Project 1 (res18_v1)

August 5, 2019

1 Painter classification

1.1 Project 1

Author: Fan Wenxiao

School: Beijing Institute of Technology First, we classify the problem. The problem of classification of painters is a process of supervising learning and a classification problem. So for such a problem, we used the more established model resnet-18 on the network for training. The final accuracy is stable at 80%.

Secondly, in this problem, there is a problem that the data set is too small, so we enhance the data set by means of random rotation and mirror flipping, and increase the size of the data set from 400 to 4000. This is conducive to the generalization of the model.

Load data that has been data-enhanced and convert it to a tensor.

[12]:
```
import torch
import torch.nn as nn
import torchvision.transforms as transforms
from torchvision.datasets import ImageFolder
import torch.optim as optim
import numpy as np
import math
import torch.utils.model_zoo as model_zoo

data_transform = transforms.Compose([
        #transforms.Resize(299),
        #transforms.CenterCrop(299),
        transforms.RandomResizedCrop(224),
        transforms.RandomHorizontalFlip(),
        transforms.ToTensor(),
        transforms.Normalize(mean=[0.5, 0.5, 0.5],
                             std=[0.5, 0.5, 0.5])
    ])
train_dataset = ImageFolder(root='/content/drive/My Drive/Colab Notebooks/
 ↪artist_distribute_new',transform=data_transform)
```

```
trainloader = torch.utils.data.DataLoader(train_dataset,batch_size=128,␣
 ↪shuffle=True)

print(train_dataset.class_to_idx)
#print(len(train_dataset[0]))
```

```
{'canaletto': 0, 'claude_monet': 1, 'george_romney': 2, 'jmw_turner': 3,
'john_robert_cozens': 4, 'paul_cezanne': 5, 'paul_gauguin': 6, 'paul_sandby': 7,
'peter_paul_rubens': 8, 'rembrandt': 9, 'richard_wilson': 10}
```

Define the network structure.

[0]:
```
__all__ = ['ResNet', 'resnet18_cbam', 'resnet34_cbam', 'resnet50_cbam',␣
 ↪'resnet101_cbam',
           'resnet152_cbam']


model_urls = {
    'resnet18': 'https://download.pytorch.org/models/resnet18-5c106cde.pth',
    'resnet34': 'https://download.pytorch.org/models/resnet34-333f7ec4.pth',
    'resnet50': 'https://download.pytorch.org/models/resnet50-19c8e357.pth',
    'resnet101': 'https://download.pytorch.org/models/resnet101-5d3b4d8f.pth',
    'resnet152': 'https://download.pytorch.org/models/resnet152-b121ed2d.pth',
}


def conv3x3(in_planes, out_planes, stride=1):
    "3x3 convolution with padding"
    return nn.Conv2d(in_planes, out_planes, kernel_size=3, stride=stride,
                     padding=1, bias=False)

class ChannelAttention(nn.Module):
    def __init__(self, in_planes, ratio=16):
        super(ChannelAttention, self).__init__()
        self.avg_pool = nn.AdaptiveAvgPool2d(1)
        self.max_pool = nn.AdaptiveMaxPool2d(1)

        self.fc1   = nn.Conv2d(in_planes, in_planes // 16, 1, bias=False)
        self.relu1 = nn.ReLU()
        self.fc2   = nn.Conv2d(in_planes // 16, in_planes, 1, bias=False)

        self.sigmoid = nn.Sigmoid()

    def forward(self, x):
        avg_out = self.fc2(self.relu1(self.fc1(self.avg_pool(x))))
        max_out = self.fc2(self.relu1(self.fc1(self.max_pool(x))))
        out = avg_out + max_out
        return self.sigmoid(out)
```

```
class SpatialAttention(nn.Module):
    def __init__(self, kernel_size=7):
        super(SpatialAttention, self).__init__()

        assert kernel_size in (3, 7), 'kernel size must be 3 or 7'
        padding = 3 if kernel_size == 7 else 1

        self.conv1 = nn.Conv2d(2, 1, kernel_size, padding=padding, bias=False)
        self.sigmoid = nn.Sigmoid()

    def forward(self, x):
        avg_out = torch.mean(x, dim=1, keepdim=True)
        max_out, _ = torch.max(x, dim=1, keepdim=True)
        x = torch.cat([avg_out, max_out], dim=1)
        x = self.conv1(x)
        return self.sigmoid(x)

class BasicBlock(nn.Module):
    expansion = 1

    def __init__(self, inplanes, planes, stride=1, downsample=None):
        super(BasicBlock, self).__init__()
        self.conv1 = conv3x3(inplanes, planes, stride)
        self.bn1 = nn.BatchNorm2d(planes)
        self.relu = nn.ReLU(inplace=True)
        self.conv2 = conv3x3(planes, planes)
        self.bn2 = nn.BatchNorm2d(planes)

        self.ca = ChannelAttention(planes)
        self.sa = SpatialAttention()

        self.downsample = downsample
        self.stride = stride

    def forward(self, x):
        residual = x

        out = self.conv1(x)
        out = self.bn1(out)
        out = self.relu(out)

        out = self.conv2(out)
        out = self.bn2(out)

        out = self.ca(out) * out
        out = self.sa(out) * out
```

```
        if self.downsample is not None:
            residual = self.downsample(x)

        out += residual
        out = self.relu(out)

        return out


class Bottleneck(nn.Module):
    expansion = 4

    def __init__(self, inplanes, planes, stride=1, downsample=None):
        super(Bottleneck, self).__init__()
        self.conv1 = nn.Conv2d(inplanes, planes, kernel_size=1, bias=False)
        self.bn1 = nn.BatchNorm2d(planes)
        self.conv2 = nn.Conv2d(planes, planes, kernel_size=3, stride=stride,
                               padding=1, bias=False)
        self.bn2 = nn.BatchNorm2d(planes)
        self.conv3 = nn.Conv2d(planes, planes * 4, kernel_size=1, bias=False)
        self.bn3 = nn.BatchNorm2d(planes * 4)
        self.relu = nn.ReLU(inplace=True)

        self.ca = ChannelAttention(planes * 4)
        self.sa = SpatialAttention()

        self.downsample = downsample
        self.stride = stride

    def forward(self, x):
        residual = x

        out = self.conv1(x)
        out = self.bn1(out)
        out = self.relu(out)

        out = self.conv2(out)
        out = self.bn2(out)
        out = self.relu(out)

        out = self.conv3(out)
        out = self.bn3(out)

        out = self.ca(out) * out
        out = self.sa(out) * out
```

```
        if self.downsample is not None:
            residual = self.downsample(x)

        out += residual
        out = self.relu(out)

        return out


class ResNet(nn.Module):

    def __init__(self, block, layers, num_classes=11):
        self.inplanes = 64
        super(ResNet, self).__init__()
        self.conv1 = nn.Conv2d(3, 64, kernel_size=7, stride=2, padding=3,
                               bias=False)
        self.bn1 = nn.BatchNorm2d(64)
        self.relu = nn.ReLU(inplace=True)
        self.maxpool = nn.MaxPool2d(kernel_size=3, stride=2, padding=1)
        self.layer1 = self._make_layer(block, 64, layers[0])
        self.layer2 = self._make_layer(block, 128, layers[1], stride=2)
        self.layer3 = self._make_layer(block, 256, layers[2], stride=2)
        self.layer4 = self._make_layer(block, 512, layers[3], stride=2)
        self.avgpool = nn.AvgPool2d(7, stride=1)
        self.fc = nn.Linear(512 * block.expansion, num_classes)

        for m in self.modules():
            if isinstance(m, nn.Conv2d):
                n = m.kernel_size[0] * m.kernel_size[1] * m.out_channels
                m.weight.data.normal_(0, math.sqrt(2. / n))
            elif isinstance(m, nn.BatchNorm2d):
                m.weight.data.fill_(1)
                m.bias.data.zero_()

    def _make_layer(self, block, planes, blocks, stride=1):
        downsample = None
        if stride != 1 or self.inplanes != planes * block.expansion:
            downsample = nn.Sequential(
                nn.Conv2d(self.inplanes, planes * block.expansion,
                          kernel_size=1, stride=stride, bias=False),
                nn.BatchNorm2d(planes * block.expansion),
            )

        layers = []
        layers.append(block(self.inplanes, planes, stride, downsample))
        self.inplanes = planes * block.expansion
        for i in range(1, blocks):
```

```
            layers.append(block(self.inplanes, planes))

        return nn.Sequential(*layers)

    def forward(self, x):
        x = self.conv1(x)
        x = self.bn1(x)
        x = self.relu(x)
        x = self.maxpool(x)

        x = self.layer1(x)
        x = self.layer2(x)
        x = self.layer3(x)
        x = self.layer4(x)

        x = self.avgpool(x)
        x = x.view(x.size(0), -1)
        x = self.fc(x)

        return x


def resnet18_cbam(pretrained=False, **kwargs):
    """Constructs a ResNet-18 model.
    Args:
        pretrained (bool): If True, returns a model pre-trained on ImageNet
    """
    #model = ResNet(BasicBlock, [2, 2, 2, 2], **kwargs)
    model = ResNet(BasicBlock, [1, 1, 1, 1], **kwargs)
    if pretrained:
        pretrained_state_dict = model_zoo.load_url(model_urls['resnet18'])
        now_state_dict        = model.state_dict()
        now_state_dict.update(pretrained_state_dict)
        model.load_state_dict(now_state_dict)
    return model


def resnet34_cbam(pretrained=False, **kwargs):
    """Constructs a ResNet-34 model.
    Args:
        pretrained (bool): If True, returns a model pre-trained on ImageNet
    """
    model = ResNet(BasicBlock, [3, 4, 6, 3], **kwargs)
    if pretrained:
        pretrained_state_dict = model_zoo.load_url(model_urls['resnet34'])
        now_state_dict        = model.state_dict()
        now_state_dict.update(pretrained_state_dict)
```

```
        model.load_state_dict(now_state_dict)
    return model


def resnet50_cbam(pretrained=False, **kwargs):
    """Constructs a ResNet-50 model.
    Args:
        pretrained (bool): If True, returns a model pre-trained on ImageNet
    """
    model = ResNet(Bottleneck, [3, 4, 6, 3], **kwargs)
    if pretrained:
        pretrained_state_dict = model_zoo.load_url(model_urls['resnet50'])
        now_state_dict        = model.state_dict()
        now_state_dict.update(pretrained_state_dict)
        model.load_state_dict(now_state_dict)
    return model


def resnet101_cbam(pretrained=False, **kwargs):
    """Constructs a ResNet-101 model.
    Args:
        pretrained (bool): If True, returns a model pre-trained on ImageNet
    """
    model = ResNet(Bottleneck, [3, 4, 23, 3], **kwargs)
    if pretrained:
        pretrained_state_dict = model_zoo.load_url(model_urls['resnet101'])
        now_state_dict        = model.state_dict()
        now_state_dict.update(pretrained_state_dict)
        model.load_state_dict(now_state_dict)
    return model


def resnet152_cbam(pretrained=False, **kwargs):
    """Constructs a ResNet-152 model.
    Args:
        pretrained (bool): If True, returns a model pre-trained on ImageNet
    """
    model = ResNet(Bottleneck, [3, 8, 36, 3], **kwargs)
    if pretrained:
        pretrained_state_dict = model_zoo.load_url(model_urls['resnet152'])
        now_state_dict        = model.state_dict()
        now_state_dict.update(pretrained_state_dict)
        model.load_state_dict(now_state_dict)
    return model
```

Define the loss function and optimizer.

