Epoch [25/30], Step [1/4], Loss: 1.0106
Epoch [25/30], Step [2/4], Loss: 0.9335
Epoch [25/30], Step [3/4], Loss: 0.6450
Epoch [25/30], Step [4/4], Loss: 0.3730
Epoch Train Time: 0.57 secs
test_accuracy=  100.0
Epoch [26/30], Step [1/4], Loss: 0.3482
Epoch [26/30], Step [2/4], Loss: 0.2678
Epoch [26/30], Step [3/4], Loss: 0.2681
Epoch [26/30], Step [4/4], Loss: 0.3100
Epoch Train Time: 0.57 secs
test_accuracy=  66.66666666666666
Epoch [27/30], Step [1/4], Loss: 0.1945
Epoch [27/30], Step [2/4], Loss: 0.2622
Epoch [27/30], Step [3/4], Loss: 0.1571
Epoch [27/30], Step [4/4], Loss: 0.3119
Epoch Train Time: 0.57 secs
test_accuracy=  33.33333333333333
Epoch [28/30], Step [1/4], Loss: 0.2103
Epoch [28/30], Step [2/4], Loss: 0.4161
Epoch [28/30], Step [3/4], Loss: 0.7491
Epoch [28/30], Step [4/4], Loss: 0.3665
Epoch Train Time: 0.56 secs
test_accuracy=  66.66666666666666
Epoch [29/30], Step [1/4], Loss: 0.1920
Epoch [29/30], Step [2/4], Loss: 0.2419
Epoch [29/30], Step [3/4], Loss: 0.2568
Epoch [29/30], Step [4/4], Loss: 0.2541
Epoch Train Time: 0.57 secs
test_accuracy=  66.66666666666666
Epoch [30/30], Step [1/4], Loss: 0.1924
Epoch [30/30], Step [2/4], Loss: 0.1512
Epoch [30/30], Step [3/4], Loss: 0.1758
Epoch [30/30], Step [4/4], Loss: 0.2697
Epoch Train Time: 0.58 secs
test_accuracy=  33.33333333333333
Total Train Time: 0.28 mins

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:25: FutureWarning:
Slicing a positional slice with .loc is not supported, and will raise TypeError
```

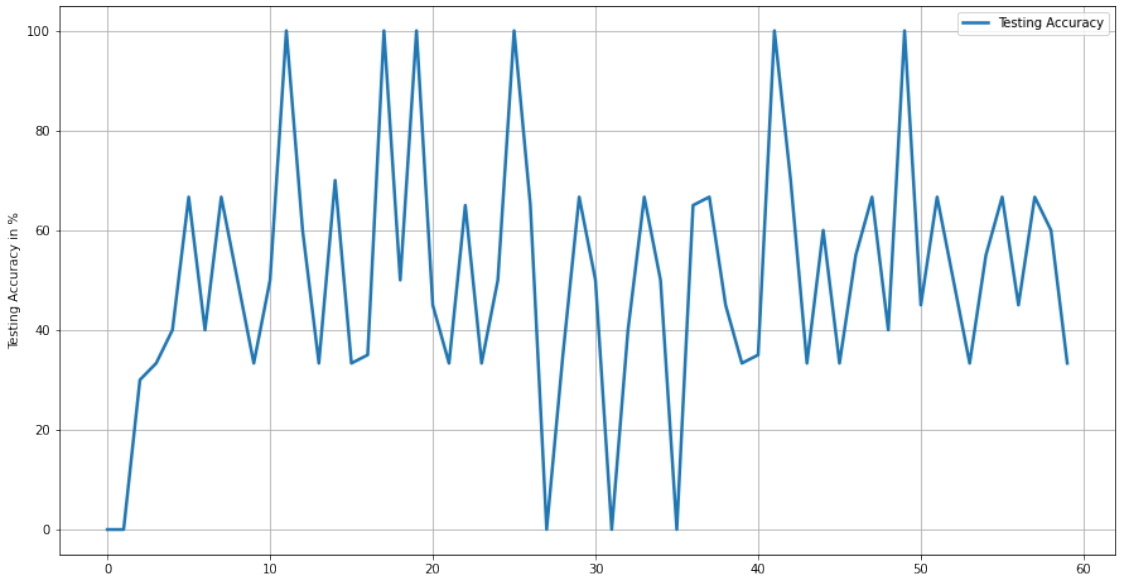
```
in a future version.  Use .loc with labels or .iloc with positions instead.
```

[18]:

	exp_name	Loss	Accuracy
0	loss: CXE, optimizer: SGD, lr: 0.001, epoch: 3...	0.35	60.87
1	loss: CXE, optimizer: SGD, lr: 0.01, epoch: 30...	0.172	100.0




```
<Figure size 432x288 with 0 Axes>
```



7 Model 3

[11]:
```
# Download the training and testing data
mini_batch = 50
train_loader = trainload(mini_batch)
test_loader = testload(mini_batch)

# Initialize model
cnn3 = CNN()
if device.type == "cuda":
    cnn3.to('cuda')

# Define loss function
loss_func = torch.nn.CrossEntropyLoss()

# Define optimizer
optimizer = torch.optim.SGD(cnn3.parameters(), lr = 0.01, momentum=0.3)

# Train the model
num_epochs = 40

train_loss, test_accuracy = train(num_epochs, cnn3, train_loader, test_loader,␣
 ↪mini_batch)


#Logging the experiments
exp_name = f"loss: CXE, optimizer: SGD, lr: 0.01, epoch: 40, mini_batch=50"
expLog_R.loc[2,:3] = [f"{exp_name}"] + [f"{np.round(train_loss[-1],3)}"] +␣
 ↪[f"{np.round(test_accuracy[-1],3)}"]
expLog_R
```

```
Epoch [1/40], Step [1/2], Loss: 6.1574
Epoch [1/40], Step [2/2], Loss: 6.1980
Epoch Train Time: 0.30 secs
test_accuracy=  0.0
Epoch [2/40], Step [1/2], Loss: 6.1755
Epoch [2/40], Step [2/2], Loss: 4.8990
Epoch Train Time: 0.48 secs
test_accuracy=  43.47826086956522
Epoch [3/40], Step [1/2], Loss: 1.8473
Epoch [3/40], Step [2/2], Loss: 1.1476
Epoch Train Time: 0.47 secs
test_accuracy=  56.52173913043478
Epoch [4/40], Step [1/2], Loss: 18.5119
Epoch [4/40], Step [2/2], Loss: 5.6510
Epoch Train Time: 0.46 secs
test_accuracy=  43.47826086956522
```

```
Epoch [5/40], Step [1/2], Loss: 5.6504
Epoch [5/40], Step [2/2], Loss: 5.0117
Epoch Train Time: 0.45 secs
test_accuracy=  43.47826086956522
Epoch [6/40], Step [1/2], Loss: 2.9347
Epoch [6/40], Step [2/2], Loss: 2.7541
Epoch Train Time: 0.45 secs
test_accuracy=  56.52173913043478
Epoch [7/40], Step [1/2], Loss: 1.5520
Epoch [7/40], Step [2/2], Loss: 1.4636
Epoch Train Time: 0.44 secs
test_accuracy=  43.47826086956522
Epoch [8/40], Step [1/2], Loss: 0.9901
Epoch [8/40], Step [2/2], Loss: 1.8018
Epoch Train Time: 0.44 secs
test_accuracy=  43.47826086956522
Epoch [9/40], Step [1/2], Loss: 1.4001
Epoch [9/40], Step [2/2], Loss: 1.2175
Epoch Train Time: 0.44 secs
test_accuracy=  43.47826086956522
Epoch [10/40], Step [1/2], Loss: 0.6944
Epoch [10/40], Step [2/2], Loss: 0.6533
Epoch Train Time: 0.44 secs
test_accuracy=  43.47826086956522
Epoch [11/40], Step [1/2], Loss: 0.6453
Epoch [11/40], Step [2/2], Loss: 0.6280
Epoch Train Time: 0.44 secs
test_accuracy=  65.21739130434783
Epoch [12/40], Step [1/2], Loss: 0.6047
Epoch [12/40], Step [2/2], Loss: 0.7173
Epoch Train Time: 0.44 secs
test_accuracy=  43.47826086956522
Epoch [13/40], Step [1/2], Loss: 1.4431
Epoch [13/40], Step [2/2], Loss: 1.1706
Epoch Train Time: 0.44 secs
test_accuracy=  52.17391304347826
Epoch [14/40], Step [1/2], Loss: 0.7932
Epoch [14/40], Step [2/2], Loss: 0.7671
Epoch Train Time: 0.44 secs
test_accuracy=  56.52173913043478
Epoch [15/40], Step [1/2], Loss: 0.6946
Epoch [15/40], Step [2/2], Loss: 0.6400
Epoch Train Time: 0.44 secs
test_accuracy=  43.47826086956522
Epoch [16/40], Step [1/2], Loss: 0.8525
Epoch [16/40], Step [2/2], Loss: 0.6431
Epoch Train Time: 0.45 secs
test_accuracy=  56.52173913043478
```

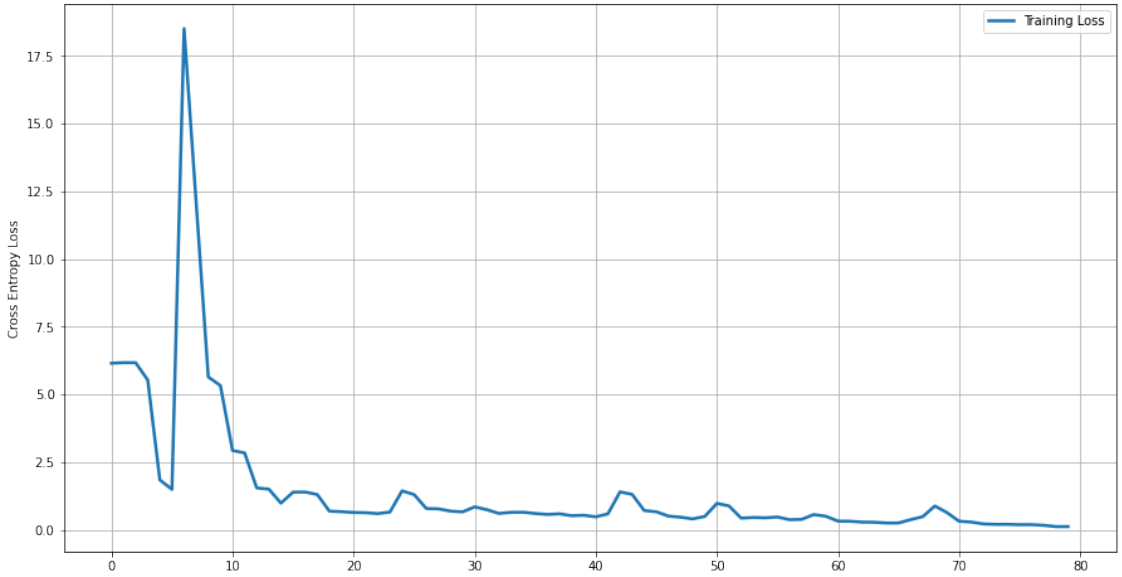
```
Epoch [17/40], Step [1/2], Loss: 0.6104
Epoch [17/40], Step [2/2], Loss: 0.6913
Epoch Train Time: 0.44 secs
test_accuracy=  56.52173913043478
Epoch [18/40], Step [1/2], Loss: 0.6549
Epoch [18/40], Step [2/2], Loss: 0.5617
Epoch Train Time: 0.46 secs
test_accuracy=  47.82608695652174
Epoch [19/40], Step [1/2], Loss: 0.5731
Epoch [19/40], Step [2/2], Loss: 0.6180
Epoch Train Time: 0.45 secs
test_accuracy=  52.17391304347826
Epoch [20/40], Step [1/2], Loss: 0.5265
Epoch [20/40], Step [2/2], Loss: 0.5587
Epoch Train Time: 0.44 secs
test_accuracy=  56.52173913043478
Epoch [21/40], Step [1/2], Loss: 0.4850
Epoch [21/40], Step [2/2], Loss: 0.7032
Epoch Train Time: 0.44 secs
test_accuracy=  56.52173913043478
Epoch [22/40], Step [1/2], Loss: 1.4069
Epoch [22/40], Step [2/2], Loss: 1.2196
Epoch Train Time: 0.44 secs
test_accuracy=  65.21739130434783
Epoch [23/40], Step [1/2], Loss: 0.7167
Epoch [23/40], Step [2/2], Loss: 0.6266
Epoch Train Time: 0.44 secs
test_accuracy=  52.17391304347826
Epoch [24/40], Step [1/2], Loss: 0.5115
Epoch [24/40], Step [2/2], Loss: 0.4315
Epoch Train Time: 0.45 secs
test_accuracy=  60.86956521739131
Epoch [25/40], Step [1/2], Loss: 0.4082
Epoch [25/40], Step [2/2], Loss: 0.5904
Epoch Train Time: 0.45 secs
test_accuracy=  56.52173913043478
Epoch [26/40], Step [1/2], Loss: 0.9805
Epoch [26/40], Step [2/2], Loss: 0.7937
Epoch Train Time: 0.45 secs
test_accuracy=  60.86956521739131
Epoch [27/40], Step [1/2], Loss: 0.4360
Epoch [27/40], Step [2/2], Loss: 0.4842
Epoch Train Time: 0.45 secs
test_accuracy=  56.52173913043478
Epoch [28/40], Step [1/2], Loss: 0.4485
Epoch [28/40], Step [2/2], Loss: 0.5085
Epoch Train Time: 0.44 secs
test_accuracy=  60.86956521739131
```

