

## cnn\_mini\_lenet

November 27, 2025

```
[362]: import sys
import numpy as np
import matplotlib.pyplot as plt
import pickle
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
from torchvision import transforms
from torch.utils.data import Dataset
from torch.utils.data import DataLoader, random_split
import os

np.random.seed(0)
```

```
[363]: if torch.backends.mps.is_available():
    device = torch.device("mps")
    use_mps = True
else:
    device = torch.device("cpu")
    use_mps = False

print(device)
```

mps

```
[364]: class PKLDataset(Dataset):
    def __init__(self, path, transform=None):
        with open(path, "rb") as f:
            data = pickle.load(f)

            self.images = data["images"]          # shape: (N, 28, 28, 3)
            self.labels = data["labels"].reshape(-1)  # shape: (N,) instead of ↵
            ↵ (N, 1)
            self.transform = transform

    def __len__(self):
        return len(self.images)
```

```

def __getitem__(self, idx):
    img = self.images[idx]           # numpy array (28,28,3)
    label = int(self.labels[idx])    # convert to Python int

    # Convert to tensor and permute to (C, H, W)
    img = torch.tensor(img, dtype=torch.float32).permute(2, 0, 1) / 255.0

    if self.transform:
        img = self.transform(img)

    return img, label

```

```

[365]: import pickle

with open("ift-3395-6390-kaggle-2-competition-fall-2025/train_data.pkl", "rb") as f:
    data = pickle.load(f)

print(type(data))
print(len(data) if hasattr(data, "__len__") else "no len")
print(data)

```

```

<class 'dict'>
2
{'images': array([[[[ 6,  4,  0],
   [ 9,  5,  0],
   [ 8,  4,  0],
   ...,
   [ 9,  6,  0],
   [ 9,  6,  0],
   [ 7,  4,  0]],

  [[11,  6,  0],
   [ 4,  4,  0],
   [ 3,  3,  0],
   ...,
   [ 9,  6,  0],
   [ 6,  4,  0],
   [ 4,  2,  0]],

  [[11,  6,  0],
   [ 4,  4,  0],
   [ 3,  3,  0],
   ...,
   [ 6,  4,  0],
   [ 6,  4,  0],
   [ 4,  2,  0]]],
```

```

...,

[[ 1,  1,  0],
 [ 0,  0,  0],
 [ 0,  0,  1],
 ...,
 [ 5,  4,  0],
 [ 6,  5,  0],
 [ 6,  5,  0]],

[[ 3,  1,  1],
 [ 0,  0,  0],
 [ 0,  0,  1],
 ...,
 [ 6,  5,  0],
 [ 6,  5,  0],
 [ 7,  6,  0]],

[[[10,  2,  2],
 [ 0,  0,  1],
 [ 0,  0,  1],
 ...,
 [ 6,  5,  0],
 [ 7,  6,  0],
 [ 7,  6,  0]]],


[[[11,  9,  0],
 [ 9,  7,  0],
 [ 9,  7,  0],
 ...,
 [ 0,  0,  1],
 [ 0,  0,  1],
 [ 0,  0,  1]],

[[12,  9,  0],
 [11,  7,  0],
 [ 9,  6,  0],
 ...,
 [ 0,  0,  2],
 [ 0,  0,  1],
 [ 0,  0,  1]],

[[ 9,  7,  0],
 [12,  8,  0],
 [10,  6,  0],
 ...,

```

```

[ 0,  0,  1],
[ 0,  0,  1],
[ 0,  0,  0]],

...,

[[ 0,  0,  3],
[ 0,  0,  3],
[ 0,  0,  4],
...,

[ 0,  0,  2],
[ 0,  0,  1],
[ 0,  0,  0]],

[[ 0,  0,  2],
[ 0,  0,  2],
[ 0,  0,  5],
...,

[ 0,  0,  2],
[ 0,  0,  1],
[ 1,  0,  0]],

[[ 0,  0,  1],
[ 0,  0,  2],
[ 0,  0,  4],
...,

[ 0,  0,  1],
[ 1,  0,  0],
[ 1,  0,  0]]],


[[[18, 12,  0],
[12,  9,  0],
[17, 11,  0],
...,

[ 0,  0,  3],
[ 5,  0,  2],
[ 5,  0,  2]],

[[15, 11,  0],
[10,  9,  0],
[10,  8,  0],
...,

[ 2,  0,  2],
[ 7,  0,  1],
[ 8,  0,  1]],

[[17, 10,  0],

```

```

[ 7, 6, 0],
[ 4, 4, 0],
...,
[ 0, 0, 2],
[ 5, 0, 1],
[ 8, 0, 1]],

...,

[[ 2, 0, 0],
[ 1, 0, 0],
[ 0, 0, 0],
...,
[ 0, 0, 1],
[ 0, 0, 0],
[ 0, 0, 1]],

[[ 2, 0, 0],
[ 3, 1, 0],
[ 0, 0, 0],
...,
[ 0, 0, 0],
[ 0, 0, 1],
[ 1, 0, 0]],

[[ 3, 0, 0],
[ 2, 0, 0],
[ 2, 1, 0],
...,
[ 0, 0, 1],
[ 1, 0, 0],
[ 1, 0, 0]],

...,

[[[56, 8, 20],
[19, 0, 11],
[ 0, 0, 1],
...,
[ 6, 3, 0],
[11, 7, 0],
[16, 8, 0]],

[[19, 0, 11],
[ 5, 0, 7],
[ 0, 0, 3],
```

```

...,
[[ 5,  3,  0],
 [ 5,  3,  0],
 [ 8,  4,  0]],

[[ 0,  0,  0],
 [ 0,  0,  2],
 [ 1,  0,  6],
 ...,
 [ 7,  4,  0],
 [ 5,  3,  0],
 [ 4,  2,  0]],

...,

[[ 4,  3,  0],
 [ 4,  2,  0],
 [ 4,  2,  0],
 ...,
 [ 0,  0,  1],
 [ 1,  0,  0],
 [ 0,  0,  0]],

[[ 1,  1,  0],
 [ 1,  1,  0],
 [ 4,  2,  0],
 ...,
 [ 0,  0,  0],
 [ 3,  1,  0],
 [ 1,  1,  0]],

[[ 1,  1,  0],
 [ 1,  1,  0],
 [ 0,  0,  0],
 ...,
 [ 1,  0,  0],
 [ 3,  2,  0],
 [ 3,  3,  0]]],


[[[ 9,  6,  0],
 [ 8,  6,  0],
 [ 6,  4,  0],
 ...,
 [ 3,  2,  0],
 [ 3,  2,  0],
 [ 0,  0,  0]],
```

```

[[ 6,  6,  0],
 [ 4,  4,  0],
 [ 6,  4,  0],
 ...,
 [ 3,  2,  0],
 [ 1,  0,  0],
 [ 2,  0,  0]],

[[ 4,  4,  0],
 [ 6,  4,  0],
 [ 6,  4,  0],
 ...,
 [ 1,  0,  0],
 [ 2,  0,  0],
 [ 3,  0,  0]],

...,

[[ 3,  3,  0],
 [ 3,  3,  0],
 [ 3,  1,  0],
 ...,
 [ 0,  0,  1],
 [ 0,  0,  0],
 [ 0,  0,  0]],

[[ 3,  3,  0],
 [ 3,  3,  0],
 [ 4,  2,  0],
 ...,
 [ 2,  1,  0],
 [ 2,  1,  0],
 [ 1,  1,  0]],

[[ 3,  3,  0],
 [ 3,  3,  0],
 [ 4,  2,  0],
 ...,
 [ 2,  1,  0],
 [ 2,  1,  0],
 [ 1,  1,  0]]],


[[[ 6,  3,  0],
 [ 3,  2,  0],
 [ 2,  0,  2],
 ...,
 [ 7,  6,  0],

```

```

[ 7, 6, 0],
[ 6, 6, 0]],

[[ 4, 2, 0],
[ 1, 0, 1],
[ 2, 0, 2],
...,
[ 8, 5, 0],
[ 7, 5, 0],
[ 6, 4, 0]],

[[ 2, 0, 0],
[ 0, 0, 1],
[ 0, 0, 3],
...,
[ 5, 3, 0],
[ 7, 4, 0],
[ 7, 4, 0]],

...,

[[ 4, 1, 0],
[ 2, 0, 0],
[ 1, 0, 0],
...,
[ 4, 2, 0],
[ 6, 4, 0],
[ 6, 4, 0]],

[[ 7, 3, 0],
[ 7, 4, 0],
[ 6, 2, 0],
...,
[ 6, 4, 0],
[ 7, 4, 0],
[ 7, 4, 0]],

[[11, 6, 0],
[10, 5, 0],
[12, 5, 0],
...,
[ 7, 4, 0],
[ 7, 4, 0],
[ 7, 5, 0]]], shape=(1080, 28, 28, 3), dtype=uint8), 'labels':
array([[0],
       [0],
       [0],
       ...,

```

```

[2],
[2],
[3]], shape=(1080, 1), dtype=uint8) }

[366]: from sklearn.model_selection import train_test_split
from torchvision import transforms
from torch.utils.data import Subset

# Charger le dataset SANS transformation d'abord
dataset = PKLDataset(
    "ift-3395-6390-kaggle-2-competition-fall-2025/train_data.pkl",
)

loader = DataLoader(dataset, batch_size=64, shuffle=False)

d_mean = torch.zeros(3)
d_std = torch.zeros(3)
nb_samples = 0.0

for images, _ in loader:
    batch_samples = images.size(0)

    d_mean += images.mean(dim=[0,2,3]) * batch_samples
    d_std += images.std(dim=[0,2,3]) * batch_samples
    nb_samples += batch_samples

d_mean /= nb_samples
d_std /= nb_samples

d_mean = d_mean.tolist()
d_std = d_std.tolist()

print("Mean:", d_mean)
print("Std:", d_std)