[14]:
```
# In[]
net = resnet18_cbam()
net = torch.load('/content/drive/My Drive/Colab Notebooks/res18_40_128_old.pkl')
net = net.cuda()
print(net)
criterion = nn.CrossEntropyLoss()
criterion = criterion.cuda()
optimizer = optim.Adam(net.parameters(), lr=0.001)
```

```
ResNet(
  (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3),
bias=False)
  (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  (relu): ReLU(inplace)
  (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1,
ceil_mode=False)
  (layer1): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (ca): ChannelAttention(
        (avg_pool): AdaptiveAvgPool2d(output_size=1)
        (max_pool): AdaptiveMaxPool2d(output_size=1)
        (fc1): Conv2d(64, 4, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (relu1): ReLU()
        (fc2): Conv2d(4, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (sigmoid): Sigmoid()
      )
      (sa): SpatialAttention(
        (conv1): Conv2d(2, 1, kernel_size=(7, 7), stride=(1, 1), padding=(3, 3),
bias=False)
        (sigmoid): Sigmoid()
      )
    )
    (1): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
```

```
      (relu): ReLU(inplace)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (ca): ChannelAttention(
        (avg_pool): AdaptiveAvgPool2d(output_size=1)
        (max_pool): AdaptiveMaxPool2d(output_size=1)
        (fc1): Conv2d(64, 4, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (relu1): ReLU()
        (fc2): Conv2d(4, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (sigmoid): Sigmoid()
      )
      (sa): SpatialAttention(
        (conv1): Conv2d(2, 1, kernel_size=(7, 7), stride=(1, 1), padding=(3, 3),
bias=False)
        (sigmoid): Sigmoid()
      )
    )
  )
  (layer2): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (ca): ChannelAttention(
        (avg_pool): AdaptiveAvgPool2d(output_size=1)
        (max_pool): AdaptiveMaxPool2d(output_size=1)
        (fc1): Conv2d(128, 8, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (relu1): ReLU()
        (fc2): Conv2d(8, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (sigmoid): Sigmoid()
      )
      (sa): SpatialAttention(
        (conv1): Conv2d(2, 1, kernel_size=(7, 7), stride=(1, 1), padding=(3, 3),
bias=False)
        (sigmoid): Sigmoid()
      )
      (downsample): Sequential(
        (0): Conv2d(64, 128, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
```

```
      )
    )
    (1): BasicBlock(
      (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (ca): ChannelAttention(
        (avg_pool): AdaptiveAvgPool2d(output_size=1)
        (max_pool): AdaptiveMaxPool2d(output_size=1)
        (fc1): Conv2d(128, 8, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (relu1): ReLU()
        (fc2): Conv2d(8, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (sigmoid): Sigmoid()
      )
      (sa): SpatialAttention(
        (conv1): Conv2d(2, 1, kernel_size=(7, 7), stride=(1, 1), padding=(3, 3),
bias=False)
        (sigmoid): Sigmoid()
      )
    )
  )
  (layer3): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (ca): ChannelAttention(
        (avg_pool): AdaptiveAvgPool2d(output_size=1)
        (max_pool): AdaptiveMaxPool2d(output_size=1)
        (fc1): Conv2d(256, 16, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (relu1): ReLU()
        (fc2): Conv2d(16, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (sigmoid): Sigmoid()
      )
      (sa): SpatialAttention(
        (conv1): Conv2d(2, 1, kernel_size=(7, 7), stride=(1, 1), padding=(3, 3),
```

```
bias=False)
        (sigmoid): Sigmoid()
      )
      (downsample): Sequential(
        (0): Conv2d(128, 256, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
    )
    (1): BasicBlock(
      (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (ca): ChannelAttention(
        (avg_pool): AdaptiveAvgPool2d(output_size=1)
        (max_pool): AdaptiveMaxPool2d(output_size=1)
        (fc1): Conv2d(256, 16, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (relu1): ReLU()
        (fc2): Conv2d(16, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (sigmoid): Sigmoid()
      )
      (sa): SpatialAttention(
        (conv1): Conv2d(2, 1, kernel_size=(7, 7), stride=(1, 1), padding=(3, 3),
bias=False)
        (sigmoid): Sigmoid()
      )
    )
  )
  (layer4): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (ca): ChannelAttention(
        (avg_pool): AdaptiveAvgPool2d(output_size=1)
        (max_pool): AdaptiveMaxPool2d(output_size=1)
```

```
        (fc1): Conv2d(512, 32, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (relu1): ReLU()
        (fc2): Conv2d(32, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (sigmoid): Sigmoid()
      )
      (sa): SpatialAttention(
        (conv1): Conv2d(2, 1, kernel_size=(7, 7), stride=(1, 1), padding=(3, 3),
bias=False)
        (sigmoid): Sigmoid()
      )
      (downsample): Sequential(
        (0): Conv2d(256, 512, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
    )
    (1): BasicBlock(
      (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (ca): ChannelAttention(
        (avg_pool): AdaptiveAvgPool2d(output_size=1)
        (max_pool): AdaptiveMaxPool2d(output_size=1)
        (fc1): Conv2d(512, 32, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (relu1): ReLU()
        (fc2): Conv2d(32, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (sigmoid): Sigmoid()
      )
      (sa): SpatialAttention(
        (conv1): Conv2d(2, 1, kernel_size=(7, 7), stride=(1, 1), padding=(3, 3),
bias=False)
        (sigmoid): Sigmoid()
      )
    )
  )
  (avgpool): AvgPool2d(kernel_size=7, stride=1, padding=0)
  (fc): Linear(in_features=512, out_features=11, bias=True)
)
```

Start training, one batch is 256, train 20 epochs.

[16]:
```
print("Start Training...")
LOSS = []
num_epoch = 20
for epoch in range(num_epoch):
    # 10batchloss
    loss10 = 0.0
    # dataloader
    for i, data in enumerate(trainloader):
        inputs, labels = data
        #print(labels)
        optimizer.zero_grad()
        inputs = inputs.cuda()
        labels = labels.cuda()
        #print(torch.cuda.current_device())
        outputs = net(inputs)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
        loss10 += loss.item()
        if i % 10 == 9:
            print('[Epoch %d, Batch %5d] loss: %.3f' %
                  (epoch + 1, i + 1, loss10 / 10))
            LOSS.append(loss10/10)
            loss10 = 0.0

print("Done Training!")
```

```
Start Training...
[Epoch 1, Batch    10] loss: 0.061
[Epoch 1, Batch    20] loss: 0.068
[Epoch 2, Batch    10] loss: 0.069
[Epoch 2, Batch    20] loss: 0.092
[Epoch 3, Batch    10] loss: 0.086
[Epoch 3, Batch    20] loss: 0.062
[Epoch 4, Batch    10] loss: 0.078
[Epoch 4, Batch    20] loss: 0.087
[Epoch 5, Batch    10] loss: 0.087
[Epoch 5, Batch    20] loss: 0.084
[Epoch 6, Batch    10] loss: 0.081
[Epoch 6, Batch    20] loss: 0.059
[Epoch 7, Batch    10] loss: 0.083
[Epoch 7, Batch    20] loss: 0.062
[Epoch 8, Batch    10] loss: 0.067
[Epoch 8, Batch    20] loss: 0.073
[Epoch 9, Batch    10] loss: 0.089
[Epoch 9, Batch    20] loss: 0.057
[Epoch 10, Batch    10] loss: 0.070
```

```
[Epoch 10, Batch    20] loss: 0.078
[Epoch 11, Batch    10] loss: 0.097
[Epoch 11, Batch    20] loss: 0.075
[Epoch 12, Batch    10] loss: 0.080
[Epoch 12, Batch    20] loss: 0.079
[Epoch 13, Batch    10] loss: 0.055
[Epoch 13, Batch    20] loss: 0.066
[Epoch 14, Batch    10] loss: 0.081
[Epoch 14, Batch    20] loss: 0.081
[Epoch 15, Batch    10] loss: 0.083
[Epoch 15, Batch    20] loss: 0.067
[Epoch 16, Batch    10] loss: 0.072
[Epoch 16, Batch    20] loss: 0.061
[Epoch 17, Batch    10] loss: 0.082
[Epoch 17, Batch    20] loss: 0.077
[Epoch 18, Batch    10] loss: 0.086
[Epoch 18, Batch    20] loss: 0.101
[Epoch 19, Batch    10] loss: 0.068
[Epoch 19, Batch    20] loss: 0.057
[Epoch 20, Batch    10] loss: 0.046
[Epoch 20, Batch    20] loss: 0.081
Done Training!
```