```
Epoch [29/40], Step [1/2], Loss: 0.3798
Epoch [29/40], Step [2/2], Loss: 0.4027
Epoch Train Time: 0.46 secs
test_accuracy=  56.52173913043478
Epoch [30/40], Step [1/2], Loss: 0.5678
Epoch [30/40], Step [2/2], Loss: 0.4432
Epoch Train Time: 0.44 secs
test_accuracy=  60.86956521739131
Epoch [31/40], Step [1/2], Loss: 0.3263
Epoch [31/40], Step [2/2], Loss: 0.3159
Epoch Train Time: 0.44 secs
test_accuracy=  47.82608695652174
Epoch [32/40], Step [1/2], Loss: 0.2902
Epoch [32/40], Step [2/2], Loss: 0.2795
Epoch Train Time: 0.43 secs
test_accuracy=  47.82608695652174
Epoch [33/40], Step [1/2], Loss: 0.2562
Epoch [33/40], Step [2/2], Loss: 0.2612
Epoch Train Time: 0.44 secs
test_accuracy=  43.47826086956522
Epoch [34/40], Step [1/2], Loss: 0.3811
Epoch [34/40], Step [2/2], Loss: 0.6023
Epoch Train Time: 0.45 secs
test_accuracy=  43.47826086956522
Epoch [35/40], Step [1/2], Loss: 0.8852
Epoch [35/40], Step [2/2], Loss: 0.3965
Epoch Train Time: 0.44 secs
test_accuracy=  56.52173913043478
Epoch [36/40], Step [1/2], Loss: 0.3220
Epoch [36/40], Step [2/2], Loss: 0.2607
Epoch Train Time: 0.44 secs
test_accuracy=  60.86956521739131
Epoch [37/40], Step [1/2], Loss: 0.2233
Epoch [37/40], Step [2/2], Loss: 0.1919
Epoch Train Time: 0.44 secs
test_accuracy=  47.82608695652174
Epoch [38/40], Step [1/2], Loss: 0.2072
Epoch [38/40], Step [2/2], Loss: 0.1853
Epoch Train Time: 0.47 secs
test_accuracy=  47.82608695652174
Epoch [39/40], Step [1/2], Loss: 0.1967
Epoch [39/40], Step [2/2], Loss: 0.1454
Epoch Train Time: 0.44 secs
test_accuracy=  47.82608695652174
Epoch [40/40], Step [1/2], Loss: 0.1235
Epoch [40/40], Step [2/2], Loss: 0.1240
Epoch Train Time: 0.44 secs
test_accuracy=  39.130434782608695
```

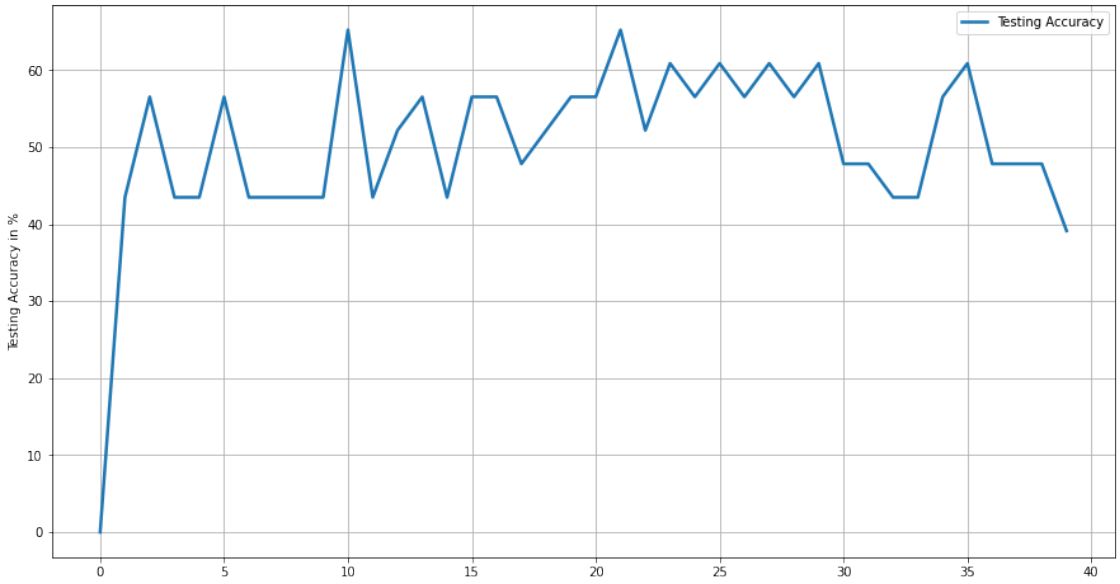
```
Total Train Time: 0.30 mins

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:25: FutureWarning:
Slicing a positional slice with .loc is not supported, and will raise TypeError
in a future version.  Use .loc with labels or .iloc with positions instead.
```

[11]:

	exp_name	Loss	Accuracy
0	loss: CXE, optimizer: SGD, lr: 0.001, epoch: 3...	0.559	56.522
2	loss: CXE, optimizer: SGD, lr: 0.01, epoch: 40...	0.124	39.13




```
<Figure size 432x288 with 0 Axes>
```



8 Model 4

[12]:
```
# Download the training and testing data
mini_batch = 35
train_loader = trainload(mini_batch)
test_loader = testload(mini_batch)

# Initialize model
cnn4 = CNN()
if device.type == "cuda":
    cnn4.to('cuda')

# Define loss function
loss_func = torch.nn.CrossEntropyLoss()

# Define optimizer
optimizer = torch.optim.SGD(cnn4.parameters(), lr = 0.01)

# Train the model
num_epochs = 40

train_loss, test_accuracy = train(num_epochs, cnn4, train_loader, test_loader,␣
 ↪mini_batch)


#Logging the experiments
exp_name = f"loss: CXE, optimizer: SGD, lr: 0.01, epoch: 40, mini_batch=35"
expLog_R.loc[3,:3] = [f"{exp_name}"] + [f"{np.round(train_loss[-1],3)}"] +␣
 ↪[f"{np.round(test_accuracy[-1],3)}"]
expLog_R
```

```
Epoch [1/40], Step [1/3], Loss: 6.1826
Epoch [1/40], Step [2/3], Loss: 6.1734
Epoch [1/40], Step [3/3], Loss: 6.1770
Epoch Train Time: 0.33 secs
test_accuracy=  0.0
Epoch [2/40], Step [1/3], Loss: 6.1773
Epoch [2/40], Step [2/3], Loss: 5.2992
Epoch [2/40], Step [3/3], Loss: 3.8482
Epoch Train Time: 0.53 secs
test_accuracy=  43.47826086956522
Epoch [3/40], Step [1/3], Loss: 3.6521
Epoch [3/40], Step [2/3], Loss: 2.9824
Epoch [3/40], Step [3/3], Loss: 2.0369
```

```
Epoch Train Time: 0.52 secs
test_accuracy=  43.47826086956522
Epoch [4/40], Step [1/3], Loss: 0.7755
Epoch [4/40], Step [2/3], Loss: 0.8082
Epoch [4/40], Step [3/3], Loss: 2.0007
Epoch Train Time: 0.50 secs
test_accuracy=  56.52173913043478
Epoch [5/40], Step [1/3], Loss: 2.0067
Epoch [5/40], Step [2/3], Loss: 1.7002
Epoch [5/40], Step [3/3], Loss: 0.4287
Epoch Train Time: 0.49 secs
test_accuracy=  56.52173913043478
Epoch [6/40], Step [1/3], Loss: 0.6907
Epoch [6/40], Step [2/3], Loss: 1.0833
Epoch [6/40], Step [3/3], Loss: 2.2005
Epoch Train Time: 0.49 secs
test_accuracy=  43.47826086956522
Epoch [7/40], Step [1/3], Loss: 1.0021
Epoch [7/40], Step [2/3], Loss: 0.6230
Epoch [7/40], Step [3/3], Loss: 0.7622
Epoch Train Time: 0.48 secs
test_accuracy=  56.52173913043478
Epoch [8/40], Step [1/3], Loss: 1.4832
Epoch [8/40], Step [2/3], Loss: 1.0773
Epoch [8/40], Step [3/3], Loss: 0.7884
Epoch Train Time: 0.51 secs
test_accuracy=  56.52173913043478
Epoch [9/40], Step [1/3], Loss: 0.6236
Epoch [9/40], Step [2/3], Loss: 0.6676
Epoch [9/40], Step [3/3], Loss: 1.0569
Epoch Train Time: 0.49 secs
test_accuracy=  43.47826086956522
Epoch [10/40], Step [1/3], Loss: 2.2309
Epoch [10/40], Step [2/3], Loss: 1.5930
Epoch [10/40], Step [3/3], Loss: 0.5067
Epoch Train Time: 0.49 secs
test_accuracy=  43.47826086956522
Epoch [11/40], Step [1/3], Loss: 0.5877
Epoch [11/40], Step [2/3], Loss: 0.6708
Epoch [11/40], Step [3/3], Loss: 2.3086
Epoch Train Time: 0.49 secs
test_accuracy=  56.52173913043478
Epoch [12/40], Step [1/3], Loss: 1.7735
Epoch [12/40], Step [2/3], Loss: 1.4267
Epoch [12/40], Step [3/3], Loss: 0.5175
Epoch Train Time: 0.49 secs
test_accuracy=  43.47826086956522
Epoch [13/40], Step [1/3], Loss: 0.9620
```

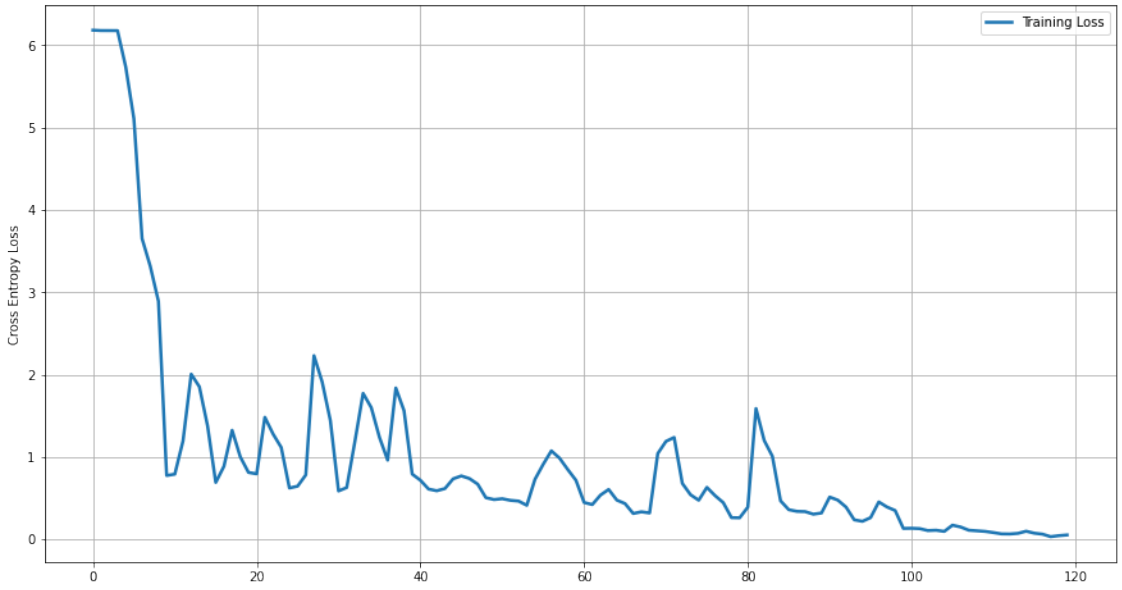
```
Epoch [13/40], Step [2/3], Loss: 2.7147
Epoch [13/40], Step [3/3], Loss: 1.0125
Epoch Train Time: 0.49 secs
test_accuracy=  43.47826086956522
Epoch [14/40], Step [1/3], Loss: 0.7927
Epoch [14/40], Step [2/3], Loss: 0.6432
Epoch [14/40], Step [3/3], Loss: 0.3966
Epoch Train Time: 0.48 secs
test_accuracy=  52.17391304347826
Epoch [15/40], Step [1/3], Loss: 0.5901
Epoch [15/40], Step [2/3], Loss: 0.6441
Epoch [15/40], Step [3/3], Loss: 0.9718
Epoch Train Time: 0.48 secs
test_accuracy=  56.52173913043478
Epoch [16/40], Step [1/3], Loss: 0.7709
Epoch [16/40], Step [2/3], Loss: 0.7061
Epoch [16/40], Step [3/3], Loss: 0.5356
Epoch Train Time: 0.49 secs
test_accuracy=  56.52173913043478
Epoch [17/40], Step [1/3], Loss: 0.5061
Epoch [17/40], Step [2/3], Loss: 0.4612
Epoch [17/40], Step [3/3], Loss: 0.5137
Epoch Train Time: 0.47 secs
test_accuracy=  60.86956521739131
Epoch [18/40], Step [1/3], Loss: 0.4740
Epoch [18/40], Step [2/3], Loss: 0.4560
Epoch [18/40], Step [3/3], Loss: 0.3106
Epoch Train Time: 0.48 secs
test_accuracy=  39.130434782608695
Epoch [19/40], Step [1/3], Loss: 0.7299
Epoch [19/40], Step [2/3], Loss: 1.0920
Epoch [19/40], Step [3/3], Loss: 1.4036
Epoch Train Time: 0.48 secs
test_accuracy=  56.52173913043478
Epoch [20/40], Step [1/3], Loss: 0.9867
Epoch [20/40], Step [2/3], Loss: 0.7111
Epoch [20/40], Step [3/3], Loss: 0.4572
Epoch Train Time: 0.48 secs
test_accuracy=  65.21739130434783
Epoch [21/40], Step [1/3], Loss: 0.4479
Epoch [21/40], Step [2/3], Loss: 0.3972
Epoch [21/40], Step [3/3], Loss: 0.7679
Epoch Train Time: 0.48 secs
test_accuracy=  43.47826086956522
Epoch [22/40], Step [1/3], Loss: 0.6065
Epoch [22/40], Step [2/3], Loss: 0.3466
Epoch [22/40], Step [3/3], Loss: 0.3511
Epoch Train Time: 0.48 secs
```

```
test_accuracy=  56.52173913043478
Epoch [23/40], Step [1/3], Loss: 0.3142
Epoch [23/40], Step [2/3], Loss: 0.3555
Epoch [23/40], Step [3/3], Loss: 0.2920
Epoch Train Time: 0.48 secs
test_accuracy=  43.47826086956522
Epoch [24/40], Step [1/3], Loss: 1.0425
Epoch [24/40], Step [2/3], Loss: 1.3381
Epoch [24/40], Step [3/3], Loss: 1.3333
Epoch Train Time: 0.47 secs
test_accuracy=  56.52173913043478
Epoch [25/40], Step [1/3], Loss: 0.6799
Epoch [25/40], Step [2/3], Loss: 0.4047
Epoch [25/40], Step [3/3], Loss: 0.3407
Epoch Train Time: 0.49 secs
test_accuracy=  56.52173913043478
Epoch [26/40], Step [1/3], Loss: 0.6316
Epoch [26/40], Step [2/3], Loss: 0.4310
Epoch [26/40], Step [3/3], Loss: 0.2761
Epoch Train Time: 0.50 secs
test_accuracy=  47.82608695652174
Epoch [27/40], Step [1/3], Loss: 0.2642
Epoch [27/40], Step [2/3], Loss: 0.2594
Epoch [27/40], Step [3/3], Loss: 0.6502
Epoch Train Time: 0.49 secs
test_accuracy=  56.52173913043478
Epoch [28/40], Step [1/3], Loss: 1.5890
Epoch [28/40], Step [2/3], Loss: 0.8140
Epoch [28/40], Step [3/3], Loss: 0.6255
Epoch Train Time: 0.49 secs
test_accuracy=  56.52173913043478
Epoch [29/40], Step [1/3], Loss: 0.4682
Epoch [29/40], Step [2/3], Loss: 0.2528
Epoch [29/40], Step [3/3], Loss: 0.2999
Epoch Train Time: 0.49 secs
test_accuracy=  52.17391304347826
Epoch [30/40], Step [1/3], Loss: 0.3374
Epoch [30/40], Step [2/3], Loss: 0.2747
Epoch [30/40], Step [3/3], Loss: 0.3482
Epoch Train Time: 0.48 secs
test_accuracy=  47.82608695652174
Epoch [31/40], Step [1/3], Loss: 0.5141
Epoch [31/40], Step [2/3], Loss: 0.4390
Epoch [31/40], Step [3/3], Loss: 0.2230
Epoch Train Time: 0.47 secs
test_accuracy=  52.17391304347826
Epoch [32/40], Step [1/3], Loss: 0.2375
Epoch [32/40], Step [2/3], Loss: 0.2012
```

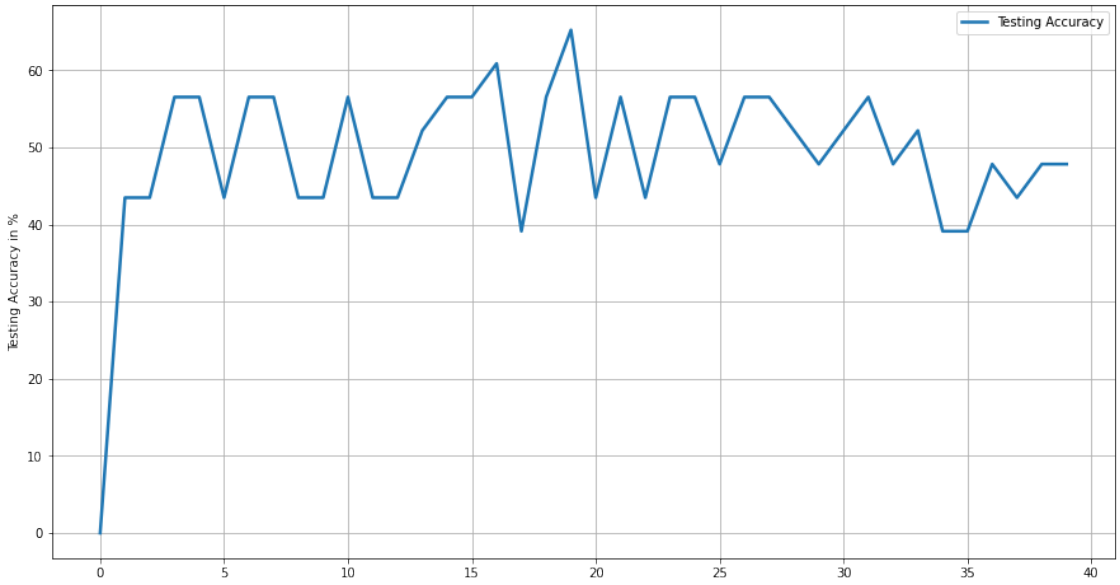
```
Epoch [32/40], Step [3/3], Loss: 0.3538
Epoch Train Time: 0.47 secs
test_accuracy=  56.52173913043478
Epoch [33/40], Step [1/3], Loss: 0.4550
Epoch [33/40], Step [2/3], Loss: 0.3319
Epoch [33/40], Step [3/3], Loss: 0.2651
Epoch Train Time: 0.48 secs
test_accuracy=  47.82608695652174
Epoch [34/40], Step [1/3], Loss: 0.1330
Epoch [34/40], Step [2/3], Loss: 0.1371
Epoch [34/40], Step [3/3], Loss: 0.1222
Epoch Train Time: 0.48 secs
test_accuracy=  52.17391304347826
Epoch [35/40], Step [1/3], Loss: 0.1072
Epoch [35/40], Step [2/3], Loss: 0.1151
Epoch [35/40], Step [3/3], Loss: 0.0686
Epoch Train Time: 0.50 secs
test_accuracy=  39.130434782608695
Epoch [36/40], Step [1/3], Loss: 0.1727
Epoch [36/40], Step [2/3], Loss: 0.1278
Epoch [36/40], Step [3/3], Loss: 0.0330
Epoch Train Time: 0.49 secs
test_accuracy=  39.130434782608695
Epoch [37/40], Step [1/3], Loss: 0.1040
Epoch [37/40], Step [2/3], Loss: 0.0896
Epoch [37/40], Step [3/3], Loss: 0.0544
Epoch Train Time: 0.49 secs
test_accuracy=  47.82608695652174
Epoch [38/40], Step [1/3], Loss: 0.0669
Epoch [38/40], Step [2/3], Loss: 0.0643
Epoch [38/40], Step [3/3], Loss: 0.0884
Epoch Train Time: 0.49 secs
test_accuracy=  43.47826086956522
Epoch [39/40], Step [1/3], Loss: 0.0992
Epoch [39/40], Step [2/3], Loss: 0.0515
Epoch [39/40], Step [3/3], Loss: 0.0401
Epoch Train Time: 0.49 secs
test_accuracy=  47.82608695652174
Epoch [40/40], Step [1/3], Loss: 0.0330
Epoch [40/40], Step [2/3], Loss: 0.0579
Epoch [40/40], Step [3/3], Loss: 0.0705
Epoch Train Time: 0.48 secs
test_accuracy=  47.82608695652174
Total Train Time: 0.32 mins

