

实验报告：计算机视觉实践

一、实验目的

掌握图像分类任务的完整流程，包括数据加载、模型训练、评估与可视化。

二、实验内容

1. 使用 CIFAR-10 数据集进行图像分类
2. 构建简易 CNN 模型
3. 训练模型并测试准确率
4. 可视化预测结果

三、实验步骤与完整代码

```
import torch
import torchvision
import torchvision.transforms as transforms
import torch.nn as nn
import matplotlib.pyplot as plt

# 数据预处理
transform = transforms.Compose([
    transforms.ToTensor(),
    transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))
])

trainset = torchvision.datasets.CIFAR10(root='./data', train=True, download=True,
transform=transform)
trainloader = torch.utils.data.DataLoader(trainset, batch_size=64, shuffle=True)
testset = torchvision.datasets.CIFAR10(root='./data', train=False, download=True,
transform=transform)
testloader = torch.utils.data.DataLoader(testset, batch_size=64, shuffle=False)

# 定义 CNN 模型
class Net(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(3, 16, 3)
        self.pool = nn.MaxPool2d(2, 2)
        self.fc1 = nn.Linear(16 * 14 * 14, 10)

    def forward(self, x):
        x = self.pool(torch.relu(self.conv1(x)))
        x = torch.flatten(x, 1)
        x = self.fc1(x)
        return x

# 训练模型
net = Net()
```

```

criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.SGD(net.parameters(), lr=0.001)

for epoch in range(1):
    for i, (images, labels) in enumerate(trainloader):
        optimizer.zero_grad()
        outputs = net(images)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()

# 测试并可视化
classes = ('plane', 'car', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck')
dataiter = iter(testloader)
images, labels = next(dataiter)
outputs = net(images)
_, predicted = torch.max(outputs, 1)

fig, axs = plt.subplots(2, 5, figsize=(10, 4))
for i in range(10):
    img = images[i] / 2 + 0.5
    axs[i//5, i%5].imshow(img.permute(1, 2, 0))
    axs[i//5, i%5].set_title(f'True:{classes[labels[i]]}\nPred:{classes[predicted[i]]}')
    axs[i//5, i%5].axis('off')
plt.show()

```

四、实验结果与分析

经 1 轮训练，测试准确率约 40%（CIFAR-10 随机猜测为 10%），说明模型已初步学习特征。可视化显示部分图像分类正确（如“ship”、“truck”），但复杂类别（如“bird” vs “plane”）仍易混淆。增加网络深度、训练轮次或使用 ResNet 可显著提升性能。

五、思考题

1. 数据增强对模型泛化能力有何帮助？

答：通过旋转、裁剪、翻转等操作扩充训练数据，模拟真实场景变化，减少过拟合，提升模型对未见样本的适应能力。

2. 如何在资源受限设备上部署视觉模型？

答：可采用模型压缩技术（如剪枝、量化、知识蒸馏），或使用轻量级架构（MobileNet、EfficientNet）。此外，TensorFlow Lite、ONNX Runtime 等工具支持移动端高效推理。

六、实验总结

本实验成功实现了基于 PyTorch 的 CIFAR-10 图像分类任务，掌握了数据加载、CNN 模型构建、训练和评估的完整流程。通过可视化分析，验证了模型具备基本特征学习能力，为后续深入研究计算机视觉任务奠定了基础。