Evaluation

[17]:
```
# In[eval]
summ = 0
for ii in range(100):
  test_dataset = ImageFolder(root='/content/drive/My Drive/Colab Notebooks/
 ↪test_set',transform=data_transform)
  testloader = torch.utils.data.DataLoader(test_dataset,batch_size=46,␣
 ↪shuffle=True)
  eval_loss = 0
  correct = 0
  total = 0
  for i, data in enumerate(testloader):
    inputs, labels = data
    #print(labels)

    inputs = inputs.cuda()
    labels = labels.cuda()
    outputs = net(inputs)
    loss = criterion(outputs, labels)
    eval_loss += loss.item()
    total += labels.size(0)
    pred = torch.max(outputs, 1)[1]
    print('pred: ',pred,' labels: ',labels)
    correct += (pred == labels).sum().item()
```

```
    summ += (100 * correct / total)
  print('eval_loss: ',eval_loss)
  print('correct: ',correct)
  print('Accuracy of the network on the test images: %d %%' % (100 * correct /␣
 ↪total))
print("total_acc: ", summ/100)
```

```
pred:  tensor([ 9,  3,  9,  0,  9,  3,  1,  4,  4,  6, 10,  7, 10,  2,  4,  7,
3,  7,
         7,  5,  1,  8,  5,  3,  2,  2,  1,  3,  6,  2,  3,  5,  3,  4,  1,  2,
         3, 10,  1,  8,  2,  6,  9,  9,  6,  2], device='cuda:0')  labels:
tensor([ 9,  3,  3,  0,  9,  3,  1,  4,  4,  5, 10,  7,  3,  2,  4,  7,  3,  7,
         1,  5,  1,  8,  5, 10,  2,  2,  5,  3,  6,  2,  0,  1,  3,  4,  1,  2,
         3,  3,  1,  8,  2,  6,  9,  9,  6,  2], device='cuda:0')
eval_loss:  1.1506589651107788
correct:  37
Accuracy of the network on the test images: 80 %
pred:  tensor([ 1,  2,  9,  0,  3,  2,  2,  5,  9,  3,  3,  4,  1,  4,  2, 10,
6,  7,
         8,  3,  0,  4,  2,  3,  7,  1,  5,  1,  8,  2, 10,  5,  3,  9,  7,  7,
         6,  1,  8,  4,  2,  3,  3,  0,  9,  6], device='cuda:0')  labels:
tensor([ 1,  2,  9,  3,  3,  2,  2,  5,  9,  3,  3,  4,  1,  4,  2,  3,  6,  3,
         8,  0,  0,  4,  2,  3,  7,  1,  1,  1,  5,  2, 10,  5,  3,  9,  7,  7,
         6,  5,  8,  4,  2,  3, 10,  1,  9,  6], device='cuda:0')
eval_loss:  0.8940167427062988
correct:  37
Accuracy of the network on the test images: 80 %
pred:  tensor([ 3,  9,  2,  2,  9,  0,  7,  6,  7,  3,  6,  8,  3,  4,  6,  2,
0, 10,
         2,  3,  3,  5, 10,  1,  3,  3,  7,  4,  1,  8,  3,  5,  1,  6,  4,  0,
         5,  9,  2,  2,  1,  4,  9,  3,  2,  8], device='cuda:0')  labels:
tensor([ 3,  9,  2,  2,  9,  0,  7,  5,  7, 10,  6,  8,  3,  4,  6,  2,  0,  3,
         2,  3,  3,  5, 10,  1,  3,  3,  7,  4,  1,  8,  3,  1,  1,  6,  4,  1,
         5,  9,  2,  2,  1,  4,  9,  3,  2,  5], device='cuda:0')
eval_loss:  0.8593901991844177
correct:  40
Accuracy of the network on the test images: 86 %
pred:  tensor([ 2,  6,  6,  4,  1,  3,  6,  1,  3,  9,  1,  5,  4,  4,  1,  3,
10,  9,
         9,  5, 10,  3,  1,  2,  0,  2,  9,  8,  2,  7,  4,  3,  6, 10,  2,  3,
         2,  8,  8,  3,  3,  2,  7,  7,  1,  8], device='cuda:0')  labels:
tensor([ 2,  6,  6,  4,  1,  3,  6,  1,  3,  9,  1,  5,  4,  4,  5,  3,  3,  9,
         9,  1, 10,  3,  1,  2,  0,  2,  9,  3,  2,  7,  4,  3,  5,  0,  2, 10,
         2,  5,  8,  3,  3,  2,  7,  7,  1,  8], device='cuda:0')
eval_loss:  0.9626793265342712
correct:  38
Accuracy of the network on the test images: 82 %
```

```
pred:  tensor([ 1,  5,  6,  2,  4,  4,  9,  3,  7,  3,  5,  7,  3, 10,  9,  6,
1,  3,
        9,  6,  5,  3,  9,  2,  2,  3,  6,  2,  1,  3, 10,  8,  3,  1,  2,  7,
        1,  3,  2,  4,  4,  3,  9,  2,  0,  1], device='cuda:0')  labels:
tensor([ 1,  1,  6,  2,  4,  4,  9,  3,  7,  3,  5,  7,  3, 10,  9,  6,  1,  0,
        8,  6,  5,  3,  9,  2,  2, 10,  5,  2,  1,  3,  3,  8,  3,  5,  2,  7,
        1,  3,  2,  4,  4,  3,  9,  2,  0,  1], device='cuda:0')
eval_loss:  0.9515205025672913
correct:  39
Accuracy of the network on the test images: 84 %
pred:  tensor([ 5,  1,  7, 10,  0,  2,  9,  4,  1,  2,  1,  4,  4,  3,  6,  9,
3,  9,
        5,  4,  6,  6,  6,  2,  3,  2,  1,  8,  3,  4,  3,  7,  1,  3,  0,  2,
        3,  9,  7,  5,  9,  9,  2, 10,  5,  3], device='cuda:0')  labels:
tensor([ 1,  5,  7,  3,  0,  2,  9,  4,  1,  2,  1,  4,  4,  3,  6,  8,  3,  9,
        5,  2,  5,  6,  6,  2,  3,  2,  1,  8,  3,  4,  0,  7,  1,  3, 10,  2,
        3,  3,  7,  1,  9,  9,  2, 10,  5,  3], device='cuda:0')
eval_loss:  1.2341374158859253
correct:  36
Accuracy of the network on the test images: 78 %
pred:  tensor([ 2,  5,  7,  4,  3,  4,  4,  3,  6,  3,  2,  3,  2,  6,  7,  3,
1,  9,
        1,  6,  0,  9,  9,  2,  3, 10,  7,  1, 10,  5,  9,  8,  2,  2,  1,  2,
        9,  8,  4,  5,  3,  3,  1,  6,  3,  1], device='cuda:0')  labels:
tensor([ 2,  1,  7,  4,  0,  4,  4,  3,  6,  3,  2,  3,  2,  6,  7,  3,  1, 10,
        1,  6,  0,  9,  9,  2,  3, 10,  7,  1,  3,  5,  9,  8,  2,  2,  1,  2,
        9,  8,  4,  5,  3,  3,  5,  5,  3,  1], device='cuda:0')
eval_loss:  0.7958880066871643
correct:  40
Accuracy of the network on the test images: 86 %
pred:  tensor([ 8,  6,  1, 10,  9,  0,  6,  7,  3,  3,  3,  1,  2,  3,  2,  1,
6,  7,
        6,  9,  9,  2,  4,  1,  2, 10,  3,  6,  3,  4,  1,  2,  3,  3,  3,  5,
        7,  0,  2,  2,  5,  9,  4,  8,  3,  4], device='cuda:0')  labels:
tensor([ 8,  6,  1,  3,  9,  3,  6,  7,  0,  3,  3,  1,  2,  3,  2,  1,  5,  7,
        5,  9,  9,  2,  4,  1,  2, 10,  3,  6,  3,  4,  5,  2,  3, 10,  3,  1,
        7,  0,  2,  2,  5,  9,  4,  8,  1,  4], device='cuda:0')
eval_loss:  1.0551252365112305
correct:  37
Accuracy of the network on the test images: 80 %
pred:  tensor([ 4,  6,  2,  6,  8,  1,  7,  4,  2,  7,  3,  1,  3, 10,  5,  4,
2,  8,
        3,  1,  3,  6,  9,  2,  1,  3,  3,  9,  3,  2,  9,  6,  2,  4,  9,  8,
        2,  7,  3,  3,  5,  0,  3,  1,  1, 10], device='cuda:0')  labels:
tensor([ 4,  6,  2,  5,  5,  5,  7,  4,  2,  7,  3,  1, 10, 10,  5,  4,  2,  8,
        3,  1,  3,  6,  9,  2,  1,  3,  3,  9,  3,  2,  9,  6,  2,  4,  9,  8,
        2,  7,  3,  0,  1,  0,  3,  1,  1,  3], device='cuda:0')
eval_loss:  1.237066388130188
```

```
correct:  39
Accuracy of the network on the test images: 84 %
pred:  tensor([ 6,  6,  7,  2,  4,  1,  2,  2,  4,  6,  1,  3,  1,  0,  3,  9,
3,  4,
         2,  9,  5,  9,  1,  2,  1,  5,  3,  3,  6,  9,  2,  1, 10,  7,  8,  0,
         4,  2,  3,  3,  5,  1,  9,  7,  7,  3], device='cuda:0')  labels:
tensor([ 6,  6,  7,  2,  4,  1,  2,  2,  4,  6,  5,  3,  1,  0,  3,  8,  3,  4,
         2,  9,  5,  9,  3,  2,  1,  5,  3,  3,  5,  9,  2,  1, 10,  7,  8, 10,
         4,  2,  0,  3,  1,  1,  9,  3,  7,  3], device='cuda:0')
eval_loss:  0.6804027557373047
correct:  38
Accuracy of the network on the test images: 82 %
pred:  tensor([ 2,  4,  7,  4,  2,  9,  6,  9,  9,  4,  2,  3,  8,  7, 10, 10,
3,  3,
         3,  7,  7,  1,  1,  8,  2,  7,  5,  4,  5,  0,  2,  3,  1,  9,  1, 10,
         1,  6,  3,  5,  6,  3,  2,  6,  3,  1], device='cuda:0')  labels:
tensor([ 2,  4,  2,  4,  2,  9,  6,  9,  9,  4,  2, 10,  8,  7,  3,  0,  3,  3,
         3,  3,  7,  1,  1,  8,  2,  7,  5,  4,  5,  0,  2,  3,  1,  9,  1, 10,
         1,  6,  3,  1,  6,  3,  2,  5,  3,  5], device='cuda:0')
eval_loss:  0.9073984622955322
correct:  38
Accuracy of the network on the test images: 82 %
pred:  tensor([ 3,  2,  0,  0,  5,  2,  5,  4, 10,  6,  2,  3,  9,  0,  4,  2,
7,  6,
         6,  2,  6,  7,  3,  1,  4,  2,  3,  3,  8,  4,  9,  3,  3,  1,  9, 10,
         2,  1,  3,  9,  1,  7,  6,  3,  9,  1], device='cuda:0')  labels:
tensor([ 3,  2,  0, 10,  5,  2,  1,  4,  3,  5,  2,  3,  9,  1,  4,  2,  7,  6,
         5,  2,  6,  7,  3,  1,  4,  2,  3,  3,  8,  4,  9,  3,  3,  5,  9, 10,
         2,  1,  0,  8,  1,  7,  6,  3,  9,  1], device='cuda:0')
eval_loss:  1.0935393571853638
correct:  37
Accuracy of the network on the test images: 80 %
pred:  tensor([ 0,  9,  9, 10,  4,  0,  2,  6,  3,  3,  5,  6,  1,  3,  9,  7,
2,  2,
         5, 10,  3,  3,  3,  4,  2, 10,  6,  7,  8,  2,  1,  3,  3,  6,  2,  2,
         1,  9,  9,  1,  4,  6,  7,  1,  1,  4], device='cuda:0')  labels:
tensor([ 0,  9,  8,  3,  4,  3,  2,  6,  3,  3,  5,  6,  5, 10,  9,  7,  2,  2,
         1, 10,  3,  3,  3,  4,  2,  0,  5,  7,  8,  2,  1,  3,  3,  5,  2,  2,
         1,  9,  9,  1,  4,  6,  7,  1,  1,  4], device='cuda:0')
eval_loss:  1.1818592548370361
correct:  37
Accuracy of the network on the test images: 80 %
pred:  tensor([ 2,  5,  2,  1,  4,  4,  4,  3,  3,  2,  6,  1,  3,  7,  3,  6,
1,  9,
         2,  9,  8,  6,  3,  3,  2,  4,  9, 10, 10,  2,  3,  6, 10,  3,  3,  1,
         2,  6,  7,  1,  1,  9,  1,  9,  3,  5], device='cuda:0')  labels:
tensor([ 2,  5,  2,  5,  4,  4,  4,  0,  3,  2,  5,  7,  3,  7,  3,  6,  1,  9,
         2,  9,  8,  6,  3,  3,  2,  4,  8, 10,  3,  2,  3,  6,  0,  3, 10,  1,
```

```
        2,  5,  7,  1,  1,  9,  1,  9,  3,  1], device='cuda:0')
eval_loss:  0.8753384351730347
correct:  36
Accuracy of the network on the test images: 78 %
pred:  tensor([ 9,  9,  9,  3,  8,  6,  1,  7,  5,  4,  5,  6,  4, 10,  2,  3,
2,  7,
        4,  3,  7,  4,  3,  9,  6,  2,  1, 10,  3,  9,  6,  2,  2,  3,  2,  1,
       10,  3,  1,  3,  2,  1,  9,  3,  6,  9], device='cuda:0')  labels:
tensor([ 3,  8,  9,  3,  3,  5,  5,  7,  5,  4,  1,  6,  4,  3,  2,  3,  2,  7,
        4,  3,  7,  4,  3,  9,  5,  2,  1, 10, 10,  8,  6,  2,  2,  3,  2,  1,
        0,  0,  1,  1,  2,  1,  9,  3,  6,  9], device='cuda:0')
eval_loss:  1.6672890186309814
correct:  33
Accuracy of the network on the test images: 71 %
pred:  tensor([ 8,  4,  7,  9,  8,  3,  2,  4,  0, 10,  5,  8,  1,  8,  9,  6,
1,  3,
        1,  2,  6,  3,  2,  5,  3,  2,  9,  5,  1,  1,  2,  3,  3,  7,  3,  3,
        1,  7,  2,  9,  2,  6,  9,  4,  4, 10], device='cuda:0')  labels:
tensor([10,  4,  7,  9,  3,  3,  2,  4,  0,  3,  1,  8,  1,  5,  9,  6,  1,  3,