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:25: FutureWarning:
Slicing a positional slice with .loc is not supported, and will raise TypeError
in a future version.  Use .loc with labels or .iloc with positions instead.
```

[12]:

	exp_name	Loss	Accuracy
0	loss: CXE, optimizer: SGD, lr: 0.001, epoch: 3...	0.559	56.522
2	loss: CXE, optimizer: SGD, lr: 0.01, epoch: 40...	0.124	39.13
3	loss: CXE, optimizer: SGD, lr: 0.01, epoch: 40...	0.054	47.826



<Figure size 432x288 with 0 Axes>



9 Model 5

[13]:
```
##Lets focus on zero loss:
# Download the training and testing data
mini_batch = 30
train_loader = trainload(mini_batch)
test_loader = testload(mini_batch)

# Initialize model
cnn5 = CNN()
if device.type == "cuda":
    cnn5.to('cuda')

# Define loss function
loss_func = torch.nn.CrossEntropyLoss()

# Define optimizer
optimizer = torch.optim.SGD(cnn5.parameters(), lr = 0.01, momentum=0.3)

# Train the model
num_epochs = 30

train_loss, test_accuracy = train(num_epochs, cnn5, train_loader, test_loader,␣
 ↪mini_batch)


#Logging the experiments
exp_name = f"loss: CXE, optimizer: SGD, lr: 0.01, epoch: 30, mini_batch=30,␣
 ↪momentum=0.3"
expLog_R.loc[4,:3] = [f"{exp_name}"] + [f"{np.round(train_loss[-1],3)}"] +␣
 ↪[f"{np.round(test_accuracy[-1],3)}"]
expLog_R
```

```
Epoch [1/30], Step [1/3], Loss: 6.2161
Epoch [1/30], Step [2/3], Loss: 6.2167
Epoch [1/30], Step [3/3], Loss: 6.2425
Epoch Train Time: 0.29 secs
test_accuracy=  0.0
Epoch [2/30], Step [1/3], Loss: 6.2433
Epoch [2/30], Step [2/3], Loss: 5.7449
Epoch [2/30], Step [3/3], Loss: 4.8223
Epoch Train Time: 0.55 secs
test_accuracy=  69.56521739130434
Epoch [3/30], Step [1/3], Loss: 1.8911
Epoch [3/30], Step [2/3], Loss: 1.0819
Epoch [3/30], Step [3/3], Loss: 10.9624
Epoch Train Time: 0.53 secs
```

```
test_accuracy=  43.47826086956522
Epoch [4/30], Step [1/3], Loss: 3.7499
Epoch [4/30], Step [2/3], Loss: 2.5191
Epoch [4/30], Step [3/3], Loss: 2.2704
Epoch Train Time: 0.53 secs
test_accuracy=  43.47826086956522
Epoch [5/30], Step [1/3], Loss: 1.2591
Epoch [5/30], Step [2/3], Loss: 0.9362
Epoch [5/30], Step [3/3], Loss: 0.6670
Epoch Train Time: 0.55 secs
test_accuracy=  56.52173913043478
Epoch [6/30], Step [1/3], Loss: 0.9748
Epoch [6/30], Step [2/3], Loss: 1.5936
Epoch [6/30], Step [3/3], Loss: 0.7793
Epoch Train Time: 0.54 secs
test_accuracy=  56.52173913043478
Epoch [7/30], Step [1/3], Loss: 0.8104
Epoch [7/30], Step [2/3], Loss: 0.7088
Epoch [7/30], Step [3/3], Loss: 0.6763
Epoch Train Time: 0.55 secs
test_accuracy=  60.86956521739131
Epoch [8/30], Step [1/3], Loss: 0.6827
Epoch [8/30], Step [2/3], Loss: 0.9554
Epoch [8/30], Step [3/3], Loss: 1.1993
Epoch Train Time: 0.54 secs
test_accuracy=  43.47826086956522
Epoch [9/30], Step [1/3], Loss: 0.6424
Epoch [9/30], Step [2/3], Loss: 0.8590
Epoch [9/30], Step [3/3], Loss: 0.6279
Epoch Train Time: 0.55 secs
test_accuracy=  56.52173913043478
Epoch [10/30], Step [1/3], Loss: 0.9523
Epoch [10/30], Step [2/3], Loss: 0.8090
Epoch [10/30], Step [3/3], Loss: 0.6199
Epoch Train Time: 0.53 secs
test_accuracy=  56.52173913043478
Epoch [11/30], Step [1/3], Loss: 0.6977
Epoch [11/30], Step [2/3], Loss: 0.5991
Epoch [11/30], Step [3/3], Loss: 0.6376
Epoch Train Time: 0.54 secs
test_accuracy=  60.86956521739131
Epoch [12/30], Step [1/3], Loss: 0.6609
Epoch [12/30], Step [2/3], Loss: 0.6151
Epoch [12/30], Step [3/3], Loss: 0.5514
Epoch Train Time: 0.53 secs
test_accuracy=  56.52173913043478
Epoch [13/30], Step [1/3], Loss: 0.8082
Epoch [13/30], Step [2/3], Loss: 1.3392
```

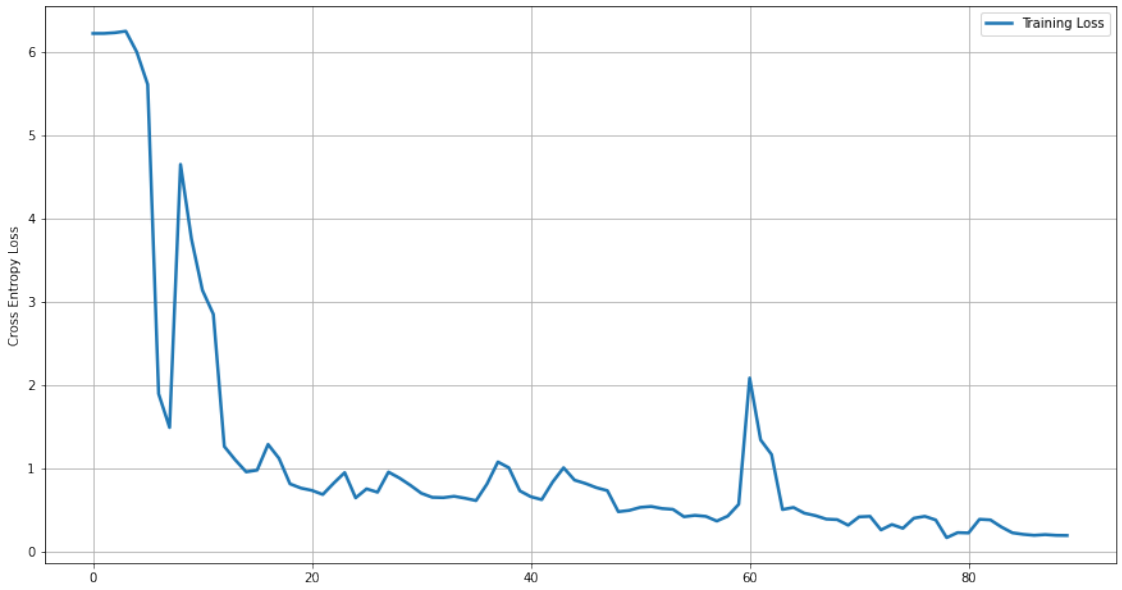
```
Epoch [13/30], Step [3/3], Loss: 0.8641
Epoch Train Time: 0.54 secs
test_accuracy=  56.52173913043478
Epoch [14/30], Step [1/3], Loss: 0.7266
Epoch [14/30], Step [2/3], Loss: 0.5859
Epoch [14/30], Step [3/3], Loss: 0.5503
Epoch Train Time: 0.54 secs
test_accuracy=  56.52173913043478
Epoch [15/30], Step [1/3], Loss: 0.8324
Epoch [15/30], Step [2/3], Loss: 1.1755
Epoch [15/30], Step [3/3], Loss: 0.5583
Epoch Train Time: 0.54 secs
test_accuracy=  56.52173913043478
Epoch [16/30], Step [1/3], Loss: 0.8150
Epoch [16/30], Step [2/3], Loss: 0.7140
Epoch [16/30], Step [3/3], Loss: 0.6588
Epoch Train Time: 0.53 secs
test_accuracy=  52.17391304347826
Epoch [17/30], Step [1/3], Loss: 0.4762
Epoch [17/30], Step [2/3], Loss: 0.5072
Epoch [17/30], Step [3/3], Loss: 0.6009
Epoch Train Time: 0.53 secs
test_accuracy=  43.47826086956522
Epoch [18/30], Step [1/3], Loss: 0.5392
Epoch [18/30], Step [2/3], Loss: 0.4891
Epoch [18/30], Step [3/3], Loss: 0.4848
Epoch Train Time: 0.53 secs
test_accuracy=  65.21739130434783
Epoch [19/30], Step [1/3], Loss: 0.4156
Epoch [19/30], Step [2/3], Loss: 0.4487
Epoch [19/30], Step [3/3], Loss: 0.3961
Epoch Train Time: 0.53 secs
test_accuracy=  56.52173913043478
Epoch [20/30], Step [1/3], Loss: 0.3638
Epoch [20/30], Step [2/3], Loss: 0.4829
Epoch [20/30], Step [3/3], Loss: 0.8487
Epoch Train Time: 0.54 secs
test_accuracy=  56.52173913043478
Epoch [21/30], Step [1/3], Loss: 2.0807
Epoch [21/30], Step [2/3], Loss: 0.5964
Epoch [21/30], Step [3/3], Loss: 0.8112
Epoch Train Time: 0.53 secs
test_accuracy=  56.52173913043478
Epoch [22/30], Step [1/3], Loss: 0.5017
Epoch [22/30], Step [2/3], Loss: 0.5531
Epoch [22/30], Step [3/3], Loss: 0.3201
Epoch Train Time: 0.55 secs
test_accuracy=  47.82608695652174
```

```
Epoch [23/30], Step [1/3], Loss: 0.4296
Epoch [23/30], Step [2/3], Loss: 0.3453
Epoch [23/30], Step [3/3], Loss: 0.3684
Epoch Train Time: 0.54 secs
test_accuracy=  52.17391304347826
Epoch [24/30], Step [1/3], Loss: 0.3136
Epoch [24/30], Step [2/3], Loss: 0.5150
Epoch [24/30], Step [3/3], Loss: 0.4329
Epoch Train Time: 0.52 secs
test_accuracy=  47.82608695652174
Epoch [25/30], Step [1/3], Loss: 0.2576
Epoch [25/30], Step [2/3], Loss: 0.3871
Epoch [25/30], Step [3/3], Loss: 0.1854
Epoch Train Time: 0.53 secs
test_accuracy=  47.82608695652174
Epoch [26/30], Step [1/3], Loss: 0.3983
Epoch [26/30], Step [2/3], Loss: 0.4433
Epoch [26/30], Step [3/3], Loss: 0.2902
Epoch Train Time: 0.53 secs
test_accuracy=  56.52173913043478
Epoch [27/30], Step [1/3], Loss: 0.1644
Epoch [27/30], Step [2/3], Loss: 0.2858
Epoch [27/30], Step [3/3], Loss: 0.2131
Epoch Train Time: 0.54 secs
test_accuracy=  34.78260869565217
Epoch [28/30], Step [1/3], Loss: 0.3852
Epoch [28/30], Step [2/3], Loss: 0.3702
Epoch [28/30], Step [3/3], Loss: 0.1231
Epoch Train Time: 0.53 secs
test_accuracy=  43.47826086956522
Epoch [29/30], Step [1/3], Loss: 0.2231
Epoch [29/30], Step [2/3], Loss: 0.1855
Epoch [29/30], Step [3/3], Loss: 0.1697
Epoch Train Time: 0.53 secs
test_accuracy=  47.82608695652174
Epoch [30/30], Step [1/3], Loss: 0.2013
Epoch [30/30], Step [2/3], Loss: 0.1835
Epoch [30/30], Step [3/3], Loss: 0.1872
Epoch Train Time: 0.54 secs
test_accuracy=  52.17391304347826
Total Train Time: 0.26 mins

