

Final_Flower_Classification_2025

June 9, 2025

1 Flower Classification Competition (30%)

For this competition, we will use the Flower Recognition. This dataset contains 4317 images of flowers. The data collection is based on the data from flickr, google images, yandex images. The aim is to recognize flower species from a photo.

The pictures are divided into five classes: chamomile, tulip, rose, sunflower, dandelion. For each class there are about 800 photos. Photos are not high resolution, about 320x240 pixels. Note that for this dataset the photos are not all at a fixed size, they have different proportions.

We provide baseline code that includes the following features:

- Loading and Analysing the Flowers dataset using torchvision.
- Providing some augmentations (on loading)
- Defining a simple convolutional neural network.
- How to use existing loss function for the model learning.
- Train the network on the training data.
- Test the trained network on the testing data.
- Generate prediction for the random test image(s).

1.1 The following improvements could be considered:

1. Change of advanced training parameters: Learning Rate, Optimizer, Batch-size, Number of Max Epochs, and Drop-out.
2. Use of a new loss function.
3. Additional/better data augmentation
4. Architectural Changes: Batch Normalization, Residual layers, Attention Block, and other variants.

Your code should be modified from the provided baseline. A pdf report of a maximum of two pages is required to explain the changes you made from the baseline, why you chose those changes, and the improvements they achieved.

1.2 Marking Rules:

We will mark the competition based on the final test accuracy on testing images and your report.

Final mark (out of 50) = acc_mark + efficiency mark + report mark ###Acc_mark 10:

We will rank all the submission results based on their test accuracy. Zero improvement over the baseline yields 0 marks. Maximum improvement over the baseline will yield 10 marks. There will be a sliding scale applied in between.

###Efficiency mark 10:

Efficiency considers not only the accuracy, but the computational cost of running the model (flops: <https://en.wikipedia.org/wiki/FLOPS>). Efficiency for our purposes is defined to be the ratio of accuracy (in %) to Gflops. Please report the computational cost for your final model and include the efficiency calculation in your report. Maximum improvement over the baseline will yield 10 marks. Zero improvement over the baseline yields zero marks, with a sliding scale in between.

###Report mark 30: Your report should comprise: 1. An introduction showing your understanding of the task and of the baseline model: [10 marks]

2. A description of how you have modified aspects of the system to improve performance. [10 marks]

A recommended way to present a summary of this is via an “ablation study” table, eg:

| Method1 | Method2 | Method3 | Accuracy |
|---------|---------|---------|----------|
| N | N | N | 60% |
| Y | N | N | 65% |
| Y | Y | N | 77% |
| Y | Y | Y | 82% |

3. Explanation of the methods for reducing the computational cost and/or improve the trade-off between accuracy and cost: [5 marks]
4. Limitations/Conclusions: [5 marks]

```
[ ]: #####
### Subject: Computer Vision
### Year: 2025
### Student Name: Nethmi Ranathunga, Emily Carey
### Student ID: a1895261, a1851030
### Competition Name: Flowers Classification Competition
### Final Results:
### ACC:89.00%          GFLOPs:4.11058
#####
```

```
[ ]: # Importing libraries.

import torch
import torch.nn as nn
import torch.nn.functional as F
from typing import Optional #added
from torchvision import models #added
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
import random
```

```
# To avoid non-essential warnings
import warnings
warnings.filterwarnings('ignore')

%matplotlib inline
from torch.utils.data import DataLoader
from torchvision import datasets, transforms, models
from torchvision.utils import make_grid
from sklearn.model_selection import train_test_split
```

```
[ ]: # Performing Image Transformations.

train_transform=transforms.Compose([
    transforms.Resize(224),          # resize shortest side to 224 pixels
    transforms.CenterCrop(224),      # crop longest side to 224 pixels
    ↪at center
    # transforms.ColorJitter(brightness=0.4, contrast=0.4, saturation=0.4,
    ↪hue=0.1),
    transforms.RandomRotation(15),
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406],
                        [0.229, 0.224, 0.225])
])
```

```
[ ]: # Mounting G-Drive to get your dataset.
# To access Google Colab GPU; Go To: Edit >>> Network Settings >>> Hardware
    ↪Accelerator: Select GPU.
# Reference: https://towardsdatascience.com/google-colab-import-and-export-datasets-eccf801e2971
    ↪google-colab-import-and-export-datasets-eccf801e2971
from google.colab import drive
drive.mount('/content/drive')

# Dataset path. Ensure that the file path correspond to the path you have here.
    ↪It is expected that you unzip the data folders before running the notebook.
data_directory = '/content/drive/MyDrive/Datasets/flower/flowers'
dataset=datasets.ImageFolder(root=data_directory,transform=train_transform)
dataset
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call `drive.mount("/content/drive", force_remount=True)`.

```
[ ]: Dataset ImageFolder
      Number of datapoints: 4317
      Root location: /content/drive/MyDrive/Datasets/flower/flowers
      StandardTransform
      Transform: Compose(
          Resize(size=224, interpolation=bilinear, max_size=None,
```

```

antialias=True)
        CenterCrop(size=(224, 224))
        RandomRotation(degrees=[-15.0, 15.0], interpolation=nearest,
expand=False, fill=0)
        ToTensor()
        Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
    )

```

```

[ ]: # Checking the flower class types.
class_names=dataset.classes
print(class_names)
print(len(class_names))

```

```

['daisy', 'dandelion', 'rose', 'sunflower', 'tulip']
5

```

```

[ ]: # Train and Test data split.
train_indices, test_indices = train_test_split(list(range(len(dataset.
    ↪targets))), test_size=0.2, stratify=dataset.targets)
train_data = torch.utils.data.Subset(dataset, train_indices)
test_data = torch.utils.data.Subset(dataset, test_indices)

```

```

[ ]: # To check whether Google Colab GPU has been assigned/not.
torch.cuda.is_available()
def get_default_device():
    """Pick GPU if available, else CPU"""
    if torch.cuda.is_available():
        return torch.device('cuda')
    else:
        return None
device = get_default_device()
device

```

```

[ ]: device(type='cuda')

```

```

[ ]: train_loader=DataLoader(train_data,batch_size=8,shuffle=True)
test_loader=DataLoader(test_data,batch_size=8)

```

```

[ ]: def to_device(data, device):
    """Move tensor(s) to chosen device"""
    if isinstance(data, (list,tuple)):
        return [to_device(x, device) for x in data]
    return data.to(device, non_blocking=True)

class DeviceDataLoader():
    """Wrap a dataloader to move data to a device"""
    def __init__(self, dl, device):
        self.dl = dl

```

```

        self.device = device

    def __iter__(self):
        """Yield a batch of data after moving it to device"""
        for b in self.dl:
            yield to_device(b, self.device)

    def __len__(self):
        """Number of batches"""
        return len(self.dl)

```

```
[ ]: print(len(train_data))
      print(len(test_data))
```

```

3453
864

```

```
[ ]: # Preview of the datasets.
      for images, labels in train_loader:
          break
      #print the labels
      print('Label:', labels.numpy())
      print('Class:', *np.array([class_names[i] for i in labels]))

      im=make_grid(images,nrow=5)

```

```

Label: [2 2 0 4 4 0 1 4]
Class: rose rose daisy tulip tulip daisy dandelion tulip

```

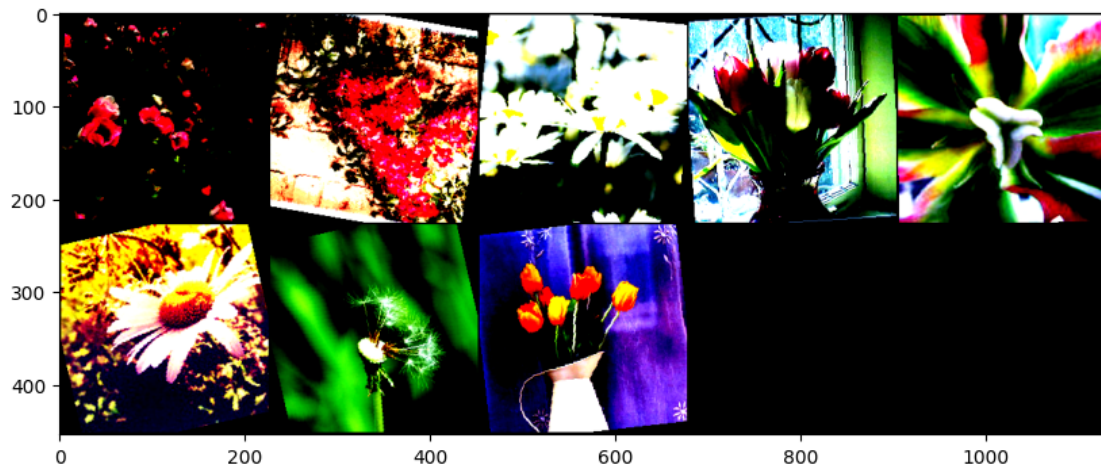
```
[ ]: plt.figure(figsize=(10,10))
      plt.imshow(np.transpose(im.numpy(),(1,2,0)))
```

```

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers). Got range
[-2.117904..2.64].

```

```
[ ]: <matplotlib.image.AxesImage at 0x7d30cdcb0650>
```



```
[ ]: # Inverse Normalization.
inv_normalize=transforms.Normalize(mean=[-0.485/0.229,-0.456/0.224,-0.406/0.
↪225],
                                std=[1/0.229,1/0.224,1/0.225])
im=inv_normalize(im)
```

```
[ ]: plt.figure(figsize=(10,10))
plt.imshow(np.transpose(im.numpy(),(1,2,0)))
```

```
[ ]: <matplotlib.image.AxesImage at 0x7d30cc14b650>
```



```
[ ]: #ADDED
# define focal loss class
```

```

class FocalLoss(nn.Module):
    def __init__(self,
                  alpha: Optional[torch.Tensor] = None,
                  gamma: float = 2.0,
                  reduction: str = 'mean',
                  ignore_index: int = -100):
        super().__init__()
        self.alpha = alpha
        self.gamma = gamma
        self.ignore_index = ignore_index
        self.reduction = reduction
        self.nll_loss = nn.NLLLoss(weight=alpha, reduction='none',
        ↪ ignore_index=ignore_index)

    def forward(self, x, y):
        log_pt = F.log_softmax(x, dim=1)
        pt = torch.exp(log_pt)
        log_pt = log_pt.gather(1, y.view(-1, 1)).squeeze(1)
        pt = pt.gather(1, y.view(-1, 1)).squeeze(1)
        loss = -self.alpha[y] * (1 - pt) ** self.gamma * log_pt

        if self.reduction == 'mean':
            return loss.mean()
        elif self.reduction == 'sum':
            return loss.sum()
        else:
            return loss

```

[]: *# #ADDED - architectural change - residual linear block*

```

class ResidualLinearBlock(nn.Module):
    def __init__(self, input_dim, hidden_dim, output_dim):
        super().__init__()
        self.block = nn.Sequential(
            nn.Linear(input_dim, hidden_dim),
            nn.BatchNorm1d(hidden_dim),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(hidden_dim, output_dim)
        )
        self.shortcut = nn.Linear(input_dim, output_dim)

    def forward(self, x):
        return self.block(x) + self.shortcut(x)

```

[]: `class ResNetFlowerModel(nn.Module):`
`def __init__(self, num_classes):`
`super().__init__()`

```

base_model = models.resnet50(pretrained=True)
base_model = base_model.to(device)

# Freeze pretrained layers
for param in base_model.parameters():
    param.requires_grad = False

# Replace final FC layer with custom residual block
base_model.fc = ResidualLinearBlock(2048, 500, num_classes)
self.network = base_model

def forward(self, xb):
    return self.network(xb)

def training_step(self, batch):
    images, name = batch
    out = self(images)
    return F.cross_entropy(out, name)

def validation_step(self, batch):
    images, name = batch
    out = self(images)
    loss = F.cross_entropy(out, name)
    acc = (torch.argmax(out, dim=1) == labels).float().mean()
    return {'val_loss': loss.detach(), 'val_acc': acc}

def validation_epoch_end(self, outputs):
    batch_losses = [x['val_loss'] for x in outputs]
    batch_accs = [x['val_acc'] for x in outputs]
    return {
        'val_loss': torch.stack(batch_losses).mean().item(),
        'val_acc': torch.stack(batch_accs).mean().item()
    }

def epoch_end(self, epoch, result): #
    print(f"Epoch [{epoch+1}], train_loss: {result['train_loss']:.4f}, "
          f"val_loss: {result['val_loss']:.4f}, val_acc: {result['val_acc']:.4f}")

```

```

[ ]: import time
import torch.optim as optim

```

```

start_time = time.time()
model = ResNetFlowerModel(num_classes=5).to(device)

```



```

optimizer = optim.SGD(model.network.fc.parameters(), lr=0.001, momentum=0.9)
criterion = FocalLoss(alpha=torch.tensor([1, 1, 2, 1, 1], dtype=torch.float32,
↪device=device), gamma=2)

train_losses = []
test_losses = []
train_correct = []
test_correct = []

epochs = 30

for i in range(epochs):
    trn_corr = 0
    tst_corr = 0
    model.train()
    for b, (X_train, y_train) in enumerate(train_loader):
        b+=1
        X_train = X_train.to(device)
        y_train = y_train.to(device)

        y_pred = model(X_train)
        loss = criterion(y_pred, y_train)

        predicted = torch.max(y_pred.data, 1)[1]
        batch_corr = (predicted == y_train).sum()
        trn_corr += batch_corr

        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        if b%10==0:
            print(f"epoch: {i} loss: {loss.item():.4f} batch: {b} accuracy:
↪{trn_corr.item()*100/(b*batch_size):7.3f}%")

    train_losses.append(loss.item())
    train_correct.append(trn_corr.item() / len(train_loader.dataset))
    model.eval()
    with torch.no_grad():
        for b, (X_test, y_test) in enumerate(test_loader):
            X_test = X_test.to(device)
            y_test = y_test.to(device)

            y_val = model(X_test)

```

```

        loss = criterion(y_val, y_test)

        predicted = torch.max(y_val.data, 1)[1]
        batch_corr = (predicted == y_test).sum()
        tst_corr += batch_corr

    loss = loss.detach().cpu().numpy()
    test_losses.append(loss)
    test_correct.append(tst_corr.cpu().numpy())
    print(f'epoch: {i}, test accuracy: {tst_corr*100/len(test_data):7.3f}%')
    print(f'\nDuration: {time.time() - start_time:.0f} seconds')

```