        5,  2,  6,  3,  2,  5,  3,  2,  9,  5,  1,  1,  2,  3,  3,  7,  0,  3,
        1,  7,  2,  9,  2,  6,  8,  4,  4, 10], device='cuda:0')
eval_loss:  0.8089241981506348
correct:  38
Accuracy of the network on the test images: 82 %
pred:  tensor([ 4,  2,  3,  3,  4,  6,  6,  3,  2,  6,  4,  2,  1,  3,  6,  7,
1,  5,
        8,  4,  2,  9,  7,  2,  3,  1,  3,  0, 10,  3,  7,  1,  9,  2,  1,  5,
        1,  9,  9, 10,  3,  8,  6,  2,  3,  3], device='cuda:0')  labels:
tensor([ 4,  2,  3,  3,  4,  6,  6,  3,  2,  5,  4,  2,  1,  3,  6,  7,  1,  5,
        8,  4,  2,  9,  7,  2,  3,  1,  3,  0,  3, 10,  7,  1,  9,  2,  5,  1,
        1,  9,  9, 10,  3,  8,  5,  2,  3,  0], device='cuda:0')
eval_loss:  1.0011593103408813
correct:  39
Accuracy of the network on the test images: 84 %
pred:  tensor([ 6,  2,  1,  7,  3,  9,  3,  4,  1,  2,  1,  7,  4,  5, 10,  6,
3,  3,
       10,  6,  3,  3,  2,  0,  1,  3,  2,  5, 10,  2,  9,  6,  3,  7,  2,  2,
        3,  5,  1,  6,  4,  9,  9,  4,  9,  9], device='cuda:0')  labels:
tensor([ 6,  2,  1,  7,  3,  9,  3,  4,  1,  2,  1,  7,  4,  1,  3,  6,  3,  3,
       10,  5, 10,  1,  2,  0,  5,  3,  2,  5,  0,  2,  8,  5,  3,  7,  2,  2,
        3,  8,  1,  6,  4,  9,  9,  4,  9,  3], device='cuda:0')
eval_loss:  1.1158725023269653
correct:  35
Accuracy of the network on the test images: 76 %
pred:  tensor([ 7,  3,  1,  3,  0,  3, 10,  5,  0, 10,  2,  7,  2,  8,  8,  1,
1,  4,
        5,  2,  9,  5,  8,  3,  9,  2,  3,  3,  1,  9,  3,  3,  1,  7,  4,  6,
        4,  4, 10,  2,  6,  2,  9,  2,  6,  6], device='cuda:0')  labels:
```

```
tensor([ 7,  3,  1,  3, 10,  3, 10,  1,  3,  3,  2,  7,  2,  8,  8,  5,  1,  4,
         5,  2,  9,  5,  1,  3,  9,  2,  0,  3,  1,  9,  3,  3,  1,  7,  4,  6,
         4,  4,  0,  2,  5,  2,  9,  2,  6,  6], device='cuda:0')
eval_loss:  1.2914353609085083
correct:  37
Accuracy of the network on the test images: 80 %
pred:  tensor([ 2,  4,  9,  2,  4, 10,  0,  3,  9,  2,  2,  6,  6,  6,  9,  4,
0,  3,
         9,  7,  7,  3,  7,  1, 10,  8, 10,  5,  2,  1,  4,  1,  3,  6,  1,  8,
         3,  1,  3,  5,  9,  2,  5,  2,  1,  3], device='cuda:0')  labels:
tensor([ 2,  4,  3,  2,  4,  3,  0,  3,  9,  2,  2,  6,  5,  6,  9,  4, 10,  3,
         9,  7,  7,  3,  7,  1,  3,  8, 10,  5,  2,  1,  4,  1,  0,  6,  5,  8,
         3,  1,  3,  1,  9,  2,  5,  2,  1,  3], device='cuda:0')
eval_loss:  1.1329810619354248
correct:  38
Accuracy of the network on the test images: 82 %
pred:  tensor([ 9,  3,  1,  9,  3,  3,  2,  3,  1,  1,  7,  6,  6,  2,  3,  9,
4,  6,
         3,  5,  7,  3,  3,  5,  8,  5,  4,  9,  7,  1,  3,  0,  6,  2,  2,  2,
        10,  7,  2,  4,  4,  7,  1,  1,  2, 10], device='cuda:0')  labels:
tensor([ 9,  3,  1,  9,  3,  0,  2,  3,  1,  1,  7,  6,  5,  2, 10,  8,  4,  6,
         3,  5,  3,  3,  3,  5,  8,  1,  4,  9,  7,  1,  3,  0,  6,  2,  2,  2,
         9,  7,  2,  4,  4,  3,  1,  5,  2, 10], device='cuda:0')
eval_loss:  1.0164874792099
correct:  37
Accuracy of the network on the test images: 80 %
pred:  tensor([ 6,  4,  3,  3,  8,  9,  5,  8,  6,  5,  3, 10,  1,  3,  9,  1,
2, 10,
         3,  6,  9,  2,  3,  3,  2,  7,  3,  2,  9,  7,  1,  2,  4, 10,  0,  2,
         1,  8,  4,  2,  5,  1,  7,  9,  4,  3], device='cuda:0')  labels:
tensor([ 6,  4,  3,  3,  5,  8,  5,  3,  6,  5,  3, 10,  1,  3,  9,  1,  2,  3,
         0,  6,  9,  2,  3,  3,  2,  7,  3,  2,  9,  7,  5,  2,  4,  0,  1,  2,
         1,  8,  4,  2,  1,  1,  7,  9,  4, 10], device='cuda:0')
eval_loss:  1.1444367170333862
correct:  36
Accuracy of the network on the test images: 78 %
pred:  tensor([ 6,  0,  1,  4,  3,  1,  5,  4,  6,  9,  5,  3, 10,  3,  7,  9,
8,  9,
         3,  2,  8,  0,  8,  1,  4,  6,  3,  1,  3,  2, 10,  0,  3,  2,  1,  2,
         4,  6,  2,  9,  9,  2,  7,  2,  7,  5], device='cuda:0')  labels:
tensor([ 6,  3,  5,  4,  3,  1,  5,  4,  6,  8,  5,  3,  3,  0,  7,  9,  3,  9,
         3,  2,  1, 10,  8,  1,  4,  6,  3,  1,  3,  2, 10,  0,  3,  2,  1,  2,
         4,  5,  2,  9,  9,  2,  7,  2,  7,  1], device='cuda:0')
eval_loss:  1.147971749305725
correct:  36
Accuracy of the network on the test images: 78 %
pred:  tensor([ 1,  4,  9,  7,  2,  3,  2,  6,  2,  6,  8,  9,  6,  9,  3,  8,
1,  7,
```

```
        4,  1,  5,  2,  1, 10, 10,  5,  6,  3,  3,  4,  1,  2,  4,  3,  1,  0,
        2,  0,  7,  9,  9,  3,  5,  2,  8,  3], device='cuda:0')  labels:
tensor([ 1,  4,  9,  7,  2,  3,  2,  5,  2,  6,  8,  9,  6,  3,  3,  8,  1,  7,
        4,  5,  5,  2,  1, 10,  3,  1,  6,  3,  3,  4,  1,  2,  4,  3,  1,  0,
        2,  0,  7,  9,  9, 10,  5,  2,  3,  3], device='cuda:0')
eval_loss:  0.9417474865913391
correct:  39
Accuracy of the network on the test images: 84 %
pred:  tensor([ 9,  3,  2,  4,  3,  9,  5,  2,  6,  3,  3,  6,  7,  4,  4,  3,
0,  7,
        2,  2,  3,  1,  1,  6,  2,  6,  0,  3, 10,  8,  9,  7, 10,  9,  5,  1,
        2,  6,  1,  1,  4,  3,  9,  8,  2,  1], device='cuda:0')  labels:
tensor([ 9,  3,  2,  4,  0,  9,  1,  2,  6,  3,  3,  5,  7,  4,  4,  3, 10,  7,
        2,  2,  3,  1,  1,  6,  2,  5,  0,  3, 10,  8,  3,  7,  3,  9,  5,  1,
        2,  6,  1,  5,  4,  3,  9,  8,  2,  1], device='cuda:0')
eval_loss:  1.0315037965774536
correct:  38
Accuracy of the network on the test images: 82 %
pred:  tensor([ 9,  2,  3,  2,  1,  7,  4,  3,  3,  8,  4,  2,  3,  2,  1,  1,
3,  0,
        3,  8,  6,  5,  6,  4,  6,  3,  8,  2,  2,  1,  3,  1, 10,  1,  9,  0,
        9, 10,  5,  4,  7,  9,  6,  7,  9,  2], device='cuda:0')  labels:
tensor([ 9,  2,  3,  2,  5,  7,  4,  0,  3,  8,  4,  2,  3,  2,  1,  1,  3,  3,
        3,  8,  6,  5,  5,  4,  6,  1,  5,  2,  2,  1, 10,  3, 10,  1,  9,  0,
        9,  3,  1,  4,  7,  3,  6,  7,  9,  2], device='cuda:0')
eval_loss:  1.277437448501587
correct:  35
Accuracy of the network on the test images: 76 %
pred:  tensor([ 3,  9,  1,  0,  9,  2,  3,  0,  3,  2,  1,  2,  9,  6,  7,  6,
3,  3,
        5,  2,  7,  4,  1,  9,  3,  3,  9,  1,  5,  8, 10,  0,  7,  6,  2,  4,
        4,  1,  2,  6,  0,  6,  7,  2,  4, 10], device='cuda:0')  labels:
tensor([ 0,  9,  1,  0,  8,  2,  3, 10,  3,  2,  5,  2,  9,  5,  7,  8,  3,  3,
        1,  2,  3,  4,  1,  9,  3,  3,  9,  1,  5,  5, 10,  3,  7,  6,  2,  4,
        4,  1,  2,  6,  1,  6,  7,  2,  4,  3], device='cuda:0')
eval_loss:  0.9665833115577698
correct:  34
Accuracy of the network on the test images: 73 %
pred:  tensor([ 3,  7,  2,  0,  4,  9,  9,  2,  3,  6, 10,  1,  9,  1,  1,  2,
5,  2,
        4,  6,  7,  8,  1,  9,  4,  3,  3,  5,  2,  2,  2,  7,  6,  6,  9,  4,
        7, 10,  3,  8,  6,  1,  3,  1,  3,  3], device='cuda:0')  labels:
tensor([ 3,  7,  2,  0,  4,  9,  8,  2, 10,  5,  3,  1,  9,  5,  1,  2,  1,  2,
        4,  6,  3,  3,  1,  9,  4,  3,  3,  5,  2,  2,  2,  7,  6,  5,  9,  4,
        7, 10,  3,  8,  6,  1,  3,  1,  3,  0], device='cuda:0')
eval_loss:  0.8976683616638184
correct:  36
Accuracy of the network on the test images: 78 %
```

```
pred:  tensor([ 3,  3,  9,  9,  9,  2,  3,  2,  7,  1,  1,  5,  1,  2,  2,  9,
3, 10,
         3,  2,  4,  7,  1,  0,  3,  2, 10,  6,  7,  3,  5,  4,  4,  2,  1,  8,
        10,  4,  3,  8,  6,  6,  3,  3,  8,  6], device='cuda:0')  labels:
tensor([ 3,  3,  9,  9,  9,  2,  3,  2,  7,  1,  1,  5,  1,  2,  2,  9,  3,  3,
         0,  2,  4,  7,  1,  1, 10,  2,  0,  6,  7,  3,  1,  4,  4,  2,  5,  8,
        10,  4,  3,  5,  6,  5,  3,  3,  8,  6], device='cuda:0')
eval_loss:  1.0257455110549927
correct:  37
Accuracy of the network on the test images: 80 %
pred:  tensor([ 5,  9,  1,  3,  3,  1,  6,  5,  4,  6,  3,  8,  1,  0,  9,  1,
6,  7,
         2,  4,  3,  2,  3,  2, 10, 10,  8,  2,  7,  2,  3,  3,  9,  1,  6,  7,
         1,  2,  9,  9,  7,  4,  4,  5,  3,  2], device='cuda:0')  labels:
tensor([ 5,  9,  1,  3, 10,  1,  6,  1,  4,  5,  3,  8,  1,  0,  9,  1,  6,  7,
         2,  4,  3,  2,  3,  2, 10,  3,  0,  2,  7,  2,  3,  3,  9,  1,  6,  7,
         5,  2,  9,  8,  3,  4,  4,  5,  3,  2], device='cuda:0')
eval_loss:  1.0082179307937622
correct:  38
Accuracy of the network on the test images: 82 %
pred:  tensor([10,  1,  3,  9,  5,  4,  1,  3,  3,  4,  2,  2,  3,  9,  0,  2,
1,  2,
         3,  2,  3,  7,  8,  2,  3,  6,  7,  0,  1,  5,  3,  2,  4,  1,  4,  6,
         1,  6, 10,  9,  7,  9,  3,  6,  3,  9], device='cuda:0')  labels:
tensor([ 3,  1,  3,  9,  5,  4,  1,  3,  0,  4,  2,  2,  3,  9,  1,  2,  1,  2,
         3,  2,  3,  7,  8,  2, 10,  6,  7,  0,  1,  5,  3,  2,  4,  1,  4,  5,
         5,  6, 10,  9,  7,  8,  3,  6,  3,  9], device='cuda:0')
eval_loss:  0.9767537117004395
correct:  39
Accuracy of the network on the test images: 84 %
pred:  tensor([ 3,  1,  7,  0,  9,  7,  2,  2, 10,  6,  9,  1,  7,  9,  8,  1,
3,  5,
         3, 10,  2,  3,  6,  2,  8,  6,  2,  2,  5,  3,  3,  3,  3,  4,  5,  4,
         4,  2,  1,  1,  9,  6,  0,  6,  4,  3], device='cuda:0')  labels:
tensor([ 3,  1,  7,  0,  9,  7,  2,  2,  3,  6,  9,  1,  7,  9,  8,  1,  3,  5,
        10, 10,  2,  3,  5,  2,  8,  6,  2,  2,  1,  3,  3,  3,  3,  4,  5,  4,
         4,  2,  1,  1,  9,  5,  0,  6,  4,  3], device='cuda:0')
eval_loss:  0.8576213121414185
correct:  41
Accuracy of the network on the test images: 89 %
pred:  tensor([ 5,  6,  0,  8,  4, 10,  2,  7,  5, 10,  9,  1,  8,  3,  2,  3,
2,  3,
         4,  6, 10,  8,  3,  4,  4,  1,  2,  3,  6,  9,  2,  7,  2,  9,  6,  3,
         6,  1,  7,  7,  9,  1,  3,  1,  9,  2], device='cuda:0')  labels:
tensor([ 1,  6,  0,  3,  4,  3,  2,  3,  5,  0,  9,  1,  3,  1,  2,  3,  2,  3,
         4,  6, 10,  8, 10,  4,  4,  1,  2,  3,  5,  9,  2,  7,  2,  8,  5,  3,