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:26: FutureWarning:
Slicing a positional slice with .loc is not supported, and will raise TypeError
in a future version.  Use .loc with labels or .iloc with positions instead.
```

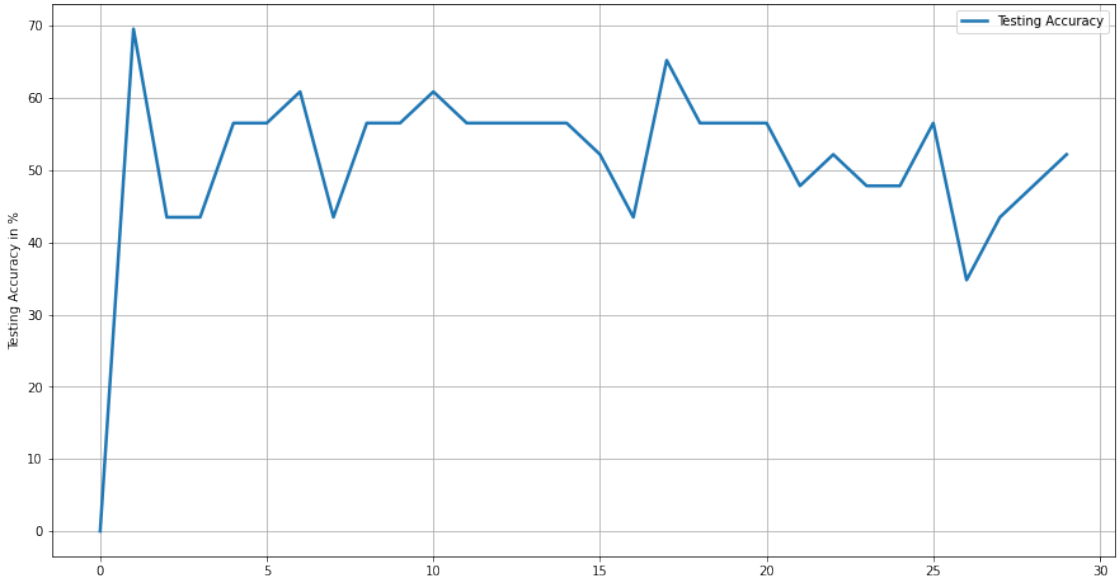
[13]:

	exp_name	Loss	Accuracy
0	loss: CXE, optimizer: SGD, lr: 0.001, epoch: 3...	0.559	56.522

```
2  loss: CXE, optimizer: SGD, lr: 0.01, epoch: 40...  0.124    39.13
3  loss: CXE, optimizer: SGD, lr: 0.01, epoch: 40...  0.054   47.826
4  loss: CXE, optimizer: SGD, lr: 0.01, epoch: 30...  0.191   52.174
```




```
<Figure size 432x288 with 0 Axes>
```



10 Model 6

[14]:
```
##Lets focus on zero loss:
# Download the training and testing data
mini_batch = 30
train_loader = trainload(mini_batch)
test_loader = testload(mini_batch)

# Initialize model
cnn6 = CNN()
if device.type == "cuda":
    cnn6.to('cuda')

# Define loss function
loss_func = torch.nn.CrossEntropyLoss()

# Define optimizer
optimizer = torch.optim.SGD(cnn6.parameters(), lr = 0.01, momentum=0.2)

# Train the model
num_epochs = 30

train_loss, test_accuracy = train(num_epochs, cnn6, train_loader, test_loader,␣
 ↪mini_batch)


#Logging the experiments
exp_name = f"loss: CXE, optimizer: SGD, lr: 0.01, epoch: 30, mini_batch=30,␣
 ↪momentum=0.2"
expLog_R.loc[5,:3] = [f"{exp_name}"] + [f"{np.round(train_loss[-1],3)}"] +␣
 ↪[f"{np.round(test_accuracy[-1],3)}"]
expLog_R
```

```
Epoch [1/30], Step [1/3], Loss: 6.1663
Epoch [1/30], Step [2/3], Loss: 6.1725
Epoch [1/30], Step [3/3], Loss: 6.1681
Epoch Train Time: 0.28 secs
test_accuracy=  0.0
Epoch [2/30], Step [1/3], Loss: 6.1686
Epoch [2/30], Step [2/3], Loss: 5.2125
Epoch [2/30], Step [3/3], Loss: 2.8989
Epoch Train Time: 0.51 secs
test_accuracy=  43.47826086956522
Epoch [3/30], Step [1/3], Loss: 1.0123
Epoch [3/30], Step [2/3], Loss: 9.6271
Epoch [3/30], Step [3/3], Loss: 3.9276
Epoch Train Time: 0.54 secs
```

```
test_accuracy=  43.47826086956522
Epoch [4/30], Step [1/3], Loss: 2.2714
Epoch [4/30], Step [2/3], Loss: 2.0666
Epoch [4/30], Step [3/3], Loss: 1.4776
Epoch Train Time: 0.54 secs
test_accuracy=  65.21739130434783
Epoch [5/30], Step [1/3], Loss: 0.7156
Epoch [5/30], Step [2/3], Loss: 0.7852
Epoch [5/30], Step [3/3], Loss: 0.9503
Epoch Train Time: 0.55 secs
test_accuracy=  43.47826086956522
Epoch [6/30], Step [1/3], Loss: 0.8821
Epoch [6/30], Step [2/3], Loss: 0.7656
Epoch [6/30], Step [3/3], Loss: 0.6774
Epoch Train Time: 0.55 secs
test_accuracy=  56.52173913043478
Epoch [7/30], Step [1/3], Loss: 0.7667
Epoch [7/30], Step [2/3], Loss: 1.0812
Epoch [7/30], Step [3/3], Loss: 1.4066
Epoch Train Time: 0.53 secs
test_accuracy=  43.47826086956522
Epoch [8/30], Step [1/3], Loss: 0.9885
Epoch [8/30], Step [2/3], Loss: 0.7224
Epoch [8/30], Step [3/3], Loss: 0.6796
Epoch Train Time: 0.54 secs
test_accuracy=  56.52173913043478
Epoch [9/30], Step [1/3], Loss: 0.7903
Epoch [9/30], Step [2/3], Loss: 0.7072
Epoch [9/30], Step [3/3], Loss: 0.5915
Epoch Train Time: 0.55 secs
test_accuracy=  60.86956521739131
Epoch [10/30], Step [1/3], Loss: 0.5627
Epoch [10/30], Step [2/3], Loss: 0.8503
Epoch [10/30], Step [3/3], Loss: 1.7288
Epoch Train Time: 0.54 secs
test_accuracy=  43.47826086956522
Epoch [11/30], Step [1/3], Loss: 1.2661
Epoch [11/30], Step [2/3], Loss: 0.7718
Epoch [11/30], Step [3/3], Loss: 0.6545
Epoch Train Time: 0.54 secs
test_accuracy=  43.47826086956522
Epoch [12/30], Step [1/3], Loss: 1.0134
Epoch [12/30], Step [2/3], Loss: 1.0138
Epoch [12/30], Step [3/3], Loss: 0.5061
Epoch Train Time: 0.54 secs
test_accuracy=  43.47826086956522
Epoch [13/30], Step [1/3], Loss: 0.7345
Epoch [13/30], Step [2/3], Loss: 0.5993
```

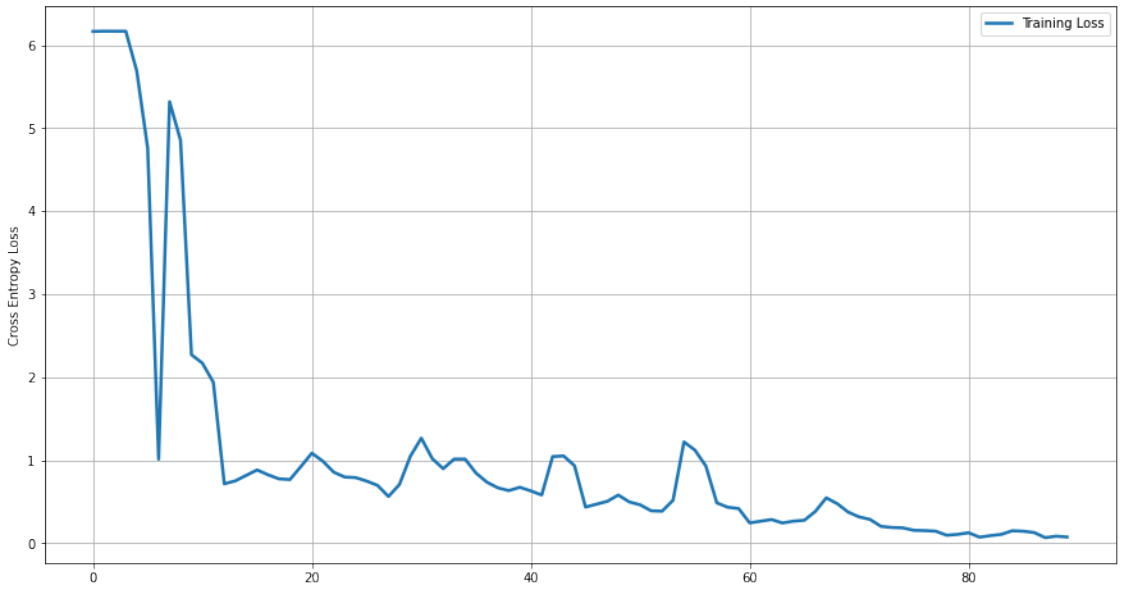
```
Epoch [13/30], Step [3/3], Loss: 0.5670
Epoch Train Time: 0.54 secs
test_accuracy=  56.52173913043478
Epoch [14/30], Step [1/3], Loss: 0.6737
Epoch [14/30], Step [2/3], Loss: 0.5865
Epoch [14/30], Step [3/3], Loss: 0.4820
Epoch Train Time: 0.54 secs
test_accuracy=  56.52173913043478
Epoch [15/30], Step [1/3], Loss: 1.0440
Epoch [15/30], Step [2/3], Loss: 1.0592
Epoch [15/30], Step [3/3], Loss: 0.6968
Epoch Train Time: 0.54 secs
test_accuracy=  52.17391304347826
Epoch [16/30], Step [1/3], Loss: 0.4352
Epoch [16/30], Step [2/3], Loss: 0.5037
Epoch [16/30], Step [3/3], Loss: 0.5753
Epoch Train Time: 0.53 secs
test_accuracy=  43.47826086956522
Epoch [17/30], Step [1/3], Loss: 0.5801
Epoch [17/30], Step [2/3], Loss: 0.4145
Epoch [17/30], Step [3/3], Loss: 0.3910
Epoch Train Time: 0.55 secs
test_accuracy=  56.52173913043478
Epoch [18/30], Step [1/3], Loss: 0.3903
Epoch [18/30], Step [2/3], Loss: 0.3804
Epoch [18/30], Step [3/3], Loss: 0.7807
Epoch Train Time: 0.55 secs
test_accuracy=  56.52173913043478
Epoch [19/30], Step [1/3], Loss: 1.2202
Epoch [19/30], Step [2/3], Loss: 1.0229
Epoch [19/30], Step [3/3], Loss: 0.5450
Epoch Train Time: 0.53 secs
test_accuracy=  56.52173913043478
Epoch [20/30], Step [1/3], Loss: 0.4855
Epoch [20/30], Step [2/3], Loss: 0.3809
Epoch [20/30], Step [3/3], Loss: 0.3843
Epoch Train Time: 0.53 secs
test_accuracy=  47.82608695652174
Epoch [21/30], Step [1/3], Loss: 0.2443
Epoch [21/30], Step [2/3], Loss: 0.2852
Epoch [21/30], Step [3/3], Loss: 0.3246
Epoch Train Time: 0.53 secs
test_accuracy=  43.47826086956522
Epoch [22/30], Step [1/3], Loss: 0.2433
Epoch [22/30], Step [2/3], Loss: 0.2875
Epoch [22/30], Step [3/3], Loss: 0.2981
Epoch Train Time: 0.53 secs
test_accuracy=  43.47826086956522
```

```
Epoch [23/30], Step [1/3], Loss: 0.3822
Epoch [23/30], Step [2/3], Loss: 0.7103
Epoch [23/30], Step [3/3], Loss: 0.3364
Epoch Train Time: 0.53 secs
test_accuracy=  39.130434782608695
Epoch [24/30], Step [1/3], Loss: 0.3744
Epoch [24/30], Step [2/3], Loss: 0.2592
Epoch [24/30], Step [3/3], Loss: 0.2252
Epoch Train Time: 0.54 secs
test_accuracy=  39.130434782608695
Epoch [25/30], Step [1/3], Loss: 0.2023
Epoch [25/30], Step [2/3], Loss: 0.1763
Epoch [25/30], Step [3/3], Loss: 0.1740
Epoch Train Time: 0.53 secs
test_accuracy=  39.130434782608695
Epoch [26/30], Step [1/3], Loss: 0.1552
Epoch [26/30], Step [2/3], Loss: 0.1477
Epoch [26/30], Step [3/3], Loss: 0.1308
Epoch Train Time: 0.53 secs
test_accuracy=  43.47826086956522
Epoch [27/30], Step [1/3], Loss: 0.0957
Epoch [27/30], Step [2/3], Loss: 0.1158
Epoch [27/30], Step [3/3], Loss: 0.1665
Epoch Train Time: 0.53 secs
test_accuracy=  43.47826086956522
Epoch [28/30], Step [1/3], Loss: 0.0724
Epoch [28/30], Step [2/3], Loss: 0.1095
Epoch [28/30], Step [3/3], Loss: 0.1369
Epoch Train Time: 0.55 secs
test_accuracy=  52.17391304347826
Epoch [29/30], Step [1/3], Loss: 0.1497
Epoch [29/30], Step [2/3], Loss: 0.1382
Epoch [29/30], Step [3/3], Loss: 0.0962
Epoch Train Time: 0.54 secs
test_accuracy=  47.82608695652174
Epoch [30/30], Step [1/3], Loss: 0.0664
Epoch [30/30], Step [2/3], Loss: 0.1017
Epoch [30/30], Step [3/3], Loss: 0.0552
Epoch Train Time: 0.55 secs
test_accuracy=  47.82608695652174
Total Train Time: 0.26 mins

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:26: FutureWarning:
Slicing a positional slice with .loc is not supported, and will raise TypeError
in a future version.  Use .loc with labels or .iloc with positions instead.
```

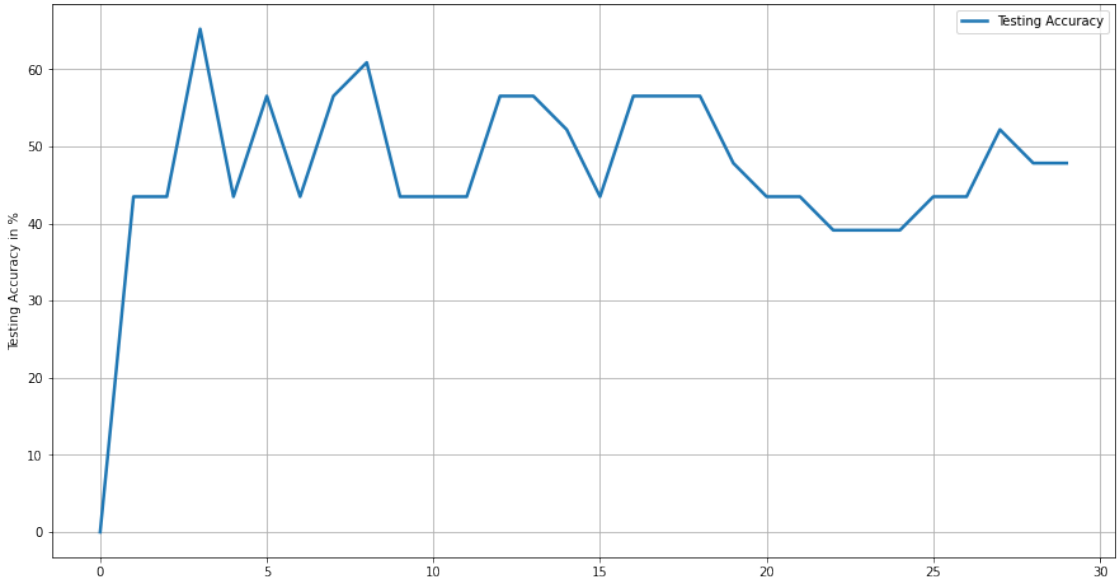
[14]:

	exp_name	Loss	Accuracy
0	loss: CXE, optimizer: SGD, lr: 0.001, epoch: 3...	0.559	56.522