```

Mean: [0.21014535427093506, 0.005330359563231468, 0.2285669893026352]  
Std: [0.18871904909610748, 0.01642582379281521, 0.16962255537509918]

```

[367]: # Définir les transformations
transform_train = transforms.Compose([
    transforms.Normalize(mean=d_mean, std=d_std),
    #transforms.RandomHorizontalFlip(p=0.5),
    #transforms.RandomVerticalFlip(),
    #transforms.RandomRotation(10),
    #transforms.RandomAffine(degrees=0, translate=(0.1, 0.1)),

```

```

        #transforms.ColorJitter(brightness=0.1, contrast=0.1),
        #transforms.GaussianBlur(3),
    ])

transform_val = transforms.Compose([
    transforms.Normalize(mean=d_mean, std=d_std),
])

```

[368]: labels = dataset.labels

```

# Split stratifié
indices = np.arange(len(dataset))
train_idx, valid_idx = train_test_split(
    indices,
    test_size=0.2,
    random_state=42,
    stratify=labels
)

# Créer des subsets
train_data = Subset(dataset, train_idx)
valid_data = Subset(dataset, valid_idx)

class TransformSubset(Dataset):
    def __init__(self, subset, transform=None):
        self.subset = subset
        self.transform = transform

    def __getitem__(self, idx):
        image, label = self.subset[idx]
        if self.transform:
            image = self.transform(image)
        return image, label

    def __len__(self):
        return len(self.subset)

# Appliquer les transformations
train_data = TransformSubset(train_data, transform=transform_train)
valid_data = TransformSubset(valid_data, transform=transform_val)

# DataLoaders
train_loader = DataLoader(train_data, batch_size=8, shuffle=True)
valid_loader = DataLoader(valid_data, batch_size=16, shuffle=False)

```

[369]: import numpy as np

```

labels = dataset.labels
classes, counts = np.unique(labels, return_counts=True)

from sklearn.utils.class_weight import compute_class_weight

weights = compute_class_weight(
    class_weight="balanced",
    classes=np.unique(labels),
    y=labels
)
class_weights = torch.tensor(weights, dtype=torch.float32).to(device)

loss_fn = nn.CrossEntropyLoss()

test_loss_fn = nn.CrossEntropyLoss(reduction='sum')

# spot to save your learning curves, and potentially checkpoint your models
savedir = 'results'
if not os.path.exists(savedir):
    os.makedirs(savedir)

```

```

[370]: def train(model, train_loader, optimizer, epoch):
    """Perform one epoch of training."""
    model.train()

    for batch_idx, (inputs, target) in enumerate(train_loader):
        inputs, target = inputs.to(device), target.to(device)

        # 1) Reset gradients
        optimizer.zero_grad()

        # 2) Forward pass
        output = model(inputs)

        # 3) Compute loss
        loss = loss_fn(output, target)

        # 4) Backpropagation
        loss.backward()

        # 5) Update weights
        optimizer.step()

        # Logging
        if batch_idx % 10 == 0:
            print('Train Epoch: {} [{}/{} ({:.0f}%)]\tLoss: {:.6f}'.format(
                epoch,

```

```
    batch_idx * len(inputs),
    len(train_loader.dataset),
    100. * batch_idx / len(train_loader),
    loss.item()
))
```

```
[371]: def test(model, test_loader):
    """Evaluate the model by doing one pass over a dataset"""
    model.eval()

    test_loss = 0      # total loss over test set
    correct = 0       # total number of correct test predictions
    test_size = 0      # number of test samples used

    with torch.no_grad(): # no backprop, faster evaluation
        for inputs, target in test_loader:
            inputs, target = inputs.to(device), target.to(device)

            # Forward pass
            output = model(inputs)

            # Accumulate loss (sum, not mean)
            loss = test_loss_fn(output, target) # already reduction='sum'
            test_loss += loss.item()

            # Predictions
            pred = output.argmax(dim=1) # index of highest logit
            correct += (pred == target).sum().item()

            # Keep track of sample count
            test_size += target.size(0)

    # Final metrics
    test_loss /= test_size
    accuracy = correct / test_size

    print('Test set: Average loss: {:.4f}, Accuracy: {}/{} ({:.0f}%)\n'.format(
        test_loss, correct, test_size, 100. * accuracy))

    return test_loss, accuracy
```

```
[376]: fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 5))

for filename in os.listdir(savedir):
    if filename.endswith('.pkl'):
        with open(os.path.join(savedir, filename), 'rb') as fin:
            results = pickle.load(fin)
```

```

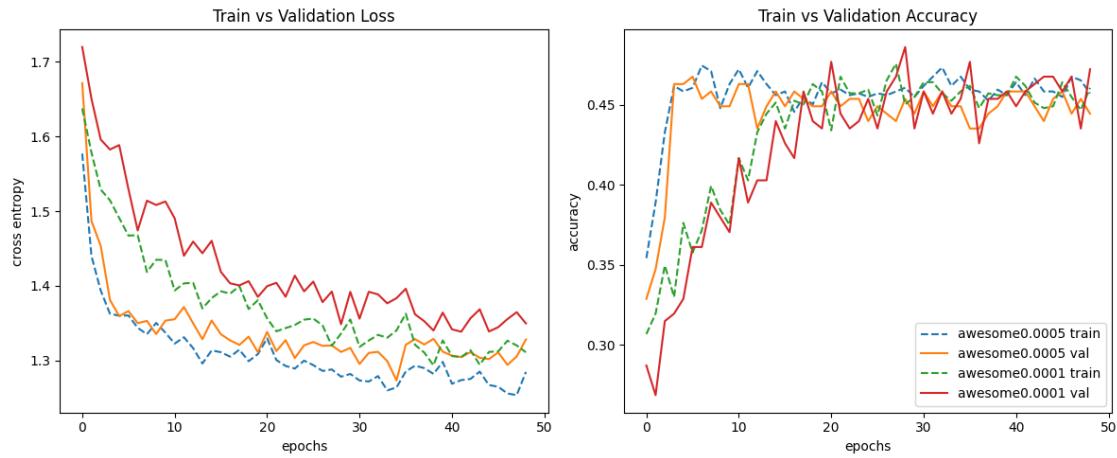
label = filename[:-4] # nom sans .pkl

# ----- Courbes de LOSS -----
ax1.plot(results['train_loss'], '--', label=f'{label} train')
ax1.plot(results['val_loss'], '--', label=f'{label} val')
ax1.set_ylabel('cross entropy')
ax1.set_xlabel('epochs')
ax1.set_title('Train vs Validation Loss')

# ----- Courbes d'ACCURACY -----
ax2.plot(results['train_acc'], '--', label=f'{label} train')
ax2.plot(results['val_acc'], '--', label=f'{label} val')
ax2.set_ylabel('accuracy')
ax2.set_xlabel('epochs')
ax2.set_title('Train vs Validation Accuracy')

plt.legend()
plt.tight_layout()
plt.show()

```



```

[373]: import torch
import torch.nn as nn
import torch.nn.functional as F

class XCos(nn.Module):
    def forward(self, x):
        # clamp to avoid division by very small cos values
        x = torch.clamp(x, -1.4, 1.4)
        return x / torch.cos(x)

```

```

class CNNNet(nn.Module):
    def __init__(self, num_classes=5):
        super().__init__()

        self.act = XCos()

        # --- Conv Block 1 ---
        self.conv1 = nn.Conv2d(3, 6, kernel_size=5, padding=2) # keep same
        ↵spatial size
        self.bn1 = nn.BatchNorm2d(6)

        # --- Conv Block 2 ---
        self.conv2 = nn.Conv2d(6, 16, kernel_size=5) # no padding + shrink
        self.bn2 = nn.BatchNorm2d(16)

        # --- Fully connected layers ---
        # After two pools: 28x28 → 14x14 → 5x5
        self.fc1 = nn.Linear(16 * 5 * 5, 120)
        self.bn3 = nn.BatchNorm1d(120)

        self.fc2 = nn.Linear(120, 84)
        self.bn4 = nn.BatchNorm1d(84)

        self.fc3 = nn.Linear(84, num_classes)

        self.dropout = nn.Dropout(0.5)

    def forward(self, x):
        # Block 1
        x = self.act(self.bn1(self.conv1(x)))
        x = F.max_pool2d(x, 2)

        # Block 2
        x = self.act(self.bn2(self.conv2(x)))
        x = F.max_pool2d(x, 2)

        # Flatten
        x = x.view(x.size(0), -1)

        # FC layers
        x = self.dropout(self.act(self.bn3(self.fc1(x))))
        x = self.act(self.bn4(self.fc2(x)))

        x = self.fc3(x)
        return x

```

```
[374]: subset = Subset(train_data, list(range(50)))
subset_loader = DataLoader(subset, batch_size=10, shuffle=True)
model = CNNNet().to(device)
lr = 0.0005
optimizer = optim.Adam(model.parameters(), lr=lr)

for epoch in range(50):
    train(model, subset_loader, optimizer, epoch)
    loss, acc = test(model, subset_loader)
    print(loss, acc)
```

Train Epoch: 0 [0/50 (0%)] Loss: 3.246361  
Test set: Average loss: 1.9643, Accuracy: 17/50 (34%)

1.9642770385742188 0.34  
Train Epoch: 1 [0/50 (0%)] Loss: 3.578418  
Test set: Average loss: 1.7150, Accuracy: 15/50 (30%)