         6,  1,  7,  7,  9,  5,  3,  1,  9,  2], device='cuda:0')
eval_loss:  1.1682772636413574
```

```
correct:  34
Accuracy of the network on the test images: 73 %
pred:  tensor([ 8,  3,  2,  2,  0,  9,  2,  3,  5,  0,  4,  7, 10,  2,  5,  1,
9,  7,
         3,  6,  1,  3,  5,  3,  2,  6,  4,  3,  4,  8,  7,  9,  1,  3,  9,  4,
         1,  7,  6,  3,  3,  2,  1,  2, 10,  6], device='cuda:0')  labels:
tensor([ 8,  3,  2,  2,  0,  9,  2,  3,  5,  0,  4,  7, 10,  2,  1,  5,  9,  1,
        10,  6,  1,  3,  5,  3,  2,  6,  4,  3,  4,  8,  7,  9,  1,  7,  9,  4,
         1,  3,  5,  3,  3,  2,  1,  2,  3,  6], device='cuda:0')
eval_loss:  0.7388781905174255
correct:  38
Accuracy of the network on the test images: 82 %
pred:  tensor([ 6,  3,  3,  3, 10,  2,  7,  9,  9,  1,  6,  4,  1, 10,  7,  4,
1,  2,
        10,  0,  4,  3,  5,  3,  2,  1,  5,  3,  1,  2,  2,  8,  8,  9,  5,  3,
         6,  9,  0,  2,  3,  6,  4,  7,  3,  2], device='cuda:0')  labels:
tensor([ 6,  3,  3,  3,  0,  2,  7,  9,  9,  1,  5,  4,  1,  3,  7,  4,  1,  2,
        10,  1,  4,  3,  5,  3,  2,  1,  5, 10,  5,  2,  2,  8,  8,  9,  1,  3,
         6,  9,  0,  2,  3,  6,  4,  7,  3,  2], device='cuda:0')
eval_loss:  1.0103248357772827
correct:  39
Accuracy of the network on the test images: 84 %
pred:  tensor([ 4,  2,  3, 10,  2,  8,  6,  7,  2,  9,  2,  9,  1,  2,  5,  9,
9,  1,
         3,  2,  3,  1, 10,  5,  4,  2,  3,  7,  1,  5,  0,  6,  4,  1,  3,  3,
         9,  9,  7, 10,  4,  6,  3,  3,  3,  1], device='cuda:0')  labels:
tensor([ 4,  2,  3, 10,  2,  5,  6,  7,  2,  8,  2,  9,  1,  2,  5,  9,  9,  1,
         3,  2,  3,  5,  3,  5,  4,  2,  3,  7,  1,  1, 10,  6,  4,  1,  0,  3,
         8,  9,  7,  0,  4,  6,  3,  3,  3,  1], device='cuda:0')
eval_loss:  1.0838357210159302
correct:  37
Accuracy of the network on the test images: 80 %
pred:  tensor([ 3,  6,  2,  3,  8,  2,  4,  9,  1,  2,  7,  1,  9,  3,  3,  1,
10,  6,
         3,  7,  5,  5,  3,  2,  6,  9,  7,  4,  3, 10,  2,  1,  3,  8,  1,  6,
         2,  9,  6,  1,  4,  4,  0,  8,  2,  3], device='cuda:0')  labels:
tensor([ 3,  5,  2,  3,  8,  2,  4,  9,  1,  2,  7,  1,  9, 10,  3,  1,  3,  6,
         3,  7,  1,  5,  3,  2,  6,  9,  7,  4,  0, 10,  2,  1,  3,  3,  1,  6,
         2,  9,  5,  5,  4,  4,  0,  8,  2,  3], device='cuda:0')
eval_loss:  1.0529603958129883
correct:  38
Accuracy of the network on the test images: 82 %
pred:  tensor([ 7,  7,  7,  5,  9,  2,  9,  2,  4,  1,  3,  9,  7,  6,  6,  3,
8,  3,
         3,  3,  3,  8,  4,  9,  2,  3, 10,  3,  2,  1,  4,  1,  4,  1,  3,  0,
         2,  8,  1,  2,  5,  6,  2, 10,  6,  6], device='cuda:0')  labels:
tensor([ 7,  7,  7,  5,  8,  2,  9,  2,  4,  1,  3,  9,  3,  5,  6,  3,  3,  0,
         0,  3,  3,  9,  4,  9,  2,  3, 10, 10,  2,  1,  4,  5,  4,  1,  3,  1,
```

```
        2,  8,  1,  2,  1,  5,  2,  3,  6,  6], device='cuda:0')
eval_loss:  1.502564787864685
correct:  33
Accuracy of the network on the test images: 71 %
pred:  tensor([ 3,  9,  1,  6,  2,  5,  4,  4,  7,  9,  0,  7,  9,  9,  7,  4,
1,  0,
        3,  6,  3,  3,  1,  4,  4,  3, 10,  6,  9,  5, 10,  1,  5, 10,  3,  3,
        2,  9,  2,  6,  2,  1,  3,  2,  4,  3], device='cuda:0')  labels:
tensor([ 0,  9,  1,  5,  2,  1,  2,  4,  7,  8, 10,  7,  8,  9,  7,  4,  1,  1,
        3,  6,  3,  3,  5,  4,  2,  3,  3,  6,  9,  5, 10,  1,  5,  0,  3,  3,
        2,  9,  2,  6,  2,  1,  3,  2,  4,  3], device='cuda:0')
eval_loss:  0.9957242012023926
correct:  34
Accuracy of the network on the test images: 73 %
pred:  tensor([ 6,  3,  1,  5,  4,  7,  3,  3,  2,  9,  6,  7,  3,  4,  3,  9,
1,  9,
        7,  3,  8,  1,  4,  6,  5, 10,  0,  4,  7,  1,  2, 10,  6,  5,  1,  9,
        2,  9,  2,  3,  0,  3,  9,  3,  2,  2], device='cuda:0')  labels:
tensor([ 5,  3,  1,  1,  4,  2,  3,  3,  2,  8,  6,  7,  3,  4,  0,  9,  1,  9,
        7,  3,  8,  5,  4,  6,  5,  3,  0,  4,  7,  1,  2, 10,  6,  5,  1,  9,
        2,  9,  2,  3,  1,  3, 10,  3,  2,  2], device='cuda:0')
eval_loss:  0.8666501045227051
correct:  37
Accuracy of the network on the test images: 80 %
pred:  tensor([ 5,  6,  7,  3,  3,  3, 10,  4,  2,  2,  5,  7,  5,  0,  2,  9,
9,  4,
        3,  9,  4,  2,  6,  2,  1,  8,  1,  7,  0,  1,  9,  2,  6,  2, 10,  4,
        3,  5,  6,  1,  3,  1, 10,  3,  9,  3], device='cuda:0')  labels:
tensor([ 5,  6,  7,  3,  3,  3,  0,  4,  2,  2,  1,  7,  5,  3,  2,  9,  8,  4,
        3,  9,  4,  2,  6,  2,  1,  8,  1,  7,  0,  1,  9,  2,  5,  2, 10,  4,
       10,  5,  6,  1,  3,  1,  3,  3,  9,  3], device='cuda:0')
eval_loss:  0.8872841000556946
correct:  39
Accuracy of the network on the test images: 84 %
pred:  tensor([10, 10,  6,  3,  3,  5,  1,  6,  3,  7,  4,  3,  9,  7,  4,  3,
1,  1,
        2,  3,  5,  5, 10,  7,  2,  6,  9,  6,  2,  5,  2,  8,  4,  3,  2,  1,
        3,  3,  4,  2,  9,  8,  3,  9,  2,  9], device='cuda:0')  labels:
tensor([ 3,  0,  6, 10,  3,  1,  1,  6,  3,  7,  4,  1,  9,  7,  4,  3,  1,  5,
        2,  3,  1,  5, 10,  7,  2,  5,  3,  6,  2,  5,  2,  8,  4,  3,  2,  1,
        0,  3,  4,  2,  9,  8,  3,  9,  2,  9], device='cuda:0')
eval_loss:  1.0251601934432983
correct:  36
Accuracy of the network on the test images: 78 %
pred:  tensor([ 2,  7,  3,  9,  3,  5,  3,  3,  9,  6,  2,  5,  1,  8,  1,  1,
2,  1,
        5, 10,  2,  7,  7,  2,  3,  4,  1,  3,  2,  3,  2,  4,  4,  9,  6,  4,
       10,  8,  9,  3,  3,  0,  3,  6,  0,  6], device='cuda:0')  labels:
```

```
tensor([ 2,  7,  0,  9,  3,  5,  3,  3,  9,  6,  2,  5,  1,  8,  5,  1,  2,  1,
         1,  3,  2,  7,  7,  2,  3,  4,  1,  3,  2, 10,  2,  4,  4,  9,  6,  4,
        10,  8,  9,  3,  3,  0,  3,  6,  1,  5], device='cuda:0')
eval_loss:  0.7986611127853394
correct:  39
Accuracy of the network on the test images: 84 %
pred:  tensor([ 0,  1,  6,  1,  4,  3,  6,  8,  7,  4,  2, 10,  4,  2,  0,  3,
7,  3,
         1,  5,  4,  3,  2,  5,  6, 10,  2,  0,  9,  3,  5,  9,  3,  9,  1,  2,
         8,  2,  3,  3,  2,  9,  7,  6,  3,  1], device='cuda:0')  labels:
tensor([ 1,  1,  5,  1,  4,  3,  5,  8,  7,  4,  2,  3,  4,  2,  0,  3,  7,  3,
         5,  5,  4,  3,  2,  1,  6, 10,  2,  0,  9, 10,  6,  9,  3,  9,  1,  2,
         8,  2,  3,  3,  2,  9,  7,  6,  3,  1], device='cuda:0')
eval_loss:  0.9747252464294434
correct:  38
Accuracy of the network on the test images: 82 %
pred:  tensor([ 3,  2,  9,  4,  3,  0,  8,  9,  2,  6,  5,  9,  2,  3, 10,  3,
1, 10,
         5,  2,  3,  7,  4,  2,  1,  3,  3,  1,  9,  4,  6,  4,  1,  3,  8,  7,
         6,  1,  2,  7,  5,  6,  0,  2, 10,  3], device='cuda:0')  labels:
tensor([ 3,  2,  9,  4,  3, 10,  8,  9,  2,  6,  5,  9,  2,  3, 10,  3,  1,  0,
         5,  2,  3,  7,  4,  2,  1,  3,  3,  5,  9,  4,  6,  4,  1,  3,  8,  7,
         5,  1,  2,  7,  1,  6,  1,  2,  3,  0], device='cuda:0')
eval_loss:  1.233949899673462
correct:  38
Accuracy of the network on the test images: 82 %
pred:  tensor([ 5,  8,  7,  1, 10,  3,  3,  7,  2,  1,  3,  3,  6,  2,  1,  2,
3,  6,
         6,  4,  5,  7,  2,  3,  2,  1,  9,  9,  3,  3,  6,  3,  9,  8,  3,  4,
         3,  2,  4,  7,  8,  5,  4,  2,  7,  1], device='cuda:0')  labels:
tensor([ 5,  8,  7,  5, 10,  0,  3,  7,  2,  1,  3,  3,  5,  2,  1,  2,  3,  6,
         6,  4,  5,  7,  2,  3,  2,  1,  9,  9, 10,  0,  6,  3,  9,  9,  1,  4,
         3,  2,  4,  3,  8,  1,  4,  2,  3,  1], device='cuda:0')
eval_loss:  0.9906989932060242
correct:  36
Accuracy of the network on the test images: 78 %
pred:  tensor([ 9, 10,  3,  3,  1,  4,  1,  3,  1,  4,  2,  2, 10,  3,  5,  9,
3,  2,
         3,  2,  8,  7,  8,  4,  3,  3,  9,  7,  6,  5,  7,  7,  4,  0,  0,  1,
         2,  6,  6,  6,  2,  1,  6,  2,  9,  0], device='cuda:0')  labels:
tensor([ 9,  3,  3, 10,  1,  4,  5,  3,  1,  4,  2,  2, 10,  3,  5,  9,  3,  2,
         3,  2,  8,  7,  8,  4,  3,  3,  9,  7,  6,  1,  1,  7,  4,  3,  0,  1,
         2,  6,  5,  5,  2,  1,  6,  2,  9,  0], device='cuda:0')
eval_loss:  1.064706802368164
correct:  38
Accuracy of the network on the test images: 82 %
pred:  tensor([ 3,  2,  1,  8,  5,  2,  1,  9,  7,  6,  2,  3,  3,  9,  7,  3,
3,  2,
```

```
        10,  3,  3,  5, 10,  4,  8,  7,  1,  7,  4,  3,  6,  1,  2,  2,  4,  4,
         6,  5,  9,  9,  5,  2,  0,  1,  6,  1], device='cuda:0')  labels:
tensor([ 3,  2,  1,  8,  1,  2,  1,  9,  7,  5,  2,  3,  3,  9,  7,  3,  3,  2,
         3,  0,  3,  5, 10,  4,  8,  7,  1,  3,  4, 10,  6,  1,  2,  2,  4,  4,
         6,  3,  9,  9,  5,  2,  0,  5,  6,  1], device='cuda:0')
eval_loss:  1.037197232246399
correct:  38
Accuracy of the network on the test images: 82 %
pred:  tensor([ 8,  7,  3,  9,  3,  2,  2,  3,  3,  9,  9,  4,  6,  5,  0,  2,
1,  3,
         4,  1,  6, 10,  3,  1,  2,  4,  9,  2,  3,  7,  1,  5, 10,  4,  3,  2,
         2,  3,  0,  5, 10,  8,  1,  6,  9,  7], device='cuda:0')  labels:
tensor([ 8,  7, 10,  9,  3,  2,  2,  3,  3,  9,  9,  4,  6,  5,  1,  2,  5,  3,
         4,  1,  6,  0,  3,  1,  2,  4,  8,  2,  3,  7,  1,  1,  3,  4,  3,  2,
         2,  3,  0,  5, 10,  5,  1,  6,  9,  7], device='cuda:0')
eval_loss:  1.206048846244812
correct:  38
Accuracy of the network on the test images: 82 %
pred:  tensor([ 4,  2,  9,  2,  3,  2,  6,  9,  5, 10,  4,  4,  1,  3,  1,  2,
3,  5,
         9,  2,  1,  3,  6,  6,  1,  0,  3,  3,  8,  2,  1,  3,  7,  2,  9,  8,
         4,  3,  3,  0,  6, 10,  7,  7,  5,  1], device='cuda:0')  labels:
tensor([ 4,  2,  9,  2, 10,  2,  6,  9,  1,  3,  4,  4,  1,  0,  5,  2,  3,  5,
         9,  2,  1,  3,  6,  5,  1,  1,  3,  3,  8,  2,  1,  3,  7,  2,  9,  8,
         4,  3,  3,  0,  6, 10,  7,  7,  5,  3], device='cuda:0')
eval_loss:  1.0668964385986328
correct:  38