```
2  loss: CXE, optimizer: SGD, lr: 0.01, epoch: 40...  0.124    39.13
3  loss: CXE, optimizer: SGD, lr: 0.01, epoch: 40...  0.054   47.826
4  loss: CXE, optimizer: SGD, lr: 0.01, epoch: 30...  0.191   52.174
5  loss: CXE, optimizer: SGD, lr: 0.01, epoch: 30...  0.074   47.826
```




```
<Figure size 432x288 with 0 Axes>
```



11 Model 7

[15]:
```
# Download the training and testing data
mini_batch = 30
train_loader = trainload(mini_batch)
test_loader = testload(mini_batch)

# Initialize model
cnn7 = CNN()
if device.type == "cuda":
    cnn7.to('cuda')

# Define loss function
loss_func = torch.nn.CrossEntropyLoss()

# Define optimizer
optimizer = torch.optim.SGD(cnn7.parameters(), lr = 0.01, momentum=0.02)

# Train the model
num_epochs = 30
train_loss, test_accuracy = train(num_epochs, cnn7, train_loader, test_loader,␣
 ↪mini_batch)

#Logging the experiments
exp_name = f"loss: CXE, optimizer: SGD, lr: 0.01, epoch: 30, mini_batch=30,␣
 ↪momentum=0.02"
expLog_R.loc[6,:3] = [f"{exp_name}"] + [f"{np.round(train_loss[-1],3)}"] +␣
 ↪[f"{np.round(test_accuracy[-1],3)}"]
expLog_R
```

```
Epoch [1/30], Step [1/3], Loss: 6.2097
Epoch [1/30], Step [2/3], Loss: 6.2051
Epoch [1/30], Step [3/3], Loss: 6.2008
Epoch Train Time: 0.27 secs
test_accuracy=  0.0
Epoch [2/30], Step [1/3], Loss: 6.2108
Epoch [2/30], Step [2/3], Loss: 4.6824
Epoch [2/30], Step [3/3], Loss: 2.0032
Epoch Train Time: 0.51 secs
test_accuracy=  43.47826086956522
Epoch [3/30], Step [1/3], Loss: 2.3284
Epoch [3/30], Step [2/3], Loss: 3.3959
Epoch [3/30], Step [3/3], Loss: 1.9539
Epoch Train Time: 0.55 secs
test_accuracy=  43.47826086956522
Epoch [4/30], Step [1/3], Loss: 0.7851
Epoch [4/30], Step [2/3], Loss: 4.7394
```

```
Epoch [4/30], Step [3/3], Loss: 3.3252
Epoch Train Time: 0.53 secs
test_accuracy=  43.47826086956522
Epoch [5/30], Step [1/3], Loss: 3.0373
Epoch [5/30], Step [2/3], Loss: 3.1235
Epoch [5/30], Step [3/3], Loss: 1.8485
Epoch Train Time: 0.54 secs
test_accuracy=  56.52173913043478
Epoch [6/30], Step [1/3], Loss: 0.8905
Epoch [6/30], Step [2/3], Loss: 0.8328
Epoch [6/30], Step [3/3], Loss: 0.8364
Epoch Train Time: 0.53 secs
test_accuracy=  43.47826086956522
Epoch [7/30], Step [1/3], Loss: 0.6798
Epoch [7/30], Step [2/3], Loss: 0.6544
Epoch [7/30], Step [3/3], Loss: 0.6289
Epoch Train Time: 0.53 secs
test_accuracy=  39.130434782608695
Epoch [8/30], Step [1/3], Loss: 0.6707
Epoch [8/30], Step [2/3], Loss: 1.1982
Epoch [8/30], Step [3/3], Loss: 1.6650
Epoch Train Time: 0.53 secs
test_accuracy=  56.52173913043478
Epoch [9/30], Step [1/3], Loss: 0.9136
Epoch [9/30], Step [2/3], Loss: 0.6404
Epoch [9/30], Step [3/3], Loss: 0.8159
Epoch Train Time: 0.72 secs
test_accuracy=  56.52173913043478
Epoch [10/30], Step [1/3], Loss: 0.8582
Epoch [10/30], Step [2/3], Loss: 0.8394
Epoch [10/30], Step [3/3], Loss: 0.6991
Epoch Train Time: 0.53 secs
test_accuracy=  43.47826086956522
Epoch [11/30], Step [1/3], Loss: 0.6891
Epoch [11/30], Step [2/3], Loss: 0.6466
Epoch [11/30], Step [3/3], Loss: 0.8034
Epoch Train Time: 0.54 secs
test_accuracy=  56.52173913043478
Epoch [12/30], Step [1/3], Loss: 1.1369
Epoch [12/30], Step [2/3], Loss: 1.4656
Epoch [12/30], Step [3/3], Loss: 0.9115
Epoch Train Time: 0.54 secs
test_accuracy=  56.52173913043478
Epoch [13/30], Step [1/3], Loss: 0.7602
Epoch [13/30], Step [2/3], Loss: 0.8107
Epoch [13/30], Step [3/3], Loss: 0.4965
Epoch Train Time: 0.55 secs
test_accuracy=  56.52173913043478
```

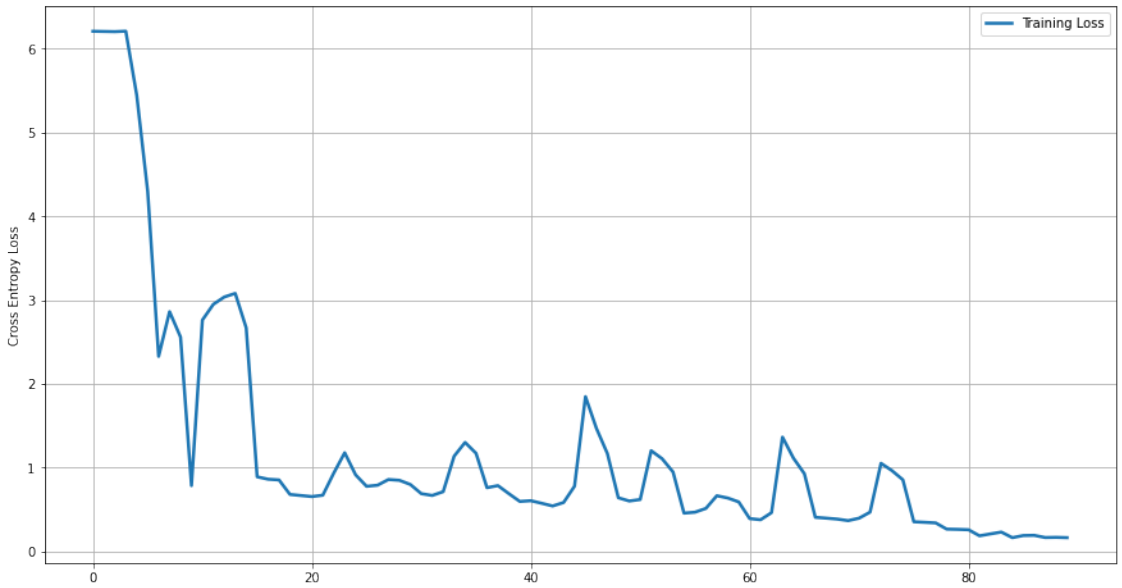
```
Epoch [14/30], Step [1/3], Loss: 0.5962
Epoch [14/30], Step [2/3], Loss: 0.6138
Epoch [14/30], Step [3/3], Loss: 0.5131
Epoch Train Time: 0.54 secs
test_accuracy=  65.21739130434783
Epoch [15/30], Step [1/3], Loss: 0.5419
Epoch [15/30], Step [2/3], Loss: 0.6250
Epoch [15/30], Step [3/3], Loss: 1.1683
Epoch Train Time: 0.55 secs
test_accuracy=  56.52173913043478
Epoch [16/30], Step [1/3], Loss: 1.8478
Epoch [16/30], Step [2/3], Loss: 1.0895
Epoch [16/30], Step [3/3], Loss: 0.5676
Epoch Train Time: 0.56 secs
test_accuracy=  56.52173913043478
Epoch [17/30], Step [1/3], Loss: 0.6405
Epoch [17/30], Step [2/3], Loss: 0.5624
Epoch [17/30], Step [3/3], Loss: 0.6550
Epoch Train Time: 0.56 secs
test_accuracy=  43.47826086956522
Epoch [18/30], Step [1/3], Loss: 1.2024
Epoch [18/30], Step [2/3], Loss: 1.0110
Epoch [18/30], Step [3/3], Loss: 0.6310
Epoch Train Time: 0.54 secs
test_accuracy=  60.86956521739131
Epoch [19/30], Step [1/3], Loss: 0.4569
Epoch [19/30], Step [2/3], Loss: 0.4784
Epoch [19/30], Step [3/3], Loss: 0.6036
Epoch Train Time: 0.55 secs
test_accuracy=  56.52173913043478
Epoch [20/30], Step [1/3], Loss: 0.6644
Epoch [20/30], Step [2/3], Loss: 0.6100
Epoch [20/30], Step [3/3], Loss: 0.4981
Epoch Train Time: 0.54 secs
test_accuracy=  43.47826086956522
Epoch [21/30], Step [1/3], Loss: 0.3916
Epoch [21/30], Step [2/3], Loss: 0.3637
Epoch [21/30], Step [3/3], Loss: 0.6329
Epoch Train Time: 0.56 secs
test_accuracy=  43.47826086956522
Epoch [22/30], Step [1/3], Loss: 1.3645
Epoch [22/30], Step [2/3], Loss: 0.8623
Epoch [22/30], Step [3/3], Loss: 0.5554
Epoch Train Time: 0.55 secs
test_accuracy=  56.52173913043478
Epoch [23/30], Step [1/3], Loss: 0.4066
Epoch [23/30], Step [2/3], Loss: 0.3870
Epoch [23/30], Step [3/3], Loss: 0.3634
```

```
Epoch Train Time: 0.55 secs
test_accuracy=  60.86956521739131
Epoch [24/30], Step [1/3], Loss: 0.3671
Epoch [24/30], Step [2/3], Loss: 0.4258
Epoch [24/30], Step [3/3], Loss: 0.6120
Epoch Train Time: 0.55 secs
test_accuracy=  43.47826086956522
Epoch [25/30], Step [1/3], Loss: 1.0518
Epoch [25/30], Step [2/3], Loss: 0.8773
Epoch [25/30], Step [3/3], Loss: 0.6356
Epoch Train Time: 0.57 secs
test_accuracy=  52.17391304347826
Epoch [26/30], Step [1/3], Loss: 0.3536
Epoch [26/30], Step [2/3], Loss: 0.3406
Epoch [26/30], Step [3/3], Loss: 0.3256
Epoch Train Time: 0.55 secs
test_accuracy=  52.17391304347826
Epoch [27/30], Step [1/3], Loss: 0.2660
Epoch [27/30], Step [2/3], Loss: 0.2600
Epoch [27/30], Step [3/3], Loss: 0.2492
Epoch Train Time: 0.55 secs
test_accuracy=  56.52173913043478
Epoch [28/30], Step [1/3], Loss: 0.1862
Epoch [28/30], Step [2/3], Loss: 0.2317
Epoch [28/30], Step [3/3], Loss: 0.2739
Epoch Train Time: 0.55 secs
test_accuracy=  43.47826086956522
Epoch [29/30], Step [1/3], Loss: 0.1638
Epoch [29/30], Step [2/3], Loss: 0.2148
Epoch [29/30], Step [3/3], Loss: 0.1959
Epoch Train Time: 0.55 secs
test_accuracy=  52.17391304347826
Epoch [30/30], Step [1/3], Loss: 0.1655
Epoch [30/30], Step [2/3], Loss: 0.1697
Epoch [30/30], Step [3/3], Loss: 0.1567
Epoch Train Time: 0.54 secs
test_accuracy=  34.78260869565217
Total Train Time: 0.27 mins

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:23: FutureWarning:
Slicing a positional slice with .loc is not supported, and will raise TypeError
in a future version.  Use .loc with labels or .iloc with positions instead.
```

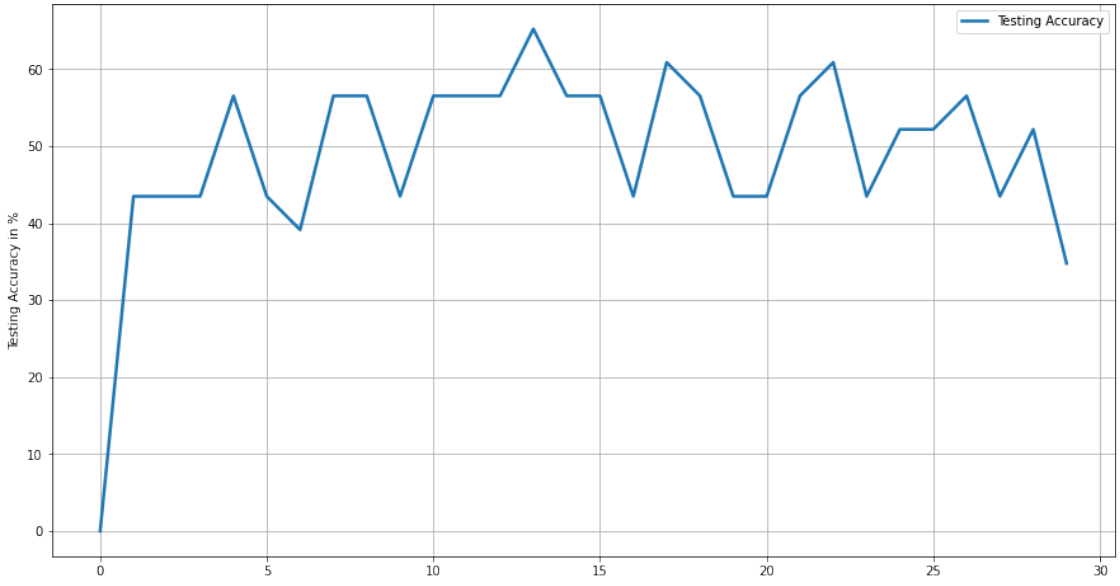
[15]:

	exp_name	Loss	Accuracy
0	loss: CXE, optimizer: SGD, lr: 0.001, epoch: 3...	0.559	56.522
2	loss: CXE, optimizer: SGD, lr: 0.01, epoch: 40...	0.124	39.13
3	loss: CXE, optimizer: SGD, lr: 0.01, epoch: 40...	0.054	47.826
4	loss: CXE, optimizer: SGD, lr: 0.01, epoch: 30...	0.191	52.174

```
5  loss: CXE, optimizer: SGD, lr: 0.01, epoch: 30...  0.074   47.826
6  loss: CXE, optimizer: SGD, lr: 0.01, epoch: 30...  0.164   34.783
```




```
<Figure size 432x288 with 0 Axes>
```



12 Model 8

[16]:
```
# Download the training and testing data
mini_batch = 50
train_loader = trainload(mini_batch)
test_loader = testload(mini_batch)

# Initialize model
cnn8 = CNN()
if device.type == "cuda":
    cnn8.to('cuda')

# Define loss function
loss_func = torch.nn.CrossEntropyLoss()

# Define optimizer
optimizer = torch.optim.SGD(cnn8.parameters(), lr = 0.01, momentum=0.02)

# Train the model
num_epochs = 40

train_loss, test_accuracy = train(num_epochs, cnn8, train_loader, test_loader,␣
 ↪mini_batch)