1.7150343894958495 0.3  
Train Epoch: 2 [0/50 (0%)] Loss: 2.358491  
Test set: Average loss: 1.6351, Accuracy: 9/50 (18%)

1.6351260375976562 0.18  
Train Epoch: 3 [0/50 (0%)] Loss: 2.062575  
Test set: Average loss: 1.5834, Accuracy: 15/50 (30%)

1.5834258270263672 0.3  
Train Epoch: 4 [0/50 (0%)] Loss: 2.081959  
Test set: Average loss: 1.3954, Accuracy: 20/50 (40%)

1.395391731262207 0.4  
Train Epoch: 5 [0/50 (0%)] Loss: 2.382929  
Test set: Average loss: 1.2726, Accuracy: 27/50 (54%)

1.2726461219787597 0.54  
Train Epoch: 6 [0/50 (0%)] Loss: 1.831971  
Test set: Average loss: 1.2889, Accuracy: 23/50 (46%)

1.2888614273071288 0.46  
Train Epoch: 7 [0/50 (0%)] Loss: 1.452326  
Test set: Average loss: 1.2878, Accuracy: 24/50 (48%)

1.2877804374694823 0.48  
Train Epoch: 8 [0/50 (0%)] Loss: 1.693329  
Test set: Average loss: 1.2817, Accuracy: 24/50 (48%)

1.281693286895752 0.48  
Train Epoch: 9 [0/50 (0%)] Loss: 1.603549

Test set: Average loss: 1.3787, Accuracy: 22/50 (44%)

1.378739776611328 0.44

Train Epoch: 10 [0/50 (0%)] Loss: 1.393107

Test set: Average loss: 1.2681, Accuracy: 25/50 (50%)

1.268112678527832 0.5

Train Epoch: 11 [0/50 (0%)] Loss: 1.788290

Test set: Average loss: 1.2631, Accuracy: 26/50 (52%)

1.2630924034118651 0.52

Train Epoch: 12 [0/50 (0%)] Loss: 1.974171

Test set: Average loss: 1.1803, Accuracy: 30/50 (60%)

1.180320873260498 0.6

Train Epoch: 13 [0/50 (0%)] Loss: 1.595053

Test set: Average loss: 1.1178, Accuracy: 29/50 (58%)

1.117806510925293 0.58

Train Epoch: 14 [0/50 (0%)] Loss: 2.129482

Test set: Average loss: 1.1300, Accuracy: 28/50 (56%)

1.1299814319610595 0.56

Train Epoch: 15 [0/50 (0%)] Loss: 1.017410

Test set: Average loss: 1.0780, Accuracy: 29/50 (58%)

1.0779643630981446 0.58

Train Epoch: 16 [0/50 (0%)] Loss: 1.120140

Test set: Average loss: 1.0369, Accuracy: 29/50 (58%)

1.0369363021850586 0.58

Train Epoch: 17 [0/50 (0%)] Loss: 1.648139

Test set: Average loss: 0.9956, Accuracy: 31/50 (62%)

0.9956360244750977 0.62

Train Epoch: 18 [0/50 (0%)] Loss: 1.648045

Test set: Average loss: 0.9959, Accuracy: 34/50 (68%)

0.9959151458740234 0.68

Train Epoch: 19 [0/50 (0%)] Loss: 1.436266

Test set: Average loss: 0.9527, Accuracy: 31/50 (62%)

0.9526643753051758 0.62

Train Epoch: 20 [0/50 (0%)] Loss: 1.115302

Test set: Average loss: 0.9639, Accuracy: 34/50 (68%)

0.9638616275787354 0.68

Train Epoch: 21 [0/50 (0%)] Loss: 0.857325

Test set: Average loss: 0.9160, Accuracy: 33/50 (66%)

0.9160395050048828 0.66

Train Epoch: 22 [0/50 (0%)] Loss: 1.501520

Test set: Average loss: 0.9454, Accuracy: 35/50 (70%)

0.9454392433166504 0.7

Train Epoch: 23 [0/50 (0%)] Loss: 1.300773

Test set: Average loss: 0.9624, Accuracy: 35/50 (70%)

0.9623659515380859 0.7

Train Epoch: 24 [0/50 (0%)] Loss: 0.935182

Test set: Average loss: 0.9232, Accuracy: 36/50 (72%)

0.9231879997253418 0.72

Train Epoch: 25 [0/50 (0%)] Loss: 0.710264

Test set: Average loss: 0.9227, Accuracy: 36/50 (72%)

0.9226731109619141 0.72

Train Epoch: 26 [0/50 (0%)] Loss: 0.872690

Test set: Average loss: 0.9690, Accuracy: 35/50 (70%)

0.968971939086914 0.7

Train Epoch: 27 [0/50 (0%)] Loss: 1.070462

Test set: Average loss: 0.9194, Accuracy: 35/50 (70%)

0.9193580436706543 0.7

Train Epoch: 28 [0/50 (0%)] Loss: 1.743356

Test set: Average loss: 0.9314, Accuracy: 39/50 (78%)

0.9314253807067872 0.78

Train Epoch: 29 [0/50 (0%)] Loss: 0.992703

Test set: Average loss: 0.8581, Accuracy: 41/50 (82%)

0.8580945491790771 0.82

Train Epoch: 30 [0/50 (0%)] Loss: 0.802396

Test set: Average loss: 0.8449, Accuracy: 41/50 (82%)

0.8448917865753174 0.82

Train Epoch: 31 [0/50 (0%)] Loss: 1.752178

Test set: Average loss: 0.7921, Accuracy: 41/50 (82%)

0.7920825099945068 0.82

Train Epoch: 32 [0/50 (0%)] Loss: 1.877849

Test set: Average loss: 0.7837, Accuracy: 42/50 (84%)

0.7837172794342041 0.84

Train Epoch: 33 [0/50 (0%)] Loss: 1.455213

Test set: Average loss: 0.8000, Accuracy: 40/50 (80%)

0.800006046295166 0.8

Train Epoch: 34 [0/50 (0%)] Loss: 0.997139

Test set: Average loss: 0.7747, Accuracy: 39/50 (78%)

0.7747241306304932 0.78

Train Epoch: 35 [0/50 (0%)] Loss: 1.035955

Test set: Average loss: 0.8033, Accuracy: 37/50 (74%)

0.8033082962036133 0.74

Train Epoch: 36 [0/50 (0%)] Loss: 0.757934

Test set: Average loss: 0.7935, Accuracy: 41/50 (82%)

0.7934531593322753 0.82

Train Epoch: 37 [0/50 (0%)] Loss: 1.046719

Test set: Average loss: 0.7888, Accuracy: 42/50 (84%)

0.7887678909301757 0.84

Train Epoch: 38 [0/50 (0%)] Loss: 1.096822

Test set: Average loss: 0.7554, Accuracy: 42/50 (84%)

0.7554468059539795 0.84

Train Epoch: 39 [0/50 (0%)] Loss: 0.296361

Test set: Average loss: 0.7875, Accuracy: 42/50 (84%)

0.7874512672424316 0.84

Train Epoch: 40 [0/50 (0%)] Loss: 0.647872

Test set: Average loss: 0.7516, Accuracy: 42/50 (84%)

0.7515550136566163 0.84

Train Epoch: 41 [0/50 (0%)] Loss: 1.345892

Test set: Average loss: 0.7046, Accuracy: 43/50 (86%)

0.7045807361602783 0.86

Train Epoch: 42 [0/50 (0%)] Loss: 0.939490

Test set: Average loss: 0.6752, Accuracy: 44/50 (88%)

0.6752281093597412 0.88

Train Epoch: 43 [0/50 (0%)] Loss: 1.435943

Test set: Average loss: 0.6908, Accuracy: 41/50 (82%)

0.690772180557251 0.82

Train Epoch: 44 [0/50 (0%)] Loss: 0.869362

Test set: Average loss: 0.6881, Accuracy: 41/50 (82%)

0.6880568504333496 0.82

Train Epoch: 45 [0/50 (0%)] Loss: 1.198344

```

Test set: Average loss: 0.6975, Accuracy: 39/50 (78%)

0.6974782276153565 0.78
Train Epoch: 46 [0/50 (0%)]      Loss: 1.135091
Test set: Average loss: 0.7387, Accuracy: 41/50 (82%)

0.7386514854431152 0.82
Train Epoch: 47 [0/50 (0%)]      Loss: 0.511344
Test set: Average loss: 0.7364, Accuracy: 43/50 (86%)

0.736418514251709 0.86
Train Epoch: 48 [0/50 (0%)]      Loss: 0.947497
Test set: Average loss: 0.7673, Accuracy: 40/50 (80%)

0.7672672939300537 0.8
Train Epoch: 49 [0/50 (0%)]      Loss: 1.160058
Test set: Average loss: 0.7261, Accuracy: 41/50 (82%)

0.7260635662078857 0.82

```

```

[375]: # TRAINING
model = CNNNet().to(device)

lr = 0.0005
optimizer = optim.Adam(model.parameters(), lr=lr)

results = {'name':'awesome', 'lr': lr, 'train_loss': [],
           'train_acc': [],
           'val_loss': [],
           'val_acc': []
         }
savefile = os.path.join(savedir, results['name']+str(results['lr'])+'.pkl' )

for epoch in range(1, 50):
    train(model, train_loader, optimizer, epoch)

    train_loss, train_acc = test(model, train_loader)
    val_loss, val_acc = test(model, valid_loader)

    results['train_loss'].append(train_loss)
    results['train_acc'].append(train_acc)

    results['val_loss'].append(val_loss)
    results['val_acc'].append(val_acc)

    with open(savefile, 'wb') as fout:
        pickle.dump(results, fout)

```

```
Train Epoch: 1 [0/864 (0%)]      Loss: 4.049563
Train Epoch: 1 [80/864 (9%)]     Loss: 3.197384
Train Epoch: 1 [160/864 (19%)]    Loss: 2.956885
Train Epoch: 1 [240/864 (28%)]    Loss: 3.540251
Train Epoch: 1 [320/864 (37%)]    Loss: 1.928679
Train Epoch: 1 [400/864 (46%)]    Loss: 3.062601
Train Epoch: 1 [480/864 (56%)]    Loss: 2.459350
Train Epoch: 1 [560/864 (65%)]    Loss: 2.516848
Train Epoch: 1 [640/864 (74%)]    Loss: 2.654123
Train Epoch: 1 [720/864 (83%)]    Loss: 2.273489
Train Epoch: 1 [800/864 (93%)]    Loss: 2.630432
Test set: Average loss: 1.5772, Accuracy: 306/864 (35%)
```

Test set: Average loss: 1.6716, Accuracy: 71/216 (33%)

```
Train Epoch: 2 [0/864 (0%)]      Loss: 1.893392
Train Epoch: 2 [80/864 (9%)]     Loss: 3.317691
Train Epoch: 2 [160/864 (19%)]    Loss: 2.696556
Train Epoch: 2 [240/864 (28%)]    Loss: 0.984051
Train Epoch: 2 [320/864 (37%)]    Loss: 2.132479
Train Epoch: 2 [400/864 (46%)]    Loss: 1.612870
Train Epoch: 2 [480/864 (56%)]    Loss: 2.096712
Train Epoch: 2 [560/864 (65%)]    Loss: 1.542459
Train Epoch: 2 [640/864 (74%)]    Loss: 2.352394
Train Epoch: 2 [720/864 (83%)]    Loss: 1.424272
Train Epoch: 2 [800/864 (93%)]    Loss: 2.128184
Test set: Average loss: 1.4398, Accuracy: 336/864 (39%)
```

Test set: Average loss: 1.4867, Accuracy: 75/216 (35%)

```
Train Epoch: 3 [0/864 (0%)]      Loss: 2.000822
Train Epoch: 3 [80/864 (9%)]     Loss: 1.446383
Train Epoch: 3 [160/864 (19%)]    Loss: 2.012389
Train Epoch: 3 [240/864 (28%)]    Loss: 1.342784
Train Epoch: 3 [320/864 (37%)]    Loss: 1.427157
Train Epoch: 3 [400/864 (46%)]    Loss: 1.801977
Train Epoch: 3 [480/864 (56%)]    Loss: 2.145951
Train Epoch: 3 [560/864 (65%)]    Loss: 2.301106
Train Epoch: 3 [640/864 (74%)]    Loss: 1.065500
Train Epoch: 3 [720/864 (83%)]    Loss: 1.319576
Train Epoch: 3 [800/864 (93%)]    Loss: 1.492340
Test set: Average loss: 1.3937, Accuracy: 374/864 (43%)
```

Test set: Average loss: 1.4535, Accuracy: 82/216 (38%)

```
Train Epoch: 4 [0/864 (0%)]      Loss: 1.851864
Train Epoch: 4 [80/864 (9%)]     Loss: 1.561983
Train Epoch: 4 [160/864 (19%)]    Loss: 1.522848
```

```
Train Epoch: 4 [240/864 (28%)] Loss: 2.275672
Train Epoch: 4 [320/864 (37%)] Loss: 1.591223
Train Epoch: 4 [400/864 (46%)] Loss: 1.287260
Train Epoch: 4 [480/864 (56%)] Loss: 2.617677
Train Epoch: 4 [560/864 (65%)] Loss: 1.283360
Train Epoch: 4 [640/864 (74%)] Loss: 2.069630
Train Epoch: 4 [720/864 (83%)] Loss: 2.198033
Train Epoch: 4 [800/864 (93%)] Loss: 1.946620
Test set: Average loss: 1.3625, Accuracy: 399/864 (46%)
```

Test set: Average loss: 1.3811, Accuracy: 100/216 (46%)

```
Train Epoch: 5 [0/864 (0%)] Loss: 2.053038
Train Epoch: 5 [80/864 (9%)] Loss: 2.277802
Train Epoch: 5 [160/864 (19%)] Loss: 1.190848
Train Epoch: 5 [240/864 (28%)] Loss: 1.727794
Train Epoch: 5 [320/864 (37%)] Loss: 1.333760
Train Epoch: 5 [400/864 (46%)] Loss: 1.393091
Train Epoch: 5 [480/864 (56%)] Loss: 1.052721
Train Epoch: 5 [560/864 (65%)] Loss: 1.105615
Train Epoch: 5 [640/864 (74%)] Loss: 1.097529
Train Epoch: 5 [720/864 (83%)] Loss: 1.454311
Train Epoch: 5 [800/864 (93%)] Loss: 1.706666
Test set: Average loss: 1.3599, Accuracy: 396/864 (46%)
```

Test set: Average loss: 1.3593, Accuracy: 100/216 (46%)

```
Train Epoch: 6 [0/864 (0%)] Loss: 1.859358
Train Epoch: 6 [80/864 (9%)] Loss: 1.082456
Train Epoch: 6 [160/864 (19%)] Loss: 2.024524
Train Epoch: 6 [240/864 (28%)] Loss: 1.392885
Train Epoch: 6 [320/864 (37%)] Loss: 1.107300
Train Epoch: 6 [400/864 (46%)] Loss: 2.078992
Train Epoch: 6 [480/864 (56%)] Loss: 1.488758
Train Epoch: 6 [560/864 (65%)] Loss: 1.423407
Train Epoch: 6 [640/864 (74%)] Loss: 1.628813
Train Epoch: 6 [720/864 (83%)] Loss: 1.266345
Train Epoch: 6 [800/864 (93%)] Loss: 1.387010
Test set: Average loss: 1.3603, Accuracy: 398/864 (46%)
```

Test set: Average loss: 1.3661, Accuracy: 101/216 (47%)

```
Train Epoch: 7 [0/864 (0%)] Loss: 1.552865
Train Epoch: 7 [80/864 (9%)] Loss: 1.307847
Train Epoch: 7 [160/864 (19%)] Loss: 2.291791
Train Epoch: 7 [240/864 (28%)] Loss: 2.112608
Train Epoch: 7 [320/864 (37%)] Loss: 1.708389
Train Epoch: 7 [400/864 (46%)] Loss: 1.119328
```

```
Train Epoch: 7 [480/864 (56%)] Loss: 1.510254
Train Epoch: 7 [560/864 (65%)] Loss: 2.025012
Train Epoch: 7 [640/864 (74%)] Loss: 1.725900
Train Epoch: 7 [720/864 (83%)] Loss: 2.041892
Train Epoch: 7 [800/864 (93%)] Loss: 1.053825
Test set: Average loss: 1.3436, Accuracy: 410/864 (47%)
```

Test set: Average loss: 1.3499, Accuracy: 98/216 (45%)

```
Train Epoch: 8 [0/864 (0%)] Loss: 1.193032
Train Epoch: 8 [80/864 (9%)] Loss: 1.351406
Train Epoch: 8 [160/864 (19%)] Loss: 1.565144
Train Epoch: 8 [240/864 (28%)] Loss: 2.048691
Train Epoch: 8 [320/864 (37%)] Loss: 1.004503
Train Epoch: 8 [400/864 (46%)] Loss: 1.523039
Train Epoch: 8 [480/864 (56%)] Loss: 1.400535
Train Epoch: 8 [560/864 (65%)] Loss: 1.142431
Train Epoch: 8 [640/864 (74%)] Loss: 1.268573
Train Epoch: 8 [720/864 (83%)] Loss: 0.835496
Train Epoch: 8 [800/864 (93%)] Loss: 2.001235
Test set: Average loss: 1.3351, Accuracy: 407/864 (47%)
```

Test set: Average loss: 1.3529, Accuracy: 99/216 (46%)

```
Train Epoch: 9 [0/864 (0%)] Loss: 1.603295
Train Epoch: 9 [80/864 (9%)] Loss: 1.012705
Train Epoch: 9 [160/864 (19%)] Loss: 1.571244
Train Epoch: 9 [240/864 (28%)] Loss: 1.473482
Train Epoch: 9 [320/864 (37%)] Loss: 2.595198
Train Epoch: 9 [400/864 (46%)] Loss: 1.369501
Train Epoch: 9 [480/864 (56%)] Loss: 1.631327
Train Epoch: 9 [560/864 (65%)] Loss: 1.226314
Train Epoch: 9 [640/864 (74%)] Loss: 1.154537
Train Epoch: 9 [720/864 (83%)] Loss: 1.192116
Train Epoch: 9 [800/864 (93%)] Loss: 1.420991
Test set: Average loss: 1.3503, Accuracy: 387/864 (45%)
```

Test set: Average loss: 1.3352, Accuracy: 97/216 (45%)

```
Train Epoch: 10 [0/864 (0%)] Loss: 1.845751
Train Epoch: 10 [80/864 (9%)] Loss: 1.435316
Train Epoch: 10 [160/864 (19%)] Loss: 1.120949
Train Epoch: 10 [240/864 (28%)] Loss: 1.166934
Train Epoch: 10 [320/864 (37%)] Loss: 1.831757
Train Epoch: 10 [400/864 (46%)] Loss: 1.189580
Train Epoch: 10 [480/864 (56%)] Loss: 1.773858
Train Epoch: 10 [560/864 (65%)] Loss: 1.431038
Train Epoch: 10 [640/864 (74%)] Loss: 1.380459
```

```
Train Epoch: 10 [720/864 (83%)] Loss: 1.280839
Train Epoch: 10 [800/864 (93%)] Loss: 1.427181
Test set: Average loss: 1.3371, Accuracy: 400/864 (46%)
```

```
Test set: Average loss: 1.3534, Accuracy: 97/216 (45%)
```

```
Train Epoch: 11 [0/864 (0%)] Loss: 1.079228
Train Epoch: 11 [80/864 (9%)] Loss: 1.378886
Train Epoch: 11 [160/864 (19%)] Loss: 1.461481
Train Epoch: 11 [240/864 (28%)] Loss: 1.554431
Train Epoch: 11 [320/864 (37%)] Loss: 1.528032
Train Epoch: 11 [400/864 (46%)] Loss: 1.136062
Train Epoch: 11 [480/864 (56%)] Loss: 1.388611
Train Epoch: 11 [560/864 (65%)] Loss: 2.167915
Train Epoch: 11 [640/864 (74%)] Loss: 1.290220
Train Epoch: 11 [720/864 (83%)] Loss: 1.320116
Train Epoch: 11 [800/864 (93%)] Loss: 1.496343
Test set: Average loss: 1.3225, Accuracy: 408/864 (47%)
```

```
Test set: Average loss: 1.3552, Accuracy: 100/216 (46%)
```

```
Train Epoch: 12 [0/864 (0%)] Loss: 1.782002
Train Epoch: 12 [80/864 (9%)] Loss: 1.470469
Train Epoch: 12 [160/864 (19%)] Loss: 1.477746
Train Epoch: 12 [240/864 (28%)] Loss: 1.354188
Train Epoch: 12 [320/864 (37%)] Loss: 1.140474
Train Epoch: 12 [400/864 (46%)] Loss: 1.991795
Train Epoch: 12 [480/864 (56%)] Loss: 1.282259
Train Epoch: 12 [560/864 (65%)] Loss: 1.753293
Train Epoch: 12 [640/864 (74%)] Loss: 1.484847
Train Epoch: 12 [720/864 (83%)] Loss: 1.320451
Train Epoch: 12 [800/864 (93%)] Loss: 1.002370
Test set: Average loss: 1.3311, Accuracy: 398/864 (46%)
```

```
Test set: Average loss: 1.3716, Accuracy: 100/216 (46%)
```

```
Train Epoch: 13 [0/864 (0%)] Loss: 1.464574
Train Epoch: 13 [80/864 (9%)] Loss: 1.702637
Train Epoch: 13 [160/864 (19%)] Loss: 1.014907
Train Epoch: 13 [240/864 (28%)] Loss: 1.863691
Train Epoch: 13 [320/864 (37%)] Loss: 1.662147
Train Epoch: 13 [400/864 (46%)] Loss: 1.318960
Train Epoch: 13 [480/864 (56%)] Loss: 1.675079
Train Epoch: 13 [560/864 (65%)] Loss: 1.111277
Train Epoch: 13 [640/864 (74%)] Loss: 0.796897
Train Epoch: 13 [720/864 (83%)] Loss: 1.273110
Train Epoch: 13 [800/864 (93%)] Loss: 1.004088
Test set: Average loss: 1.3164, Accuracy: 407/864 (47%)
```

Test set: Average loss: 1.3497, Accuracy: 94/216 (44%)

```
Train Epoch: 14 [0/864 (0%)] Loss: 1.730787
Train Epoch: 14 [80/864 (9%)] Loss: 2.306227
Train Epoch: 14 [160/864 (19%)] Loss: 1.459943
Train Epoch: 14 [240/864 (28%)] Loss: 2.035325
Train Epoch: 14 [320/864 (37%)] Loss: 1.352701
Train Epoch: 14 [400/864 (46%)] Loss: 1.560421
Train Epoch: 14 [480/864 (56%)] Loss: 1.064653
Train Epoch: 14 [560/864 (65%)] Loss: 1.684895
Train Epoch: 14 [640/864 (74%)] Loss: 1.285944
Train Epoch: 14 [720/864 (83%)] Loss: 1.258256
Train Epoch: 14 [800/864 (93%)] Loss: 1.362808
Test set: Average loss: 1.2958, Accuracy: 400/864 (46%)
```

Test set: Average loss: 1.3284, Accuracy: 97/216 (45%)

```
Train Epoch: 15 [0/864 (0%)] Loss: 1.639161
Train Epoch: 15 [80/864 (9%)] Loss: 2.050541
Train Epoch: 15 [160/864 (19%)] Loss: 1.356549
Train Epoch: 15 [240/864 (28%)] Loss: 1.764887
Train Epoch: 15 [320/864 (37%)] Loss: 1.287584
Train Epoch: 15 [400/864 (46%)] Loss: 1.566497
Train Epoch: 15 [480/864 (56%)] Loss: 1.379402
Train Epoch: 15 [560/864 (65%)] Loss: 1.678130
Train Epoch: 15 [640/864 (74%)] Loss: 1.562755
Train Epoch: 15 [720/864 (83%)] Loss: 1.432615
Train Epoch: 15 [800/864 (93%)] Loss: 1.403230
Test set: Average loss: 1.3133, Accuracy: 394/864 (46%)
```

Test set: Average loss: 1.3535, Accuracy: 99/216 (46%)

```
Train Epoch: 16 [0/864 (0%)] Loss: 1.096343
Train Epoch: 16 [80/864 (9%)] Loss: 1.735525
Train Epoch: 16 [160/864 (19%)] Loss: 1.207095
Train Epoch: 16 [240/864 (28%)] Loss: 1.599390
Train Epoch: 16 [320/864 (37%)] Loss: 0.992145
Train Epoch: 16 [400/864 (46%)] Loss: 1.068582
Train Epoch: 16 [480/864 (56%)] Loss: 1.544440
Train Epoch: 16 [560/864 (65%)] Loss: 0.967910
Train Epoch: 16 [640/864 (74%)] Loss: 1.822186
Train Epoch: 16 [720/864 (83%)] Loss: 1.721234
Train Epoch: 16 [800/864 (93%)] Loss: 1.536056
Test set: Average loss: 1.3110, Accuracy: 396/864 (46%)
```

Test set: Average loss: 1.3351, Accuracy: 97/216 (45%)

```
Train Epoch: 17 [0/864 (0%)] Loss: 1.115463
Train Epoch: 17 [80/864 (9%)] Loss: 1.061137
Train Epoch: 17 [160/864 (19%)] Loss: 1.296701
Train Epoch: 17 [240/864 (28%)] Loss: 1.151989
Train Epoch: 17 [320/864 (37%)] Loss: 1.402959
Train Epoch: 17 [400/864 (46%)] Loss: 1.008626
Train Epoch: 17 [480/864 (56%)] Loss: 1.192462
Train Epoch: 17 [560/864 (65%)] Loss: 1.744173
Train Epoch: 17 [640/864 (74%)] Loss: 0.981551
Train Epoch: 17 [720/864 (83%)] Loss: 1.432586
Train Epoch: 17 [800/864 (93%)] Loss: 1.194800
Test set: Average loss: 1.3050, Accuracy: 385/864 (45%)
```

Test set: Average loss: 1.3268, Accuracy: 99/216 (46%)

```
Train Epoch: 18 [0/864 (0%)] Loss: 1.163602
Train Epoch: 18 [80/864 (9%)] Loss: 1.441919
Train Epoch: 18 [160/864 (19%)] Loss: 1.787188
Train Epoch: 18 [240/864 (28%)] Loss: 1.079360
Train Epoch: 18 [320/864 (37%)] Loss: 1.119056
Train Epoch: 18 [400/864 (46%)] Loss: 1.672909
Train Epoch: 18 [480/864 (56%)] Loss: 1.282390
Train Epoch: 18 [560/864 (65%)] Loss: 1.053414
Train Epoch: 18 [640/864 (74%)] Loss: 1.132046
Train Epoch: 18 [720/864 (83%)] Loss: 1.522172
Train Epoch: 18 [800/864 (93%)] Loss: 1.539015
Test set: Average loss: 1.3143, Accuracy: 393/864 (45%)
```

Test set: Average loss: 1.3208, Accuracy: 98/216 (45%)

```
Train Epoch: 19 [0/864 (0%)] Loss: 1.072097
Train Epoch: 19 [80/864 (9%)] Loss: 0.999088
Train Epoch: 19 [160/864 (19%)] Loss: 1.031597
Train Epoch: 19 [240/864 (28%)] Loss: 1.362321
Train Epoch: 19 [320/864 (37%)] Loss: 1.374459
Train Epoch: 19 [400/864 (46%)] Loss: 1.678142
Train Epoch: 19 [480/864 (56%)] Loss: 1.509708
Train Epoch: 19 [560/864 (65%)] Loss: 1.191978
Train Epoch: 19 [640/864 (74%)] Loss: 1.229449
Train Epoch: 19 [720/864 (83%)] Loss: 1.261681
Train Epoch: 19 [800/864 (93%)] Loss: 0.914052
Test set: Average loss: 1.2989, Accuracy: 389/864 (45%)
```

Test set: Average loss: 1.3318, Accuracy: 97/216 (45%)

```
Train Epoch: 20 [0/864 (0%)] Loss: 1.184922
Train Epoch: 20 [80/864 (9%)] Loss: 1.284894
Train Epoch: 20 [160/864 (19%)] Loss: 1.734905
```

```
Train Epoch: 20 [240/864 (28%)] Loss: 1.130274
Train Epoch: 20 [320/864 (37%)] Loss: 1.160266
Train Epoch: 20 [400/864 (46%)] Loss: 1.518228
Train Epoch: 20 [480/864 (56%)] Loss: 1.834978
Train Epoch: 20 [560/864 (65%)] Loss: 1.832897
Train Epoch: 20 [640/864 (74%)] Loss: 1.488452
Train Epoch: 20 [720/864 (83%)] Loss: 1.107299
Train Epoch: 20 [800/864 (93%)] Loss: 0.934298
Test set: Average loss: 1.3084, Accuracy: 401/864 (46%)
```

Test set: Average loss: 1.3111, Accuracy: 97/216 (45%)

```
Train Epoch: 21 [0/864 (0%)] Loss: 1.382961
Train Epoch: 21 [80/864 (9%)] Loss: 1.529423
Train Epoch: 21 [160/864 (19%)] Loss: 1.208969
Train Epoch: 21 [240/864 (28%)] Loss: 1.138039
Train Epoch: 21 [320/864 (37%)] Loss: 1.524592
Train Epoch: 21 [400/864 (46%)] Loss: 1.097987
Train Epoch: 21 [480/864 (56%)] Loss: 1.204393
Train Epoch: 21 [560/864 (65%)] Loss: 1.160232
Train Epoch: 21 [640/864 (74%)] Loss: 1.480703
Train Epoch: 21 [720/864 (83%)] Loss: 1.301433
Train Epoch: 21 [800/864 (93%)] Loss: 1.649361
Test set: Average loss: 1.3296, Accuracy: 395/864 (46%)
```

Test set: Average loss: 1.3382, Accuracy: 99/216 (46%)

```
Train Epoch: 22 [0/864 (0%)] Loss: 1.596512
Train Epoch: 22 [80/864 (9%)] Loss: 1.012561
Train Epoch: 22 [160/864 (19%)] Loss: 1.060703
Train Epoch: 22 [240/864 (28%)] Loss: 2.180164
Train Epoch: 22 [320/864 (37%)] Loss: 1.429795
Train Epoch: 22 [400/864 (46%)] Loss: 0.935326
Train Epoch: 22 [480/864 (56%)] Loss: 1.244877
Train Epoch: 22 [560/864 (65%)] Loss: 1.211668
Train Epoch: 22 [640/864 (74%)] Loss: 2.017658
Train Epoch: 22 [720/864 (83%)] Loss: 1.310220
Train Epoch: 22 [800/864 (93%)] Loss: 1.298620
Test set: Average loss: 1.3005, Accuracy: 397/864 (46%)
```

Test set: Average loss: 1.3125, Accuracy: 97/216 (45%)

```
Train Epoch: 23 [0/864 (0%)] Loss: 1.653414
Train Epoch: 23 [80/864 (9%)] Loss: 0.996565
Train Epoch: 23 [160/864 (19%)] Loss: 1.017189
Train Epoch: 23 [240/864 (28%)] Loss: 1.543460
Train Epoch: 23 [320/864 (37%)] Loss: 0.843443
Train Epoch: 23 [400/864 (46%)] Loss: 1.184456
```

```
Train Epoch: 23 [480/864 (56%)] Loss: 1.691575
Train Epoch: 23 [560/864 (65%)] Loss: 1.136047
Train Epoch: 23 [640/864 (74%)] Loss: 1.213310
Train Epoch: 23 [720/864 (83%)] Loss: 1.421741
Train Epoch: 23 [800/864 (93%)] Loss: 1.651903
Test set: Average loss: 1.2925, Accuracy: 394/864 (46%)
```

Test set: Average loss: 1.3271, Accuracy: 98/216 (45%)

```
Train Epoch: 24 [0/864 (0%)] Loss: 1.284871
Train Epoch: 24 [80/864 (9%)] Loss: 1.145159
Train Epoch: 24 [160/864 (19%)] Loss: 1.251008
Train Epoch: 24 [240/864 (28%)] Loss: 1.120037
Train Epoch: 24 [320/864 (37%)] Loss: 1.738264
Train Epoch: 24 [400/864 (46%)] Loss: 1.068857
Train Epoch: 24 [480/864 (56%)] Loss: 1.079917
Train Epoch: 24 [560/864 (65%)] Loss: 0.904408
Train Epoch: 24 [640/864 (74%)] Loss: 1.370974
Train Epoch: 24 [720/864 (83%)] Loss: 0.895933
Train Epoch: 24 [800/864 (93%)] Loss: 1.260490
Test set: Average loss: 1.2891, Accuracy: 395/864 (46%)
```

Test set: Average loss: 1.3031, Accuracy: 98/216 (45%)

```
Train Epoch: 25 [0/864 (0%)] Loss: 1.143201
Train Epoch: 25 [80/864 (9%)] Loss: 1.114883
Train Epoch: 25 [160/864 (19%)] Loss: 1.224770
Train Epoch: 25 [240/864 (28%)] Loss: 1.536052
Train Epoch: 25 [320/864 (37%)] Loss: 1.334335
Train Epoch: 25 [400/864 (46%)] Loss: 1.102354
Train Epoch: 25 [480/864 (56%)] Loss: 1.750090
Train Epoch: 25 [560/864 (65%)] Loss: 1.430602
Train Epoch: 25 [640/864 (74%)] Loss: 1.405973
Train Epoch: 25 [720/864 (83%)] Loss: 1.371124
Train Epoch: 25 [800/864 (93%)] Loss: 1.073006
Test set: Average loss: 1.2996, Accuracy: 393/864 (45%)
```

Test set: Average loss: 1.3202, Accuracy: 95/216 (44%)

```
Train Epoch: 26 [0/864 (0%)] Loss: 1.337963
Train Epoch: 26 [80/864 (9%)] Loss: 1.218190
Train Epoch: 26 [160/864 (19%)] Loss: 1.378709
Train Epoch: 26 [240/864 (28%)] Loss: 1.146011
Train Epoch: 26 [320/864 (37%)] Loss: 0.877971
Train Epoch: 26 [400/864 (46%)] Loss: 1.733360
Train Epoch: 26 [480/864 (56%)] Loss: 1.417835
Train Epoch: 26 [560/864 (65%)] Loss: 1.800625
Train Epoch: 26 [640/864 (74%)] Loss: 1.257079
```

```
Train Epoch: 26 [720/864 (83%)] Loss: 1.191964
Train Epoch: 26 [800/864 (93%)] Loss: 1.204626
Test set: Average loss: 1.2937, Accuracy: 395/864 (46%)
```

```
Test set: Average loss: 1.3247, Accuracy: 97/216 (45%)
```

```
Train Epoch: 27 [0/864 (0%)] Loss: 1.226477
Train Epoch: 27 [80/864 (9%)] Loss: 1.280884
Train Epoch: 27 [160/864 (19%)] Loss: 1.107193
Train Epoch: 27 [240/864 (28%)] Loss: 1.233109
Train Epoch: 27 [320/864 (37%)] Loss: 1.720418
Train Epoch: 27 [400/864 (46%)] Loss: 1.729921
Train Epoch: 27 [480/864 (56%)] Loss: 1.352268
Train Epoch: 27 [560/864 (65%)] Loss: 0.882496
Train Epoch: 27 [640/864 (74%)] Loss: 1.270927
Train Epoch: 27 [720/864 (83%)] Loss: 1.102807
Train Epoch: 27 [800/864 (93%)] Loss: 1.026173
Test set: Average loss: 1.2858, Accuracy: 394/864 (46%)
```

```
Test set: Average loss: 1.3194, Accuracy: 96/216 (44%)
```

```
Train Epoch: 28 [0/864 (0%)] Loss: 1.636341
Train Epoch: 28 [80/864 (9%)] Loss: 1.058550
Train Epoch: 28 [160/864 (19%)] Loss: 1.622057
Train Epoch: 28 [240/864 (28%)] Loss: 1.131012
Train Epoch: 28 [320/864 (37%)] Loss: 1.188351
Train Epoch: 28 [400/864 (46%)] Loss: 0.874984
Train Epoch: 28 [480/864 (56%)] Loss: 1.144302
Train Epoch: 28 [560/864 (65%)] Loss: 1.614762
Train Epoch: 28 [640/864 (74%)] Loss: 1.011088
Train Epoch: 28 [720/864 (83%)] Loss: 0.974365
Train Epoch: 28 [800/864 (93%)] Loss: 1.025636
Test set: Average loss: 1.2879, Accuracy: 396/864 (46%)
```

```
Test set: Average loss: 1.3201, Accuracy: 95/216 (44%)
```

```
Train Epoch: 29 [0/864 (0%)] Loss: 1.218605
Train Epoch: 29 [80/864 (9%)] Loss: 1.109848
Train Epoch: 29 [160/864 (19%)] Loss: 1.302086
Train Epoch: 29 [240/864 (28%)] Loss: 1.431918
Train Epoch: 29 [320/864 (37%)] Loss: 0.853084
Train Epoch: 29 [400/864 (46%)] Loss: 0.938006
Train Epoch: 29 [480/864 (56%)] Loss: 1.447058
Train Epoch: 29 [560/864 (65%)] Loss: 1.191972
Train Epoch: 29 [640/864 (74%)] Loss: 2.569841
Train Epoch: 29 [720/864 (83%)] Loss: 1.461305
Train Epoch: 29 [800/864 (93%)] Loss: 0.987091
Test set: Average loss: 1.2781, Accuracy: 398/864 (46%)
```

Test set: Average loss: 1.3113, Accuracy: 98/216 (45%)

Train Epoch: 30 [0/864 (0%)] Loss: 1.016550  
Train Epoch: 30 [80/864 (9%)] Loss: 1.341541  
Train Epoch: 30 [160/864 (19%)] Loss: 1.250875  
Train Epoch: 30 [240/864 (28%)] Loss: 1.312182  
Train Epoch: 30 [320/864 (37%)] Loss: 1.023608  
Train Epoch: 30 [400/864 (46%)] Loss: 1.027994  
Train Epoch: 30 [480/864 (56%)] Loss: 1.301787  
Train Epoch: 30 [560/864 (65%)] Loss: 1.051680  
Train Epoch: 30 [640/864 (74%)] Loss: 1.318416  
Train Epoch: 30 [720/864 (83%)] Loss: 1.059525  
Train Epoch: 30 [800/864 (93%)] Loss: 1.280990  
Test set: Average loss: 1.2818, Accuracy: 393/864 (45%)

Test set: Average loss: 1.3167, Accuracy: 96/216 (44%)

Train Epoch: 31 [0/864 (0%)] Loss: 1.179945  
Train Epoch: 31 [80/864 (9%)] Loss: 1.091726  
Train Epoch: 31 [160/864 (19%)] Loss: 1.122880  
Train Epoch: 31 [240/864 (28%)] Loss: 1.368983  
Train Epoch: 31 [320/864 (37%)] Loss: 1.408426  
Train Epoch: 31 [400/864 (46%)] Loss: 1.109519  
Train Epoch: 31 [480/864 (56%)] Loss: 0.974769  
Train Epoch: 31 [560/864 (65%)] Loss: 1.015256  
Train Epoch: 31 [640/864 (74%)] Loss: 1.151330  
Train Epoch: 31 [720/864 (83%)] Loss: 1.257093  
Train Epoch: 31 [800/864 (93%)] Loss: 1.179801  
Test set: Average loss: 1.2731, Accuracy: 399/864 (46%)

Test set: Average loss: 1.2952, Accuracy: 99/216 (46%)

Train Epoch: 32 [0/864 (0%)] Loss: 1.233731  
Train Epoch: 32 [80/864 (9%)] Loss: 1.234777  
Train Epoch: 32 [160/864 (19%)] Loss: 1.251807  
Train Epoch: 32 [240/864 (28%)] Loss: 1.343108  
Train Epoch: 32 [320/864 (37%)] Loss: 2.267174  
Train Epoch: 32 [400/864 (46%)] Loss: 0.909154  
Train Epoch: 32 [480/864 (56%)] Loss: 1.006966  
Train Epoch: 32 [560/864 (65%)] Loss: 1.473215  
Train Epoch: 32 [640/864 (74%)] Loss: 1.416740  
Train Epoch: 32 [720/864 (83%)] Loss: 1.889057  
Train Epoch: 32 [800/864 (93%)] Loss: 1.453809  
Test set: Average loss: 1.2717, Accuracy: 404/864 (47%)

Test set: Average loss: 1.3100, Accuracy: 97/216 (45%)

```
Train Epoch: 33 [0/864 (0%)] Loss: 1.072263
Train Epoch: 33 [80/864 (9%)] Loss: 1.439752
Train Epoch: 33 [160/864 (19%)] Loss: 1.225325
Train Epoch: 33 [240/864 (28%)] Loss: 1.216747
Train Epoch: 33 [320/864 (37%)] Loss: 0.976359
Train Epoch: 33 [400/864 (46%)] Loss: 1.114221
Train Epoch: 33 [480/864 (56%)] Loss: 1.393740
Train Epoch: 33 [560/864 (65%)] Loss: 1.149943
Train Epoch: 33 [640/864 (74%)] Loss: 0.989325
Train Epoch: 33 [720/864 (83%)] Loss: 1.414813
Train Epoch: 33 [800/864 (93%)] Loss: 1.895261
Test set: Average loss: 1.2790, Accuracy: 409/864 (47%)
```

Test set: Average loss: 1.3115, Accuracy: 99/216 (46%)

```
Train Epoch: 34 [0/864 (0%)] Loss: 1.037127
Train Epoch: 34 [80/864 (9%)] Loss: 1.057356
Train Epoch: 34 [160/864 (19%)] Loss: 1.339152
Train Epoch: 34 [240/864 (28%)] Loss: 1.356348
Train Epoch: 34 [320/864 (37%)] Loss: 1.046067
Train Epoch: 34 [400/864 (46%)] Loss: 1.085610
Train Epoch: 34 [480/864 (56%)] Loss: 1.462431
Train Epoch: 34 [560/864 (65%)] Loss: 1.414763
Train Epoch: 34 [640/864 (74%)] Loss: 1.270340
Train Epoch: 34 [720/864 (83%)] Loss: 1.396055
Train Epoch: 34 [800/864 (93%)] Loss: 1.395530
Test set: Average loss: 1.2598, Accuracy: 399/864 (46%)
```

Test set: Average loss: 1.2990, Accuracy: 97/216 (45%)

```
Train Epoch: 35 [0/864 (0%)] Loss: 1.458727
Train Epoch: 35 [80/864 (9%)] Loss: 1.505372
Train Epoch: 35 [160/864 (19%)] Loss: 1.215223
Train Epoch: 35 [240/864 (28%)] Loss: 1.246634
Train Epoch: 35 [320/864 (37%)] Loss: 1.284001
Train Epoch: 35 [400/864 (46%)] Loss: 1.373031
Train Epoch: 35 [480/864 (56%)] Loss: 1.511626
Train Epoch: 35 [560/864 (65%)] Loss: 0.964805
Train Epoch: 35 [640/864 (74%)] Loss: 2.183046
Train Epoch: 35 [720/864 (83%)] Loss: 1.813673
Train Epoch: 35 [800/864 (93%)] Loss: 1.252797
Test set: Average loss: 1.2637, Accuracy: 404/864 (47%)
```

Test set: Average loss: 1.2732, Accuracy: 97/216 (45%)

```
Train Epoch: 36 [0/864 (0%)] Loss: 1.341001
Train Epoch: 36 [80/864 (9%)] Loss: 1.575984
Train Epoch: 36 [160/864 (19%)] Loss: 1.039371
```

```
Train Epoch: 36 [240/864 (28%)] Loss: 1.630593
Train Epoch: 36 [320/864 (37%)] Loss: 1.288849
Train Epoch: 36 [400/864 (46%)] Loss: 1.063427
Train Epoch: 36 [480/864 (56%)] Loss: 1.245236
Train Epoch: 36 [560/864 (65%)] Loss: 1.100206
Train Epoch: 36 [640/864 (74%)] Loss: 2.205206
Train Epoch: 36 [720/864 (83%)] Loss: 1.257340
Train Epoch: 36 [800/864 (93%)] Loss: 1.453465
Test set: Average loss: 1.2851, Accuracy: 397/864 (46%)
```

Test set: Average loss: 1.3210, Accuracy: 94/216 (44%)

```
Train Epoch: 37 [0/864 (0%)] Loss: 1.383440
Train Epoch: 37 [80/864 (9%)] Loss: 0.942218
Train Epoch: 37 [160/864 (19%)] Loss: 1.483813
Train Epoch: 37 [240/864 (28%)] Loss: 1.190874
Train Epoch: 37 [320/864 (37%)] Loss: 1.204611
Train Epoch: 37 [400/864 (46%)] Loss: 1.448608
Train Epoch: 37 [480/864 (56%)] Loss: 1.016123
Train Epoch: 37 [560/864 (65%)] Loss: 1.512648
Train Epoch: 37 [640/864 (74%)] Loss: 1.569131
Train Epoch: 37 [720/864 (83%)] Loss: 1.818055
Train Epoch: 37 [800/864 (93%)] Loss: 0.957214
Test set: Average loss: 1.2929, Accuracy: 396/864 (46%)
```

Test set: Average loss: 1.3285, Accuracy: 94/216 (44%)

```
Train Epoch: 38 [0/864 (0%)] Loss: 1.377250
Train Epoch: 38 [80/864 (9%)] Loss: 1.090276
Train Epoch: 38 [160/864 (19%)] Loss: 1.666580
Train Epoch: 38 [240/864 (28%)] Loss: 1.453249
Train Epoch: 38 [320/864 (37%)] Loss: 0.950091
Train Epoch: 38 [400/864 (46%)] Loss: 1.499230
Train Epoch: 38 [480/864 (56%)] Loss: 1.337368
Train Epoch: 38 [560/864 (65%)] Loss: 1.076520
Train Epoch: 38 [640/864 (74%)] Loss: 0.950393
Train Epoch: 38 [720/864 (83%)] Loss: 1.258301
Train Epoch: 38 [800/864 (93%)] Loss: 1.352594
Test set: Average loss: 1.2897, Accuracy: 391/864 (45%)
```

Test set: Average loss: 1.3212, Accuracy: 96/216 (44%)

```
Train Epoch: 39 [0/864 (0%)] Loss: 1.712719
Train Epoch: 39 [80/864 (9%)] Loss: 1.257202
Train Epoch: 39 [160/864 (19%)] Loss: 1.138534
Train Epoch: 39 [240/864 (28%)] Loss: 1.281910
Train Epoch: 39 [320/864 (37%)] Loss: 1.042250
Train Epoch: 39 [400/864 (46%)] Loss: 1.306417
```

```
Train Epoch: 39 [480/864 (56%)] Loss: 1.100107
Train Epoch: 39 [560/864 (65%)] Loss: 1.874016
Train Epoch: 39 [640/864 (74%)] Loss: 1.410880
Train Epoch: 39 [720/864 (83%)] Loss: 0.863386
Train Epoch: 39 [800/864 (93%)] Loss: 0.952679
Test set: Average loss: 1.2819, Accuracy: 397/864 (46%)
```

Test set: Average loss: 1.3289, Accuracy: 97/216 (45%)

```
Train Epoch: 40 [0/864 (0%)] Loss: 1.637710
Train Epoch: 40 [80/864 (9%)] Loss: 1.441450
Train Epoch: 40 [160/864 (19%)] Loss: 0.998155
Train Epoch: 40 [240/864 (28%)] Loss: 1.334983
Train Epoch: 40 [320/864 (37%)] Loss: 0.950550
Train Epoch: 40 [400/864 (46%)] Loss: 0.998067
Train Epoch: 40 [480/864 (56%)] Loss: 1.220629
Train Epoch: 40 [560/864 (65%)] Loss: 1.083066
Train Epoch: 40 [640/864 (74%)] Loss: 1.943546
Train Epoch: 40 [720/864 (83%)] Loss: 1.519876
Train Epoch: 40 [800/864 (93%)] Loss: 1.702074
Test set: Average loss: 1.2980, Accuracy: 394/864 (46%)
```

Test set: Average loss: 1.3118, Accuracy: 99/216 (46%)

```
Train Epoch: 41 [0/864 (0%)] Loss: 1.926388
Train Epoch: 41 [80/864 (9%)] Loss: 1.555738
Train Epoch: 41 [160/864 (19%)] Loss: 0.942415
Train Epoch: 41 [240/864 (28%)] Loss: 1.421025
Train Epoch: 41 [320/864 (37%)] Loss: 1.225998
Train Epoch: 41 [400/864 (46%)] Loss: 1.633295
Train Epoch: 41 [480/864 (56%)] Loss: 1.273133
Train Epoch: 41 [560/864 (65%)] Loss: 1.464369
Train Epoch: 41 [640/864 (74%)] Loss: 0.913436
Train Epoch: 41 [720/864 (83%)] Loss: 1.696296
Train Epoch: 41 [800/864 (93%)] Loss: 1.050156
Test set: Average loss: 1.2687, Accuracy: 401/864 (46%)
```

Test set: Average loss: 1.3064, Accuracy: 99/216 (46%)

```
Train Epoch: 42 [0/864 (0%)] Loss: 1.457688
Train Epoch: 42 [80/864 (9%)] Loss: 1.330248
Train Epoch: 42 [160/864 (19%)] Loss: 1.312354
Train Epoch: 42 [240/864 (28%)] Loss: 1.005483
Train Epoch: 42 [320/864 (37%)] Loss: 1.417515
Train Epoch: 42 [400/864 (46%)] Loss: 1.068891
Train Epoch: 42 [480/864 (56%)] Loss: 1.072143
Train Epoch: 42 [560/864 (65%)] Loss: 1.391063
Train Epoch: 42 [640/864 (74%)] Loss: 1.125016
```

```
Train Epoch: 42 [720/864 (83%)] Loss: 1.445016
Train Epoch: 42 [800/864 (93%)] Loss: 1.178821
Test set: Average loss: 1.2736, Accuracy: 394/864 (46%)
```

```
Test set: Average loss: 1.3044, Accuracy: 99/216 (46%)
```

```
Train Epoch: 43 [0/864 (0%)] Loss: 1.338542
Train Epoch: 43 [80/864 (9%)] Loss: 1.203204
Train Epoch: 43 [160/864 (19%)] Loss: 1.284897
Train Epoch: 43 [240/864 (28%)] Loss: 1.551943
Train Epoch: 43 [320/864 (37%)] Loss: 1.200897
Train Epoch: 43 [400/864 (46%)] Loss: 1.041782
Train Epoch: 43 [480/864 (56%)] Loss: 1.074599
Train Epoch: 43 [560/864 (65%)] Loss: 1.233359
Train Epoch: 43 [640/864 (74%)] Loss: 1.643083
Train Epoch: 43 [720/864 (83%)] Loss: 1.097064
Train Epoch: 43 [800/864 (93%)] Loss: 2.055210
Test set: Average loss: 1.2752, Accuracy: 403/864 (47%)
```

```
Test set: Average loss: 1.3110, Accuracy: 97/216 (45%)
```

```
Train Epoch: 44 [0/864 (0%)] Loss: 1.124158
Train Epoch: 44 [80/864 (9%)] Loss: 1.369982
Train Epoch: 44 [160/864 (19%)] Loss: 1.326723
Train Epoch: 44 [240/864 (28%)] Loss: 1.559821
Train Epoch: 44 [320/864 (37%)] Loss: 1.158859
Train Epoch: 44 [400/864 (46%)] Loss: 1.528338
Train Epoch: 44 [480/864 (56%)] Loss: 1.328083
Train Epoch: 44 [560/864 (65%)] Loss: 0.977862
Train Epoch: 44 [640/864 (74%)] Loss: 1.267318
Train Epoch: 44 [720/864 (83%)] Loss: 1.049466
Train Epoch: 44 [800/864 (93%)] Loss: 1.856247
Test set: Average loss: 1.2850, Accuracy: 396/864 (46%)
```

```
Test set: Average loss: 1.3035, Accuracy: 95/216 (44%)
```

```
Train Epoch: 45 [0/864 (0%)] Loss: 1.563077
Train Epoch: 45 [80/864 (9%)] Loss: 1.356148
Train Epoch: 45 [160/864 (19%)] Loss: 1.133268
Train Epoch: 45 [240/864 (28%)] Loss: 1.051093
Train Epoch: 45 [320/864 (37%)] Loss: 1.149900
Train Epoch: 45 [400/864 (46%)] Loss: 1.681302
Train Epoch: 45 [480/864 (56%)] Loss: 1.795173
Train Epoch: 45 [560/864 (65%)] Loss: 1.069951
Train Epoch: 45 [640/864 (74%)] Loss: 1.492081
Train Epoch: 45 [720/864 (83%)] Loss: 1.547793
Train Epoch: 45 [800/864 (93%)] Loss: 1.834395
Test set: Average loss: 1.2669, Accuracy: 396/864 (46%)
```

Test set: Average loss: 1.3017, Accuracy: 98/216 (45%)

Train Epoch: 46 [0/864 (0%)] Loss: 1.528476  
Train Epoch: 46 [80/864 (9%)] Loss: 1.141604  
Train Epoch: 46 [160/864 (19%)] Loss: 0.955937  
Train Epoch: 46 [240/864 (28%)] Loss: 1.134751  
Train Epoch: 46 [320/864 (37%)] Loss: 1.163779  
Train Epoch: 46 [400/864 (46%)] Loss: 1.098299  
Train Epoch: 46 [480/864 (56%)] Loss: 1.499730  
Train Epoch: 46 [560/864 (65%)] Loss: 1.860631  
Train Epoch: 46 [640/864 (74%)] Loss: 1.389573  
Train Epoch: 46 [720/864 (83%)] Loss: 0.969483  
Train Epoch: 46 [800/864 (93%)] Loss: 1.230116  
Test set: Average loss: 1.2647, Accuracy: 393/864 (45%)

Test set: Average loss: 1.3108, Accuracy: 99/216 (46%)

Train Epoch: 47 [0/864 (0%)] Loss: 1.522315  
Train Epoch: 47 [80/864 (9%)] Loss: 1.266650  
Train Epoch: 47 [160/864 (19%)] Loss: 1.369256  
Train Epoch: 47 [240/864 (28%)] Loss: 1.833878  
Train Epoch: 47 [320/864 (37%)] Loss: 0.943606  
Train Epoch: 47 [400/864 (46%)] Loss: 1.289729  
Train Epoch: 47 [480/864 (56%)] Loss: 1.326138  
Train Epoch: 47 [560/864 (65%)] Loss: 1.556635  
Train Epoch: 47 [640/864 (74%)] Loss: 1.191510  
Train Epoch: 47 [720/864 (83%)] Loss: 1.541106  
Train Epoch: 47 [800/864 (93%)] Loss: 1.230581  
Test set: Average loss: 1.2557, Accuracy: 404/864 (47%)

Test set: Average loss: 1.2940, Accuracy: 96/216 (44%)

Train Epoch: 48 [0/864 (0%)] Loss: 1.743913  
Train Epoch: 48 [80/864 (9%)] Loss: 1.529886  
Train Epoch: 48 [160/864 (19%)] Loss: 1.135280  
Train Epoch: 48 [240/864 (28%)] Loss: 1.275874  
Train Epoch: 48 [320/864 (37%)] Loss: 1.104076  
Train Epoch: 48 [400/864 (46%)] Loss: 1.228148  
Train Epoch: 48 [480/864 (56%)] Loss: 1.092661  
Train Epoch: 48 [560/864 (65%)] Loss: 1.327121  
Train Epoch: 48 [640/864 (74%)] Loss: 1.662200  
Train Epoch: 48 [720/864 (83%)] Loss: 1.171509  
Train Epoch: 48 [800/864 (93%)] Loss: 1.239546  
Test set: Average loss: 1.2534, Accuracy: 402/864 (47%)

Test set: Average loss: 1.3052, Accuracy: 98/216 (45%)

```
Train Epoch: 49 [0/864 (0%)]    Loss: 0.878574
Train Epoch: 49 [80/864 (9%)]   Loss: 1.634493
Train Epoch: 49 [160/864 (19%)]  Loss: 1.832926
Train Epoch: 49 [240/864 (28%)]  Loss: 1.476380
Train Epoch: 49 [320/864 (37%)]  Loss: 1.231911
Train Epoch: 49 [400/864 (46%)]  Loss: 1.439614
Train Epoch: 49 [480/864 (56%)]  Loss: 1.297813
Train Epoch: 49 [560/864 (65%)]  Loss: 1.301319
Train Epoch: 49 [640/864 (74%)]  Loss: 1.964117
Train Epoch: 49 [720/864 (83%)]  Loss: 1.401947
Train Epoch: 49 [800/864 (93%)]  Loss: 1.439896
Test set: Average loss: 1.2845, Accuracy: 395/864 (46%)
```

```
Test set: Average loss: 1.3281, Accuracy: 96/216 (44%)
```