Accuracy of the network on the test images: 82 %
pred:  tensor([ 9,  1,  3,  2,  4,  6,  3,  6,  9, 10,  9,  7,  3,  8,  3,  3,
1,  3,
         6,  7,  1,  5,  2,  0,  1,  9,  0,  4, 10,  3,  2,  6,  4,  9,  6,  2,
         3,  2,  6,  2,  5,  7,  2,  3,  5,  4], device='cuda:0')  labels:
tensor([ 9,  1,  3,  2,  4,  6,  3,  5,  9,  3,  8,  7,  0,  8, 10,  3,  1,  3,
         5,  7,  1,  5,  2,  1,  1,  9,  0,  4, 10,  3,  2,  6,  4,  9,  5,  2,
         3,  2,  6,  2,  1,  7,  2,  3,  3,  4], device='cuda:0')
eval_loss:  1.1516402959823608
correct:  36
Accuracy of the network on the test images: 78 %
pred:  tensor([ 3,  6,  2,  3,  5,  9,  3, 10,  2,  2,  1,  3,  6, 10, 10,  7,
7,  6,
         4,  2,  3,  8,  8,  9,  2,  9,  1,  5,  9,  4,  8,  1,  4,  3,  6,  3,
         9,  4,  1,  1,  3,  7,  2,  2,  3,  1], device='cuda:0')  labels:
tensor([ 3,  6,  2,  0,  5,  9,  3,  0,  2,  2,  1,  3,  6, 10,  3,  7,  7,  5,
         4,  2,  3,  5,  8,  9,  2,  9,  1,  1,  3,  4,  8,  1,  4, 10,  6,  3,
         9,  4,  5,  1,  3,  7,  2,  2,  3,  1], device='cuda:0')
eval_loss:  1.1532646417617798
correct:  37
Accuracy of the network on the test images: 80 %
```

```
pred:  tensor([ 9,  2,  2,  1,  3,  9,  9,  2,  3,  9,  4,  6, 10,  6,  1,  6,
4,  5,
         9,  5,  1,  8,  2,  3,  3,  3,  3,  2,  1,  4,  0,  3,  5,  3,  6,  7,
         4,  7,  2,  2,  9,  7, 10,  3, 10,  1], device='cuda:0')  labels:
tensor([ 9,  2,  2,  5,  3,  8,  9,  2,  3,  3,  4,  6,  0,  6,  1,  5,  4,  5,
         9,  5,  1,  8,  2,  3,  0,  3,  3,  2,  1,  4,  1, 10,  1,  3,  6,  7,
         4,  7,  2,  2,  9,  7,  3,  3, 10,  1], device='cuda:0')
eval_loss:  1.5579661130905151
correct:  36
Accuracy of the network on the test images: 78 %
pred:  tensor([ 3,  1,  6,  3,  1,  4,  9,  2,  5,  3,  9,  4,  3, 10,  7,  2,
1,  6,
         3,  5, 10,  2,  0,  2,  9,  1,  6,  2,  9,  1,  8,  9,  3,  7,  2,  0,
         3,  4,  3,  7,  2,  3,  6,  5,  3,  4], device='cuda:0')  labels:
tensor([ 3,  1,  6,  3,  1,  4,  8,  2,  1,  3,  9,  4,  0, 10,  7,  2,  5,  6,
         0,  5,  3,  2,  1,  2,  9,  1,  5,  2,  9,  1,  8,  9,  3,  7,  2,  3,
        10,  4,  3,  7,  2,  3,  6,  5,  3,  4], device='cuda:0')
eval_loss:  1.2122275829315186
correct:  36
Accuracy of the network on the test images: 78 %
pred:  tensor([ 7,  4,  1, 10,  6,  2,  3,  9, 10,  1,  4,  7,  8,  3,  2,  3,
3,  8,
         3,  2,  6,  9,  6,  7,  8,  9,  6,  5,  2,  2,  5,  3,  8,  1,  1,  6,
         2,  1,  4,  2,  3,  4,  0,  9,  1,  3], device='cuda:0')  labels:
tensor([ 7,  4,  1, 10,  5,  2,  0,  9,  3,  5,  4,  7,  3,  3,  2, 10,  3,  8,
         3,  2,  6,  9,  6,  7,  0,  9,  5,  1,  2,  2,  5,  3,  8,  1,  1,  6,
         2,  1,  4,  2,  3,  4,  3,  9,  1,  3], device='cuda:0')
eval_loss:  1.1486889123916626
correct:  36
Accuracy of the network on the test images: 78 %
pred:  tensor([ 2,  1,  3,  3,  0,  2,  4,  9,  3,  4,  3,  7, 10,  3,  6,  4,
0,  0,
         4,  6,  3,  1,  9,  6,  1,  2, 10,  2,  2,  8,  3,  1,  1,  5,  8, 10,
         3,  3,  8,  7,  5,  9,  2,  2, 10,  1], device='cuda:0')  labels:
tensor([ 2,  5,  3,  3,  7,  2,  4,  9,  3,  4,  3,  7,  3, 10,  6,  4,  0,  0,
         4,  6,  3,  1,  9,  6,  1,  2,  5,  2,  2,  8,  3,  1,  1,  5,  8,  9,
         3,  3,  5,  7,  1,  9,  2,  2, 10,  1], device='cuda:0')
eval_loss:  0.9474493861198425
correct:  38
Accuracy of the network on the test images: 82 %
pred:  tensor([ 7,  7,  2,  9,  4,  6,  7,  8, 10,  7,  2,  5,  8,  1,  1,  9,
3,  2,
         3,  2,  3,  9,  1,  4,  2,  5,  1,  6,  8,  3,  5,  4,  3,  6,  6,  3,
         8, 10,  3,  4,  2,  6,  9,  0,  2,  1], device='cuda:0')  labels:
tensor([ 7,  7,  2,  9,  4,  5,  7,  8,  3,  3,  2,  5,  3,  1,  1,  9,  3,  2,
        10,  2,  3,  9,  1,  4,  2,  1,  1,  6,  8,  3,  3,  4,  0,  5,  6,  3,
         0, 10,  3,  4,  2,  6,  9,  1,  2,  5], device='cuda:0')
eval_loss:  1.6288849115371704
```

```
correct:  34
Accuracy of the network on the test images: 73 %
pred:  tensor([ 4,  5,  2,  5,  9,  3,  1,  3, 10,  9,  4,  7,  6,  5,  6,  3,
1,  3,
        7,  2,  1,  4,  2,  3,  2,  3,  6,  9,  2,  5,  1,  9,  3,  4,  1,  3,
        1,  9,  8, 10,  2,  9,  7, 10,  2,  8], device='cuda:0')  labels:
tensor([ 4,  5,  2,  3,  9, 10,  1,  0, 10,  8,  4,  7,  6,  1,  6,  3,  1,  3,
        7,  2,  1,  4,  2,  3,  2,  3,  6,  8,  2,  5,  1,  9,  3,  4,  5,  3,
        1,  9,  5,  3,  2,  9,  7,  0,  2,  3], device='cuda:0')
eval_loss:  0.8752208948135376
correct:  35
Accuracy of the network on the test images: 76 %
pred:  tensor([ 2,  3,  3,  7,  5,  8,  9,  9,  1,  3, 10,  7,  6,  4,  7,  2,
6,  9,
        2,  6,  3,  3,  1,  9,  3,  3,  2,  4,  6,  6,  3, 10,  1,  4,  1,  3,
        8,  4,  2,  5,  2,  3,  0,  2,  1,  1], device='cuda:0')  labels:
tensor([ 2,  3,  3,  7,  1,  8,  9,  9,  1,  3, 10,  7,  6,  4,  7,  2,  6,  9,
        2,  5,  3, 10,  1,  9,  0,  3,  2,  4,  5,  6,  3,  3,  1,  4,  1,  3,
        8,  4,  2,  5,  2,  3,  0,  2,  1,  5], device='cuda:0')
eval_loss:  0.9905914664268494
correct:  39
Accuracy of the network on the test images: 84 %
pred:  tensor([ 2,  7,  7,  3,  1,  3,  9, 10,  3,  1,  6,  3,  4,  5,  1,  2,
3,  8,
        5,  3,  6,  9,  9,  3,  1,  2,  2,  0,  1,  7,  4,  2,  2,  2,  4,  9,
        5,  4, 10,  6,  3,  6,  7,  9,  1,  8], device='cuda:0')  labels:
tensor([ 2,  7,  7,  3,  1,  0,  3, 10,  3,  1,  6, 10,  4,  5,  5,  2,  3,  8,
        5,  3,  6,  9,  9,  3,  1,  2,  2,  0,  1,  3,  4,  2,  2,  2,  4,  9,
        1,  4,  3,  6,  3,  5,  7,  9,  1,  8], device='cuda:0')
eval_loss:  1.0439469814300537
correct:  38
Accuracy of the network on the test images: 82 %
pred:  tensor([ 1,  5,  4,  5,  9,  2,  2,  4, 10,  3,  7,  6, 10,  2,  6,  8,
7,  2,
        9,  1,  3,  2,  3,  7,  4,  0,  3,  8,  9,  1,  3,  6,  2,  3,  1,  9,
        4,  6,  6,  3,  3,  9,  1,  3,  8,  2], device='cuda:0')  labels:
tensor([ 5,  1,  4,  5,  9,  2,  2,  4, 10,  3,  7,  5,  3,  2,  6,  8,  7,  2,
        9,  1,  3,  2,  0,  7,  4,  0,  3,  3,  9,  1, 10,  6,  2,  3,  1,  8,
        4,  5,  6,  3,  3,  9,  1,  3,  1,  2], device='cuda:0')
eval_loss:  1.0832140445709229
correct:  36
Accuracy of the network on the test images: 78 %
pred:  tensor([ 8,  2, 10,  9,  6,  4,  1,  7,  5, 10,  2,  3,  0,  3,  3,  2,
1,  8,
        4,  3,  6,  0,  1,  9,  5,  3,  4,  9,  8,  7,  7,  2,  1,  1,  3,  3,
        1, 10,  6,  4,  3,  2,  5,  2,  9,  4], device='cuda:0')  labels:
tensor([ 5,  2,  0,  9,  6,  4,  1,  7,  1, 10,  2,  3,  0, 10,  3,  2,  1,  8,
        2,  3,  6,  3,  1,  9,  5,  3,  4,  9,  8,  7,  7,  2,  1,  1,  3,  3,
```

```
        5,  3,  6,  4,  3,  2,  5,  2,  9,  4], device='cuda:0')
eval_loss:  0.9652303457260132
correct:  38
Accuracy of the network on the test images: 82 %
pred:  tensor([ 8,  2,  1,  6,  9,  3,  5,  9,  7,  9, 10,  3,  3,  1,  3,  4,
4,  4,
        2,  3,  3,  8,  9,  5,  9,  6,  2, 10,  1,  1, 10,  3,  1,  2,  2,  6,
        5,  7,  7,  3,  1,  4,  0,  6,  2,  2], device='cuda:0')  labels:
tensor([ 8,  2,  1,  6,  3,  3,  1,  9,  7,  9,  3,  3,  3,  1,  3,  4,  4,  4,
        2,  3,  3,  8,  9,  5,  9,  6,  2,  0,  5,  1, 10, 10,  1,  2,  2,  5,
        5,  7,  7,  3,  1,  4,  0,  6,  2,  2], device='cuda:0')
eval_loss:  1.1043858528137207
correct:  39
Accuracy of the network on the test images: 84 %
pred:  tensor([ 2,  1,  3,  1,  1,  2,  7,  0,  5,  4,  8,  3,  6,  1,  4,  0,
3,  2,
        3,  2,  4,  9,  5,  3,  9,  3,  7,  9,  4, 10,  2,  9,  3,  7, 10,  3,
        8,  4,  5,  1,  6,  1,  2,  6,  6, 10], device='cuda:0')  labels:
tensor([ 2,  5,  3,  1,  1,  2,  7,  0,  1,  4,  8, 10,  5,  1,  4,  0,  3,  2,
        3,  2,  4,  9,  5,  3,  9,  3,  7,  9,  4, 10,  2,  9,  3,  7,  3,  3,
        8,  2,  5,  1,  6,  1,  2,  6,  6,  3], device='cuda:0')
eval_loss:  0.9570050239562988
correct:  39
Accuracy of the network on the test images: 84 %
pred:  tensor([ 8,  3,  2,  1,  1,  1,  9,  7,  4,  2,  3,  6,  2,  7,  0,  5,
2, 10,
        3,  1,  5,  6, 10,  4,  4,  2,  4,  9,  2,  1,  2,  5,  9,  1,  7,  3,
        3,  3,  9,  9,  6,  6,  3,  3,  3,  8], device='cuda:0')  labels:
tensor([ 3,  0,  2,  1,  1,  1,  9,  7,  4,  2,  3,  6,  2,  7,  0,  5,  2, 10,
        3,  1,  5,  6,  3,  4,  4,  2,  4,  9,  2,  3,  2,  1,  8,  5,  7,  1,
        3,  3,  9,  9,  6,  5, 10,  3,  3,  8], device='cuda:0')
eval_loss:  1.045567274093628
correct:  36
Accuracy of the network on the test images: 78 %
pred:  tensor([ 9,  4,  7, 10,  3,  2,  9,  1, 10,  2,  0,  1,  5,  1,  1,  9,
2,  3,
        7,  7,  2,  1,  4,  9,  3,  7,  6,  8,  5,  2,  8,  3,  3,  3,  3,  6,
        6,  4,  2,  2,  4,  6, 10,  8,  3,  5], device='cuda:0')  labels:
tensor([ 9,  4,  7,  0,  3,  2,  9,  1,  3,  2,  1,  5,  5,  1,  1,  9,  2,  3,
        7,  7,  2,  1,  4,  9,  3,  3,  6,  8,  5,  2,  8,  3,  3,  3,  0,  6,
        5,  4,  2,  2,  4,  6, 10,  3, 10,  1], device='cuda:0')
eval_loss:  1.2394914627075195
correct:  36
Accuracy of the network on the test images: 78 %
pred:  tensor([ 2,  9, 10,  3,  8,  1,  4,  3,  2,  2,  3,  3,  1,  3,  7,  6,
2, 10,
        4,  4,  2,  6,  5,  3,  3,  0,  9,  3,  6,  3,  5,  6, 10,  1,  4,  2,
        9,  1,  7,  6,  2,  1,  9, 10,  7,  1], device='cuda:0')  labels:
```

```
tensor([ 2,  9,  8,  3,  3,  1,  4,  0,  2,  2, 10,  3,  1,  3,  7,  5,  2,  3,
         4,  4,  2,  6,  5,  3,  3,  0,  9,  3,  6,  3,  1,  6,  9,  1,  4,  2,
         8,  1,  7,  5,  2,  1,  9, 10,  7,  5], device='cuda:0')
eval_loss:  1.315332055091858
correct:  35
Accuracy of the network on the test images: 76 %
pred:  tensor([ 1,  7,  9,  0,  3,  2,  2,  5,  3,  2,  3,  5,  1,  2,  6,  8,
4,  6,