#Logging the experiments
exp_name = f"loss: CXE, optimizer: SGD, lr: 0.01, epoch: 40, mini_batch=50,␣
 ↪momentum=0.02"
expLog_R.loc[7,:3] = [f"{exp_name}"] + [f"{np.round(train_loss[-1],3)}"] +␣
 ↪[f"{np.round(test_accuracy[-1],3)}"]
expLog_R
```

```
Epoch [1/40], Step [1/2], Loss: 6.2153
Epoch [1/40], Step [2/2], Loss: 6.2121
Epoch Train Time: 0.28 secs
test_accuracy=  0.0
Epoch [2/40], Step [1/2], Loss: 6.2220
Epoch [2/40], Step [2/2], Loss: 5.2762
Epoch Train Time: 0.49 secs
test_accuracy=  43.47826086956522
Epoch [3/40], Step [1/2], Loss: 3.4066
Epoch [3/40], Step [2/2], Loss: 1.0019
Epoch Train Time: 0.50 secs
test_accuracy=  56.52173913043478
Epoch [4/40], Step [1/2], Loss: 2.1721
Epoch [4/40], Step [2/2], Loss: 6.1310
Epoch Train Time: 0.48 secs
test_accuracy=  43.47826086956522
```

```
Epoch [5/40], Step [1/2], Loss: 4.1023
Epoch [5/40], Step [2/2], Loss: 2.1064
Epoch Train Time: 0.45 secs
test_accuracy=  56.52173913043478
Epoch [6/40], Step [1/2], Loss: 0.8552
Epoch [6/40], Step [2/2], Loss: 0.7042
Epoch Train Time: 0.46 secs
test_accuracy=  43.47826086956522
Epoch [7/40], Step [1/2], Loss: 1.7307
Epoch [7/40], Step [2/2], Loss: 2.5275
Epoch Train Time: 0.43 secs
test_accuracy=  43.47826086956522
Epoch [8/40], Step [1/2], Loss: 1.5097
Epoch [8/40], Step [2/2], Loss: 0.8551
Epoch Train Time: 0.43 secs
test_accuracy=  56.52173913043478
Epoch [9/40], Step [1/2], Loss: 2.6282
Epoch [9/40], Step [2/2], Loss: 1.5915
Epoch Train Time: 0.45 secs
test_accuracy=  43.47826086956522
Epoch [10/40], Step [1/2], Loss: 0.8792
Epoch [10/40], Step [2/2], Loss: 0.7393
Epoch Train Time: 0.44 secs
test_accuracy=  43.47826086956522
Epoch [11/40], Step [1/2], Loss: 0.9930
Epoch [11/40], Step [2/2], Loss: 0.8073
Epoch Train Time: 0.45 secs
test_accuracy=  43.47826086956522
Epoch [12/40], Step [1/2], Loss: 0.7644
Epoch [12/40], Step [2/2], Loss: 1.0126
Epoch Train Time: 0.43 secs
test_accuracy=  43.47826086956522
Epoch [13/40], Step [1/2], Loss: 1.1721
Epoch [13/40], Step [2/2], Loss: 0.9123
Epoch Train Time: 0.45 secs
test_accuracy=  39.130434782608695
Epoch [14/40], Step [1/2], Loss: 0.6767
Epoch [14/40], Step [2/2], Loss: 0.6941
Epoch Train Time: 0.45 secs
test_accuracy=  43.47826086956522
Epoch [15/40], Step [1/2], Loss: 0.7307
Epoch [15/40], Step [2/2], Loss: 1.1159
Epoch Train Time: 0.45 secs
test_accuracy=  43.47826086956522
Epoch [16/40], Step [1/2], Loss: 1.1338
Epoch [16/40], Step [2/2], Loss: 0.7208
Epoch Train Time: 0.44 secs
test_accuracy=  56.52173913043478
```

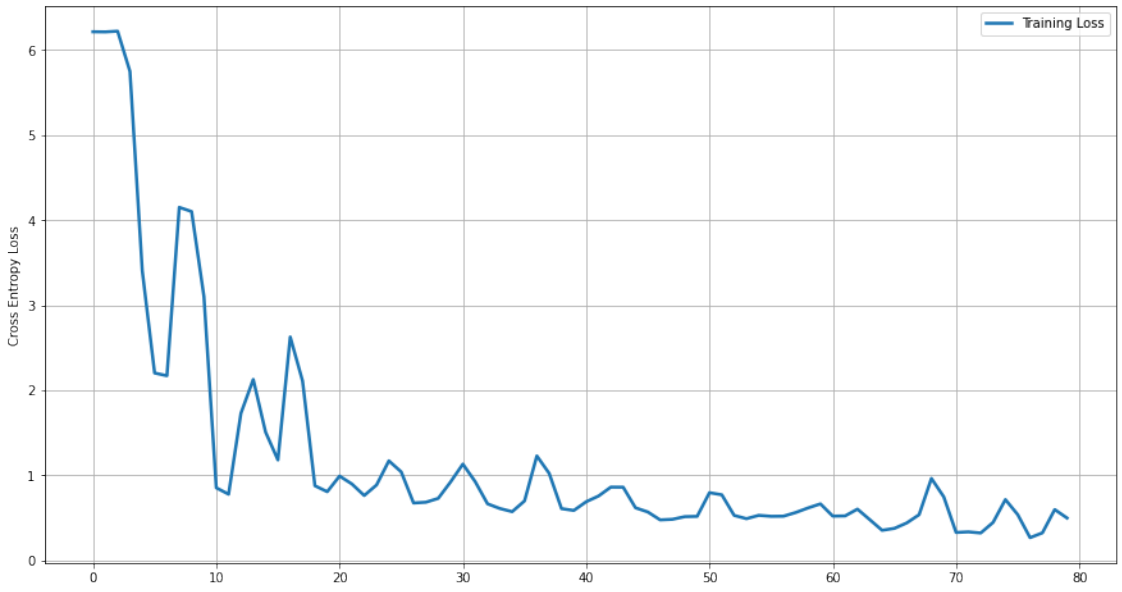
```
Epoch [17/40], Step [1/2], Loss: 0.6668
Epoch [17/40], Step [2/2], Loss: 0.5566
Epoch Train Time: 0.44 secs
test_accuracy=  56.52173913043478
Epoch [18/40], Step [1/2], Loss: 0.5746
Epoch [18/40], Step [2/2], Loss: 0.8265
Epoch Train Time: 0.44 secs
test_accuracy=  56.52173913043478
Epoch [19/40], Step [1/2], Loss: 1.2298
Epoch [19/40], Step [2/2], Loss: 0.8195
Epoch Train Time: 0.44 secs
test_accuracy=  56.52173913043478
Epoch [20/40], Step [1/2], Loss: 0.6101
Epoch [20/40], Step [2/2], Loss: 0.5677
Epoch Train Time: 0.44 secs
test_accuracy=  43.47826086956522
Epoch [21/40], Step [1/2], Loss: 0.6912
Epoch [21/40], Step [2/2], Loss: 0.8251
Epoch Train Time: 0.43 secs
test_accuracy=  43.47826086956522
Epoch [22/40], Step [1/2], Loss: 0.8631
Epoch [22/40], Step [2/2], Loss: 0.8614
Epoch Train Time: 0.43 secs
test_accuracy=  39.130434782608695
Epoch [23/40], Step [1/2], Loss: 0.6200
Epoch [23/40], Step [2/2], Loss: 0.5216
Epoch Train Time: 0.45 secs
test_accuracy=  52.17391304347826
Epoch [24/40], Step [1/2], Loss: 0.4778
Epoch [24/40], Step [2/2], Loss: 0.4917
Epoch Train Time: 0.43 secs
test_accuracy=  60.86956521739131
Epoch [25/40], Step [1/2], Loss: 0.5159
Epoch [25/40], Step [2/2], Loss: 0.5228
Epoch Train Time: 0.45 secs
test_accuracy=  56.52173913043478
Epoch [26/40], Step [1/2], Loss: 0.7978
Epoch [26/40], Step [2/2], Loss: 0.7481
Epoch Train Time: 0.44 secs
test_accuracy=  56.52173913043478
Epoch [27/40], Step [1/2], Loss: 0.5307
Epoch [27/40], Step [2/2], Loss: 0.4534
Epoch Train Time: 0.44 secs
test_accuracy=  39.130434782608695
Epoch [28/40], Step [1/2], Loss: 0.5315
Epoch [28/40], Step [2/2], Loss: 0.5084
Epoch Train Time: 0.44 secs
test_accuracy=  39.130434782608695
```

```
Epoch [29/40], Step [1/2], Loss: 0.5213
Epoch [29/40], Step [2/2], Loss: 0.6087
Epoch Train Time: 0.45 secs
test_accuracy=  43.47826086956522
Epoch [30/40], Step [1/2], Loss: 0.6186
Epoch [30/40], Step [2/2], Loss: 0.7139
Epoch Train Time: 0.44 secs
test_accuracy=  43.47826086956522
Epoch [31/40], Step [1/2], Loss: 0.5222
Epoch [31/40], Step [2/2], Loss: 0.5272
Epoch Train Time: 0.43 secs
test_accuracy=  43.47826086956522
Epoch [32/40], Step [1/2], Loss: 0.6046
Epoch [32/40], Step [2/2], Loss: 0.3566
Epoch Train Time: 0.44 secs
test_accuracy=  60.86956521739131
Epoch [33/40], Step [1/2], Loss: 0.3555
Epoch [33/40], Step [2/2], Loss: 0.4009
Epoch Train Time: 0.43 secs
test_accuracy=  56.52173913043478
Epoch [34/40], Step [1/2], Loss: 0.4421
Epoch [34/40], Step [2/2], Loss: 0.6352
Epoch Train Time: 0.43 secs
test_accuracy=  56.52173913043478
Epoch [35/40], Step [1/2], Loss: 0.9653
Epoch [35/40], Step [2/2], Loss: 0.5332
Epoch Train Time: 0.44 secs
test_accuracy=  52.17391304347826
Epoch [36/40], Step [1/2], Loss: 0.3303
Epoch [36/40], Step [2/2], Loss: 0.3438
Epoch Train Time: 0.43 secs
test_accuracy=  47.82608695652174
Epoch [37/40], Step [1/2], Loss: 0.3239
Epoch [37/40], Step [2/2], Loss: 0.5724
Epoch Train Time: 0.44 secs
test_accuracy=  43.47826086956522
Epoch [38/40], Step [1/2], Loss: 0.7168
Epoch [38/40], Step [2/2], Loss: 0.3625
Epoch Train Time: 0.43 secs
test_accuracy=  47.82608695652174
Epoch [39/40], Step [1/2], Loss: 0.2689
Epoch [39/40], Step [2/2], Loss: 0.3819
Epoch Train Time: 0.46 secs
test_accuracy=  56.52173913043478
Epoch [40/40], Step [1/2], Loss: 0.5995
Epoch [40/40], Step [2/2], Loss: 0.4003
Epoch Train Time: 0.44 secs
test_accuracy=  56.52173913043478
```

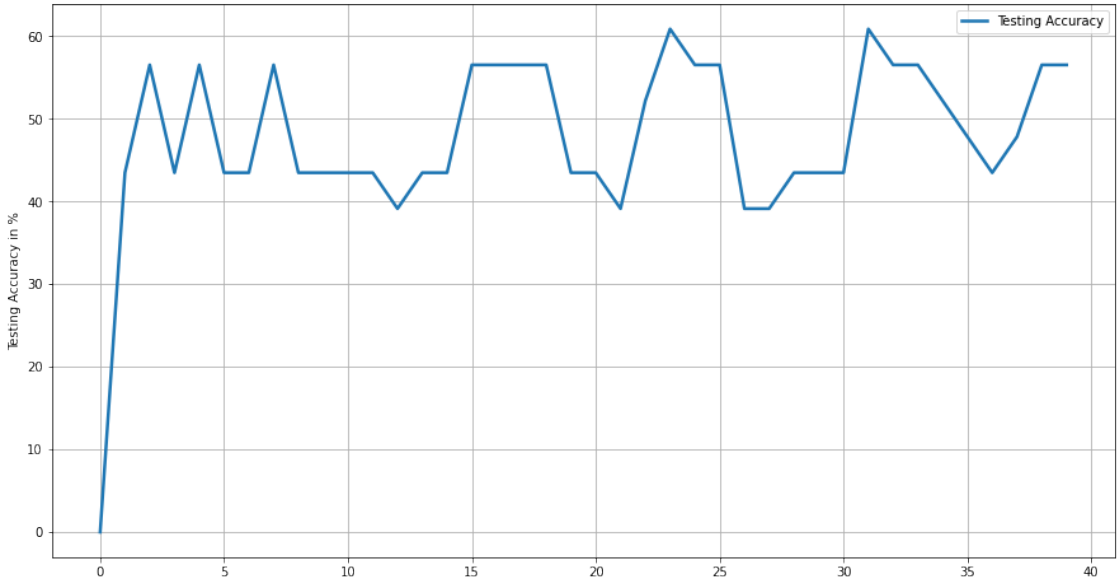
```
Total Train Time: 0.29 mins

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:24: FutureWarning:
Slicing a positional slice with .loc is not supported, and will raise TypeError
in a future version.  Use .loc with labels or .iloc with positions instead.
```

[16]:

	exp_name	Loss	Accuracy
0	loss: CXE, optimizer: SGD, lr: 0.001, epoch: 3...	0.559	56.522
2	loss: CXE, optimizer: SGD, lr: 0.01, epoch: 40...	0.124	39.13
3	loss: CXE, optimizer: SGD, lr: 0.01, epoch: 40...	0.054	47.826
4	loss: CXE, optimizer: SGD, lr: 0.01, epoch: 30...	0.191	52.174
5	loss: CXE, optimizer: SGD, lr: 0.01, epoch: 30...	0.074	47.826
6	loss: CXE, optimizer: SGD, lr: 0.01, epoch: 30...	0.164	34.783
7	loss: CXE, optimizer: SGD, lr: 0.01, epoch: 40...	0.5	56.522




```
<Figure size 432x288 with 0 Axes>
```



13 Model 9

[9]:
```
# Download the training and testing data
mini_batch = 30
train_loader = trainload(mini_batch)
test_loader = testload(mini_batch)

# Initialize model
cnn9 = CNN()
if device.type == "cuda":
    cnn9.to('cuda')

# Define loss function
loss_func = torch.nn.CrossEntropyLoss()

# Define optimizer
optimizer = torch.optim.Adam(cnn9.parameters(), lr = 0.01)


# Train the model
num_epochs = 30

train_loss, test_accuracy = train(num_epochs, cnn9, train_loader, test_loader,␣
 ↪mini_batch)

#Logging the experiments
exp_name = f"loss: CXE, optimizer: Adam, lr: 0.01, epoch: 30, mini_batch=30"
```

```
expLog_R.loc[8,:3] = [f"{exp_name}"] + [f"{np.round(train_loss[-1],3)}"] +␣
↪[f"{np.round(test_accuracy[-1],3)}"]
expLog_R
```