```

epoch: 0 loss: 0.9561 batch: 10 accuracy: 26.250%
epoch: 0 loss: 0.5221 batch: 20 accuracy: 35.000%
epoch: 0 loss: 1.7720 batch: 30 accuracy: 43.333%
epoch: 0 loss: 0.2596 batch: 40 accuracy: 49.375%
epoch: 0 loss: 0.4310 batch: 50 accuracy: 51.250%
epoch: 0 loss: 1.2876 batch: 60 accuracy: 53.125%
epoch: 0 loss: 0.5325 batch: 70 accuracy: 54.286%
epoch: 0 loss: 0.1850 batch: 80 accuracy: 57.031%
epoch: 0 loss: 0.3536 batch: 90 accuracy: 59.306%
epoch: 0 loss: 0.3189 batch: 100 accuracy: 60.250%
epoch: 0 loss: 1.1438 batch: 110 accuracy: 61.477%
epoch: 0 loss: 1.1100 batch: 120 accuracy: 61.979%
epoch: 0 loss: 0.4181 batch: 130 accuracy: 63.365%
epoch: 0 loss: 0.4713 batch: 140 accuracy: 64.018%
epoch: 0 loss: 1.0556 batch: 150 accuracy: 64.000%
epoch: 0 loss: 0.4434 batch: 160 accuracy: 64.219%
epoch: 0 loss: 0.2592 batch: 170 accuracy: 65.368%
epoch: 0 loss: 0.0432 batch: 180 accuracy: 65.764%
epoch: 0 loss: 0.0937 batch: 190 accuracy: 65.987%
epoch: 0 loss: 0.6471 batch: 200 accuracy: 66.312%
epoch: 0 loss: 0.2548 batch: 210 accuracy: 66.667%
epoch: 0 loss: 0.4985 batch: 220 accuracy: 66.932%
epoch: 0 loss: 0.1213 batch: 230 accuracy: 67.011%
epoch: 0 loss: 0.3971 batch: 240 accuracy: 67.188%
epoch: 0 loss: 1.4414 batch: 250 accuracy: 67.100%
epoch: 0 loss: 0.2103 batch: 260 accuracy: 67.356%
epoch: 0 loss: 0.8730 batch: 270 accuracy: 67.546%
epoch: 0 loss: 0.3277 batch: 280 accuracy: 67.455%
epoch: 0 loss: 0.2531 batch: 290 accuracy: 67.716%
epoch: 0 loss: 0.0301 batch: 300 accuracy: 67.875%
epoch: 0 loss: 0.0053 batch: 310 accuracy: 68.306%
epoch: 0 loss: 0.1846 batch: 320 accuracy: 68.477%
epoch: 0 loss: 0.1531 batch: 330 accuracy: 68.636%
epoch: 0 loss: 0.9014 batch: 340 accuracy: 68.934%
epoch: 0 loss: 1.4089 batch: 350 accuracy: 69.036%
epoch: 0 loss: 0.3719 batch: 360 accuracy: 69.236%

```

epoch: 0 loss: 0.6836 batch: 370 accuracy: 69.358%
 epoch: 0 loss: 0.1723 batch: 380 accuracy: 69.309%
 epoch: 0 loss: 0.2618 batch: 390 accuracy: 69.327%
 epoch: 0 loss: 1.7125 batch: 400 accuracy: 69.375%
 epoch: 0 loss: 0.2328 batch: 410 accuracy: 69.573%
 epoch: 0 loss: 1.1706 batch: 420 accuracy: 69.554%
 epoch: 0 loss: 0.4792 batch: 430 accuracy: 69.680%
 epoch: 0, test accuracy: 79.630%
 epoch: 1 loss: 0.1869 batch: 10 accuracy: 73.750%
 epoch: 1 loss: 0.0354 batch: 20 accuracy: 77.500%
 epoch: 1 loss: 0.2220 batch: 30 accuracy: 76.667%
 epoch: 1 loss: 0.1267 batch: 40 accuracy: 76.875%
 epoch: 1 loss: 0.5832 batch: 50 accuracy: 75.750%
 epoch: 1 loss: 0.1138 batch: 60 accuracy: 76.042%
 epoch: 1 loss: 0.0220 batch: 70 accuracy: 77.500%
 epoch: 1 loss: 0.3501 batch: 80 accuracy: 76.719%
 epoch: 1 loss: 0.2233 batch: 90 accuracy: 75.972%
 epoch: 1 loss: 0.0703 batch: 100 accuracy: 76.250%
 epoch: 1 loss: 0.1751 batch: 110 accuracy: 76.364%
 epoch: 1 loss: 0.0413 batch: 120 accuracy: 76.354%
 epoch: 1 loss: 0.0401 batch: 130 accuracy: 76.442%
 epoch: 1 loss: 0.2464 batch: 140 accuracy: 76.875%
 epoch: 1 loss: 0.1168 batch: 150 accuracy: 76.667%
 epoch: 1 loss: 0.0264 batch: 160 accuracy: 76.875%
 epoch: 1 loss: 0.4786 batch: 170 accuracy: 76.618%
 epoch: 1 loss: 0.0404 batch: 180 accuracy: 76.597%
 epoch: 1 loss: 0.2228 batch: 190 accuracy: 76.184%
 epoch: 1 loss: 1.3322 batch: 200 accuracy: 76.188%
 epoch: 1 loss: 0.4687 batch: 210 accuracy: 75.893%
 epoch: 1 loss: 0.5110 batch: 220 accuracy: 75.455%
 epoch: 1 loss: 0.5862 batch: 230 accuracy: 75.707%
 epoch: 1 loss: 1.1444 batch: 240 accuracy: 75.781%
 epoch: 1 loss: 1.2021 batch: 250 accuracy: 75.500%
 epoch: 1 loss: 0.2749 batch: 260 accuracy: 75.817%
 epoch: 1 loss: 0.2373 batch: 270 accuracy: 75.787%
 epoch: 1 loss: 0.1518 batch: 280 accuracy: 76.027%
 epoch: 1 loss: 0.3791 batch: 290 accuracy: 75.948%
 epoch: 1 loss: 0.1814 batch: 300 accuracy: 76.250%
 epoch: 1 loss: 0.3373 batch: 310 accuracy: 76.048%
 epoch: 1 loss: 0.1422 batch: 320 accuracy: 76.250%
 epoch: 1 loss: 0.2164 batch: 330 accuracy: 76.364%
 epoch: 1 loss: 0.3432 batch: 340 accuracy: 76.397%
 epoch: 1 loss: 0.2491 batch: 350 accuracy: 76.679%
 epoch: 1 loss: 0.3181 batch: 360 accuracy: 76.840%
 epoch: 1 loss: 0.2356 batch: 370 accuracy: 76.824%
 epoch: 1 loss: 0.2572 batch: 380 accuracy: 76.612%
 epoch: 1 loss: 0.9558 batch: 390 accuracy: 76.603%
 epoch: 1 loss: 0.9215 batch: 400 accuracy: 76.375%

epoch: 1 loss: 0.1983 batch: 410 accuracy: 76.372%
 epoch: 1 loss: 1.0086 batch: 420 accuracy: 76.339%
 epoch: 1 loss: 0.3146 batch: 430 accuracy: 76.279%
 epoch: 1, test accuracy: 84.722%
 epoch: 2 loss: 0.0554 batch: 10 accuracy: 86.250%
 epoch: 2 loss: 0.7013 batch: 20 accuracy: 78.125%
 epoch: 2 loss: 0.3522 batch: 30 accuracy: 78.333%
 epoch: 2 loss: 0.5349 batch: 40 accuracy: 79.375%
 epoch: 2 loss: 0.0068 batch: 50 accuracy: 79.000%
 epoch: 2 loss: 0.2982 batch: 60 accuracy: 78.958%
 epoch: 2 loss: 1.0170 batch: 70 accuracy: 78.393%
 epoch: 2 loss: 0.3061 batch: 80 accuracy: 77.656%
 epoch: 2 loss: 0.0241 batch: 90 accuracy: 76.667%
 epoch: 2 loss: 0.1519 batch: 100 accuracy: 76.750%
 epoch: 2 loss: 0.2154 batch: 110 accuracy: 76.591%
 epoch: 2 loss: 0.0953 batch: 120 accuracy: 75.938%
 epoch: 2 loss: 0.3298 batch: 130 accuracy: 75.962%
 epoch: 2 loss: 0.5266 batch: 140 accuracy: 76.161%
 epoch: 2 loss: 0.1339 batch: 150 accuracy: 76.583%
 epoch: 2 loss: 1.0017 batch: 160 accuracy: 76.797%
 epoch: 2 loss: 0.1564 batch: 170 accuracy: 77.279%
 epoch: 2 loss: 0.6924 batch: 180 accuracy: 77.222%
 epoch: 2 loss: 0.4171 batch: 190 accuracy: 77.566%
 epoch: 2 loss: 0.7857 batch: 200 accuracy: 77.500%
 epoch: 2 loss: 0.2124 batch: 210 accuracy: 77.262%
 epoch: 2 loss: 0.2131 batch: 220 accuracy: 77.273%
 epoch: 2 loss: 1.4204 batch: 230 accuracy: 77.174%
 epoch: 2 loss: 0.6646 batch: 240 accuracy: 77.448%
 epoch: 2 loss: 0.6860 batch: 250 accuracy: 77.600%
 epoch: 2 loss: 0.0126 batch: 260 accuracy: 77.452%
 epoch: 2 loss: 0.4046 batch: 270 accuracy: 77.315%
 epoch: 2 loss: 0.3245 batch: 280 accuracy: 77.500%
 epoch: 2 loss: 0.6781 batch: 290 accuracy: 77.457%
 epoch: 2 loss: 0.0955 batch: 300 accuracy: 77.542%
 epoch: 2 loss: 0.0004 batch: 310 accuracy: 77.984%
 epoch: 2 loss: 0.1965 batch: 320 accuracy: 77.891%
 epoch: 2 loss: 0.1370 batch: 330 accuracy: 77.879%
 epoch: 2 loss: 1.0120 batch: 340 accuracy: 77.794%
 epoch: 2 loss: 0.5955 batch: 350 accuracy: 78.036%
 epoch: 2 loss: 0.1337 batch: 360 accuracy: 78.160%
 epoch: 2 loss: 0.1738 batch: 370 accuracy: 78.311%
 epoch: 2 loss: 0.0330 batch: 380 accuracy: 78.289%
 epoch: 2 loss: 0.3796 batch: 390 accuracy: 78.269%
 epoch: 2 loss: 0.0262 batch: 400 accuracy: 78.312%
 epoch: 2 loss: 0.4952 batch: 410 accuracy: 78.201%
 epoch: 2 loss: 0.4816 batch: 420 accuracy: 78.363%
 epoch: 2 loss: 0.3169 batch: 430 accuracy: 77.994%
 epoch: 2, test accuracy: 82.986%

epoch: 3 loss: 0.2684 batch: 10 accuracy: 85.000%
 epoch: 3 loss: 0.4050 batch: 20 accuracy: 83.125%
 epoch: 3 loss: 0.4210 batch: 30 accuracy: 81.250%
 epoch: 3 loss: 1.8995 batch: 40 accuracy: 79.062%
 epoch: 3 loss: 0.0376 batch: 50 accuracy: 81.000%
 epoch: 3 loss: 0.0235 batch: 60 accuracy: 81.042%
 epoch: 3 loss: 0.3419 batch: 70 accuracy: 79.643%
 epoch: 3 loss: 0.2536 batch: 80 accuracy: 79.844%
 epoch: 3 loss: 0.4901 batch: 90 accuracy: 79.861%
 epoch: 3 loss: 0.2132 batch: 100 accuracy: 79.750%
 epoch: 3 loss: 0.0037 batch: 110 accuracy: 80.227%
 epoch: 3 loss: 0.3527 batch: 120 accuracy: 80.104%
 epoch: 3 loss: 0.0169 batch: 130 accuracy: 80.865%
 epoch: 3 loss: 0.6347 batch: 140 accuracy: 81.429%
 epoch: 3 loss: 0.8931 batch: 150 accuracy: 81.583%
 epoch: 3 loss: 0.0901 batch: 160 accuracy: 81.406%
 epoch: 3 loss: 0.1349 batch: 170 accuracy: 81.029%
 epoch: 3 loss: 0.0137 batch: 180 accuracy: 81.042%
 epoch: 3 loss: 0.3535 batch: 190 accuracy: 81.118%
 epoch: 3 loss: 0.3593 batch: 200 accuracy: 81.125%
 epoch: 3 loss: 0.4972 batch: 210 accuracy: 81.190%
 epoch: 3 loss: 0.0557 batch: 220 accuracy: 81.136%
 epoch: 3 loss: 0.4353 batch: 230 accuracy: 81.304%
 epoch: 3 loss: 0.0108 batch: 240 accuracy: 81.354%
 epoch: 3 loss: 0.0309 batch: 250 accuracy: 80.950%
 epoch: 3 loss: 0.0046 batch: 260 accuracy: 81.202%
 epoch: 3 loss: 0.5662 batch: 270 accuracy: 81.296%
 epoch: 3 loss: 0.6802 batch: 280 accuracy: 81.205%
 epoch: 3 loss: 0.3940 batch: 290 accuracy: 81.121%
 epoch: 3 loss: 0.1120 batch: 300 accuracy: 81.042%
 epoch: 3 loss: 0.2481 batch: 310 accuracy: 81.089%
 epoch: 3 loss: 0.3448 batch: 320 accuracy: 81.055%
 epoch: 3 loss: 0.0077 batch: 330 accuracy: 81.174%
 epoch: 3 loss: 0.1227 batch: 340 accuracy: 81.176%
 epoch: 3 loss: 1.6356 batch: 350 accuracy: 81.036%
 epoch: 3 loss: 0.1450 batch: 360 accuracy: 80.764%
 epoch: 3 loss: 0.2518 batch: 370 accuracy: 80.608%
 epoch: 3 loss: 0.8252 batch: 380 accuracy: 80.428%
 epoch: 3 loss: 0.2275 batch: 390 accuracy: 80.256%
 epoch: 3 loss: 0.4931 batch: 400 accuracy: 80.250%
 epoch: 3 loss: 0.3859 batch: 410 accuracy: 80.305%
 epoch: 3 loss: 0.8668 batch: 420 accuracy: 80.357%
 epoch: 3 loss: 0.4195 batch: 430 accuracy: 80.116%
 epoch: 3, test accuracy: 79.861%
 epoch: 4 loss: 0.2885 batch: 10 accuracy: 75.000%
 epoch: 4 loss: 0.0310 batch: 20 accuracy: 80.000%
 epoch: 4 loss: 0.2046 batch: 30 accuracy: 78.750%
 epoch: 4 loss: 0.3733 batch: 40 accuracy: 79.375%

epoch: 4 loss: 0.0253 batch: 50 accuracy: 81.000%
 epoch: 4 loss: 0.3799 batch: 60 accuracy: 81.458%
 epoch: 4 loss: 0.8302 batch: 70 accuracy: 80.893%
 epoch: 4 loss: 0.0640 batch: 80 accuracy: 81.719%
 epoch: 4 loss: 0.2064 batch: 90 accuracy: 81.667%
 epoch: 4 loss: 0.0071 batch: 100 accuracy: 81.375%
 epoch: 4 loss: 0.0184 batch: 110 accuracy: 81.364%
 epoch: 4 loss: 0.7620 batch: 120 accuracy: 81.250%
 epoch: 4 loss: 0.2722 batch: 130 accuracy: 80.192%
 epoch: 4 loss: 0.6162 batch: 140 accuracy: 80.089%
 epoch: 4 loss: 0.0147 batch: 150 accuracy: 80.250%
 epoch: 4 loss: 0.6554 batch: 160 accuracy: 80.234%
 epoch: 4 loss: 0.1657 batch: 170 accuracy: 79.853%
 epoch: 4 loss: 0.0018 batch: 180 accuracy: 79.861%
 epoch: 4 loss: 0.8310 batch: 190 accuracy: 79.342%
 epoch: 4 loss: 0.6072 batch: 200 accuracy: 79.875%
 epoch: 4 loss: 0.0768 batch: 210 accuracy: 79.643%
 epoch: 4 loss: 0.2245 batch: 220 accuracy: 79.773%
 epoch: 4 loss: 0.4640 batch: 230 accuracy: 80.000%
 epoch: 4 loss: 0.0614 batch: 240 accuracy: 80.208%
 epoch: 4 loss: 0.0155 batch: 250 accuracy: 80.300%
 epoch: 4 loss: 0.1114 batch: 260 accuracy: 80.721%
 epoch: 4 loss: 0.1190 batch: 270 accuracy: 80.694%
 epoch: 4 loss: 1.3448 batch: 280 accuracy: 80.536%
 epoch: 4 loss: 0.1908 batch: 290 accuracy: 80.302%
 epoch: 4 loss: 0.0179 batch: 300 accuracy: 80.333%
 epoch: 4 loss: 0.0006 batch: 310 accuracy: 80.323%
 epoch: 4 loss: 0.1102 batch: 320 accuracy: 80.312%
 epoch: 4 loss: 0.2990 batch: 330 accuracy: 80.227%
 epoch: 4 loss: 0.3990 batch: 340 accuracy: 80.294%
 epoch: 4 loss: 0.2329 batch: 350 accuracy: 80.357%
 epoch: 4 loss: 0.1211 batch: 360 accuracy: 80.417%
 epoch: 4 loss: 0.7148 batch: 370 accuracy: 80.405%
 epoch: 4 loss: 0.0158 batch: 380 accuracy: 80.691%
 epoch: 4 loss: 0.1719 batch: 390 accuracy: 80.737%
 epoch: 4 loss: 1.6100 batch: 400 accuracy: 80.656%
 epoch: 4 loss: 0.1736 batch: 410 accuracy: 80.640%