        10,  4,  1,  2,  4,  3,  3, 10,  3,  0,  3,  7,  2,  9,  8,  6,  3,  9,
         5,  2,  9,  4,  6,  7,  1,  3,  6,  9], device='cuda:0')  labels:
tensor([ 1,  7,  9, 10,  0,  2,  2,  5,  3,  2,  3,  5,  1,  2,  1,  8,  4,  6,
         3,  4,  1,  2,  4,  3,  3, 10,  3,  0,  3,  7,  2,  9,  1,  5,  3,  9,
         5,  2,  9,  4,  6,  7,  1,  3,  6,  8], device='cuda:0')
eval_loss:  1.1245235204696655
correct:  39
Accuracy of the network on the test images: 84 %
pred:  tensor([ 2,  6,  4,  9,  3,  2, 10,  0,  5,  7,  3,  2,  6,  2,  8,  4,
7,  2,
         2,  1,  1,  6,  4,  1,  1,  7,  7,  7, 10,  3,  4,  9,  3,  9,  9,  7,
         5,  2,  5,  3,  9,  3,  5,  8,  3,  1], device='cuda:0')  labels:
tensor([ 2,  6,  4,  9,  3,  2,  3,  0,  8,  7,  0,  2,  6,  2,  5,  4,  1,  2,
         2,  1,  1,  6,  4,  5,  1,  3,  7,  7, 10,  3,  4,  8,  3,  9,  9,  3,
         5,  2,  5, 10,  9,  3,  1,  3,  3,  1], device='cuda:0')
eval_loss:  1.0971286296844482
correct:  34
Accuracy of the network on the test images: 73 %
pred:  tensor([10,  7,  3,  3,  9, 10,  5,  3,  4,  6,  2,  2,  3,  3,  9,  2,
8,  9,
         1,  3,  4,  3,  1,  2,  4,  1,  7,  1,  3,  2,  6,  2,  5,  2,  6,  0,
         9,  8,  9,  7,  6,  3,  4,  3,  5,  0], device='cuda:0')  labels:
tensor([10,  7,  3,  3,  8,  3,  5,  0,  4,  6,  2,  2,  3,  3,  9,  2,  5,  9,
         1,  3,  4,  3,  1,  2,  4,  1,  7,  1,  3,  2,  5,  2,  1,  2,  6,  1,
         9,  8,  9,  7,  6,  3,  4,  0,  5, 10], device='cuda:0')
eval_loss:  1.3396259546279907
correct:  37
Accuracy of the network on the test images: 80 %
pred:  tensor([ 9,  4,  6,  5,  2,  9,  4,  0, 10,  3,  1,  6,  5,  3,  2,  3,
7,  3,
         7,  5,  4,  6,  3,  5,  7,  3,  9,  8,  1,  9,  3,  1, 10,  6, 10,  4,
         2,  9,  1,  2,  2,  2,  2,  3,  3,  1], device='cuda:0')  labels:
tensor([ 8,  4,  5,  5,  2,  9,  4, 10, 10,  3,  1,  6,  5,  3,  2,  3,  7,  3,
         7,  1,  4,  6,  3,  1,  7,  0,  9,  8,  1,  9,  3,  1,  0,  6,  3,  4,
         2,  9,  1,  2,  2,  2,  2,  3,  3,  5], device='cuda:0')
eval_loss:  0.9190518856048584
correct:  37
Accuracy of the network on the test images: 80 %
pred:  tensor([ 5,  4,  4,  2,  3,  2,  3,  1,  0,  4,  7,  6,  9,  3,  9,  6,
10,  3,
```

```
        1,  6,  7,  3,  3,  2,  1,  5,  4,  9,  2,  8,  5,  3,  3,  2,  2,  1,
        1, 10,  8,  7,  3,  0,  6,  9,  2,  3], device='cuda:0')  labels:
tensor([ 5,  4,  4,  2,  3,  2,  3,  1,  0,  4,  7,  6,  9,  3,  9,  5, 10,  3,
        1,  6,  7,  1,  0,  2,  1,  1,  4,  9,  2,  8,  5,  3,  3,  2,  2,  5,
        1,  3,  8,  7,  3, 10,  6,  9,  2,  3], device='cuda:0')
eval_loss:  1.0176619291305542
correct:  39
Accuracy of the network on the test images: 84 %
pred:  tensor([ 2,  3,  6,  3, 10,  9,  0,  3, 10,  4,  6,  2,  6,  1,  4,  2,
4,  3,
        2,  7,  5,  3,  3,  5,  1,  1,  9,  2,  3,  9,  1,  9,  7,  5,  4,  3,
        3,  8,  3,  8,  1,  2,  7,  8,  2,  3], device='cuda:0')  labels:
tensor([ 2,  3,  6,  3, 10,  9,  1,  0,  3,  4,  6,  2,  6,  1,  4,  2,  4,  3,
        2,  7,  1,  3, 10,  5,  1,  5,  9,  2,  3,  9,  1,  9,  7,  5,  4,  3,
        3,  8,  3,  5,  1,  2,  7,  8,  2,  0], device='cuda:0')
eval_loss:  0.8608211278915405
correct:  38
Accuracy of the network on the test images: 82 %
pred:  tensor([ 5,  2,  3, 10,  9,  3,  4,  9,  3,  2,  2,  8,  9,  7,  2,  7,
3,  5,
        1,  1,  1,  2,  3,  5,  0,  6,  8, 10,  3,  2,  1,  6,  4,  1,  7,  2,
        9,  4,  4, 10,  3,  1,  3,  6,  3,  9], device='cuda:0')  labels:
tensor([ 1,  2,  3,  3,  9,  3,  4, 10,  3,  2,  2,  8,  9,  7,  2,  7,  3,  5,
        1,  1,  1,  2,  3,  5,  0,  6,  5, 10,  3,  2,  1,  6,  4,  5,  7,  2,
        8,  4,  4,  9,  3,  1,  3,  6,  0,  9], device='cuda:0')
eval_loss:  0.9474925398826599
correct:  38
Accuracy of the network on the test images: 82 %
pred:  tensor([ 6,  5,  3,  3,  3,  1,  3,  3,  6,  6,  6,  5,  2,  8,  3,  4,
0,  2,
        4,  7,  1,  4,  9,  4,  2,  3,  8,  3,  2, 10,  4,  2,  9,  1,  9,  9,
        2,  1,  5,  0,  5, 10,  1,  7,  3,  7], device='cuda:0')  labels:
tensor([ 6,  5,  3,  3,  3,  1,  3,  3,  6,  6,  5,  5,  2,  8,  3,  4, 10,  2,
        4,  7,  1,  4,  9,  2,  2,  3,  8,  0,  2,  3,  4,  2,  9,  1,  9,  9,
        2,  1,  1,  0,  5, 10,  1,  7,  3,  7], device='cuda:0')
eval_loss:  0.9547910094261169
correct:  40
Accuracy of the network on the test images: 86 %
pred:  tensor([ 3,  8,  1,  0,  8,  2,  3,  4,  9,  2,  3,  3,  6,  6,  2, 10,
2,  2,
        7,  2,  9,  3,  9,  4,  5,  1,  4,  1,  8,  1,  3, 10,  7,  4,  5,  2,
        7,  6,  0,  3,  1,  3,  3,  5,  1,  9], device='cuda:0')  labels:
tensor([ 3,  8,  1,  0,  5,  2,  3,  4,  9,  2,  3,  3,  6,  6,  2,  3,  2,  2,
        7,  2,  9, 10,  9,  4,  5,  1,  4,  1,  8,  1,  3, 10,  7,  4,  1,  2,
        7,  6,  3,  3,  1,  0,  3,  5,  5,  9], device='cuda:0')
eval_loss:  1.0322872400283813
correct:  39
Accuracy of the network on the test images: 84 %
```

```
pred:  tensor([ 9,  5,  2,  3,  6,  4,  8,  6,  6,  9,  3,  4,  9,  5,  3,  1,
2,  5,
         3,  0,  3,  3,  9,  1,  2,  2,  6,  7,  1,  1,  4,  8,  8,  2,  7,  0,
         1, 10,  0,  4,  9,  1, 10,  2,  2,  7], device='cuda:0')  labels:
tensor([ 9,  1,  2,  3,  5,  4,  3,  6,  6,  9,  3,  4,  9,  5,  3,  1,  2,  5,
         3, 10,  3,  0,  9,  1,  2,  2,  6,  7,  1,  1,  4,  8,  8,  2,  7,  3,
         1,  3,  0,  4,  3,  5, 10,  2,  2,  7], device='cuda:0')
eval_loss:  1.2884453535079956
correct:  37
Accuracy of the network on the test images: 80 %
pred:  tensor([ 5, 10,  3,  1,  1,  4,  1,  6,  3,  8,  6,  7,  6,  6,  2,  4,
10,  1,
         3,  9,  3,  2,  4,  3,  4,  2,  3,  9,  5,  1,  3,  3,  2,  3,  5,  2,
         3,  9,  8,  0,  9,  1,  7,  2,  2,  7], device='cuda:0')  labels:
tensor([ 5, 10,  3,  1,  1,  4,  1,  5,  3,  8,  6,  7,  6,  6,  2,  4,  3,  5,
         3,  9,  3,  2,  4,  3,  4,  2,  0,  9,  5,  1,  3,  3,  2,  3,  1,  2,
        10,  9,  8,  0,  9,  1,  7,  2,  2,  7], device='cuda:0')
eval_loss:  1.117161750793457
correct:  40
Accuracy of the network on the test images: 86 %
pred:  tensor([ 3,  3,  6,  2,  4,  3,  2,  2,  2,  0,  6,  0, 10,  2,  9,  6,
9,  9,
         3,  5, 10,  1,  8,  3,  2,  2,  1,  5,  7,  0, 10,  5,  7,  8,  1,  4,
         1,  3,  1,  6,  4,  3,  9,  7,  8,  4], device='cuda:0')  labels:
tensor([ 3,  3,  5,  2,  4,  3,  2,  2,  2,  0,  6,  0, 10,  2,  9,  6,  9,  9,
         3,  5, 10,  5,  3,  3,  2,  2,  1,  1,  7,  1,  3,  5,  7,  8,  1,  4,
         1,  3,  1,  6,  4,  3,  9,  7,  8,  4], device='cuda:0')
eval_loss:  0.7924050688743591
correct:  40
Accuracy of the network on the test images: 86 %
pred:  tensor([ 7,  3,  1,  5,  1,  8,  8,  9,  9,  2,  6, 10,  3,  0,  1,  9,
9,  1,
         3,  2,  3,  4,  8,  3, 10,  0,  3,  6,  1,  2,  5,  7,  4,  6,  2,  5,
         6,  2,  8,  1,  2,  7,  2,  6,  4,  4], device='cuda:0')  labels:
tensor([ 7,  3,  1,  5,  1,  8,  9,  9,  3,  2,  5, 10,  3,  0,  1,  9,  9,  5,
         3,  2,  3,  4,  3,  3,  3,  0, 10,  1,  1,  2,  5,  7,  4,  6,  2,  3,
         6,  2,  8,  1,  2,  7,  2,  6,  4,  4], device='cuda:0')
eval_loss:  1.0367130041122437
correct:  37
Accuracy of the network on the test images: 80 %
pred:  tensor([ 3,  8,  8,  3,  8,  6,  4,  1,  4,  3,  6,  2,  4,  1,  6,  9,
0,  7,
         6,  3,  3,  1,  1,  2,  3,  3,  1,  3,  5,  2,  7,  2,  4,  2,  3,  2,
         9, 10,  5,  0,  3,  0,  2,  9, 10,  9], device='cuda:0')  labels:
tensor([10,  8,  8,  3,  5,  6,  4,  1,  4,  1,  6,  2,  4,  1,  5,  9,  1,  7,
         6,  3,  7,  1,  3,  2,  3,  9,  5,  3,  5,  2,  7,  2,  4,  2,  3,  2,
         3, 10,  1,  0,  3,  0,  2,  9,  3,  9], device='cuda:0')
eval_loss:  1.4560219049453735
```

```
correct:  34
Accuracy of the network on the test images: 73 %
pred:  tensor([ 8,  2,  3,  3,  1,  2,  1,  6, 10,  2,  6,  4,  4,  4,  4,  1,
3,  3,
         3,  1,  3,  0,  1,  6, 10,  7,  2,  2,  9,  7,  3,  3,  3,  5,  2,  2,
         5,  1,  8,  6,  9,  9,  9,  0,  7,  5], device='cuda:0')  labels:
tensor([ 8,  2,  3,  3,  1,  2,  1,  6, 10,  2,  6,  4,  4,  4,  4,  1,  3,  3,
         3,  5,  3, 10,  1,  5,  3,  7,  2,  2,  9,  7,  0,  3,  3,  5,  2,  2,
         1,  1,  8,  6,  9,  9,  9,  0,  7,  5], device='cuda:0')
eval_loss:  0.9899383187294006
correct:  40
Accuracy of the network on the test images: 86 %
pred:  tensor([ 3,  5,  9,  3,  1, 10,  3,  2,  9,  2,  6,  9,  0,  4,  7,  6,
2,  8,
         2,  6,  7,  4,  1,  6, 10,  1,  9,  9,  3,  2,  3,  3,  1,  4,  5,  3,
         3,  2,  2,  3,  7,  1,  4,  1,  8,  6], device='cuda:0')  labels:
tensor([ 3,  5,  9,  3,  1,  3,  3,  2,  8,  2,  5,  9,  0,  4,  7,  5,  2,  8,
         2,  6,  7,  4,  1,  6, 10,  1,  9,  9, 10,  2,  0,  3,  1,  4,  1,  3,
         3,  2,  2,  3,  7,  1,  4,  5,  3,  6], device='cuda:0')
eval_loss:  0.9848853945732117
correct:  37
Accuracy of the network on the test images: 80 %
pred:  tensor([ 6,  8,  2, 10,  1,  2,  4,  4,  9,  7,  3,  9,  7,  2,  4,  9,
9,  1,
         1,  1,  3,  8,  2,  3,  1,  0,  5,  3,  2,  3,  7,  5, 10,  3,  4,  3,
        10,  5,  6,  2,  6,  2,  0,  3,  3,  6], device='cuda:0')  labels:
tensor([ 5,  8,  2, 10,  1,  2,  4,  4,  9,  7,  3,  9,  7,  2,  4,  9,  9,  1,
         5,  1,  0,  8,  2,  3,  1,  1,  5,  3,  2,  3,  7,  5,  3,  3,  4,  3,
         0,  1,  6,  2,  6,  2, 10,  3,  3,  6], device='cuda:0')
eval_loss:  1.180989384651184
correct:  38
Accuracy of the network on the test images: 82 %
pred:  tensor([ 6,  4,  2,  3,  3,  7,  7,  1,  4,  3,  8,  9,  3, 10,  7,  5,
5,  3,
         9,  1,  9,  6,  3,  8,  2,  9,  1,  1,  4, 10,  2,  2,  5,  6,  7,  3,