```
Epoch [1/30], Step [1/3], Loss: 6.1492
Epoch [1/30], Step [2/3], Loss: 6.1591
Epoch [1/30], Step [3/3], Loss: 6.1707
Epoch Train Time: 11.83 secs
test_accuracy=  4.3478260869565215
Epoch [2/30], Step [1/3], Loss: 6.1608
Epoch [2/30], Step [2/3], Loss: 722.0380
Epoch [2/30], Step [3/3], Loss: 1644.6072
Epoch Train Time: 0.77 secs
test_accuracy=  56.52173913043478
Epoch [3/30], Step [1/3], Loss: 13.8190
Epoch [3/30], Step [2/3], Loss: 5.4250
Epoch [3/30], Step [3/3], Loss: 5.1476
Epoch Train Time: 0.69 secs
test_accuracy=  56.52173913043478
Epoch [4/30], Step [1/3], Loss: 4.7080
Epoch [4/30], Step [2/3], Loss: 3.7438
Epoch [4/30], Step [3/3], Loss: 2.2118
Epoch Train Time: 0.69 secs
test_accuracy=  56.52173913043478
Epoch [5/30], Step [1/3], Loss: 1.9601
Epoch [5/30], Step [2/3], Loss: 2.4570
Epoch [5/30], Step [3/3], Loss: 1.5330
Epoch Train Time: 0.69 secs
test_accuracy=  56.52173913043478
Epoch [6/30], Step [1/3], Loss: 2.3389
Epoch [6/30], Step [2/3], Loss: 2.7419
Epoch [6/30], Step [3/3], Loss: 2.2501
Epoch Train Time: 0.69 secs
test_accuracy=  56.52173913043478
Epoch [7/30], Step [1/3], Loss: 3.0170
Epoch [7/30], Step [2/3], Loss: 2.1588
Epoch [7/30], Step [3/3], Loss: 0.8180
Epoch Train Time: 0.69 secs
test_accuracy=  56.52173913043478
Epoch [8/30], Step [1/3], Loss: 1.2674
Epoch [8/30], Step [2/3], Loss: 1.1020
Epoch [8/30], Step [3/3], Loss: 0.7463
Epoch Train Time: 0.68 secs
test_accuracy=  43.47826086956522
Epoch [9/30], Step [1/3], Loss: 0.7364
Epoch [9/30], Step [2/3], Loss: 0.8680
Epoch [9/30], Step [3/3], Loss: 1.2989
```

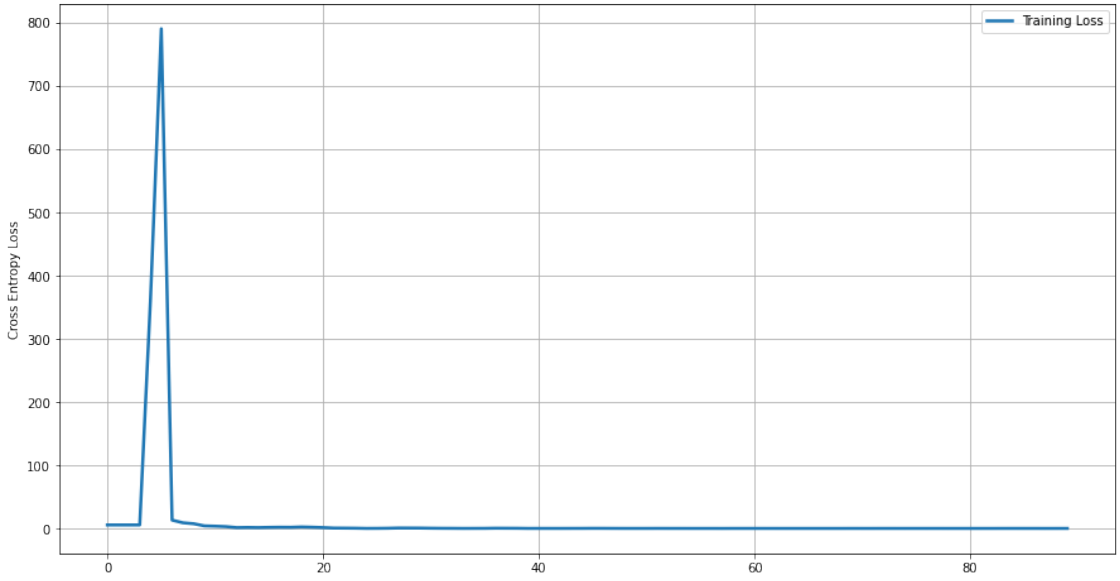
```
Epoch Train Time: 0.69 secs
test_accuracy=  43.47826086956522
Epoch [10/30], Step [1/3], Loss: 1.3569
Epoch [10/30], Step [2/3], Loss: 1.2558
Epoch [10/30], Step [3/3], Loss: 1.1495
Epoch Train Time: 0.69 secs
test_accuracy=  43.47826086956522
Epoch [11/30], Step [1/3], Loss: 1.0225
Epoch [11/30], Step [2/3], Loss: 0.7537
Epoch [11/30], Step [3/3], Loss: 0.7167
Epoch Train Time: 0.69 secs
test_accuracy=  56.52173913043478
Epoch [12/30], Step [1/3], Loss: 0.7229
Epoch [12/30], Step [2/3], Loss: 0.7977
Epoch [12/30], Step [3/3], Loss: 0.9295
Epoch Train Time: 0.70 secs
test_accuracy=  56.52173913043478
Epoch [13/30], Step [1/3], Loss: 1.0309
Epoch [13/30], Step [2/3], Loss: 0.9072
Epoch [13/30], Step [3/3], Loss: 0.8050
Epoch Train Time: 0.68 secs
test_accuracy=  56.52173913043478
Epoch [14/30], Step [1/3], Loss: 0.7101
Epoch [14/30], Step [2/3], Loss: 0.7507
Epoch [14/30], Step [3/3], Loss: 0.7022
Epoch Train Time: 0.68 secs
test_accuracy=  43.47826086956522
Epoch [15/30], Step [1/3], Loss: 0.7168
Epoch [15/30], Step [2/3], Loss: 0.7914
Epoch [15/30], Step [3/3], Loss: 0.7913
Epoch Train Time: 0.69 secs
test_accuracy=  43.47826086956522
Epoch [16/30], Step [1/3], Loss: 0.8507
Epoch [16/30], Step [2/3], Loss: 0.8009
Epoch [16/30], Step [3/3], Loss: 0.6230
Epoch Train Time: 0.69 secs
test_accuracy=  43.47826086956522
Epoch [17/30], Step [1/3], Loss: 0.7238
Epoch [17/30], Step [2/3], Loss: 0.7031
Epoch [17/30], Step [3/3], Loss: 0.7310
Epoch Train Time: 0.69 secs
test_accuracy=  56.52173913043478
Epoch [18/30], Step [1/3], Loss: 0.7623
Epoch [18/30], Step [2/3], Loss: 0.6844
Epoch [18/30], Step [3/3], Loss: 0.6928
Epoch Train Time: 0.68 secs
test_accuracy=  56.52173913043478
Epoch [19/30], Step [1/3], Loss: 0.6954
```

```
Epoch [19/30], Step [2/3], Loss: 0.6961
Epoch [19/30], Step [3/3], Loss: 0.7084
Epoch Train Time: 0.68 secs
test_accuracy=  43.47826086956522
Epoch [20/30], Step [1/3], Loss: 0.6637
Epoch [20/30], Step [2/3], Loss: 0.7560
Epoch [20/30], Step [3/3], Loss: 0.6966
Epoch Train Time: 0.68 secs
test_accuracy=  43.47826086956522
Epoch [21/30], Step [1/3], Loss: 0.7099
Epoch [21/30], Step [2/3], Loss: 0.6951
Epoch [21/30], Step [3/3], Loss: 0.6958
Epoch Train Time: 0.69 secs
test_accuracy=  56.52173913043478
Epoch [22/30], Step [1/3], Loss: 0.6911
Epoch [22/30], Step [2/3], Loss: 0.7003
Epoch [22/30], Step [3/3], Loss: 0.7007
Epoch Train Time: 0.69 secs
test_accuracy=  43.47826086956522
Epoch [23/30], Step [1/3], Loss: 0.6944
Epoch [23/30], Step [2/3], Loss: 0.6918
Epoch [23/30], Step [3/3], Loss: 0.6804
Epoch Train Time: 0.69 secs
test_accuracy=  43.47826086956522
Epoch [24/30], Step [1/3], Loss: 0.7153
Epoch [24/30], Step [2/3], Loss: 0.6757
Epoch [24/30], Step [3/3], Loss: 0.6960
Epoch Train Time: 0.69 secs
test_accuracy=  43.47826086956522
Epoch [25/30], Step [1/3], Loss: 0.7165
Epoch [25/30], Step [2/3], Loss: 0.6923
Epoch [25/30], Step [3/3], Loss: 0.6832
Epoch Train Time: 0.68 secs
test_accuracy=  43.47826086956522
Epoch [26/30], Step [1/3], Loss: 0.7043
Epoch [26/30], Step [2/3], Loss: 0.6962
Epoch [26/30], Step [3/3], Loss: 0.7102
Epoch Train Time: 0.70 secs
test_accuracy=  56.52173913043478
Epoch [27/30], Step [1/3], Loss: 0.6933
Epoch [27/30], Step [2/3], Loss: 0.6956
Epoch [27/30], Step [3/3], Loss: 0.6808
Epoch Train Time: 0.69 secs
test_accuracy=  43.47826086956522
Epoch [28/30], Step [1/3], Loss: 0.6845
Epoch [28/30], Step [2/3], Loss: 0.6971
Epoch [28/30], Step [3/3], Loss: 0.7537
Epoch Train Time: 0.69 secs
```

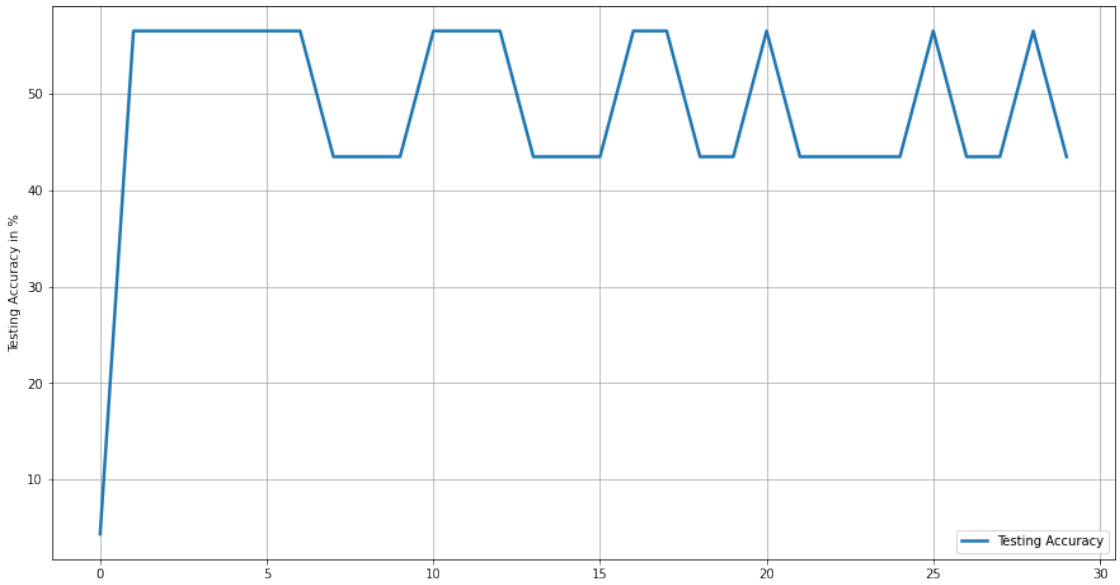
```
test_accuracy=  43.47826086956522
Epoch [29/30], Step [1/3], Loss: 0.7281
Epoch [29/30], Step [2/3], Loss: 0.6946
Epoch [29/30], Step [3/3], Loss: 0.7179
Epoch Train Time: 0.69 secs
test_accuracy=  56.52173913043478
Epoch [30/30], Step [1/3], Loss: 0.6836
Epoch [30/30], Step [2/3], Loss: 0.7083
Epoch [30/30], Step [3/3], Loss: 0.7148
Epoch Train Time: 0.68 secs
test_accuracy=  43.47826086956522
Total Train Time: 0.53 mins

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:25: FutureWarning:
Slicing a positional slice with .loc is not supported, and will raise TypeError
in a future version.  Use .loc with labels or .iloc with positions instead.
```

[9]:

	exp_name	Loss	Accuracy
8	loss: CXE, optimizer: Adam, lr: 0.01, epoch: 3...	0.702	43.478




```
<Figure size 432x288 with 0 Axes>
```



14 Model 10

[10]:
```
# Download the training and testing data
mini_batch = 50
train_loader = trainload(mini_batch)
test_loader = testload(mini_batch)

# Initialize model
cnn10 = CNN()
if device.type == "cuda":
    cnn10.to('cuda')

# Define loss function
loss_func = torch.nn.CrossEntropyLoss()

# Define optimizer
optimizer = torch.optim.Adam(cnn10.parameters(), lr = 0.001, eps=1e-08)

# Train the model
num_epochs = 30

train_loss, test_accuracy = train(num_epochs, cnn10, train_loader, test_loader,␣
 ↪mini_batch)

#Logging the experiments
exp_name = f"loss: CXE, opt: Adam, lr: 0.001, eps:1e-08 epoch: 30,␣
 ↪mini_batch=50"
```

```
expLog_R.loc[9,:3] = [f"{exp_name}"] + [f"{np.round(train_loss[-1],3)}"] +␣
↪[f"{np.round(test_accuracy[-1],3)}"]
expLog_R
```

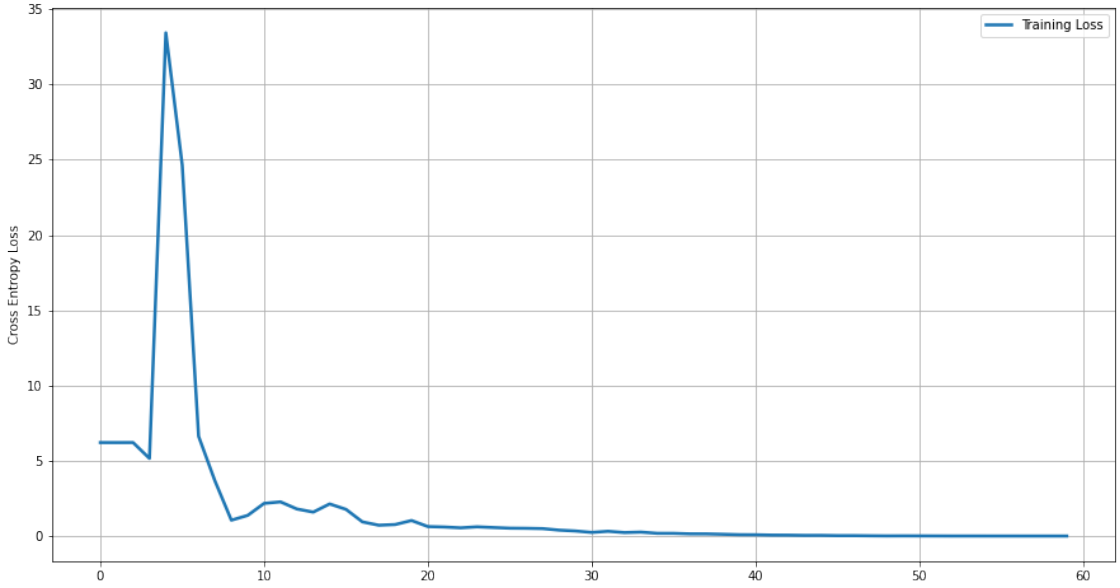
```
Epoch [1/30], Step [1/2], Loss: 6.2094
Epoch [1/30], Step [2/2], Loss: 6.2080
Epoch Train Time: 0.26 secs
test_accuracy=  0.0
Epoch [2/30], Step [1/2], Loss: 6.2117
Epoch [2/30], Step [2/2], Loss: 4.0994
Epoch Train Time: 0.60 secs
test_accuracy=  56.52173913043478
Epoch [3/30], Step [1/2], Loss: 33.4282
Epoch [3/30], Step [2/2], Loss: 15.8642
Epoch Train Time: 0.58 secs
test_accuracy=  56.52173913043478
Epoch [4/30], Step [1/2], Loss: 6.6270
Epoch [4/30], Step [2/2], Loss: 0.6853
Epoch Train Time: 0.57 secs
test_accuracy=  43.47826086956522
Epoch [5/30], Step [1/2], Loss: 1.0533
Epoch [5/30], Step [2/2], Loss: 1.6899
Epoch Train Time: 0.55 secs
test_accuracy=  43.47826086956522
Epoch [6/30], Step [1/2], Loss: 2.1701
Epoch [6/30], Step [2/2], Loss: 2.3713
Epoch Train Time: 0.55 secs
test_accuracy=  43.47826086956522
Epoch [7/30], Step [1/2], Loss: 1.7976
Epoch [7/30], Step [2/2], Loss: 1.3820
Epoch Train Time: 0.55 secs
test_accuracy=  43.47826086956522
Epoch [8/30], Step [1/2], Loss: 2.1331
Epoch [8/30], Step [2/2], Loss: 1.4125
Epoch Train Time: 0.55 secs
test_accuracy=  43.47826086956522
Epoch [9/30], Step [1/2], Loss: 0.9417
Epoch [9/30], Step [2/2], Loss: 0.4967
Epoch Train Time: 0.55 secs
test_accuracy=  56.52173913043478
Epoch [10/30], Step [1/2], Loss: 0.7641
Epoch [10/30], Step [2/2], Loss: 1.2988
Epoch Train Time: 0.55 secs
test_accuracy=  56.52173913043478
Epoch [11/30], Step [1/2], Loss: 0.6292
Epoch [11/30], Step [2/2], Loss: 0.5709
Epoch Train Time: 0.55 secs
```

```
test_accuracy=  52.17391304347826
Epoch [12/30], Step [1/2], Loss: 0.5451
Epoch [12/30], Step [2/2], Loss: 0.6798
Epoch Train Time: 0.56 secs
test_accuracy=  43.47826086956522
Epoch [13/30], Step [1/2], Loss: 0.5650
Epoch [13/30], Step [2/2], Loss: 0.4796
Epoch Train Time: 0.56 secs
test_accuracy=  43.47826086956522
Epoch [14/30], Step [1/2], Loss: 0.5142
Epoch [14/30], Step [2/2], Loss: 0.4741
Epoch Train Time: 0.55 secs
test_accuracy=  43.47826086956522
Epoch [15/30], Step [1/2], Loss: 0.3929
Epoch [15/30], Step [2/2], Loss: 0.2797
Epoch Train Time: 0.56 secs
test_accuracy=  47.82608695652174
Epoch [16/30], Step [1/2], Loss: 0.2405
Epoch [16/30], Step [2/2], Loss: 0.3930
Epoch Train Time: 0.54 secs
test_accuracy=  60.86956521739131
Epoch [17/30], Step [1/2], Loss: 0.2304
Epoch [17/30], Step [2/2], Loss: 0.2928
Epoch Train Time: 0.56 secs
test_accuracy=  60.86956521739131
Epoch [18/30], Step [1/2], Loss: 0.1818
Epoch [18/30], Step [2/2], Loss: 0.1780
Epoch Train Time: 0.55 secs
test_accuracy=  56.52173913043478
Epoch [19/30], Step [1/2], Loss: 0.1457
Epoch [19/30], Step [2/2], Loss: 0.1402
Epoch Train Time: 0.55 secs
test_accuracy=  56.52173913043478
Epoch [20/30], Step [1/2], Loss: 0.1179
Epoch [20/30], Step [2/2], Loss: 0.0609
Epoch Train Time: 0.55 secs
test_accuracy=  56.52173913043478
Epoch [21/30], Step [1/2], Loss: 0.0843
Epoch [21/30], Step [2/2], Loss: 0.0403
Epoch Train Time: 0.55 secs
test_accuracy=  65.21739130434783
Epoch [22/30], Step [1/2], Loss: 0.0587
Epoch [22/30], Step [2/2], Loss: 0.0279
Epoch Train Time: 0.55 secs
test_accuracy=  65.21739130434783
Epoch [23/30], Step [1/2], Loss: 0.0442
Epoch [23/30], Step [2/2], Loss: 0.0104
Epoch Train Time: 0.56 secs
```

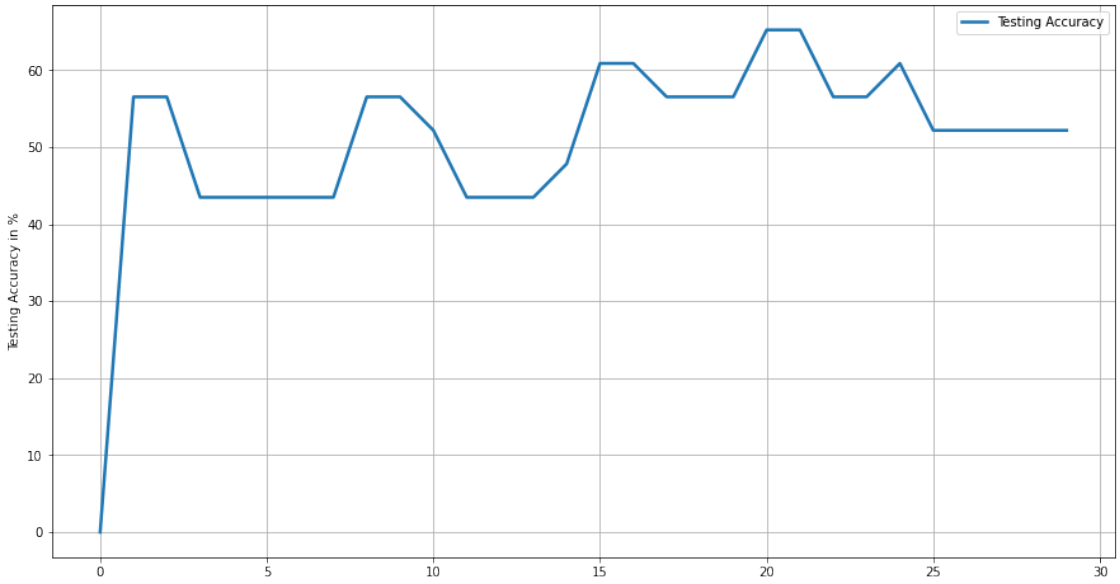
```
test_accuracy=  56.52173913043478
Epoch [24/30], Step [1/2], Loss: 0.0271
Epoch [24/30], Step [2/2], Loss: 0.0059
Epoch Train Time: 0.55 secs
test_accuracy=  56.52173913043478
Epoch [25/30], Step [1/2], Loss: 0.0084
Epoch [25/30], Step [2/2], Loss: 0.0149
Epoch Train Time: 0.56 secs
test_accuracy=  60.86956521739131
Epoch [26/30], Step [1/2], Loss: 0.0093
Epoch [26/30], Step [2/2], Loss: 0.0031
Epoch Train Time: 0.57 secs
test_accuracy=  52.17391304347826
Epoch [27/30], Step [1/2], Loss: 0.0028
Epoch [27/30], Step [2/2], Loss: 0.0046
Epoch Train Time: 0.55 secs
test_accuracy=  52.17391304347826
Epoch [28/30], Step [1/2], Loss: 0.0025
Epoch [28/30], Step [2/2], Loss: 0.0015
Epoch Train Time: 0.55 secs
test_accuracy=  52.17391304347826
Epoch [29/30], Step [1/2], Loss: 0.0016
Epoch [29/30], Step [2/2], Loss: 0.0011
Epoch Train Time: 0.56 secs
test_accuracy=  52.17391304347826
Epoch [30/30], Step [1/2], Loss: 0.0008
Epoch [30/30], Step [2/2], Loss: 0.0012
Epoch Train Time: 0.55 secs
test_accuracy=  52.17391304347826
Total Train Time: 0.27 mins

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:24: FutureWarning:
Slicing a positional slice with .loc is not supported, and will raise TypeError
in a future version.  Use .loc with labels or .iloc with positions instead.
```

[10]:

	exp_name	Loss	Accuracy
8	loss: CXE, optimizer: Adam, lr: 0.01, epoch: 3...	0.702	43.478
9	loss: CXE, opt: Adam, lr: 0.001, eps:1e-08 epo...	0.001	52.174



<Figure size 432x288 with 0 Axes>



15 Model 11

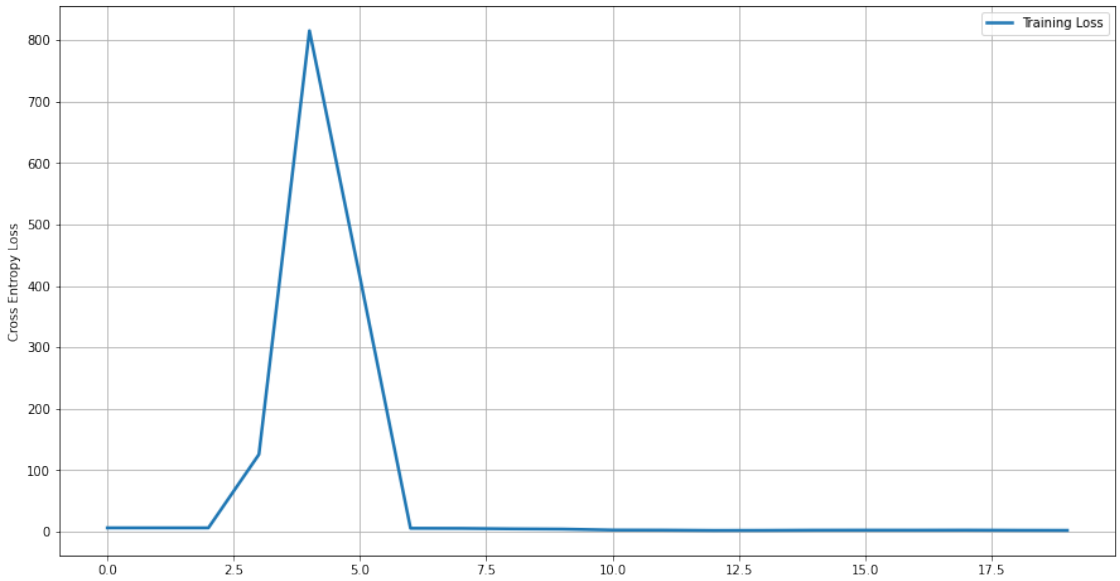
[11]:
```
# Download the training and testing data
mini_batch = 50
train_loader = trainload(mini_batch)
test_loader = testload(mini_batch)

# Initialize model
cnn11 = CNN()
if device.type == "cuda":
    cnn11.to('cuda')

# Define loss function
loss_func = torch.nn.CrossEntropyLoss()

# Define optimizer
optimizer = torch.optim.Adam(cnn11.parameters(), lr = 0.01, eps=1e-08)

# Train the model
num_epochs = 10

train_loss, test_accuracy = train(num_epochs, cnn11, train_loader, test_loader,␣
 ↪mini_batch)

#Logging the experiments
exp_name = f"loss: CXE, opt: Adam, lr: 0.01, eps:1e-08 epoch: 10, mini_batch=50"
expLog_R.loc[10,:3] = [f"{exp_name}"] + [f"{np.round(train_loss[-1],3)}"] +␣
 ↪[f"{np.round(test_accuracy[-1],3)}"]
expLog_R
```

```
Epoch [1/10], Step [1/2], Loss: 6.2266
Epoch [1/10], Step [2/2], Loss: 6.2299
Epoch Train Time: 0.26 secs
test_accuracy=  0.0
Epoch [2/10], Step [1/2], Loss: 6.2240
Epoch [2/10], Step [2/2], Loss: 245.2419
Epoch Train Time: 0.54 secs
test_accuracy=  56.52173913043478
Epoch [3/10], Step [1/2], Loss: 815.2668
Epoch [3/10], Step [2/2], Loss: 12.8128
Epoch Train Time: 0.55 secs
test_accuracy=  56.52173913043478
Epoch [4/10], Step [1/2], Loss: 5.5534
Epoch [4/10], Step [2/2], Loss: 5.2236
Epoch Train Time: 0.55 secs
test_accuracy=  56.52173913043478
Epoch [5/10], Step [1/2], Loss: 4.6255
```

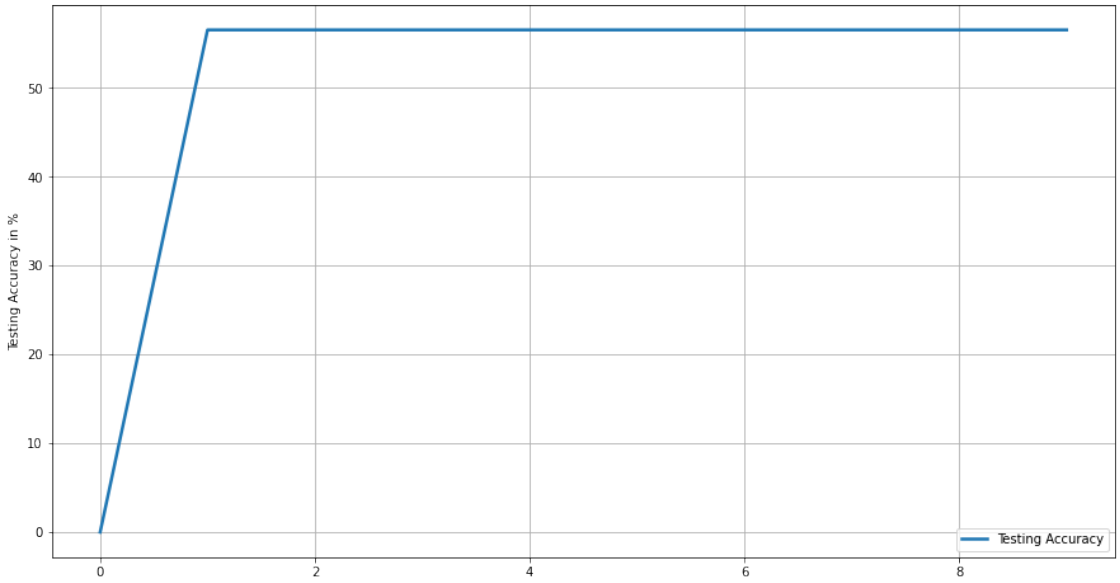
```
Epoch [5/10], Step [2/2], Loss: 3.7931
Epoch Train Time: 0.55 secs
test_accuracy=  56.52173913043478
Epoch [6/10], Step [1/2], Loss: 2.6868
Epoch [6/10], Step [2/2], Loss: 2.4278
Epoch Train Time: 0.54 secs
test_accuracy=  56.52173913043478
Epoch [7/10], Step [1/2], Loss: 2.0508
Epoch [7/10], Step [2/2], Loss: 2.2646
Epoch Train Time: 0.54 secs
test_accuracy=  56.52173913043478
Epoch [8/10], Step [1/2], Loss: 2.4589
Epoch [8/10], Step [2/2], Loss: 2.5507
Epoch Train Time: 0.54 secs
test_accuracy=  56.52173913043478
Epoch [9/10], Step [1/2], Loss: 2.4640
Epoch [9/10], Step [2/2], Loss: 2.6453
Epoch Train Time: 0.55 secs
test_accuracy=  56.52173913043478
Epoch [10/10], Step [1/2], Loss: 2.2986
Epoch [10/10], Step [2/2], Loss: 2.0342
Epoch Train Time: 0.55 secs
test_accuracy=  56.52173913043478
Total Train Time: 0.09 mins

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:24: FutureWarning:
Slicing a positional slice with .loc is not supported, and will raise TypeError
in a future version.  Use .loc with labels or .iloc with positions instead.
```

[11]:

	exp_name	Loss	Accuracy
8	loss: CXE, optimizer: Adam, lr: 0.01, epoch: 3...	0.702	43.478
9	loss: CXE, opt: Adam, lr: 0.001, eps:1e-08 epo...	0.001	52.174
10	loss: CXE, opt: Adam, lr: 0.01, eps:1e-08 epoc...	2.166	56.522




```
<Figure size 432x288 with 0 Axes>
```



[12]:
```
#%%shell
#jupyter nbconvert --to html /content/Group_19_CNN.ipynb
```