 epoch: 4 loss: 0.3191 batch: 420 accuracy: 80.536%
 epoch: 4 loss: 0.0996 batch: 430 accuracy: 80.523%
 epoch: 4, test accuracy: 87.616%
 epoch: 5 loss: 0.4783 batch: 10 accuracy: 80.000%
 epoch: 5 loss: 0.1867 batch: 20 accuracy: 77.500%
 epoch: 5 loss: 0.1783 batch: 30 accuracy: 77.500%
 epoch: 5 loss: 0.0245 batch: 40 accuracy: 76.562%
 epoch: 5 loss: 0.0215 batch: 50 accuracy: 78.250%
 epoch: 5 loss: 0.0835 batch: 60 accuracy: 78.750%
 epoch: 5 loss: 0.2156 batch: 70 accuracy: 78.929%
 epoch: 5 loss: 0.2878 batch: 80 accuracy: 79.062%

epoch: 5 loss: 0.0905 batch: 90 accuracy: 79.861%
 epoch: 5 loss: 0.1830 batch: 100 accuracy: 79.500%
 epoch: 5 loss: 0.0012 batch: 110 accuracy: 80.341%
 epoch: 5 loss: 0.3153 batch: 120 accuracy: 81.042%
 epoch: 5 loss: 0.0021 batch: 130 accuracy: 81.731%
 epoch: 5 loss: 0.0079 batch: 140 accuracy: 81.518%
 epoch: 5 loss: 0.6029 batch: 150 accuracy: 81.667%
 epoch: 5 loss: 0.0031 batch: 160 accuracy: 81.953%
 epoch: 5 loss: 0.1930 batch: 170 accuracy: 82.132%
 epoch: 5 loss: 0.3232 batch: 180 accuracy: 81.458%
 epoch: 5 loss: 0.1034 batch: 190 accuracy: 81.184%
 epoch: 5 loss: 0.0923 batch: 200 accuracy: 81.062%
 epoch: 5 loss: 0.2250 batch: 210 accuracy: 80.655%
 epoch: 5 loss: 0.5948 batch: 220 accuracy: 80.455%
 epoch: 5 loss: 0.0640 batch: 230 accuracy: 80.707%
 epoch: 5 loss: 0.4849 batch: 240 accuracy: 80.521%
 epoch: 5 loss: 0.1734 batch: 250 accuracy: 80.600%
 epoch: 5 loss: 0.3465 batch: 260 accuracy: 80.577%
 epoch: 5 loss: 0.0790 batch: 270 accuracy: 80.880%
 epoch: 5 loss: 0.7413 batch: 280 accuracy: 80.893%
 epoch: 5 loss: 0.6880 batch: 290 accuracy: 80.948%
 epoch: 5 loss: 0.0316 batch: 300 accuracy: 80.917%
 epoch: 5 loss: 0.3449 batch: 310 accuracy: 81.048%
 epoch: 5 loss: 0.0256 batch: 320 accuracy: 81.055%
 epoch: 5 loss: 0.0345 batch: 330 accuracy: 80.985%
 epoch: 5 loss: 0.3152 batch: 340 accuracy: 80.809%
 epoch: 5 loss: 0.2769 batch: 350 accuracy: 80.821%
 epoch: 5 loss: 0.7208 batch: 360 accuracy: 80.868%
 epoch: 5 loss: 0.6965 batch: 370 accuracy: 80.777%
 epoch: 5 loss: 0.0783 batch: 380 accuracy: 80.724%
 epoch: 5 loss: 0.1254 batch: 390 accuracy: 80.705%
 epoch: 5 loss: 0.2808 batch: 400 accuracy: 81.000%
 epoch: 5 loss: 0.1490 batch: 410 accuracy: 80.915%
 epoch: 5 loss: 0.0017 batch: 420 accuracy: 80.744%
 epoch: 5 loss: 0.4880 batch: 430 accuracy: 80.669%
 epoch: 5, test accuracy: 86.690%
 epoch: 6 loss: 0.5381 batch: 10 accuracy: 70.000%
 epoch: 6 loss: 0.9501 batch: 20 accuracy: 77.500%
 epoch: 6 loss: 0.0758 batch: 30 accuracy: 79.583%
 epoch: 6 loss: 0.6791 batch: 40 accuracy: 80.938%
 epoch: 6 loss: 0.2230 batch: 50 accuracy: 81.000%
 epoch: 6 loss: 0.1056 batch: 60 accuracy: 80.417%
 epoch: 6 loss: 1.0331 batch: 70 accuracy: 78.929%
 epoch: 6 loss: 0.0519 batch: 80 accuracy: 79.688%
 epoch: 6 loss: 0.2691 batch: 90 accuracy: 79.583%
 epoch: 6 loss: 0.2805 batch: 100 accuracy: 80.000%
 epoch: 6 loss: 0.3412 batch: 110 accuracy: 79.886%
 epoch: 6 loss: 0.2397 batch: 120 accuracy: 80.417%

epoch: 6 loss: 0.0564 batch: 130 accuracy: 80.865%
 epoch: 6 loss: 0.0099 batch: 140 accuracy: 80.893%
 epoch: 6 loss: 0.3766 batch: 150 accuracy: 81.167%
 epoch: 6 loss: 0.5917 batch: 160 accuracy: 81.250%
 epoch: 6 loss: 0.1378 batch: 170 accuracy: 81.103%
 epoch: 6 loss: 0.1089 batch: 180 accuracy: 81.319%
 epoch: 6 loss: 0.3969 batch: 190 accuracy: 81.447%
 epoch: 6 loss: 0.6431 batch: 200 accuracy: 81.188%
 epoch: 6 loss: 0.1209 batch: 210 accuracy: 81.071%
 epoch: 6 loss: 0.0188 batch: 220 accuracy: 81.250%
 epoch: 6 loss: 0.2138 batch: 230 accuracy: 81.467%
 epoch: 6 loss: 0.9799 batch: 240 accuracy: 81.146%
 epoch: 6 loss: 0.2152 batch: 250 accuracy: 81.050%
 epoch: 6 loss: 0.2901 batch: 260 accuracy: 81.250%
 epoch: 6 loss: 0.9399 batch: 270 accuracy: 81.019%
 epoch: 6 loss: 0.1158 batch: 280 accuracy: 81.295%
 epoch: 6 loss: 0.0724 batch: 290 accuracy: 81.379%
 epoch: 6 loss: 0.0885 batch: 300 accuracy: 81.625%
 epoch: 6 loss: 0.1647 batch: 310 accuracy: 81.694%
 epoch: 6 loss: 0.0036 batch: 320 accuracy: 81.836%
 epoch: 6 loss: 0.3750 batch: 330 accuracy: 81.856%
 epoch: 6 loss: 0.1036 batch: 340 accuracy: 81.765%
 epoch: 6 loss: 0.1251 batch: 350 accuracy: 81.714%
 epoch: 6 loss: 0.0036 batch: 360 accuracy: 81.944%
 epoch: 6 loss: 0.0689 batch: 370 accuracy: 81.993%
 epoch: 6 loss: 0.1312 batch: 380 accuracy: 81.776%
 epoch: 6 loss: 0.0021 batch: 390 accuracy: 81.795%
 epoch: 6 loss: 0.0214 batch: 400 accuracy: 81.750%
 epoch: 6 loss: 0.5211 batch: 410 accuracy: 81.646%
 epoch: 6 loss: 0.4528 batch: 420 accuracy: 81.667%
 epoch: 6 loss: 0.5812 batch: 430 accuracy: 81.686%
 epoch: 6, test accuracy: 84.606%
 epoch: 7 loss: 0.2626 batch: 10 accuracy: 80.000%
 epoch: 7 loss: 0.6299 batch: 20 accuracy: 80.625%
 epoch: 7 loss: 0.3321 batch: 30 accuracy: 81.667%
 epoch: 7 loss: 0.3944 batch: 40 accuracy: 81.875%
 epoch: 7 loss: 0.7721 batch: 50 accuracy: 81.750%
 epoch: 7 loss: 0.2476 batch: 60 accuracy: 82.083%
 epoch: 7 loss: 0.8854 batch: 70 accuracy: 80.893%
 epoch: 7 loss: 0.0984 batch: 80 accuracy: 79.531%
 epoch: 7 loss: 0.0944 batch: 90 accuracy: 79.861%
 epoch: 7 loss: 0.2448 batch: 100 accuracy: 80.250%
 epoch: 7 loss: 0.0015 batch: 110 accuracy: 80.341%
 epoch: 7 loss: 0.0468 batch: 120 accuracy: 80.625%
 epoch: 7 loss: 1.1733 batch: 130 accuracy: 79.904%
 epoch: 7 loss: 0.3026 batch: 140 accuracy: 79.732%
 epoch: 7 loss: 0.1273 batch: 150 accuracy: 80.167%
 epoch: 7 loss: 0.2090 batch: 160 accuracy: 80.312%

epoch: 7 loss: 1.0811 batch: 170 accuracy: 79.926%
 epoch: 7 loss: 0.3961 batch: 180 accuracy: 80.278%
 epoch: 7 loss: 0.0620 batch: 190 accuracy: 80.526%
 epoch: 7 loss: 0.0770 batch: 200 accuracy: 80.625%
 epoch: 7 loss: 0.0606 batch: 210 accuracy: 80.952%
 epoch: 7 loss: 0.0773 batch: 220 accuracy: 81.307%
 epoch: 7 loss: 0.1821 batch: 230 accuracy: 81.467%
 epoch: 7 loss: 0.0497 batch: 240 accuracy: 81.615%
 epoch: 7 loss: 0.4690 batch: 250 accuracy: 81.800%
 epoch: 7 loss: 0.7391 batch: 260 accuracy: 82.212%
 epoch: 7 loss: 0.1064 batch: 270 accuracy: 82.269%
 epoch: 7 loss: 0.4432 batch: 280 accuracy: 82.143%
 epoch: 7 loss: 1.1379 batch: 290 accuracy: 82.241%
 epoch: 7 loss: 0.0363 batch: 300 accuracy: 82.292%
 epoch: 7 loss: 0.1385 batch: 310 accuracy: 82.419%
 epoch: 7 loss: 0.1138 batch: 320 accuracy: 82.500%
 epoch: 7 loss: 0.0209 batch: 330 accuracy: 82.727%
 epoch: 7 loss: 0.4806 batch: 340 accuracy: 82.794%
 epoch: 7 loss: 0.1010 batch: 350 accuracy: 82.893%
 epoch: 7 loss: 0.2941 batch: 360 accuracy: 82.986%
 epoch: 7 loss: 0.0246 batch: 370 accuracy: 83.007%
 epoch: 7 loss: 0.9952 batch: 380 accuracy: 82.862%
 epoch: 7 loss: 0.0039 batch: 390 accuracy: 83.045%
 epoch: 7 loss: 0.5018 batch: 400 accuracy: 83.156%
 epoch: 7 loss: 0.1088 batch: 410 accuracy: 83.293%
 epoch: 7 loss: 0.1877 batch: 420 accuracy: 83.244%
 epoch: 7 loss: 0.7484 batch: 430 accuracy: 83.140%
 epoch: 7, test accuracy: 87.269%
 epoch: 8 loss: 0.1460 batch: 10 accuracy: 86.250%
 epoch: 8 loss: 0.0164 batch: 20 accuracy: 86.875%
 epoch: 8 loss: 0.1908 batch: 30 accuracy: 85.417%
 epoch: 8 loss: 1.5239 batch: 40 accuracy: 85.000%
 epoch: 8 loss: 0.0649 batch: 50 accuracy: 83.250%
 epoch: 8 loss: 1.2337 batch: 60 accuracy: 83.333%
 epoch: 8 loss: 0.0073 batch: 70 accuracy: 83.393%
 epoch: 8 loss: 0.0192 batch: 80 accuracy: 83.281%
 epoch: 8 loss: 0.0318 batch: 90 accuracy: 83.194%
 epoch: 8 loss: 0.0167 batch: 100 accuracy: 83.875%
 epoch: 8 loss: 0.0315 batch: 110 accuracy: 84.091%
 epoch: 8 loss: 0.6350 batch: 120 accuracy: 83.542%
 epoch: 8 loss: 0.5475 batch: 130 accuracy: 83.173%
 epoch: 8 loss: 0.3755 batch: 140 accuracy: 83.571%
 epoch: 8 loss: 0.3172 batch: 150 accuracy: 83.167%
 epoch: 8 loss: 0.2061 batch: 160 accuracy: 83.359%
 epoch: 8 loss: 0.2560 batch: 170 accuracy: 83.750%
 epoch: 8 loss: 1.0179 batch: 180 accuracy: 83.403%
 epoch: 8 loss: 0.6110 batch: 190 accuracy: 83.224%
 epoch: 8 loss: 0.0819 batch: 200 accuracy: 83.375%

epoch: 8 loss: 0.0929 batch: 210 accuracy: 82.976%
 epoch: 8 loss: 0.5976 batch: 220 accuracy: 82.500%
 epoch: 8 loss: 0.2242 batch: 230 accuracy: 82.283%
 epoch: 8 loss: 0.0185 batch: 240 accuracy: 82.448%
 epoch: 8 loss: 0.3574 batch: 250 accuracy: 82.550%
 epoch: 8 loss: 0.8091 batch: 260 accuracy: 82.260%
 epoch: 8 loss: 0.0918 batch: 270 accuracy: 82.361%
 epoch: 8 loss: 0.0473 batch: 280 accuracy: 82.321%
 epoch: 8 loss: 0.0126 batch: 290 accuracy: 82.371%
 epoch: 8 loss: 0.3017 batch: 300 accuracy: 82.292%
 epoch: 8 loss: 0.2784 batch: 310 accuracy: 82.419%
 epoch: 8 loss: 0.2783 batch: 320 accuracy: 82.422%
 epoch: 8 loss: 0.0744 batch: 330 accuracy: 82.311%
 epoch: 8 loss: 0.0579 batch: 340 accuracy: 82.463%
 epoch: 8 loss: 0.2968 batch: 350 accuracy: 82.393%
 epoch: 8 loss: 0.0121 batch: 360 accuracy: 82.188%
 epoch: 8 loss: 0.1952 batch: 370 accuracy: 82.264%
 epoch: 8 loss: 0.2051 batch: 380 accuracy: 82.138%
 epoch: 8 loss: 1.0711 batch: 390 accuracy: 82.083%
 epoch: 8 loss: 0.0541 batch: 400 accuracy: 81.969%
 epoch: 8 loss: 2.1631 batch: 410 accuracy: 81.921%
 epoch: 8 loss: 0.0198 batch: 420 accuracy: 81.935%
 epoch: 8 loss: 0.2381 batch: 430 accuracy: 81.919%
 epoch: 8, test accuracy: 87.153%
 epoch: 9 loss: 0.0036 batch: 10 accuracy: 83.750%
 epoch: 9 loss: 0.0383 batch: 20 accuracy: 84.375%
 epoch: 9 loss: 0.4857 batch: 30 accuracy: 84.583%
 epoch: 9 loss: 0.4182 batch: 40 accuracy: 85.312%
 epoch: 9 loss: 0.8476 batch: 50 accuracy: 83.500%
 epoch: 9 loss: 0.0328 batch: 60 accuracy: 83.333%
 epoch: 9 loss: 0.3854 batch: 70 accuracy: 83.393%
 epoch: 9 loss: 0.1933 batch: 80 accuracy: 82.969%
 epoch: 9 loss: 0.1548 batch: 90 accuracy: 82.361%
 epoch: 9 loss: 0.2588 batch: 100 accuracy: 83.125%
 epoch: 9 loss: 0.1439 batch: 110 accuracy: 82.727%
 epoch: 9 loss: 0.3769 batch: 120 accuracy: 82.708%
 epoch: 9 loss: 0.0687 batch: 130 accuracy: 82.981%
 epoch: 9 loss: 0.6718 batch: 140 accuracy: 82.857%
 epoch: 9 loss: 0.1137 batch: 150 accuracy: 83.083%
 epoch: 9 loss: 0.0854 batch: 160 accuracy: 82.344%
 epoch: 9 loss: 0.0222 batch: 170 accuracy: 82.279%
 epoch: 9 loss: 0.0558 batch: 180 accuracy: 82.222%
 epoch: 9 loss: 0.4211 batch: 190 accuracy: 82.105%
 epoch: 9 loss: 0.7245 batch: 200 accuracy: 82.250%
 epoch: 9 loss: 0.6115 batch: 210 accuracy: 81.845%
 epoch: 9 loss: 0.6835 batch: 220 accuracy: 81.534%
 epoch: 9 loss: 0.1838 batch: 230 accuracy: 81.793%
 epoch: 9 loss: 0.2602 batch: 240 accuracy: 81.927%

epoch: 9 loss: 0.8495 batch: 250 accuracy: 82.200%
 epoch: 9 loss: 0.4889 batch: 260 accuracy: 82.019%
 epoch: 9 loss: 0.6244 batch: 270 accuracy: 81.898%
 epoch: 9 loss: 0.0880 batch: 280 accuracy: 81.964%
 epoch: 9 loss: 0.1977 batch: 290 accuracy: 82.284%
 epoch: 9 loss: 0.0191 batch: 300 accuracy: 82.250%
 epoch: 9 loss: 0.0011 batch: 310 accuracy: 82.177%
 epoch: 9 loss: 0.3774 batch: 320 accuracy: 82.148%
 epoch: 9 loss: 0.2856 batch: 330 accuracy: 82.311%
 epoch: 9 loss: 0.0467 batch: 340 accuracy: 82.463%
 epoch: 9 loss: 0.2871 batch: 350 accuracy: 82.679%
 epoch: 9 loss: 0.1877 batch: 360 accuracy: 82.743%
 epoch: 9 loss: 0.9017 batch: 370 accuracy: 82.534%
 epoch: 9 loss: 0.2286 batch: 380 accuracy: 82.566%
 epoch: 9 loss: 0.1199 batch: 390 accuracy: 82.436%
 epoch: 9 loss: 0.1957 batch: 400 accuracy: 82.406%
 epoch: 9 loss: 0.7807 batch: 410 accuracy: 82.226%
 epoch: 9 loss: 0.2657 batch: 420 accuracy: 82.292%
 epoch: 9 loss: 0.0225 batch: 430 accuracy: 82.297%
 epoch: 9, test accuracy: 86.111%
 epoch: 10 loss: 0.0217 batch: 10 accuracy: 95.000%
 epoch: 10 loss: 0.1502 batch: 20 accuracy: 91.250%
 epoch: 10 loss: 1.2228 batch: 30 accuracy: 87.500%
 epoch: 10 loss: 0.1741 batch: 40 accuracy: 84.375%
 epoch: 10 loss: 0.0355 batch: 50 accuracy: 85.500%
 epoch: 10 loss: 0.3743 batch: 60 accuracy: 84.583%
 epoch: 10 loss: 0.4468 batch: 70 accuracy: 83.214%
 epoch: 10 loss: 0.2490 batch: 80 accuracy: 82.344%
 epoch: 10 loss: 0.7613 batch: 90 accuracy: 82.500%
 epoch: 10 loss: 1.0933 batch: 100 accuracy: 82.375%
 epoch: 10 loss: 0.0342 batch: 110 accuracy: 82.273%
 epoch: 10 loss: 0.4307 batch: 120 accuracy: 81.979%
 epoch: 10 loss: 0.3511 batch: 130 accuracy: 82.404%
 epoch: 10 loss: 0.0081 batch: 140 accuracy: 82.232%
 epoch: 10 loss: 0.2441 batch: 150 accuracy: 81.667%
 epoch: 10 loss: 0.2185 batch: 160 accuracy: 81.484%
 epoch: 10 loss: 0.0084 batch: 170 accuracy: 81.912%
 epoch: 10 loss: 0.3352 batch: 180 accuracy: 82.083%
 epoch: 10 loss: 0.0436 batch: 190 accuracy: 82.368%
 epoch: 10 loss: 0.1799 batch: 200 accuracy: 82.562%
 epoch: 10 loss: 0.3362 batch: 210 accuracy: 82.560%
 epoch: 10 loss: 0.0072 batch: 220 accuracy: 82.727%
 epoch: 10 loss: 0.0150 batch: 230 accuracy: 82.663%
 epoch: 10 loss: 0.3300 batch: 240 accuracy: 82.604%
 epoch: 10 loss: 0.0047 batch: 250 accuracy: 82.750%
 epoch: 10 loss: 0.4688 batch: 260 accuracy: 82.404%
 epoch: 10 loss: 0.4169 batch: 270 accuracy: 82.546%
 epoch: 10 loss: 0.0120 batch: 280 accuracy: 82.812%

epoch: 10 loss: 0.8349 batch: 290 accuracy: 82.845%
 epoch: 10 loss: 0.4427 batch: 300 accuracy: 82.958%
 epoch: 10 loss: 0.8825 batch: 310 accuracy: 82.863%
 epoch: 10 loss: 0.1802 batch: 320 accuracy: 82.930%
 epoch: 10 loss: 0.1607 batch: 330 accuracy: 82.765%
 epoch: 10 loss: 0.1484 batch: 340 accuracy: 82.610%
 epoch: 10 loss: 0.0259 batch: 350 accuracy: 82.571%
 epoch: 10 loss: 0.2178 batch: 360 accuracy: 82.500%
 epoch: 10 loss: 0.3946 batch: 370 accuracy: 82.568%
 epoch: 10 loss: 0.0433 batch: 380 accuracy: 82.533%
 epoch: 10 loss: 0.3605 batch: 390 accuracy: 82.436%
 epoch: 10 loss: 0.4185 batch: 400 accuracy: 82.469%
 epoch: 10 loss: 0.4611 batch: 410 accuracy: 82.439%
 epoch: 10 loss: 0.8849 batch: 420 accuracy: 82.470%
 epoch: 10 loss: 0.5961 batch: 430 accuracy: 82.529%
 epoch: 10, test accuracy: 86.806%
 epoch: 11 loss: 0.1463 batch: 10 accuracy: 85.000%
 epoch: 11 loss: 0.0040 batch: 20 accuracy: 84.375%
 epoch: 11 loss: 0.0585 batch: 30 accuracy: 84.167%
 epoch: 11 loss: 0.0259 batch: 40 accuracy: 84.062%
 epoch: 11 loss: 0.4138 batch: 50 accuracy: 84.750%
 epoch: 11 loss: 0.1158 batch: 60 accuracy: 84.583%
 epoch: 11 loss: 0.1614 batch: 70 accuracy: 84.464%
 epoch: 11 loss: 0.3631 batch: 80 accuracy: 84.531%
 epoch: 11 loss: 0.3456 batch: 90 accuracy: 83.750%
 epoch: 11 loss: 0.3555 batch: 100 accuracy: 84.250%
 epoch: 11 loss: 0.1057 batch: 110 accuracy: 84.205%
 epoch: 11 loss: 0.2413 batch: 120 accuracy: 83.646%
 epoch: 11 loss: 0.0366 batch: 130 accuracy: 83.365%
 epoch: 11 loss: 0.1438 batch: 140 accuracy: 83.661%
 epoch: 11 loss: 0.0446 batch: 150 accuracy: 83.917%
 epoch: 11 loss: 0.9359 batch: 160 accuracy: 83.594%
 epoch: 11 loss: 0.0016 batch: 170 accuracy: 83.603%
 epoch: 11 loss: 0.2242 batch: 180 accuracy: 84.167%
 epoch: 11 loss: 0.0078 batch: 190 accuracy: 83.750%
 epoch: 11 loss: 0.0744 batch: 200 accuracy: 83.750%
 epoch: 11 loss: 0.5242 batch: 210 accuracy: 83.512%
 epoch: 11 loss: 1.0871 batch: 220 accuracy: 83.523%
 epoch: 11 loss: 0.6417 batch: 230 accuracy: 83.370%
 epoch: 11 loss: 0.0515 batch: 240 accuracy: 83.125%
 epoch: 11 loss: 0.7722 batch: 250 accuracy: 82.950%
 epoch: 11 loss: 0.0769 batch: 260 accuracy: 82.788%
 epoch: 11 loss: 0.0221 batch: 270 accuracy: 82.824%
 epoch: 11 loss: 0.3587 batch: 280 accuracy: 82.545%
 epoch: 11 loss: 0.1581 batch: 290 accuracy: 82.759%
 epoch: 11 loss: 0.5146 batch: 300 accuracy: 82.875%
 epoch: 11 loss: 0.1793 batch: 310 accuracy: 82.823%
 epoch: 11 loss: 0.2438 batch: 320 accuracy: 82.773%

epoch: 11 loss: 0.3447 batch: 330 accuracy: 82.614%
 epoch: 11 loss: 0.1619 batch: 340 accuracy: 82.353%
 epoch: 11 loss: 1.2546 batch: 350 accuracy: 82.429%
 epoch: 11 loss: 0.1633 batch: 360 accuracy: 82.222%
 epoch: 11 loss: 0.0138 batch: 370 accuracy: 82.264%
 epoch: 11 loss: 0.5341 batch: 380 accuracy: 82.171%
 epoch: 11 loss: 0.3838 batch: 390 accuracy: 82.051%
 epoch: 11 loss: 0.1709 batch: 400 accuracy: 82.000%
 epoch: 11 loss: 0.3147 batch: 410 accuracy: 81.921%
 epoch: 11 loss: 0.0897 batch: 420 accuracy: 81.964%
 epoch: 11 loss: 0.9467 batch: 430 accuracy: 81.919%
 epoch: 11, test accuracy: 89.931%
 epoch: 12 loss: 0.9035 batch: 10 accuracy: 85.000%
 epoch: 12 loss: 0.0209 batch: 20 accuracy: 85.000%
 epoch: 12 loss: 0.0030 batch: 30 accuracy: 85.833%
 epoch: 12 loss: 0.6697 batch: 40 accuracy: 85.000%
 epoch: 12 loss: 0.2953 batch: 50 accuracy: 86.000%
 epoch: 12 loss: 0.0096 batch: 60 accuracy: 86.042%
 epoch: 12 loss: 0.4955 batch: 70 accuracy: 86.250%
 epoch: 12 loss: 0.0059 batch: 80 accuracy: 85.312%
 epoch: 12 loss: 1.6799 batch: 90 accuracy: 84.444%
 epoch: 12 loss: 0.4447 batch: 100 accuracy: 84.500%
 epoch: 12 loss: 0.0399 batch: 110 accuracy: 84.545%
 epoch: 12 loss: 0.0064 batch: 120 accuracy: 84.375%
 epoch: 12 loss: 0.2331 batch: 130 accuracy: 84.038%
 epoch: 12 loss: 0.0125 batch: 140 accuracy: 84.018%
 epoch: 12 loss: 0.0147 batch: 150 accuracy: 84.167%
 epoch: 12 loss: 1.1947 batch: 160 accuracy: 84.141%
 epoch: 12 loss: 0.0232 batch: 170 accuracy: 84.118%
 epoch: 12 loss: 0.0749 batch: 180 accuracy: 84.236%
 epoch: 12 loss: 0.0182 batch: 190 accuracy: 84.013%
 epoch: 12 loss: 0.0147 batch: 200 accuracy: 83.875%
 epoch: 12 loss: 0.0109 batch: 210 accuracy: 83.690%
 epoch: 12 loss: 0.8913 batch: 220 accuracy: 83.352%
 epoch: 12 loss: 0.0121 batch: 230 accuracy: 83.750%
 epoch: 12 loss: 0.4747 batch: 240 accuracy: 83.594%
 epoch: 12 loss: 0.3695 batch: 250 accuracy: 83.700%
 epoch: 12 loss: 0.0691 batch: 260 accuracy: 83.798%
 epoch: 12 loss: 0.0394 batch: 270 accuracy: 83.935%
 epoch: 12 loss: 0.0553 batch: 280 accuracy: 83.795%
 epoch: 12 loss: 0.3775 batch: 290 accuracy: 83.448%
 epoch: 12 loss: 0.0227 batch: 300 accuracy: 83.667%
 epoch: 12 loss: 0.0376 batch: 310 accuracy: 83.871%
 epoch: 12 loss: 0.0200 batch: 320 accuracy: 83.906%
 epoch: 12 loss: 0.5109 batch: 330 accuracy: 83.864%
 epoch: 12 loss: 0.2168 batch: 340 accuracy: 83.934%
 epoch: 12 loss: 0.0471 batch: 350 accuracy: 83.893%
 epoch: 12 loss: 0.1893 batch: 360 accuracy: 83.889%

epoch: 12 loss: 0.3613 batch: 370 accuracy: 83.818%
 epoch: 12 loss: 0.0107 batch: 380 accuracy: 83.684%
 epoch: 12 loss: 0.1241 batch: 390 accuracy: 83.622%
 epoch: 12 loss: 0.5082 batch: 400 accuracy: 83.562%
 epoch: 12 loss: 0.1916 batch: 410 accuracy: 83.476%
 epoch: 12 loss: 0.1844 batch: 420 accuracy: 83.304%
 epoch: 12 loss: 0.0406 batch: 430 accuracy: 83.343%
 epoch: 12, test accuracy: 86.806%
 epoch: 13 loss: 0.0770 batch: 10 accuracy: 78.750%
 epoch: 13 loss: 0.1764 batch: 20 accuracy: 83.125%
 epoch: 13 loss: 0.0604 batch: 30 accuracy: 81.667%
 epoch: 13 loss: 0.0625 batch: 40 accuracy: 82.188%
 epoch: 13 loss: 0.0377 batch: 50 accuracy: 83.000%
 epoch: 13 loss: 0.0067 batch: 60 accuracy: 84.167%
 epoch: 13 loss: 0.0507 batch: 70 accuracy: 84.643%
 epoch: 13 loss: 0.0325 batch: 80 accuracy: 85.156%
 epoch: 13 loss: 0.2826 batch: 90 accuracy: 85.139%
 epoch: 13 loss: 0.1231 batch: 100 accuracy: 85.625%
 epoch: 13 loss: 0.0283 batch: 110 accuracy: 85.114%
 epoch: 13 loss: 0.5545 batch: 120 accuracy: 84.792%
 epoch: 13 loss: 0.2072 batch: 130 accuracy: 84.231%
 epoch: 13 loss: 0.0026 batch: 140 accuracy: 83.839%
 epoch: 13 loss: 0.5651 batch: 150 accuracy: 83.667%
 epoch: 13 loss: 0.3978 batch: 160 accuracy: 83.984%
 epoch: 13 loss: 0.5639 batch: 170 accuracy: 84.265%
 epoch: 13 loss: 0.2036 batch: 180 accuracy: 83.958%
 epoch: 13 loss: 0.1897 batch: 190 accuracy: 83.618%
 epoch: 13 loss: 0.1474 batch: 200 accuracy: 83.250%
 epoch: 13 loss: 0.7164 batch: 210 accuracy: 83.631%
 epoch: 13 loss: 0.0665 batch: 220 accuracy: 83.636%
 epoch: 13 loss: 0.2951 batch: 230 accuracy: 83.750%
 epoch: 13 loss: 0.3810 batch: 240 accuracy: 83.906%
 epoch: 13 loss: 0.3626 batch: 250 accuracy: 83.900%
 epoch: 13 loss: 0.0234 batch: 260 accuracy: 84.038%
 epoch: 13 loss: 0.1318 batch: 270 accuracy: 84.028%
 epoch: 13 loss: 0.0127 batch: 280 accuracy: 84.062%
 epoch: 13 loss: 0.0029 batch: 290 accuracy: 84.138%
 epoch: 13 loss: 0.2377 batch: 300 accuracy: 84.292%
 epoch: 13 loss: 0.3130 batch: 310 accuracy: 84.355%
 epoch: 13 loss: 0.0144 batch: 320 accuracy: 84.219%
 epoch: 13 loss: 0.0334 batch: 330 accuracy: 84.432%
 epoch: 13 loss: 0.6677 batch: 340 accuracy: 84.412%
 epoch: 13 loss: 0.0400 batch: 350 accuracy: 84.250%
 epoch: 13 loss: 0.0722 batch: 360 accuracy: 84.271%
 epoch: 13 loss: 0.0656 batch: 370 accuracy: 83.986%
 epoch: 13 loss: 0.5889 batch: 380 accuracy: 83.849%
 epoch: 13 loss: 1.1031 batch: 390 accuracy: 83.878%
 epoch: 13 loss: 0.8281 batch: 400 accuracy: 83.781%

epoch: 13 loss: 0.0544 batch: 410 accuracy: 83.841%
 epoch: 13 loss: 0.0428 batch: 420 accuracy: 84.018%
 epoch: 13 loss: 1.5349 batch: 430 accuracy: 83.953%
 epoch: 13, test accuracy: 89.120%
 epoch: 14 loss: 0.0113 batch: 10 accuracy: 83.750%
 epoch: 14 loss: 0.0431 batch: 20 accuracy: 86.250%
 epoch: 14 loss: 0.0385 batch: 30 accuracy: 87.500%
 epoch: 14 loss: 0.5077 batch: 40 accuracy: 85.312%
 epoch: 14 loss: 0.1279 batch: 50 accuracy: 83.250%
 epoch: 14 loss: 0.3439 batch: 60 accuracy: 82.917%
 epoch: 14 loss: 0.1557 batch: 70 accuracy: 83.750%
 epoch: 14 loss: 0.1769 batch: 80 accuracy: 83.906%
 epoch: 14 loss: 0.9633 batch: 90 accuracy: 84.167%
 epoch: 14 loss: 0.1986 batch: 100 accuracy: 84.000%
 epoch: 14 loss: 0.0528 batch: 110 accuracy: 84.432%
 epoch: 14 loss: 0.7493 batch: 120 accuracy: 84.375%
 epoch: 14 loss: 0.0176 batch: 130 accuracy: 84.231%
 epoch: 14 loss: 0.3359 batch: 140 accuracy: 84.464%
 epoch: 14 loss: 0.0084 batch: 150 accuracy: 84.917%
 epoch: 14 loss: 0.0747 batch: 160 accuracy: 85.312%
 epoch: 14 loss: 0.0992 batch: 170 accuracy: 85.294%
 epoch: 14 loss: 0.2097 batch: 180 accuracy: 85.278%
 epoch: 14 loss: 0.3270 batch: 190 accuracy: 85.461%
 epoch: 14 loss: 0.1642 batch: 200 accuracy: 85.125%
 epoch: 14 loss: 0.4778 batch: 210 accuracy: 85.000%
 epoch: 14 loss: 0.4700 batch: 220 accuracy: 84.830%
 epoch: 14 loss: 0.2692 batch: 230 accuracy: 84.783%
 epoch: 14 loss: 0.0460 batch: 240 accuracy: 84.792%
 epoch: 14 loss: 0.9026 batch: 250 accuracy: 84.400%
 epoch: 14 loss: 0.0311 batch: 260 accuracy: 84.375%
 epoch: 14 loss: 0.1643 batch: 270 accuracy: 84.259%
 epoch: 14 loss: 0.0306 batch: 280 accuracy: 84.330%
 epoch: 14 loss: 0.2250 batch: 290 accuracy: 84.440%
 epoch: 14 loss: 0.0392 batch: 300 accuracy: 84.750%
 epoch: 14 loss: 0.0597 batch: 310 accuracy: 84.556%
 epoch: 14 loss: 0.0878 batch: 320 accuracy: 84.648%
 epoch: 14 loss: 0.0393 batch: 330 accuracy: 84.735%
 epoch: 14 loss: 0.0793 batch: 340 accuracy: 84.779%
 epoch: 14 loss: 0.0879 batch: 350 accuracy: 84.679%
 epoch: 14 loss: 0.4381 batch: 360 accuracy: 84.479%
 epoch: 14 loss: 0.6701 batch: 370 accuracy: 84.358%
 epoch: 14 loss: 0.1939 batch: 380 accuracy: 84.309%
 epoch: 14 loss: 0.6723 batch: 390 accuracy: 84.199%
 epoch: 14 loss: 1.3198 batch: 400 accuracy: 83.969%
 epoch: 14 loss: 0.2111 batch: 410 accuracy: 83.811%
 epoch: 14 loss: 0.1037 batch: 420 accuracy: 83.899%
 epoch: 14 loss: 0.3528 batch: 430 accuracy: 84.041%
 epoch: 14, test accuracy: 89.583%

epoch: 15 loss: 0.5268 batch: 10 accuracy: 73.750%
 epoch: 15 loss: 0.0117 batch: 20 accuracy: 80.625%
 epoch: 15 loss: 0.0032 batch: 30 accuracy: 82.083%
 epoch: 15 loss: 0.3290 batch: 40 accuracy: 81.875%
 epoch: 15 loss: 0.0870 batch: 50 accuracy: 81.750%
 epoch: 15 loss: 0.0112 batch: 60 accuracy: 81.667%
 epoch: 15 loss: 0.3363 batch: 70 accuracy: 81.071%
 epoch: 15 loss: 0.2485 batch: 80 accuracy: 80.938%
 epoch: 15 loss: 0.8160 batch: 90 accuracy: 81.389%
 epoch: 15 loss: 0.4877 batch: 100 accuracy: 81.750%
 epoch: 15 loss: 0.7456 batch: 110 accuracy: 81.705%
 epoch: 15 loss: 0.1853 batch: 120 accuracy: 81.875%
 epoch: 15 loss: 0.6744 batch: 130 accuracy: 82.308%
 epoch: 15 loss: 0.2096 batch: 140 accuracy: 82.143%
 epoch: 15 loss: 0.0581 batch: 150 accuracy: 82.167%
 epoch: 15 loss: 0.0042 batch: 160 accuracy: 82.344%
 epoch: 15 loss: 0.5304 batch: 170 accuracy: 82.500%
 epoch: 15 loss: 0.1066 batch: 180 accuracy: 82.361%
 epoch: 15 loss: 0.1032 batch: 190 accuracy: 82.368%
 epoch: 15 loss: 0.1135 batch: 200 accuracy: 82.000%
 epoch: 15 loss: 0.1362 batch: 210 accuracy: 81.905%
 epoch: 15 loss: 0.3021 batch: 220 accuracy: 81.989%
 epoch: 15 loss: 0.0036 batch: 230 accuracy: 82.174%
 epoch: 15 loss: 0.0144 batch: 240 accuracy: 82.500%
 epoch: 15 loss: 0.1883 batch: 250 accuracy: 82.600%
 epoch: 15 loss: 0.3456 batch: 260 accuracy: 82.548%
 epoch: 15 loss: 0.1804 batch: 270 accuracy: 82.639%
 epoch: 15 loss: 0.3866 batch: 280 accuracy: 82.679%
 epoch: 15 loss: 0.2574 batch: 290 accuracy: 82.759%
 epoch: 15 loss: 0.1857 batch: 300 accuracy: 82.917%
 epoch: 15 loss: 1.2639 batch: 310 accuracy: 83.145%
 epoch: 15 loss: 0.6254 batch: 320 accuracy: 83.047%
 epoch: 15 loss: 0.0786 batch: 330 accuracy: 83.295%
 epoch: 15 loss: 0.0700 batch: 340 accuracy: 83.309%
 epoch: 15 loss: 0.0181 batch: 350 accuracy: 83.393%
 epoch: 15 loss: 0.0863 batch: 360 accuracy: 83.299%
 epoch: 15 loss: 0.2928 batch: 370 accuracy: 83.311%
 epoch: 15 loss: 0.3753 batch: 380 accuracy: 83.421%
 epoch: 15 loss: 0.2395 batch: 390 accuracy: 83.301%
 epoch: 15 loss: 0.3222 batch: 400 accuracy: 83.188%
 epoch: 15 loss: 0.5390 batch: 410 accuracy: 83.232%
 epoch: 15 loss: 0.2010 batch: 420 accuracy: 83.423%
 epoch: 15 loss: 0.0087 batch: 430 accuracy: 83.488%
 epoch: 15, test accuracy: 88.426%
 epoch: 16 loss: 0.0189 batch: 10 accuracy: 92.500%
 epoch: 16 loss: 0.0121 batch: 20 accuracy: 91.250%
 epoch: 16 loss: 0.0340 batch: 30 accuracy: 90.000%
 epoch: 16 loss: 0.0234 batch: 40 accuracy: 89.688%

epoch: 16 loss: 0.6787 batch: 50 accuracy: 86.500%
 epoch: 16 loss: 0.0075 batch: 60 accuracy: 86.042%
 epoch: 16 loss: 0.0138 batch: 70 accuracy: 86.071%
 epoch: 16 loss: 0.2813 batch: 80 accuracy: 86.406%
 epoch: 16 loss: 0.0620 batch: 90 accuracy: 86.528%
 epoch: 16 loss: 0.3112 batch: 100 accuracy: 85.625%
 epoch: 16 loss: 0.0021 batch: 110 accuracy: 85.227%
 epoch: 16 loss: 0.0056 batch: 120 accuracy: 84.479%
 epoch: 16 loss: 1.2051 batch: 130 accuracy: 84.038%
 epoch: 16 loss: 0.2869 batch: 140 accuracy: 84.107%
 epoch: 16 loss: 0.0140 batch: 150 accuracy: 83.417%
 epoch: 16 loss: 0.8337 batch: 160 accuracy: 82.891%
 epoch: 16 loss: 0.0306 batch: 170 accuracy: 83.088%
 epoch: 16 loss: 0.0068 batch: 180 accuracy: 82.639%
 epoch: 16 loss: 0.0201 batch: 190 accuracy: 83.158%
 epoch: 16 loss: 0.3376 batch: 200 accuracy: 83.000%
 epoch: 16 loss: 0.0013 batch: 210 accuracy: 83.155%
 epoch: 16 loss: 0.0102 batch: 220 accuracy: 83.466%
 epoch: 16 loss: 0.0236 batch: 230 accuracy: 83.533%
 epoch: 16 loss: 0.5387 batch: 240 accuracy: 83.333%
 epoch: 16 loss: 0.0370 batch: 250 accuracy: 83.250%
 epoch: 16 loss: 0.0467 batch: 260 accuracy: 83.558%
 epoch: 16 loss: 0.0057 batch: 270 accuracy: 83.657%
 epoch: 16 loss: 0.1651 batch: 280 accuracy: 83.616%
 epoch: 16 loss: 0.0006 batch: 290 accuracy: 83.922%
 epoch: 16 loss: 0.8622 batch: 300 accuracy: 84.000%
 epoch: 16 loss: 0.3340 batch: 310 accuracy: 83.750%
 epoch: 16 loss: 0.3734 batch: 320 accuracy: 83.750%
 epoch: 16 loss: 0.3699 batch: 330 accuracy: 83.826%
 epoch: 16 loss: 0.2139 batch: 340 accuracy: 83.787%
 epoch: 16 loss: 0.3689 batch: 350 accuracy: 83.893%
 epoch: 16 loss: 0.1979 batch: 360 accuracy: 83.889%
 epoch: 16 loss: 0.1375 batch: 370 accuracy: 83.716%
 epoch: 16 loss: 0.1295 batch: 380 accuracy: 83.816%
 epoch: 16 loss: 0.1418 batch: 390 accuracy: 83.942%
 epoch: 16 loss: 0.3186 batch: 400 accuracy: 83.969%
 epoch: 16 loss: 0.0180 batch: 410 accuracy: 84.177%
 epoch: 16 loss: 0.4927 batch: 420 accuracy: 84.107%
 epoch: 16 loss: 0.1827 batch: 430 accuracy: 84.186%
 epoch: 16, test accuracy: 87.616%
 epoch: 17 loss: 0.2137 batch: 10 accuracy: 83.750%
 epoch: 17 loss: 0.0023 batch: 20 accuracy: 86.250%
 epoch: 17 loss: 0.1830 batch: 30 accuracy: 86.250%
 epoch: 17 loss: 0.1056 batch: 40 accuracy: 85.000%
 epoch: 17 loss: 0.5162 batch: 50 accuracy: 83.000%
 epoch: 17 loss: 0.5655 batch: 60 accuracy: 83.125%
 epoch: 17 loss: 0.0067 batch: 70 accuracy: 84.821%
 epoch: 17 loss: 0.0123 batch: 80 accuracy: 86.094%

epoch: 17 loss: 0.0407 batch: 90 accuracy: 86.389%
 epoch: 17 loss: 0.1916 batch: 100 accuracy: 86.125%
 epoch: 17 loss: 0.2566 batch: 110 accuracy: 86.364%
 epoch: 17 loss: 0.4120 batch: 120 accuracy: 86.667%
 epoch: 17 loss: 0.0287 batch: 130 accuracy: 86.635%
 epoch: 17 loss: 0.5629 batch: 140 accuracy: 86.786%
 epoch: 17 loss: 0.6299 batch: 150 accuracy: 86.500%
 epoch: 17 loss: 0.1283 batch: 160 accuracy: 87.031%
 epoch: 17 loss: 0.0693 batch: 170 accuracy: 87.206%
 epoch: 17 loss: 0.4622 batch: 180 accuracy: 87.083%
 epoch: 17 loss: 0.0114 batch: 190 accuracy: 86.776%
 epoch: 17 loss: 0.1095 batch: 200 accuracy: 87.000%
 epoch: 17 loss: 0.6718 batch: 210 accuracy: 86.726%
 epoch: 17 loss: 0.3673 batch: 220 accuracy: 86.705%
 epoch: 17 loss: 0.0042 batch: 230 accuracy: 86.685%
 epoch: 17 loss: 0.0452 batch: 240 accuracy: 86.667%
 epoch: 17 loss: 0.4965 batch: 250 accuracy: 86.800%
 epoch: 17 loss: 0.0340 batch: 260 accuracy: 86.587%
 epoch: 17 loss: 0.3284 batch: 270 accuracy: 86.343%
 epoch: 17 loss: 0.9173 batch: 280 accuracy: 85.938%
 epoch: 17 loss: 0.0722 batch: 290 accuracy: 85.862%
 epoch: 17 loss: 0.4078 batch: 300 accuracy: 85.958%
 epoch: 17 loss: 0.1378 batch: 310 accuracy: 85.766%
 epoch: 17 loss: 0.2510 batch: 320 accuracy: 85.469%
 epoch: 17 loss: 0.3920 batch: 330 accuracy: 85.530%
 epoch: 17 loss: 0.2149 batch: 340 accuracy: 85.478%
 epoch: 17 loss: 0.1796 batch: 350 accuracy: 85.429%
 epoch: 17 loss: 0.0465 batch: 360 accuracy: 85.451%
 epoch: 17 loss: 0.0031 batch: 370 accuracy: 85.439%
 epoch: 17 loss: 1.1407 batch: 380 accuracy: 85.296%
 epoch: 17 loss: 0.0004 batch: 390 accuracy: 85.192%
 epoch: 17 loss: 0.1773 batch: 400 accuracy: 85.219%
 epoch: 17 loss: 0.1602 batch: 410 accuracy: 85.305%
 epoch: 17 loss: 0.1579 batch: 420 accuracy: 85.149%
 epoch: 17 loss: 0.0904 batch: 430 accuracy: 85.174%
 epoch: 17, test accuracy: 89.468%
 epoch: 18 loss: 0.0050 batch: 10 accuracy: 85.000%
 epoch: 18 loss: 0.0047 batch: 20 accuracy: 83.750%
 epoch: 18 loss: 0.1033 batch: 30 accuracy: 82.917%
 epoch: 18 loss: 0.0005 batch: 40 accuracy: 82.188%
 epoch: 18 loss: 0.0283 batch: 50 accuracy: 83.000%
 epoch: 18 loss: 1.3418 batch: 60 accuracy: 83.333%
 epoch: 18 loss: 0.1841 batch: 70 accuracy: 83.393%
 epoch: 18 loss: 0.4440 batch: 80 accuracy: 83.906%
 epoch: 18 loss: 0.1100 batch: 90 accuracy: 84.167%
 epoch: 18 loss: 0.4470 batch: 100 accuracy: 83.875%
 epoch: 18 loss: 0.3989 batch: 110 accuracy: 84.545%
 epoch: 18 loss: 0.5133 batch: 120 accuracy: 85.208%

epoch: 18 loss: 0.1517 batch: 130 accuracy: 85.288%
 epoch: 18 loss: 0.2305 batch: 140 accuracy: 84.821%
 epoch: 18 loss: 0.1069 batch: 150 accuracy: 85.417%
 epoch: 18 loss: 0.1913 batch: 160 accuracy: 85.234%
 epoch: 18 loss: 0.0092 batch: 170 accuracy: 85.368%
 epoch: 18 loss: 0.0173 batch: 180 accuracy: 85.556%
 epoch: 18 loss: 0.0025 batch: 190 accuracy: 85.526%
 epoch: 18 loss: 0.1439 batch: 200 accuracy: 85.375%
 epoch: 18 loss: 0.0046 batch: 210 accuracy: 85.655%
 epoch: 18 loss: 0.0394 batch: 220 accuracy: 85.795%
 epoch: 18 loss: 0.1504 batch: 230 accuracy: 86.033%
 epoch: 18 loss: 0.0065 batch: 240 accuracy: 86.042%
 epoch: 18 loss: 0.3693 batch: 250 accuracy: 86.350%
 epoch: 18 loss: 0.0416 batch: 260 accuracy: 86.154%
 epoch: 18 loss: 0.3777 batch: 270 accuracy: 86.296%
 epoch: 18 loss: 0.2409 batch: 280 accuracy: 86.205%
 epoch: 18 loss: 0.3497 batch: 290 accuracy: 86.034%
 epoch: 18 loss: 0.6799 batch: 300 accuracy: 86.083%
 epoch: 18 loss: 0.0796 batch: 310 accuracy: 85.927%
 epoch: 18 loss: 0.0724 batch: 320 accuracy: 85.977%
 epoch: 18 loss: 0.0051 batch: 330 accuracy: 86.174%
 epoch: 18 loss: 0.0506 batch: 340 accuracy: 86.287%
 epoch: 18 loss: 0.1864 batch: 350 accuracy: 86.286%
 epoch: 18 loss: 0.0075 batch: 360 accuracy: 86.181%
 epoch: 18 loss: 0.0429 batch: 370 accuracy: 86.115%
 epoch: 18 loss: 0.0164 batch: 380 accuracy: 86.250%
 epoch: 18 loss: 1.8156 batch: 390 accuracy: 86.154%
 epoch: 18 loss: 0.0990 batch: 400 accuracy: 86.094%
 epoch: 18 loss: 0.2348 batch: 410 accuracy: 86.159%
 epoch: 18 loss: 0.0204 batch: 420 accuracy: 86.131%
 epoch: 18 loss: 0.7395 batch: 430 accuracy: 86.105%
 epoch: 18, test accuracy: 88.079%
 epoch: 19 loss: 0.1823 batch: 10 accuracy: 82.500%
 epoch: 19 loss: 0.0120 batch: 20 accuracy: 85.625%
 epoch: 19 loss: 0.2435 batch: 30 accuracy: 84.167%
 epoch: 19 loss: 0.1801 batch: 40 accuracy: 83.125%
 epoch: 19 loss: 0.1478 batch: 50 accuracy: 83.250%
 epoch: 19 loss: 0.6767 batch: 60 accuracy: 82.917%
 epoch: 19 loss: 0.0825 batch: 70 accuracy: 83.571%
 epoch: 19 loss: 0.2224 batch: 80 accuracy: 83.594%
 epoch: 19 loss: 0.3101 batch: 90 accuracy: 83.056%
 epoch: 19 loss: 0.0747 batch: 100 accuracy: 82.875%
 epoch: 19 loss: 0.0087 batch: 110 accuracy: 82.841%
 epoch: 19 loss: 0.0767 batch: 120 accuracy: 83.125%
 epoch: 19 loss: 0.0096 batch: 130 accuracy: 83.462%
 epoch: 19 loss: 0.4873 batch: 140 accuracy: 84.018%
 epoch: 19 loss: 0.2905 batch: 150 accuracy: 84.250%
 epoch: 19 loss: 0.0071 batch: 160 accuracy: 84.453%

epoch: 19 loss: 0.0718 batch: 170 accuracy: 84.779%
 epoch: 19 loss: 0.0281 batch: 180 accuracy: 85.139%
 epoch: 19 loss: 0.2184 batch: 190 accuracy: 85.197%
 epoch: 19 loss: 0.0619 batch: 200 accuracy: 85.125%
 epoch: 19 loss: 1.1321 batch: 210 accuracy: 85.060%
 epoch: 19 loss: 0.0483 batch: 220 accuracy: 85.114%
 epoch: 19 loss: 0.0040 batch: 230 accuracy: 85.435%
 epoch: 19 loss: 0.1252 batch: 240 accuracy: 85.208%
 epoch: 19 loss: 0.0404 batch: 250 accuracy: 85.200%
 epoch: 19 loss: 0.5200 batch: 260 accuracy: 85.000%
 epoch: 19 loss: 0.8744 batch: 270 accuracy: 84.722%
 epoch: 19 loss: 0.0964 batch: 280 accuracy: 84.777%
 epoch: 19 loss: 0.1336 batch: 290 accuracy: 84.871%
 epoch: 19 loss: 0.1321 batch: 300 accuracy: 85.000%
 epoch: 19 loss: 0.9288 batch: 310 accuracy: 84.919%
 epoch: 19 loss: 0.2372 batch: 320 accuracy: 85.000%
 epoch: 19 loss: 0.0355 batch: 330 accuracy: 85.038%
 epoch: 19 loss: 0.0110 batch: 340 accuracy: 85.037%
 epoch: 19 loss: 0.2435 batch: 350 accuracy: 85.250%
 epoch: 19 loss: 0.6701 batch: 360 accuracy: 85.139%
 epoch: 19 loss: 0.2299 batch: 370 accuracy: 85.135%
 epoch: 19 loss: 0.2244 batch: 380 accuracy: 85.132%
 epoch: 19 loss: 0.1822 batch: 390 accuracy: 85.096%
 epoch: 19 loss: 0.8421 batch: 400 accuracy: 84.969%
 epoch: 19 loss: 0.1779 batch: 410 accuracy: 85.091%
 epoch: 19 loss: 0.3162 batch: 420 accuracy: 85.060%
 epoch: 19 loss: 0.0213 batch: 430 accuracy: 85.000%
 epoch: 19, test accuracy: 88.657%
 epoch: 20 loss: 0.0767 batch: 10 accuracy: 81.250%
 epoch: 20 loss: 0.0733 batch: 20 accuracy: 86.250%
 epoch: 20 loss: 0.0306 batch: 30 accuracy: 87.500%
 epoch: 20 loss: 0.0224 batch: 40 accuracy: 87.812%
 epoch: 20 loss: 0.4205 batch: 50 accuracy: 86.000%
 epoch: 20 loss: 0.0684 batch: 60 accuracy: 85.417%
 epoch: 20 loss: 0.0107 batch: 70 accuracy: 84.464%
 epoch: 20 loss: 0.2939 batch: 80 accuracy: 84.219%
 epoch: 20 loss: 0.5003 batch: 90 accuracy: 84.028%
 epoch: 20 loss: 0.4381 batch: 100 accuracy: 84.250%
 epoch: 20 loss: 1.9300 batch: 110 accuracy: 83.295%
 epoch: 20 loss: 1.3102 batch: 120 accuracy: 83.646%
 epoch: 20 loss: 0.0350 batch: 130 accuracy: 83.750%
 epoch: 20 loss: 0.0111 batch: 140 accuracy: 83.839%
 epoch: 20 loss: 0.1477 batch: 150 accuracy: 84.167%
 epoch: 20 loss: 0.5049 batch: 160 accuracy: 84.219%
 epoch: 20 loss: 0.3813 batch: 170 accuracy: 84.412%
 epoch: 20 loss: 0.0009 batch: 180 accuracy: 84.444%
 epoch: 20 loss: 0.0758 batch: 190 accuracy: 84.605%
 epoch: 20 loss: 0.4252 batch: 200 accuracy: 84.812%

epoch: 20 loss: 0.4161 batch: 210 accuracy: 84.583%
 epoch: 20 loss: 0.0188 batch: 220 accuracy: 84.602%
 epoch: 20 loss: 1.2763 batch: 230 accuracy: 84.728%
 epoch: 20 loss: 0.0394 batch: 240 accuracy: 84.844%
 epoch: 20 loss: 0.1716 batch: 250 accuracy: 85.150%
 epoch: 20 loss: 0.3810 batch: 260 accuracy: 85.240%
 epoch: 20 loss: 0.1271 batch: 270 accuracy: 85.278%
 epoch: 20 loss: 0.0817 batch: 280 accuracy: 85.357%
 epoch: 20 loss: 0.0439 batch: 290 accuracy: 85.517%
 epoch: 20 loss: 0.0725 batch: 300 accuracy: 85.167%
 epoch: 20 loss: 0.9026 batch: 310 accuracy: 85.121%
 epoch: 20 loss: 0.6364 batch: 320 accuracy: 84.844%
 epoch: 20 loss: 0.8974 batch: 330 accuracy: 84.621%
 epoch: 20 loss: 0.6241 batch: 340 accuracy: 84.706%
 epoch: 20 loss: 0.0043 batch: 350 accuracy: 84.714%
 epoch: 20 loss: 0.5303 batch: 360 accuracy: 84.792%
 epoch: 20 loss: 0.5107 batch: 370 accuracy: 84.865%
 epoch: 20 loss: 0.0241 batch: 380 accuracy: 84.934%
 epoch: 20 loss: 0.0069 batch: 390 accuracy: 84.936%
 epoch: 20 loss: 0.3659 batch: 400 accuracy: 84.875%
 epoch: 20 loss: 1.0051 batch: 410 accuracy: 84.726%
 epoch: 20 loss: 0.2800 batch: 420 accuracy: 84.821%
 epoch: 20 loss: 0.0910 batch: 430 accuracy: 84.826%
 epoch: 20, test accuracy: 90.162%
 epoch: 21 loss: 0.0216 batch: 10 accuracy: 87.500%
 epoch: 21 loss: 0.0336 batch: 20 accuracy: 88.125%
 epoch: 21 loss: 0.0586 batch: 30 accuracy: 87.917%
 epoch: 21 loss: 0.0770 batch: 40 accuracy: 89.062%
 epoch: 21 loss: 0.0975 batch: 50 accuracy: 87.750%
 epoch: 21 loss: 0.4632 batch: 60 accuracy: 86.250%
 epoch: 21 loss: 0.0083 batch: 70 accuracy: 86.964%
 epoch: 21 loss: 0.0784 batch: 80 accuracy: 87.344%
 epoch: 21 loss: 0.4575 batch: 90 accuracy: 87.222%
 epoch: 21 loss: 0.1845 batch: 100 accuracy: 87.125%
 epoch: 21 loss: 0.0103 batch: 110 accuracy: 87.386%
 epoch: 21 loss: 0.4150 batch: 120 accuracy: 87.708%
 epoch: 21 loss: 0.4150 batch: 130 accuracy: 87.404%
 epoch: 21 loss: 0.2002 batch: 140 accuracy: 86.786%
 epoch: 21 loss: 0.6794 batch: 150 accuracy: 86.583%
 epoch: 21 loss: 0.0008 batch: 160 accuracy: 86.484%
 epoch: 21 loss: 0.0741 batch: 170 accuracy: 86.176%
 epoch: 21 loss: 0.0086 batch: 180 accuracy: 86.319%
 epoch: 21 loss: 0.6027 batch: 190 accuracy: 86.053%
 epoch: 21 loss: 0.3197 batch: 200 accuracy: 85.938%
 epoch: 21 loss: 0.0892 batch: 210 accuracy: 86.250%
 epoch: 21 loss: 0.0040 batch: 220 accuracy: 86.307%
 epoch: 21 loss: 0.0091 batch: 230 accuracy: 86.250%
 epoch: 21 loss: 0.2732 batch: 240 accuracy: 86.458%

epoch: 21 loss: 0.1762 batch: 250 accuracy: 86.350%
 epoch: 21 loss: 0.0879 batch: 260 accuracy: 86.202%
 epoch: 21 loss: 0.8212 batch: 270 accuracy: 85.972%
 epoch: 21 loss: 0.3375 batch: 280 accuracy: 86.027%
 epoch: 21 loss: 0.2009 batch: 290 accuracy: 85.862%
 epoch: 21 loss: 0.6733 batch: 300 accuracy: 85.833%
 epoch: 21 loss: 0.2962 batch: 310 accuracy: 85.806%
 epoch: 21 loss: 0.8284 batch: 320 accuracy: 85.742%
 epoch: 21 loss: 0.0003 batch: 330 accuracy: 85.492%
 epoch: 21 loss: 0.7336 batch: 340 accuracy: 85.404%
 epoch: 21 loss: 0.3029 batch: 350 accuracy: 85.393%
 epoch: 21 loss: 0.0000 batch: 360 accuracy: 85.521%
 epoch: 21 loss: 0.0125 batch: 370 accuracy: 85.507%
 epoch: 21 loss: 0.2054 batch: 380 accuracy: 85.428%
 epoch: 21 loss: 0.0166 batch: 390 accuracy: 85.160%
 epoch: 21 loss: 0.4717 batch: 400 accuracy: 85.312%
 epoch: 21 loss: 0.0333 batch: 410 accuracy: 85.366%
 epoch: 21 loss: 0.0903 batch: 420 accuracy: 85.208%
 epoch: 21 loss: 0.0380 batch: 430 accuracy: 85.291%
 epoch: 21, test accuracy: 87.037%
 epoch: 22 loss: 0.2770 batch: 10 accuracy: 83.750%
 epoch: 22 loss: 0.3940 batch: 20 accuracy: 81.875%
 epoch: 22 loss: 0.0200 batch: 30 accuracy: 83.333%
 epoch: 22 loss: 1.0768 batch: 40 accuracy: 82.812%
 epoch: 22 loss: 0.3661 batch: 50 accuracy: 83.000%
 epoch: 22 loss: 0.1133 batch: 60 accuracy: 83.750%
 epoch: 22 loss: 0.0001 batch: 70 accuracy: 83.393%
 epoch: 22 loss: 0.1188 batch: 80 accuracy: 83.281%
 epoch: 22 loss: 0.2268 batch: 90 accuracy: 83.194%
 epoch: 22 loss: 0.0002 batch: 100 accuracy: 84.125%
 epoch: 22 loss: 0.0564 batch: 110 accuracy: 84.886%
 epoch: 22 loss: 0.0731 batch: 120 accuracy: 85.000%
 epoch: 22 loss: 0.0430 batch: 130 accuracy: 85.192%
 epoch: 22 loss: 0.0254 batch: 140 accuracy: 85.536%
 epoch: 22 loss: 0.0759 batch: 150 accuracy: 85.583%
 epoch: 22 loss: 0.7955 batch: 160 accuracy: 85.703%
 epoch: 22 loss: 0.3398 batch: 170 accuracy: 85.809%
 epoch: 22 loss: 3.0846 batch: 180 accuracy: 85.556%
 epoch: 22 loss: 0.0022 batch: 190 accuracy: 85.658%
 epoch: 22 loss: 0.0938 batch: 200 accuracy: 85.625%
 epoch: 22 loss: 0.2449 batch: 210 accuracy: 85.774%
 epoch: 22 loss: 0.0243 batch: 220 accuracy: 85.511%
 epoch: 22 loss: 0.4608 batch: 230 accuracy: 85.598%
 epoch: 22 loss: 0.3408 batch: 240 accuracy: 85.833%
 epoch: 22 loss: 0.0387 batch: 250 accuracy: 85.750%
 epoch: 22 loss: 0.3528 batch: 260 accuracy: 85.817%
 epoch: 22 loss: 0.0595 batch: 270 accuracy: 86.019%
 epoch: 22 loss: 0.0099 batch: 280 accuracy: 86.205%

epoch: 22 loss: 0.4068 batch: 290 accuracy: 86.121%
 epoch: 22 loss: 0.0058 batch: 300 accuracy: 86.083%
 epoch: 22 loss: 0.3056 batch: 310 accuracy: 85.927%
 epoch: 22 loss: 0.9655 batch: 320 accuracy: 85.859%
 epoch: 22 loss: 0.0017 batch: 330 accuracy: 85.720%
 epoch: 22 loss: 0.3137 batch: 340 accuracy: 85.625%
 epoch: 22 loss: 0.0228 batch: 350 accuracy: 85.607%
 epoch: 22 loss: 0.1951 batch: 360 accuracy: 85.660%
 epoch: 22 loss: 0.0039 batch: 370 accuracy: 85.743%
 epoch: 22 loss: 0.7229 batch: 380 accuracy: 85.789%
 epoch: 22 loss: 0.0251 batch: 390 accuracy: 85.929%
 epoch: 22 loss: 0.1127 batch: 400 accuracy: 86.000%
 epoch: 22 loss: 0.0064 batch: 410 accuracy: 86.098%
 epoch: 22 loss: 0.1312 batch: 420 accuracy: 86.280%
 epoch: 22 loss: 0.0446 batch: 430 accuracy: 86.424%
 epoch: 22, test accuracy: 90.625%
 epoch: 23 loss: 0.1437 batch: 10 accuracy: 86.250%
 epoch: 23 loss: 0.1142 batch: 20 accuracy: 82.500%
 epoch: 23 loss: 0.0720 batch: 30 accuracy: 82.083%
 epoch: 23 loss: 0.0739 batch: 40 accuracy: 83.438%
 epoch: 23 loss: 0.0020 batch: 50 accuracy: 85.500%
 epoch: 23 loss: 0.0010 batch: 60 accuracy: 85.417%
 epoch: 23 loss: 0.3560 batch: 70 accuracy: 85.536%
 epoch: 23 loss: 0.2827 batch: 80 accuracy: 85.781%
 epoch: 23 loss: 0.0002 batch: 90 accuracy: 85.833%
 epoch: 23 loss: 0.1187 batch: 100 accuracy: 85.625%
 epoch: 23 loss: 0.0334 batch: 110 accuracy: 85.341%
 epoch: 23 loss: 0.2202 batch: 120 accuracy: 85.312%
 epoch: 23 loss: 0.2395 batch: 130 accuracy: 85.192%
 epoch: 23 loss: 0.1053 batch: 140 accuracy: 85.446%
 epoch: 23 loss: 0.2146 batch: 150 accuracy: 85.667%
 epoch: 23 loss: 0.3400 batch: 160 accuracy: 85.938%
 epoch: 23 loss: 0.2381 batch: 170 accuracy: 86.324%
 epoch: 23 loss: 0.0232 batch: 180 accuracy: 86.736%
 epoch: 23 loss: 0.0943 batch: 190 accuracy: 86.711%
 epoch: 23 loss: 0.1367 batch: 200 accuracy: 87.125%
 epoch: 23 loss: 0.2240 batch: 210 accuracy: 87.143%
 epoch: 23 loss: 0.0328 batch: 220 accuracy: 86.989%
 epoch: 23 loss: 0.1116 batch: 230 accuracy: 87.228%
 epoch: 23 loss: 0.0014 batch: 240 accuracy: 87.396%
 epoch: 23 loss: 0.3554 batch: 250 accuracy: 87.500%
 epoch: 23 loss: 0.0410 batch: 260 accuracy: 87.548%
 epoch: 23 loss: 0.0231 batch: 270 accuracy: 87.546%
 epoch: 23 loss: 0.0072 batch: 280 accuracy: 87.679%
 epoch: 23 loss: 0.5518 batch: 290 accuracy: 87.414%
 epoch: 23 loss: 0.3129 batch: 300 accuracy: 87.250%
 epoch: 23 loss: 0.1140 batch: 310 accuracy: 87.258%
 epoch: 23 loss: 0.1325 batch: 320 accuracy: 87.422%

epoch: 23 loss: 0.1332 batch: 330 accuracy: 87.235%
 epoch: 23 loss: 0.0041 batch: 340 accuracy: 86.985%
 epoch: 23 loss: 0.0652 batch: 350 accuracy: 86.964%
 epoch: 23 loss: 0.0028 batch: 360 accuracy: 87.083%
 epoch: 23 loss: 0.4687 batch: 370 accuracy: 87.061%
 epoch: 23 loss: 0.1231 batch: 380 accuracy: 87.039%
 epoch: 23 loss: 0.7558 batch: 390 accuracy: 86.923%
 epoch: 23 loss: 0.0144 batch: 400 accuracy: 86.844%
 epoch: 23 loss: 0.1395 batch: 410 accuracy: 86.768%
 epoch: 23 loss: 0.0069 batch: 420 accuracy: 86.756%
 epoch: 23 loss: 0.0836 batch: 430 accuracy: 86.802%
 epoch: 23, test accuracy: 86.458%
 epoch: 24 loss: 0.3437 batch: 10 accuracy: 82.500%
 epoch: 24 loss: 0.0628 batch: 20 accuracy: 81.250%
 epoch: 24 loss: 0.7785 batch: 30 accuracy: 80.833%
 epoch: 24 loss: 0.1444 batch: 40 accuracy: 82.188%
 epoch: 24 loss: 0.0334 batch: 50 accuracy: 82.000%
 epoch: 24 loss: 0.0192 batch: 60 accuracy: 82.500%
 epoch: 24 loss: 0.0034 batch: 70 accuracy: 83.929%
 epoch: 24 loss: 1.4330 batch: 80 accuracy: 82.969%
 epoch: 24 loss: 0.8733 batch: 90 accuracy: 83.194%
 epoch: 24 loss: 0.8057 batch: 100 accuracy: 83.125%
 epoch: 24 loss: 0.0317 batch: 110 accuracy: 83.636%
 epoch: 24 loss: 0.0005 batch: 120 accuracy: 83.646%
 epoch: 24 loss: 0.0326 batch: 130 accuracy: 84.519%
 epoch: 24 loss: 0.1243 batch: 140 accuracy: 84.554%
 epoch: 24 loss: 0.0006 batch: 150 accuracy: 85.250%
 epoch: 24 loss: 0.0097 batch: 160 accuracy: 85.312%
 epoch: 24 loss: 0.8039 batch: 170 accuracy: 85.441%
 epoch: 24 loss: 0.1354 batch: 180 accuracy: 85.694%
 epoch: 24 loss: 0.7540 batch: 190 accuracy: 85.658%
 epoch: 24 loss: 0.0385 batch: 200 accuracy: 85.938%
 epoch: 24 loss: 0.0341 batch: 210 accuracy: 86.071%
 epoch: 24 loss: 0.0371 batch: 220 accuracy: 85.966%
 epoch: 24 loss: 0.4340 batch: 230 accuracy: 86.250%
 epoch: 24 loss: 0.4897 batch: 240 accuracy: 86.406%
 epoch: 24 loss: 0.0556 batch: 250 accuracy: 86.200%
 epoch: 24 loss: 0.2031 batch: 260 accuracy: 86.250%
 epoch: 24 loss: 0.0084 batch: 270 accuracy: 86.343%
 epoch: 24 loss: 0.0293 batch: 280 accuracy: 86.250%
 epoch: 24 loss: 0.4956 batch: 290 accuracy: 86.466%
 epoch: 24 loss: 0.1890 batch: 300 accuracy: 86.417%
 epoch: 24 loss: 0.0269 batch: 310 accuracy: 86.250%
 epoch: 24 loss: 0.4216 batch: 320 accuracy: 86.172%
 epoch: 24 loss: 0.0008 batch: 330 accuracy: 86.136%
 epoch: 24 loss: 0.0301 batch: 340 accuracy: 86.213%
 epoch: 24 loss: 0.0582 batch: 350 accuracy: 86.107%
 epoch: 24 loss: 0.0021 batch: 360 accuracy: 86.042%

epoch: 24 loss: 0.0825 batch: 370 accuracy: 85.946%
 epoch: 24 loss: 0.3266 batch: 380 accuracy: 85.987%
 epoch: 24 loss: 0.1013 batch: 390 accuracy: 85.929%
 epoch: 24 loss: 0.1000 batch: 400 accuracy: 85.906%
 epoch: 24 loss: 0.1231 batch: 410 accuracy: 86.037%
 epoch: 24 loss: 0.0142 batch: 420 accuracy: 86.131%
 epoch: 24 loss: 0.4650 batch: 430 accuracy: 86.221%
 epoch: 24, test accuracy: 89.352%
 epoch: 25 loss: 0.2958 batch: 10 accuracy: 77.500%
 epoch: 25 loss: 0.1478 batch: 20 accuracy: 80.625%
 epoch: 25 loss: 0.6187 batch: 30 accuracy: 81.250%
 epoch: 25 loss: 0.3247 batch: 40 accuracy: 80.625%
 epoch: 25 loss: 0.4112 batch: 50 accuracy: 82.000%
 epoch: 25 loss: 0.0058 batch: 60 accuracy: 83.125%
 epoch: 25 loss: 0.1856 batch: 70 accuracy: 84.286%
 epoch: 25 loss: 0.0777 batch: 80 accuracy: 85.000%
 epoch: 25 loss: 0.0008 batch: 90 accuracy: 85.139%
 epoch: 25 loss: 0.2691 batch: 100 accuracy: 85.125%
 epoch: 25 loss: 0.0094 batch: 110 accuracy: 85.227%
 epoch: 25 loss: 0.6741 batch: 120 accuracy: 84.896%
 epoch: 25 loss: 1.2297 batch: 130 accuracy: 84.519%
 epoch: 25 loss: 0.7191 batch: 140 accuracy: 84.375%
 epoch: 25 loss: 0.0196 batch: 150 accuracy: 84.750%
 epoch: 25 loss: 0.6496 batch: 160 accuracy: 84.766%
 epoch: 25 loss: 0.0050 batch: 170 accuracy: 85.147%
 epoch: 25 loss: 0.1626 batch: 180 accuracy: 85.347%
 epoch: 25 loss: 0.1342 batch: 190 accuracy: 85.395%
 epoch: 25 loss: 1.6995 batch: 200 accuracy: 85.000%
 epoch: 25 loss: 0.8757 batch: 210 accuracy: 85.119%
 epoch: 25 loss: 0.7816 batch: 220 accuracy: 85.284%
 epoch: 25 loss: 0.5306 batch: 230 accuracy: 85.163%
 epoch: 25 loss: 0.3970 batch: 240 accuracy: 85.312%
 epoch: 25 loss: 0.0791 batch: 250 accuracy: 85.400%
 epoch: 25 loss: 0.0550 batch: 260 accuracy: 85.385%
 epoch: 25 loss: 0.0157 batch: 270 accuracy: 85.417%
 epoch: 25 loss: 0.2910 batch: 280 accuracy: 85.491%
 epoch: 25 loss: 0.0251 batch: 290 accuracy: 85.474%
 epoch: 25 loss: 0.6527 batch: 300 accuracy: 85.375%
 epoch: 25 loss: 0.0001 batch: 310 accuracy: 85.403%
 epoch: 25 loss: 0.1123 batch: 320 accuracy: 85.430%
 epoch: 25 loss: 0.3416 batch: 330 accuracy: 85.341%
 epoch: 25 loss: 0.1910 batch: 340 accuracy: 85.147%
 epoch: 25 loss: 0.2995 batch: 350 accuracy: 85.179%
 epoch: 25 loss: 0.0583 batch: 360 accuracy: 85.000%
 epoch: 25 loss: 0.0633 batch: 370 accuracy: 85.169%
 epoch: 25 loss: 0.1840 batch: 380 accuracy: 85.099%
 epoch: 25 loss: 0.0635 batch: 390 accuracy: 84.936%
 epoch: 25 loss: 0.0961 batch: 400 accuracy: 84.844%

epoch: 25 loss: 0.0245 batch: 410 accuracy: 84.878%
 epoch: 25 loss: 0.0504 batch: 420 accuracy: 85.000%
 epoch: 25 loss: 0.0103 batch: 430 accuracy: 85.145%
 epoch: 25, test accuracy: 89.120%
 epoch: 26 loss: 0.2864 batch: 10 accuracy: 91.250%
 epoch: 26 loss: 0.2534 batch: 20 accuracy: 91.250%
 epoch: 26 loss: 0.0279 batch: 30 accuracy: 92.083%
 epoch: 26 loss: 0.3130 batch: 40 accuracy: 90.312%
 epoch: 26 loss: 0.0361 batch: 50 accuracy: 89.000%
 epoch: 26 loss: 0.1601 batch: 60 accuracy: 89.375%
 epoch: 26 loss: 0.0050 batch: 70 accuracy: 89.464%
 epoch: 26 loss: 0.0000 batch: 80 accuracy: 89.844%
 epoch: 26 loss: 0.0998 batch: 90 accuracy: 89.306%
 epoch: 26 loss: 0.1818 batch: 100 accuracy: 88.750%
 epoch: 26 loss: 0.0011 batch: 110 accuracy: 88.864%
 epoch: 26 loss: 0.0247 batch: 120 accuracy: 88.958%
 epoch: 26 loss: 0.1355 batch: 130 accuracy: 88.942%
 epoch: 26 loss: 0.1860 batch: 140 accuracy: 88.750%
 epoch: 26 loss: 0.0760 batch: 150 accuracy: 88.917%
 epoch: 26 loss: 0.0073 batch: 160 accuracy: 88.750%
 epoch: 26 loss: 0.0262 batch: 170 accuracy: 88.897%
 epoch: 26 loss: 0.2826 batch: 180 accuracy: 88.819%
 epoch: 26 loss: 0.1893 batch: 190 accuracy: 88.421%
 epoch: 26 loss: 0.0143 batch: 200 accuracy: 88.562%
 epoch: 26 loss: 0.0151 batch: 210 accuracy: 88.393%
 epoch: 26 loss: 0.2449 batch: 220 accuracy: 88.182%
 epoch: 26 loss: 0.0011 batch: 230 accuracy: 88.207%
 epoch: 26 loss: 0.1031 batch: 240 accuracy: 88.281%
 epoch: 26 loss: 0.1410 batch: 250 accuracy: 88.100%
 epoch: 26 loss: 0.0264 batch: 260 accuracy: 88.221%
 epoch: 26 loss: 0.3243 batch: 270 accuracy: 88.102%
 epoch: 26 loss: 0.0512 batch: 280 accuracy: 88.080%
 epoch: 26 loss: 0.0141 batch: 290 accuracy: 88.060%
 epoch: 26 loss: 0.2864 batch: 300 accuracy: 87.875%
 epoch: 26 loss: 0.0246 batch: 310 accuracy: 87.984%
 epoch: 26 loss: 0.1081 batch: 320 accuracy: 87.852%
 epoch: 26 loss: 0.0446 batch: 330 accuracy: 87.614%
 epoch: 26 loss: 0.0399 batch: 340 accuracy: 87.574%
 epoch: 26 loss: 0.0743 batch: 350 accuracy: 87.607%
 epoch: 26 loss: 0.0329 batch: 360 accuracy: 87.708%
 epoch: 26 loss: 0.8204 batch: 370 accuracy: 87.534%
 epoch: 26 loss: 0.1151 batch: 380 accuracy: 87.533%
 epoch: 26 loss: 0.0181 batch: 390 accuracy: 87.500%
 epoch: 26 loss: 0.1063 batch: 400 accuracy: 87.438%
 epoch: 26 loss: 0.2987 batch: 410 accuracy: 87.470%
 epoch: 26 loss: 0.1163 batch: 420 accuracy: 87.560%
 epoch: 26 loss: 0.5376 batch: 430 accuracy: 87.587%
 epoch: 26, test accuracy: 86.806%

epoch: 27 loss: 0.7506 batch: 10 accuracy: 83.750%
 epoch: 27 loss: 0.1040 batch: 20 accuracy: 83.125%
 epoch: 27 loss: 0.0016 batch: 30 accuracy: 84.583%
 epoch: 27 loss: 0.0084 batch: 40 accuracy: 83.125%
 epoch: 27 loss: 0.1931 batch: 50 accuracy: 83.750%
 epoch: 27 loss: 0.1524 batch: 60 accuracy: 84.583%
 epoch: 27 loss: 0.0134 batch: 70 accuracy: 85.893%
 epoch: 27 loss: 0.4079 batch: 80 accuracy: 86.406%
 epoch: 27 loss: 0.0772 batch: 90 accuracy: 85.833%
 epoch: 27 loss: 0.4283 batch: 100 accuracy: 85.000%
 epoch: 27 loss: 0.0852 batch: 110 accuracy: 85.000%
 epoch: 27 loss: 0.3555 batch: 120 accuracy: 85.104%
 epoch: 27 loss: 0.0184 batch: 130 accuracy: 85.385%
 epoch: 27 loss: 0.0329 batch: 140 accuracy: 85.089%
 epoch: 27 loss: 0.9607 batch: 150 accuracy: 85.417%
 epoch: 27 loss: 0.2976 batch: 160 accuracy: 85.234%
 epoch: 27 loss: 0.6297 batch: 170 accuracy: 85.368%
 epoch: 27 loss: 0.2888 batch: 180 accuracy: 85.278%
 epoch: 27 loss: 0.2609 batch: 190 accuracy: 85.066%
 epoch: 27 loss: 0.0402 batch: 200 accuracy: 85.312%
 epoch: 27 loss: 0.2751 batch: 210 accuracy: 85.476%
 epoch: 27 loss: 0.2137 batch: 220 accuracy: 85.625%
 epoch: 27 loss: 0.0450 batch: 230 accuracy: 85.761%
 epoch: 27 loss: 0.3018 batch: 240 accuracy: 85.729%
 epoch: 27 loss: 0.1546 batch: 250 accuracy: 85.850%
 epoch: 27 loss: 0.5253 batch: 260 accuracy: 85.721%
 epoch: 27 loss: 0.2994 batch: 270 accuracy: 85.648%
 epoch: 27 loss: 0.2857 batch: 280 accuracy: 85.625%
 epoch: 27 loss: 0.3992 batch: 290 accuracy: 85.819%
 epoch: 27 loss: 0.0295 batch: 300 accuracy: 85.917%
 epoch: 27 loss: 0.1380 batch: 310 accuracy: 85.806%
 epoch: 27 loss: 0.4409 batch: 320 accuracy: 85.859%
 epoch: 27 loss: 0.0335 batch: 330 accuracy: 86.061%
 epoch: 27 loss: 0.3484 batch: 340 accuracy: 85.993%
 epoch: 27 loss: 0.8326 batch: 350 accuracy: 85.964%
 epoch: 27 loss: 1.2473 batch: 360 accuracy: 86.076%
 epoch: 27 loss: 0.0118 batch: 370 accuracy: 86.250%
 epoch: 27 loss: 0.3718 batch: 380 accuracy: 86.447%
 epoch: 27 loss: 0.0010 batch: 390 accuracy: 86.635%
 epoch: 27 loss: 0.0602 batch: 400 accuracy: 86.719%
 epoch: 27 loss: 0.0635 batch: 410 accuracy: 86.890%
 epoch: 27 loss: 0.1280 batch: 420 accuracy: 86.935%
 epoch: 27 loss: 0.4719 batch: 430 accuracy: 86.773%
 epoch: 27, test accuracy: 90.856%
 epoch: 28 loss: 0.2309 batch: 10 accuracy: 90.000%
 epoch: 28 loss: 1.0536 batch: 20 accuracy: 86.875%
 epoch: 28 loss: 0.2619 batch: 30 accuracy: 87.500%
 epoch: 28 loss: 0.0267 batch: 40 accuracy: 87.188%

epoch: 28 loss: 0.1281 batch: 50 accuracy: 86.000%
 epoch: 28 loss: 0.2045 batch: 60 accuracy: 85.208%
 epoch: 28 loss: 0.0721 batch: 70 accuracy: 85.536%
 epoch: 28 loss: 0.0006 batch: 80 accuracy: 85.938%
 epoch: 28 loss: 0.6621 batch: 90 accuracy: 86.528%
 epoch: 28 loss: 0.0080 batch: 100 accuracy: 87.250%
 epoch: 28 loss: 0.2416 batch: 110 accuracy: 86.818%
 epoch: 28 loss: 0.2748 batch: 120 accuracy: 87.083%
 epoch: 28 loss: 0.1277 batch: 130 accuracy: 86.827%
 epoch: 28 loss: 0.1107 batch: 140 accuracy: 87.232%
 epoch: 28 loss: 0.1529 batch: 150 accuracy: 86.833%
 epoch: 28 loss: 0.0234 batch: 160 accuracy: 86.953%
 epoch: 28 loss: 0.0406 batch: 170 accuracy: 87.206%
 epoch: 28 loss: 0.0358 batch: 180 accuracy: 87.153%
 epoch: 28 loss: 1.1309 batch: 190 accuracy: 87.171%
 epoch: 28 loss: 0.0692 batch: 200 accuracy: 87.188%
 epoch: 28 loss: 0.0116 batch: 210 accuracy: 87.083%
 epoch: 28 loss: 0.1223 batch: 220 accuracy: 86.932%
 epoch: 28 loss: 0.0284 batch: 230 accuracy: 87.174%
 epoch: 28 loss: 0.2220 batch: 240 accuracy: 86.875%
 epoch: 28 loss: 0.0148 batch: 250 accuracy: 86.900%
 epoch: 28 loss: 0.2297 batch: 260 accuracy: 86.779%
 epoch: 28 loss: 0.0017 batch: 270 accuracy: 86.620%
 epoch: 28 loss: 0.0051 batch: 280 accuracy: 86.964%
 epoch: 28 loss: 0.0148 batch: 290 accuracy: 87.155%
 epoch: 28 loss: 0.0217 batch: 300 accuracy: 87.167%
 epoch: 28 loss: 0.0570 batch: 310 accuracy: 87.177%
 epoch: 28 loss: 0.0553 batch: 320 accuracy: 87.188%
 epoch: 28 loss: 0.4028 batch: 330 accuracy: 87.197%
 epoch: 28 loss: 0.2322 batch: 340 accuracy: 87.132%
 epoch: 28 loss: 0.0735 batch: 350 accuracy: 87.107%
 epoch: 28 loss: 0.0435 batch: 360 accuracy: 87.014%
 epoch: 28 loss: 0.0270 batch: 370 accuracy: 87.196%
 epoch: 28 loss: 0.0373 batch: 380 accuracy: 87.336%
 epoch: 28 loss: 0.5225 batch: 390 accuracy: 87.308%
 epoch: 28 loss: 0.6601 batch: 400 accuracy: 87.281%
 epoch: 28 loss: 0.0695 batch: 410 accuracy: 87.287%
 epoch: 28 loss: 0.1461 batch: 420 accuracy: 87.202%
 epoch: 28 loss: 0.0010 batch: 430 accuracy: 87.297%
 epoch: 28, test accuracy: 89.236%
 epoch: 29 loss: 0.2042 batch: 10 accuracy: 85.000%
 epoch: 29 loss: 0.0079 batch: 20 accuracy: 88.125%
 epoch: 29 loss: 0.1019 batch: 30 accuracy: 88.750%
 epoch: 29 loss: 0.1632 batch: 40 accuracy: 85.938%
 epoch: 29 loss: 0.0066 batch: 50 accuracy: 87.750%
 epoch: 29 loss: 0.1648 batch: 60 accuracy: 87.292%
 epoch: 29 loss: 0.7533 batch: 70 accuracy: 86.607%
 epoch: 29 loss: 0.1710 batch: 80 accuracy: 86.875%

```

epoch: 29 loss: 0.1043 batch: 90 accuracy: 86.944%
epoch: 29 loss: 0.3212 batch: 100 accuracy: 87.000%
epoch: 29 loss: 0.0097 batch: 110 accuracy: 86.818%
epoch: 29 loss: 0.0268 batch: 120 accuracy: 87.083%
epoch: 29 loss: 0.0558 batch: 130 accuracy: 87.115%
epoch: 29 loss: 0.0177 batch: 140 accuracy: 87.500%
epoch: 29 loss: 0.1246 batch: 150 accuracy: 87.417%
epoch: 29 loss: 0.0141 batch: 160 accuracy: 87.422%
epoch: 29 loss: 0.0009 batch: 170 accuracy: 87.500%
epoch: 29 loss: 0.2699 batch: 180 accuracy: 86.806%
epoch: 29 loss: 0.1180 batch: 190 accuracy: 86.645%
epoch: 29 loss: 0.0725 batch: 200 accuracy: 86.312%
epoch: 29 loss: 0.0019 batch: 210 accuracy: 86.548%
epoch: 29 loss: 0.4749 batch: 220 accuracy: 86.307%
epoch: 29 loss: 0.0630 batch: 230 accuracy: 86.413%
epoch: 29 loss: 0.1532 batch: 240 accuracy: 86.719%
epoch: 29 loss: 0.4393 batch: 250 accuracy: 86.800%
epoch: 29 loss: 0.0462 batch: 260 accuracy: 86.731%
epoch: 29 loss: 0.0320 batch: 270 accuracy: 86.528%
epoch: 29 loss: 0.0515 batch: 280 accuracy: 86.652%
epoch: 29 loss: 0.9122 batch: 290 accuracy: 86.379%
epoch: 29 loss: 0.1690 batch: 300 accuracy: 86.417%
epoch: 29 loss: 0.0756 batch: 310 accuracy: 86.371%
epoch: 29 loss: 0.2312 batch: 320 accuracy: 86.484%
epoch: 29 loss: 1.3364 batch: 330 accuracy: 86.402%
epoch: 29 loss: 0.2197 batch: 340 accuracy: 86.360%
epoch: 29 loss: 0.0551 batch: 350 accuracy: 86.321%
epoch: 29 loss: 0.0163 batch: 360 accuracy: 86.354%
epoch: 29 loss: 0.1853 batch: 370 accuracy: 86.385%
epoch: 29 loss: 0.3536 batch: 380 accuracy: 86.316%
epoch: 29 loss: 0.0601 batch: 390 accuracy: 86.250%
epoch: 29 loss: 0.0940 batch: 400 accuracy: 86.438%
epoch: 29 loss: 0.0378 batch: 410 accuracy: 86.555%
epoch: 29 loss: 0.0947 batch: 420 accuracy: 86.577%
epoch: 29 loss: 0.0001 batch: 430 accuracy: 86.657%
epoch: 29, test accuracy: 89.005%

```

Duration: 955 seconds

COUNT PARAMETERS

```

[ ]: def count_parameters(model):
      return sum(p.numel() for p in model.parameters() if p.requires_grad)

model = ResNetFlowerModel(num_classes=5).to(device)
print(f"Total Trainable Parameters: {count_parameters(model):,}")

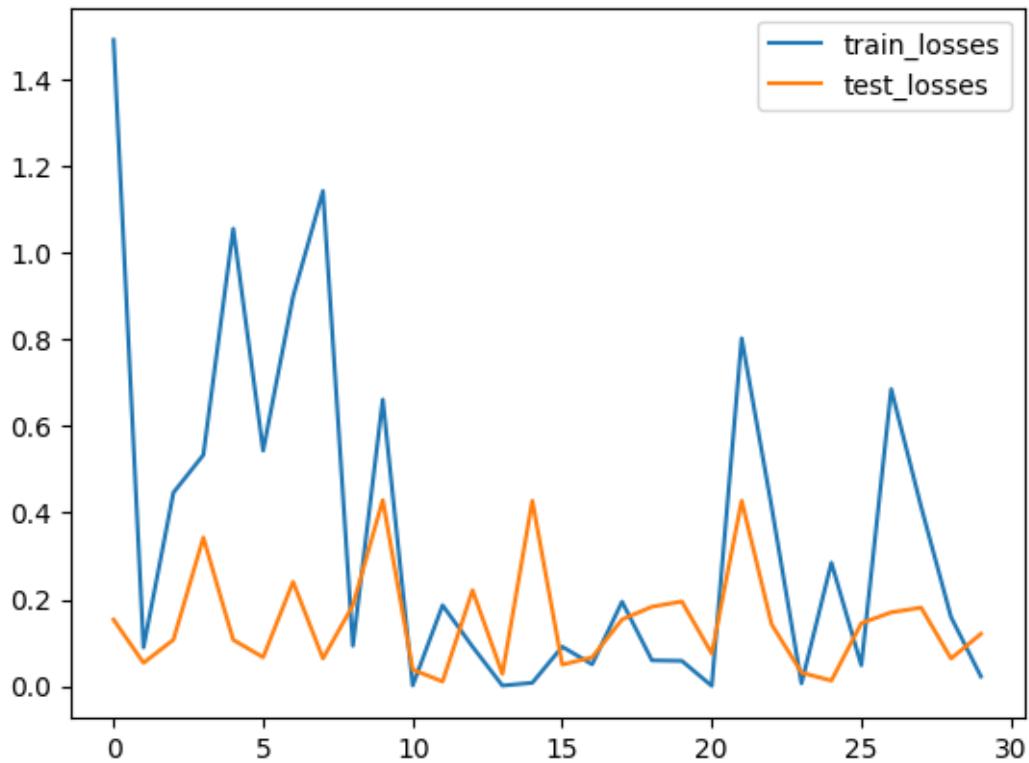
```

Total Trainable Parameters: 1,038,250

TRAINING LOSS VS TEST LOSS PER EPOCH

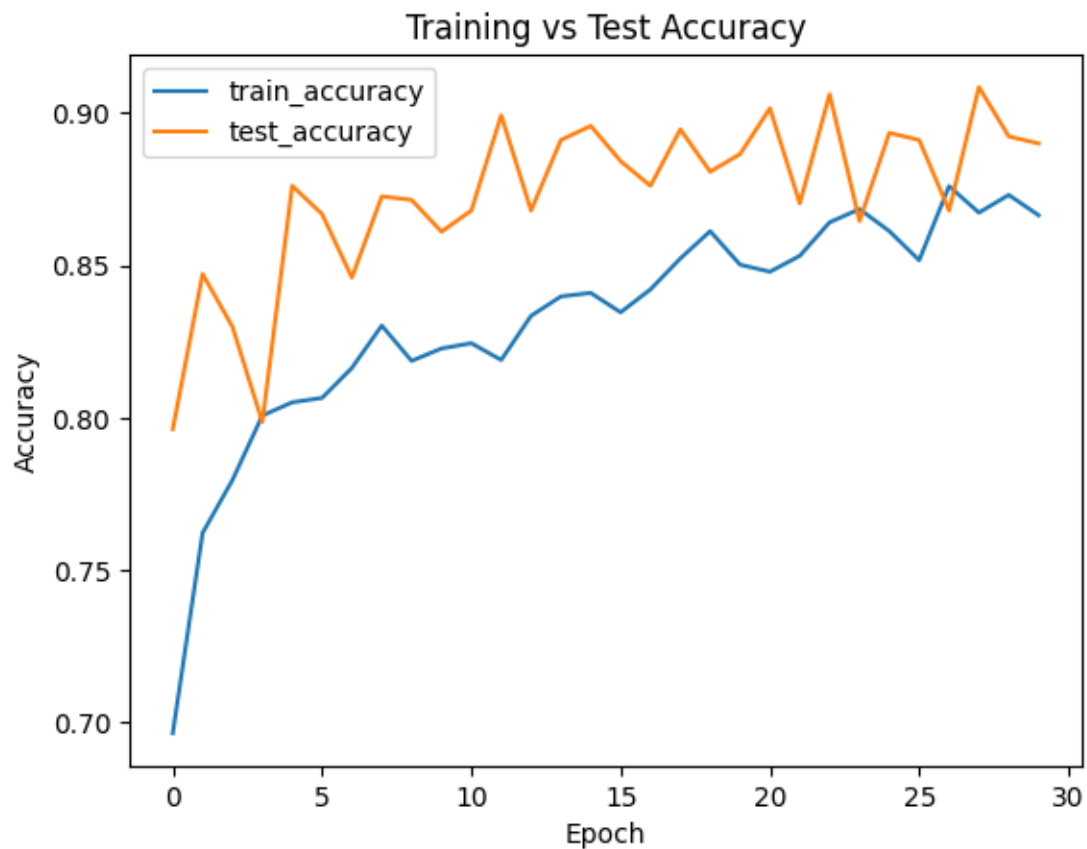
```
[ ]: # Plotting loss over time.  
plt.plot(train_losses,label="train_losses")  
plt.plot(test_losses,label="test_losses")  
plt.legend()  
train_losses
```

```
[ ]: [1.490334391593933,  
      0.08944802731275558,  
      0.44631096720695496,  
      0.533277690410614,  
      1.0550403594970703,  
      0.5432459712028503,  
      0.898073136806488,  
      1.1420806646347046,  
      0.09323965013027191,  
      0.6609362363815308,  
      0.0016257710522040725,  
      0.18657882511615753,  
      0.09122998267412186,  
      0.0013647762825712562,  
      0.007810262497514486,  
      0.09104019403457642,  
      0.050325747579336166,  
      0.1948135793209076,  
      0.0595608651638031,  
      0.058359574526548386,  
      0.0007576301577500999,  
      0.8021712303161621,  
      0.4141075611114502,  
      0.006429092492908239,  
      0.2847963273525238,  
      0.048081349581480026,  
      0.6852452158927917,  
      0.41382119059562683,  
      0.15943415462970734,  
      0.02248590812087059]
```



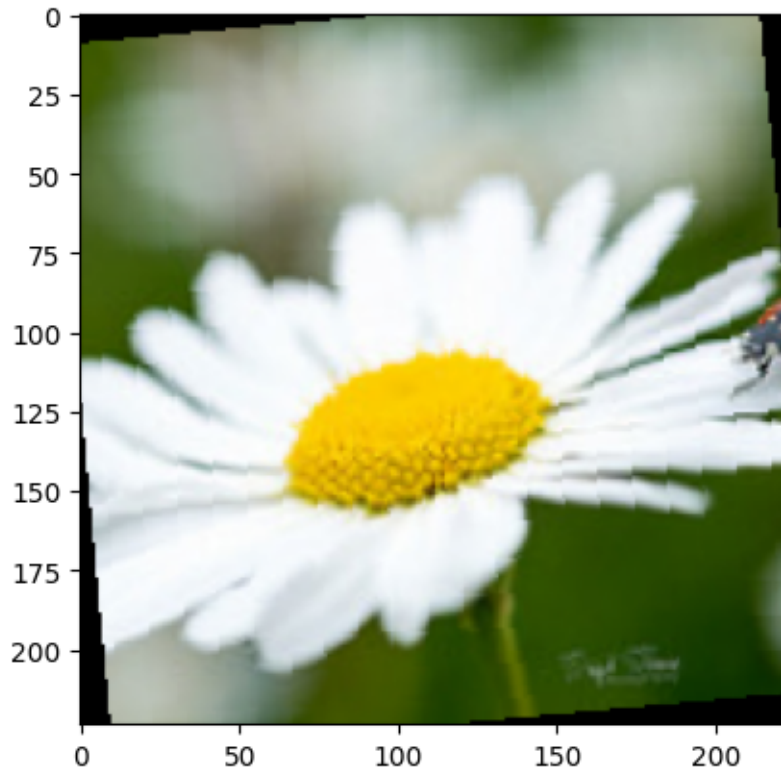
TRAIN VS TEST ACCURACY OVER TIME

```
[ ]: plt.plot(train_correct, label="train_accuracy")
plt.plot(np.array(test_correct) / len(test_loader.dataset),
         label="test_accuracy")
plt.legend()
plt.xlabel("Epoch")
plt.ylabel("Accuracy")
plt.title("Training vs Test Accuracy")
plt.show()
```



```
[ ]: x=100
      im = inv_normalize(test_data[x][0])
      plt.imshow(np.transpose(im.numpy(),(1,2,0)))
```

```
[ ]: <matplotlib.image.AxesImage at 0x7d30a5043790>
```

```
[ ]: test_data[x][0].shape
test_data[x][0].device
```

```
[ ]: device(type='cpu')
```

```
[ ]: # Prediction for one of the samples.
model.eval()
with torch.no_grad():
    new_pred=model(to_device(test_data[x][0].view(1,3,224,224),device)).argmax()
print(f'Predicted value: {new_pred.item()} {class_names[new_pred.item()]})')
```

Predicted value: 3 sunflower

##FLOPs In deep learning, FLOPs (Floating Point Operations) quantify the total number of arithmetic operations—such as additions, multiplications, and divisions—that a model performs during a single forward pass (i.e., when making a prediction). This metric serves as an indicator of a model's computational complexity. When discussing large-scale models, FLOPs are often expressed in GFLOPs (Giga Floating Point Operations), where 1 GFLOP equals one billion operations. This unit helps in comparing the computational demands of different models.

```
[ ]: # we use fvcore to calculate the FLOPs
!pip install fvcore
```

```

Collecting fvcore
  Downloading fvcore-0.1.5.post20221221.tar.gz (50 kB)
      50.2/50.2 kB
3.0 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages
(from fvcore) (2.0.2)
Collecting yacs>=0.1.6 (from fvcore)
  Downloading yacs-0.1.8-py3-none-any.whl.metadata (639 bytes)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-
packages (from fvcore) (6.0.2)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages
(from fvcore) (4.67.1)
Requirement already satisfied: termcolor>=1.1 in /usr/local/lib/python3.11/dist-
packages (from fvcore) (3.1.0)
Requirement already satisfied: Pillow in /usr/local/lib/python3.11/dist-packages
(from fvcore) (11.2.1)
Requirement already satisfied: tabulate in /usr/local/lib/python3.11/dist-
packages (from fvcore) (0.9.0)
Collecting iopath>=0.1.7 (from fvcore)
  Downloading iopath-0.1.10.tar.gz (42 kB)
      42.2/42.2 kB
4.7 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: typing_extensions in
/usr/local/lib/python3.11/dist-packages (from iopath>=0.1.7->fvcore) (4.14.0)
Collecting portalocker (from iopath>=0.1.7->fvcore)
  Downloading portalocker-3.1.1-py3-none-any.whl.metadata (8.6 kB)
Downloading yacs-0.1.8-py3-none-any.whl (14 kB)
Downloading portalocker-3.1.1-py3-none-any.whl (19 kB)
Building wheels for collected packages: fvcore, iopath
  Building wheel for fvcore (setup.py) ... done
  Created wheel for fvcore: filename=fvcore-0.1.5.post20221221-py3-none-any.whl
size=61397
sha256=6a8c8531a8c346534e839a3a741e5798b612116c5edf7cfc8d950abfad7254e4
  Stored in directory: /root/.cache/pip/wheels/65/71/95/3b8fde5c65c6e4a806e0867c
1651dcc71a1cb2f3430e8f355f
  Building wheel for iopath (setup.py) ... done
  Created wheel for iopath: filename=iopath-0.1.10-py3-none-any.whl size=31527
sha256=e7809afad861206bc4cd23a48e75a684e1cf8c9bed94142e2670ab8995e3dc58
  Stored in directory: /root/.cache/pip/wheels/ba/5e/16/6117f8fe7e9c0c161a795e10
d94645ebcf301ccbd01f66d8ec
Successfully built fvcore iopath
Installing collected packages: yacs, portalocker, iopath, fvcore
Successfully installed fvcore-0.1.5.post20221221 iopath-0.1.10 portalocker-3.1.1
yacs-0.1.8

```

```
[ ]: from fvcore.nn import FlopCountAnalysis
input = torch.randn(1, 3, 224, 224).to(device) # The input size should be the
↳ same as the size that you put into your model
#Get the network and its FLOPs
num_classes = 5
model = ResNetFlowerModel(num_classes).to(device)
model.eval()
flops = FlopCountAnalysis(model, input)
print(f"FLOPs: {flops.total()/1e9:.5f} GFLOPs")
```

WARNING:fvcore.nn.jit_analysis:Unsupported operator aten::max_pool2d encountered 1 time(s)

WARNING:fvcore.nn.jit_analysis:Unsupported operator aten::add_ encountered 16 time(s)

WARNING:fvcore.nn.jit_analysis:Unsupported operator aten::add encountered 1 time(s)

FLOPs: 4.11050 GFLOPs

EFFICIENCY AND FINAL TEST ACCURACY

```
[ ]: gflops = 4.11050
final = test_correct[-1] * 100 / len(test_loader.dataset)
efficiency = final / gflops

print(f"Final Test Accuracy: {final:.2f}%")
print(f"Efficiency Score: {efficiency:.2f}")
```

Final Test Accuracy: 89.00%

Efficiency Score: 21.65