         2,  3,  4,  6,  2,  1,  3,  7,  2,  9], device='cuda:0')  labels:
tensor([ 6,  4,  2,  3,  3,  7,  1,  5,  4,  3,  3,  9,  0,  3,  3,  5,  5,  3,
         9,  1,  9,  6, 10,  8,  2,  8,  1,  1,  4, 10,  2,  2,  1,  5,  7,  3,
         2,  0,  4,  6,  2,  1,  3,  7,  2,  9], device='cuda:0')
eval_loss:  1.1577852964401245
correct:  35
Accuracy of the network on the test images: 76 %
pred:  tensor([ 3,  1,  5,  2,  5,  6,  2,  2,  1,  7,  3,  7,  1,  2,  6,  9,
0,  6,
         3,  4,  3,  5,  7, 10, 10,  9,  4,  8,  4,  2,  9,  4,  3,  2,  2,  3,
         6,  1,  3, 10,  6,  9,  3,  1,  8,  7], device='cuda:0')  labels:
tensor([ 3,  1,  5,  2,  5,  6,  2,  2,  1,  7,  3,  7,  1,  2,  6,  9,  1,  6,
         3,  4,  3,  1,  7,  0, 10,  9,  4,  8,  4,  2,  9,  4,  3,  2,  2, 10,
```

```
        8,  5,  3,  3,  5,  9,  0,  1,  3,  3], device='cuda:0')
eval_loss:  1.4002931118011475
correct:  35
Accuracy of the network on the test images: 76 %
pred:  tensor([ 9,  6,  2,  6,  1,  3,  6,  5,  2,  1,  2,  4,  9,  3,  1,  3,
5,  7,
        4,  9,  0,  4,  2,  1,  1,  3, 10,  3,  2,  6,  3,  5, 10,  7,  1,  2,
        3,  9,  3,  0,  3,  9,  9,  4,  2,  7], device='cuda:0')  labels:
tensor([ 9,  6,  2,  6,  1,  3,  6,  5,  2,  1,  2,  4,  9,  3,  1,  3,  1,  7,
        4,  8, 10,  4,  2,  1,  1,  3, 10,  0,  2,  5,  3,  5,  3,  7,  5,  2,
        3,  8,  3,  0,  3,  9,  9,  4,  2,  7], device='cuda:0')
eval_loss:  0.8261362314224243
correct:  38
Accuracy of the network on the test images: 82 %
pred:  tensor([ 1,  2,  2,  2,  5,  3,  4,  9,  9,  3,  5,  7,  7,  3,  2,  1,
3,  8,
        6,  9,  8,  2, 10,  3,  4,  4,  3,  0, 10,  9,  2,  4,  7,  9,  2,  3,
        1,  3,  1,  8,  6,  6,  6,  3,  1,  6], device='cuda:0')  labels:
tensor([ 1,  2,  2,  2,  5,  3,  4,  9,  9,  3,  1,  7,  7,  3,  2,  1,  3,  8,
        5,  9,  8,  2,  3,  0,  4,  4,  3,  0, 10,  3,  2,  4,  7,  9,  2, 10,
        1,  3,  1,  1,  6,  5,  6,  3,  5,  6], device='cuda:0')
eval_loss:  1.2784065008163452
correct:  37
Accuracy of the network on the test images: 80 %
pred:  tensor([ 9,  0,  6,  0,  1,  3,  2,  3,  8,  7,  1,  4,  2,  5,  5, 10,
7,  2,
        9,  4,  0,  3,  9,  1,  4,  3,  2,  8, 10,  4,  3,  5,  8,  3,  3,  7,
        2,  1,  6,  6,  2,  9,  8,  2,  1,  6], device='cuda:0')  labels:
tensor([ 9, 10,  6,  0,  1,  3,  2,  3,  3,  7,  1,  4,  2,  5,  1,  3,  7,  2,
        9,  4,  0,  3,  9,  1,  4,  3,  2,  8, 10,  4,  3,  5,  8,  3,  3,  7,
        2,  1,  6,  5,  2,  9,  1,  2,  5,  6], device='cuda:0')
eval_loss:  0.8777940273284912
correct:  39
Accuracy of the network on the test images: 84 %
pred:  tensor([ 2,  8,  5,  6,  7,  3,  4,  2,  2,  3, 10,  6,  6,  4,  1,  6,
2,  9,
        7,  5,  1,  3,  3,  6,  3,  1,  3,  0,  3,  3,  8,  2,  1,  9,  9,  4,
       10, 10,  2,  2,  9,  4,  1,  3,  0,  7], device='cuda:0')  labels:
tensor([ 2,  8,  1,  6,  7,  3,  4,  2,  2,  3,  3,  5,  5,  4,  1,  6,  2,  9,
        7,  5,  1,  3, 10,  6,  3,  1,  3,  1,  3,  3,  8,  2,  5,  9,  9,  4,
        0, 10,  2,  2,  9,  4,  1,  3,  0,  7], device='cuda:0')
eval_loss:  0.86673903465271
correct:  38
Accuracy of the network on the test images: 82 %
pred:  tensor([ 1,  7,  1,  1,  9,  7,  7,  1,  2,  3,  9,  2,  3,  8,  3,  4,
3,  3,
        2,  1,  3,  8,  2,  5,  5,  5,  0,  9,  9,  3,  2, 10,  4,  2,  3,  3,
        9,  2,  5, 10,  4,  8,  3,  6,  4,  6], device='cuda:0')  labels:
```

```
tensor([ 1,  7,  5,  1,  9,  7,  7,  1,  2, 10,  9,  2,  3,  3,  3,  4,  3,  3,
         2,  1,  3,  8,  2,  5,  1,  6,  0,  9,  9,  1,  2,  3,  4,  2,  3,  3,
         8,  2,  5, 10,  4,  5,  0,  6,  4,  6], device='cuda:0')
eval_loss:  1.1383328437805176
correct:  36
Accuracy of the network on the test images: 78 %
pred:  tensor([ 1,  3,  2,  2,  3,  1,  9,  6,  8, 10,  7,  5,  9,  6,  1,  4,
2,  1,
         5,  7,  3,  3,  2,  1,  3,  3,  4,  3,  3,  5,  7,  8,  6,  2,  3,  7,
         7, 10,  4,  2,  4,  6,  9,  2,  0,  3], device='cuda:0')  labels:
tensor([ 1,  3,  2,  2,  3,  1,  9,  6,  8,  9,  7,  5,  9,  6,  5,  4,  2,  1,
         5,  7,  3,  3,  2,  1,  0,  3,  4,  3, 10,  1,  3,  8,  6,  2,  1,  7,
         3, 10,  4,  2,  4,  5,  9,  2,  0,  3], device='cuda:0')
eval_loss:  0.6582275629043579
correct:  37
Accuracy of the network on the test images: 80 %
pred:  tensor([ 6, 10,  3,  1,  6,  9,  3,  1,  7,  4,  3,  5,  6,  2,  4,  7,
5,  6,
         9,  3,  3,  4,  8,  9,  3,  9,  2,  3,  0,  2,  7,  1,  1, 10, 10,  3,
         4,  2,  2,  2,  2,  5,  7,  3,  9,  1], device='cuda:0')  labels:
tensor([ 6, 10,  3,  5,  5,  9,  3,  1,  7,  4,  3,  5,  6,  2,  4,  7,  1,  6,
         9,  3,  3,  4,  8,  9,  3,  9,  2, 10,  0,  2,  1,  1,  1,  3,  3,  0,
         4,  2,  2,  2,  2,  5,  7,  3,  8,  1], device='cuda:0')
eval_loss:  1.1199084520339966
correct:  37
Accuracy of the network on the test images: 80 %
pred:  tensor([ 8,  1,  3,  3,  3,  3,  2,  2,  0,  1,  7, 10,  4,  2,  7, 10,
10,  1,
         1,  5,  2,  3,  4,  9,  6,  5,  7,  4,  8,  2,  6,  0,  0,  0,  4,  2,
         3,  9,  1,  9,  2,  9,  9,  6,  6,  9], device='cuda:0')  labels:
tensor([ 5,  1,  3,  3,  3,  3,  2,  2,  3,  1,  7,  3,  4,  2,  7, 10,  0,  1,
         5,  5,  2,  3,  4,  3,  5,  1,  7,  4,  8,  2,  6,  1,  0, 10,  4,  2,
         3,  9,  1,  8,  2,  9,  9,  6,  6,  9], device='cuda:0')
eval_loss:  1.2202621698379517
correct:  35
Accuracy of the network on the test images: 76 %
pred:  tensor([ 4,  2,  1,  1,  2,  2,  4,  2,  9,  3,  3,  9,  1,  2,  7, 10,
3,  3,
         1, 10,  8,  7,  4,  3,  6,  3,  7,  5,  1,  3,  4,  0,  9,  6,  1,  1,
         3,  5,  9,  6,  2,  2,  3,  3,  9,  8], device='cuda:0')  labels:
tensor([ 4,  2,  5,  1,  2,  2,  4,  2,  8,  3,  3,  9,  1,  2,  7,  3,  3,  3,
         1, 10,  5,  7,  4,  3,  6, 10,  7,  5,  1,  3,  4,  0,  9,  6,  1,  1,
         3,  5,  9,  6,  2,  2,  3,  0,  9,  8], device='cuda:0')
eval_loss:  0.6494044661521912
correct:  40
Accuracy of the network on the test images: 86 %
pred:  tensor([ 7,  5,  4,  1,  2,  3,  3,  0,  2,  9,  1,  2,  1,  3,  9,  3,
3,  5,
```

```
        9,  8,  1,  3,  4,  3,  3,  3,  3,  6,  2,  2,  9,  4, 10,  1,  7, 10,
        1,  6,  7,  5,  6,  5,  2,  4,  2,  9], device='cuda:0')  labels:
tensor([ 7,  5,  4,  1,  2,  3,  3,  0,  2,  9,  1,  2,  1,  3,  9,  3,  3,  1,
        8,  5,  5,  3,  4, 10,  0,  3,  3,  6,  2,  2,  9,  4, 10,  1,  7,  3,
        1,  6,  7,  6,  5,  8,  2,  4,  2,  9], device='cuda:0')
eval_loss:  1.0793073177337646
correct:  36
Accuracy of the network on the test images: 78 %
pred:  tensor([ 4,  3,  8,  6,  0,  5,  4, 10,  9,  3,  5,  3,  2,  3,  1,  1,
1,  7,
        3,  2,  2,  3,  2,  9,  1,  2,  1,  1,  9,  0,  6,  7,  6,  3,  0,  7,
        2,  9, 10,  3,  5,  4,  6,  4,  4,  8], device='cuda:0')  labels:
tensor([ 4, 10,  8,  6,  3,  1,  4,  3,  9,  3,  5,  3,  2,  3,  5,  1,  1,  7,
        3,  2,  2,  3,  2,  9,  1,  2,  1,  1,  9,  0,  6,  7,  5,  3,  0,  7,
        2,  9, 10,  3,  5,  4,  6,  4,  2,  8], device='cuda:0')
eval_loss:  0.891823947429657
correct:  39
Accuracy of the network on the test images: 84 %
pred:  tensor([ 8,  5,  3,  4,  1,  2,  5,  4,  1, 10,  4,  9,  2,  9, 10,  1,
0,  3,
        6,  9,  6,  4,  6,  1,  3,  3,  1,  0,  8,  3,  2,  2,  7,  2,  5,  3,
        2,  8,  2,  7,  0,  9,  3,  3,  7,  3], device='cuda:0')  labels:
tensor([ 5,  1,  3,  4,  1,  2,  5,  4,  5,  3,  4,  9,  2,  9, 10,  1,  0,  3,
        6,  9,  6,  4,  6,  1,  3, 10,  1,  0,  8,  3,  2,  2,  7,  2,  5,  3,
        2,  8,  2,  7,  1,  9,  3,  3,  7,  3], device='cuda:0')
eval_loss:  0.92750483751297
correct:  40
Accuracy of the network on the test images: 86 %
pred:  tensor([ 4,  1,  5,  0,  7,  3,  9,  1, 10,  1,  9,  6,  9,  3,  2,  2,
1,  2,
        4,  1,  3,  2,  2,  8,  4,  7,  1,  6,  3,  7,  5,  2,  4,  3,  7,  9,
        3,  6,  3,  6,  3,  3,  1, 10,  9,  7], device='cuda:0')  labels:
tensor([ 4,  1,  5,  0,  7,  3,  9,  1,  3,  1,  8,  6,  9,  3,  2,  2,  1,  2,
        4,  1, 10,  2,  2,  8,  4,  7,  5,  6,  3,  2,  5,  2,  4,  3,  3,  9,
        3,  5,  3,  6,  3,  0,  1, 10,  9,  7], device='cuda:0')
eval_loss:  1.2407233715057373
correct:  38
Accuracy of the network on the test images: 82 %
pred:  tensor([ 3,  3,  3,  2,  9,  4,  1, 10,  1,  0,  7,  9,  4,  2,  2,  2,
5,  7,
        1,  9,  3,  2,  7,  8,  3,  9,  3,  0,  5,  0,  6,  3,  9,  6,  1,  2,
        4,  3,  4,  3,  1,  6,  2,  5, 10,  6], device='cuda:0')  labels:
tensor([ 3,  3,  3,  2,  9,  4,  1, 10,  1,  0,  7,  9,  4,  2,  2,  2,  1,  7,
        5,  8,  3,  2,  7,  8,  3,  9, 10,  0,  5,  1,  6,  3,  9,  6,  1,  2,
        4,  3,  4,  3,  1,  5,  2,  5,  3,  6], device='cuda:0')
eval_loss:  0.8358486890792847
correct:  39
Accuracy of the network on the test images: 84 %
```

```
total_acc:  81.0869565217391
```

[17]:
```
"\ntest_dataset = ImageFolder(root='/content/drive/My Drive/Colab
Notebooks/test_set',transform=data_transform)\ntestloader =
torch.utils.data.DataLoader(test_dataset,batch_size=46, shuffle=True)\neval_loss
= 0\ncorrect = 0\ntotal = 0\nfor i, data in enumerate(testloader):\n    inputs,
labels = data\n    #print(labels)\n    \n    inputs = inputs.cuda()\n    labels
= labels.cuda()\n    outputs = net(inputs)\n    loss = criterion(outputs,
labels)\n    eval_loss += loss.item()\n    total += labels.size(0)\n    pred =
torch.max(outputs, 1)[1]\n    print('pred: ',pred,' labels: ',labels)\n
correct += (pred == labels).sum().item()\nprint('eval_loss:
',eval_loss)\nprint('correct: ',correct)\nprint('Accuracy of the network on the
test images: %d %%' % (100 * correct / total))\n"
```

As you can see, the accuracy of the model is 81%.

[ ]: