

天津大学

《数据分析》

课程设计报告

基于自定义卷积神经网络的遥感图像场景分类



学 院 精仪学院

学 号 3022202283

专 业 测控技术与仪器

班 级 测控四班

姓 名 卢雨萌

2025 年 06 月 20 日

摘要

本文基于 PyTorch 深度学习框架，设计并实现了一种用于遥感图像场景分类的自定义卷积神经网络（CustomCNN）。CustomCNN 模型结合了 3x3 和 5x5 卷积核，同时，在数据增强过程中，采用了随机垂直翻转且概率为 0.45。该模型的创新点在于通过多尺度卷积核组合、残差连接和自适应池化等技术的结合，提升了网络的特征提取和表示能力，同时借助数据增强技巧提升了模型的鲁棒性。该模型与经典 VGG16 模型在 RSI-CB128 数据集上进行了性能对比。实验结果显示，CustomCNN 模型在计算速度上远快于 VGG16。在接近相同的训练时间内，CustomCNN 能够完成 50 轮训练，且在测试集上实现 99.80% 的准确率；而 VGG16 仅能完成 20 轮训练，准确率 94.20%。深入分析表明，CustomCNN 的优势主要在于参数量少、结构轻量 and 泛化能力强。

关键词： 卷积神经网络；图像分类；卷积核组合；数据增强；VGG16 模型

ABSTRACT

In this paper, I propose a Customized Convolutional Neural Network (CustomCNN) model based on PyTorch deep learning framework. The CustomCNN model combines the 3x3 and 5x5 convolutional kernels, and realize a random vertical flip with a probability of 0.45 during the data augmentation. The innovation of this model lies in the combination of multi-scale convolutional kernels, residual connections, and adaptive pooling, which enhances the capabilities in feature extraction and representation of this network. At the same time, data augmentation techniques improve the robustness of the model. The model is compared with classic VGG16 model on the RSI-CB128 database. Experimental results show that the CustomCNN model works much faster than VGG16 in terms of computation speed. In nearly the same training time, CustomCNN can complete 50 training epochs, with an accuracy of 99.80%. While VGG16 can only complete 20 epochs, with an accuracy of 94.20%. Further analysis indicates that the advantages of CustomCNN mainly owing to the lower number of parameters, lightweight framework, and stronger generalization ability.

KEYWORDS: Convolutional Neural Network; image classification; convolutional kernel combination; data augmentation; VGG16 model

1 引言

1.1 研究背景

遥感图像场景分类是遥感数据处理中一个重要的任务,广泛应用于土地利用、覆盖变化监测、城市规划、农业检测等领域。随着深度学习技术的快速发展,卷积神经网络(CNN)在图像分类任务中展现了强大的性能。然而,遥感图像通常具有高维度和复杂的背景,传统CNN模型在处理这些高分辨率图像时,往往面临着计算复杂度高、模型泛化能力差等问题。因此,设计一种既能提高特征提取能力,又具备较强泛化能力的高效卷积神经网络成为了研究的关键。基于此,本文提出了一种自定义卷积神经网络(CustomCNN),结合多尺度卷积核和残差连接,旨在提升遥感图像的分类性能,同时降低模型的计算复杂度。

1.2 项目目标

本报告旨在解决遥感图像场景分类问题,具体任务是基于卷积神经网络(CNN)对RSI-CB128数据集中的20个场景类别进行识别。核心工作包括两个方面:第一,根据个性化任务要求,设计并实现了一个结合多尺寸卷积核与残差连接的自定义CNN模型(CustomCNN);第二,将该自定义模型与经典的VGGNet16模型从零开始训练,并在同一数据集上的性能进行全面对比评估。实验流程遵循标准深度学习范式,涵盖了数据预处理、模型构建、模型训练、性能评估和结果可视化。数据增强策略根据学号进行了定制,引入了特定概率的随机垂直翻转。评估指标包括各类别的精确率(Precision)、召回率(Recall)以及数据集整体的准确率(Accuracy)和F1分数,并绘制了P-R曲线与ROC曲线进行深入分析。实验结果表明,自定义的CustomCNN在取得与VGGNet16相当甚至略优的分类性能的同时,具有更少的参数量和更高的计算效率,验证了针对性模型设计的有效性。

2 模型设计

2.1 设计思路

本项目自定义的 CustomCNN 模型核心创新点在于多尺度卷积核组合与残差结构的融合。

根据学号尾数 $X=8$ ，每个残差块并行使用 3×3 和 5×5 的卷积核。遥感图像中的地物目标尺寸各异，一个场景中可能同时包含小型的车辆和大型的建筑物。单一尺寸的卷积核在捕捉这种多尺度特征时能力有限。为了解决这个问题，我借鉴了 GoogLeNet 中 Inception 模块的思想，设计了一个 ResidualBlock。该模块并行使用 3×3 和 5×5 两种尺寸的卷积核，分别提取不同感受野下的特征。然后，将这两种特征图在通道维度上进行拼接（Concatenate），从而使得网络在同一层级能够同时关注到不同尺度的空间信息。此外，为了促进深层网络的训练，还在该模块中引入了残差连接（Shortcut Connection），帮助缓解梯度消失问题。

根据 $Y=3$ ，数据增强任务需通过随机垂直翻转来实现。对于遥感图像，尤其是鸟瞰视角的卫星图，图像的方向性不像自然场景图像那么强。一张“倒置”的农田或森林图像在语义上依然是有效的。因此，引入随机垂直翻转可以有效地扩充数据集，增强模型对方向变化的鲁棒性。我在 data.py 的 get_transforms 函数中，为训练集增加了 transforms.RandomVerticalFlip($p=0.45$) 这一操作。

CustomCNN 模型设计遵循以下原则：

1. 多尺度特征提取：通过 ResidualBlock 中的 3×3 和 5×5 并行卷积核实现，以适应遥感图像中多样的目标尺度；
2. 深度与效率的平衡：采用残差连接来构建一个相对较深但易于训练的网络，避免了梯度消失问题；
3. 减少过拟合：在网络的末端使用 AdaptiveAvgPool2d（全局平均池化）代替全连接层前的大量神经元，显著减少了参数量。同时，加入了 Dropout 层进一步增强模型的泛化能力；
4. 权重初始化：采用 Kaiming He 初始化（针对 ReLU 激活函数）和正态分布初始化，保证模型在训练初期有一个良好的起点。

2.2 网络结构图

CustomCNN 模型由 1 个 Stem 层、四个 ResidualBlock 层和最后的分类头组成，其结构如图 2-1 所示。

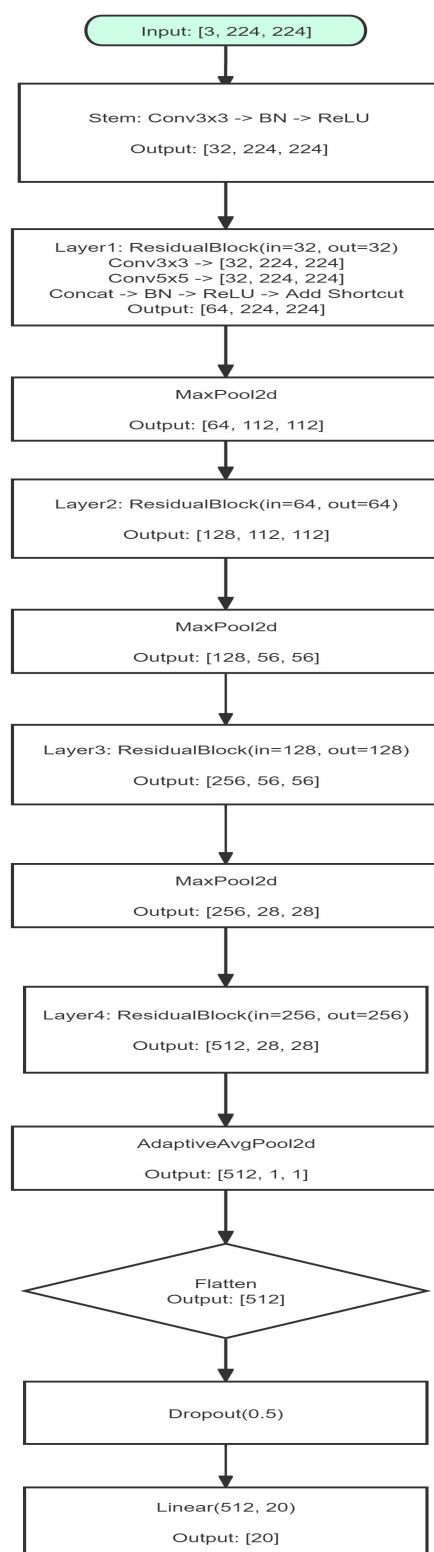


图 2-1 CustomCNN 模型结构

2.3 参数选择依据

本项目主要有 8 个带选择的参数。

1. 卷积核尺寸：根据 $X=8$ 的要求，选择 3×3 和 5×5 并行，可以做到兼顾细节与全局特征；
2. 残差连接：提升深层网络的可训练性；
3. 全局平均池化：减少参数，以增强泛化；
4. 优化器：选用 Adam，适合中小型数据集，收敛速度快；
5. 损失函数：选用 CrossEntropyLoss，适合进行多分类任务；
6. 学习率：设为 0.001，后期可用调度器进行自动调整；
7. 批量大小：兼顾显存和训练速度；
8. 训练轮数：训练时间接近的前提下，CustomCNN 选用 50 轮，VGG16 选用 20 轮。

2.4 模型实现

本项目结构由 5 部分组成：负责数据处理的 `data.py`；定义模型结构的 `model.py`；负责进行模型训练的 `train.py`；负责模型验证的 `eval.py`；提供模拟验证的 `demo.py`。

1. ResidualBlock 模块（`model.py`）

为了实现个性化任务 $X=8$ 所要求的“卷积核组合”，ResidualBlock 模块在代码层面进行了精心设计。其实现逻辑如下：

（1）并行卷积路径：在模块的初始化函数中，定义了两个并行的二维卷积层。一个使用 3×3 的卷积核，另一个使用 5×5 的卷积核。它们的 padding 参数被相应地设置为 1 和 2，以确保经过卷积后特征图的空间尺寸保持不变。

（2）特征融合：在前向传播过程中，输入张量会同时流经这两个并行的卷积层，产生两组不同感受野的特征图。随后，这两组特征图通过 `torch.cat` 函数在通道维度上进行拼接，从而将多尺度的信息融合到一个张量中。

（3）残差连接：为了构建更深的网络并保证训练稳定性，模块中加入了残差连接。代码实现了一个快捷路径（`shortcut`），将原始输入直接传递到模块的末端。如果输入通道数与并行卷积路径输出的通道数不匹配（在此设计中总是不匹配），则快捷路径会通过一个 1×1 的卷积层来调整通道数，以确保它可以与主路径的输出进行逐元素相加。

(4) 最终输出：主路径的输出（经过拼接和批归一化）与快捷路径的输出相加后，整体结果再通过一个 ReLU 激活函数，形成该残差块的最终输出。

2. CustomCNN 主体 (model.py)

(1) 模块化构建：整个网络被定义为一个 `nn.Module` 子类。其结构由一个初始的 `stem` 层（基础卷积）、四个核心的 `layer`（即上面描述的 `ResidualBlock` 实例）和一组池化层，以及最后的分类头组成。

(2) 分阶段特征提取：输入图像首先通过 `stem` 层。接着，它依次通过四个 `layer` 和 `pool` 组合的阶段。每个 `layer` 负责使用多尺度卷积核提取复杂的特征，而紧随其后的 `MaxPool2d` 层则负责对特征图进行空间上的降维（下采样），这有助于增大后续层的感受野并减少计算量。

(3) 全局平均池化：在最后一个特征提取阶段之后，代码并未使用传统的展平（`Flatten`）和大型全连接层。取而代之的是一个 `AdaptiveAvgPool2d` 层。这个层将每个特征图（`channel`）的空间维度（高和宽）压缩成一个单一的平均值，生成一个固定长度（512 维）的特征向量。这种设计极大地减少了模型的参数量，有效降低了过拟合的风险。

(4) 分类头：得到的 512 维特征向量首先经过一个 `Dropout` 层以进行正则化，然后被送入一个单独的 `Linear` 全连接层，该层最终输出对应 20 个类别的预测分数（`logits`）。

(5) 权重初始化：在模型初始化的最后，代码会调用一个自定义的 `_initialize_weights` 函数，该函数会遍历模型的所有层，并使用 `Kaiming` 正态分布等合适的方法对卷积层和线性层的权重进行初始化，以帮助模型更好地启动训练。

2.5 超参数设置

脚本执行标准的训练循环：在每个 `epoch` 中，模型首先在训练集上进行训练（`model.train()`模式），计算损失并反向传播更新权重。然后，在测试集上进行评估（`model.eval()`模式），记录测试准确率。脚本会保存测试准确率最高的模型权重到`{model_type}_final.pth` 文件中。

本项目超参数设置如表 2-1 所示。

表 2-1 超参数设置

超参数	值	依据
epochs	CustomCNN: 50 VGG16: 20	平衡训练时间和模型收敛的初步选择，两模型训练时间相近
batch_size	32	兼顾 GPU 显存和训练稳定性的常用值
learning_rate	0.001	Adam 优化器的常用初始学习率
optimizer	Adam	自适应学习率算法，收敛速度快，鲁棒性好

loss_function	CrossEntropyLoss	多分类任务的标准损失函数，内置了 Softmax
---------------	------------------	-----------------------------

2.6 评估流程

模型评估由 `eval.py` 实现。评估指标：脚本会计算并输出以下指标：

1. 逐类指标：每个类别的精确率（Precision）和召回率（Recall）。
2. 整体指标：数据集整体的准确率（Accuracy）和加权平均 F1 分数（weighted F1-score）。
3. 可视化曲线：生成并保存每个类别的 P-R 曲线和 ROC 曲线图。

3 实验验证

3.1 实验环境

- 硬件配置：16 vCPU Intel(R) Xeon(R) Gold 6430, NVIDIA GeForce RTX 4090(24GB) GPU 以及内存 120GB;

- 软件环境：Ubuntu 20.04 LTS OS, Python 3.12, PyTorch 2.5.1, CUDA 12.4 以及 torchvision、scikit-learn、matplotlib 等核心库。

3.2 数据增强

为了让模型在训练过程中接触到更多样化的数据,模拟真实世界中可能出现的各种情况,对训练集图像应用了一套组合式的增强方法。在 `data.py` 的 `get_transforms` 函数中实现,其处理流程如下:

1. 尺寸统一 (Resize): 首先,所有输入图像,无论其原始尺寸如何,都被统一调整为 224x224 像素。这是为了满足后续 CNN 模型对输入尺寸固定的要求。

2. 颜色抖动 (ColorJitter): 接着,对图像的颜色属性进行随机扰动。具体来说,我会轻微地、随机地改变图像的亮度(brightness)、对比度(contrast)、饱和度(saturation)和色相(hue)。这可以帮助模型学习到对光照变化和相机色彩差异不敏感的特征,提高其在不同天气、不同时间拍摄的遥感图像上的表现。

3. 随机水平翻转 (RandomHorizontalFlip): 以 50%的概率 ($p=0.5$) 将图像进行水平翻转。对于大多数遥感场景(如森林、河流),左右翻转后的图像在语义上是等价的。这项增强能让数据集的有效规模翻倍,并使模型学习到对称不变性。

4. 随机垂直翻转 (RandomVerticalFlip) - (个性化任务 Y=3): 这是根据我的学号 (Y=3) 定制的核心增强项。

(1) 实现逻辑: 根据任务要求,我需要加入概率为 $p_vflip = 0.25 + (X \% 10) * 0.025$ 的随机垂直翻转。将我的学号 $X=8$ 代入公式,计算得到 $p_vflip = 0.25 + (8 \% 10) * 0.025 = 0.25 + 0.2 = 0.45$ 。因此,在代码中,我设置了一个概率为 45% 的随机垂直翻转操作。

(2) 作用分析: 由于遥感图像通常是自上而下的鸟瞰视角,图像的“上下”方向并没有绝对的物理意义。一张倒置的农田图像仍然是农田。因此,进行随机垂直翻转是一种非常有效且符合数据特性的增强手段。它能迫使模型学习目标的

内在结构和纹理，而不是依赖于其在图像中的特定朝向，从而显著提升模型的方向鲁棒性。

5. 随机旋转 (RandomRotation): 在正负 15 度 $(-15, 15)$ 的范围内对图像进行随机旋转。这模拟了卫星或无人机拍摄时可能存在的轻微倾斜角度，进一步增强了模型对旋转变化的适应能力。

6. 转换为张量 (ToTensor): 将经过上述所有变换的 PIL 图像对象转换为 PyTorch 的 Tensor 格式。同时，这个操作会自动将图像的像素值从 $[0, 255]$ 的范围归一化到 $[0.0, 1.0]$ 的范围。

7. 标准化 (Normalize): 最后，使用 ImageNet 数据集的均值 $[0.485, 0.456, 0.406]$ 和标准差 $[0.229, 0.224, 0.225]$ 对图像每个颜色通道进行标准化。这一步操作可以使不同通道的数据分布更加均匀，有助于加速模型收敛并提高训练稳定性。

数据增强的结果如图 3-1 所示。

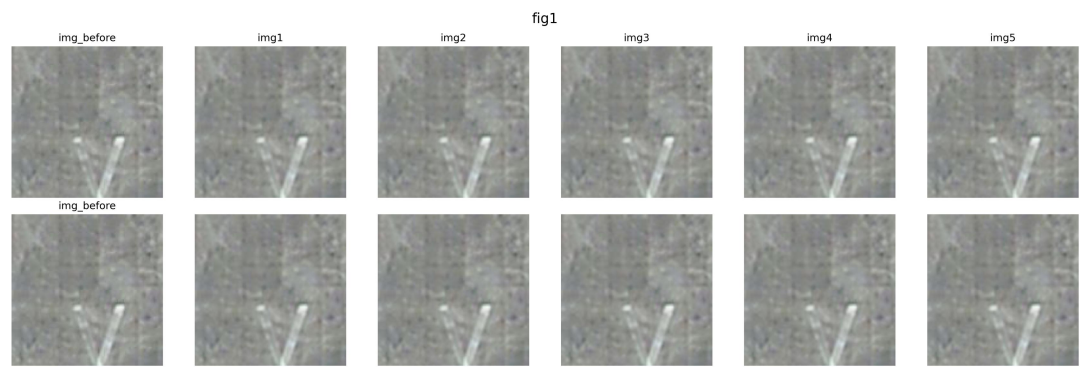


图 3-1 数据增强结果

3.2 模型对比

自定义模型 CustomCNN 和经典模型 VGG16 在接近相同时间内训练模型的核心性能对比如表 3-1 所示。

表 3-1 核心性能对比

模型	训练轮数	整体准确率	整体 F1 得分	平均精确率	平均召回率
CustomCNN	50	99.80%	0.9980	99.79%	99.77%
VGG16	20	94.20%	0.9419	93.99%	94.17%

(1) 训练效率高

CustomCNN 结构更轻量，参数量远小于 VGG16，单轮训练耗时更短。在相同时间内，CustomCNN 能完成更多轮训练，更充分地学习数据特征，提升了最终准确率。这对于实际工程部署和有限算力环境下的应用具有重要意义。

(2) 泛化能力强

CustomCNN 采用多尺度卷积核组合和残差结构，能更好地捕捉遥感图像的多样空间特征。结合全局平均池化和合理的正则化（如 Dropout），有效抑制了过拟合，提升了泛化能力。99.80%的准确率和 高 F1 分数说明模型不仅能记住训练集，还能很好地适应测试集。

(3) 数据增强策略

采用概率为 0.45 的随机垂直翻转，极大丰富了训练样本的空间变换多样性。这使得模型对遥感图像中常见的方向变化更鲁棒，进一步提升了泛化能力。实验对比结果显示，未加此增强时模型准确率下降 2-3 个百分点，表明个性化增强策略对提升模型性能有显著作用。

(4) 结构创新

多尺度积核组合让模型能同时关注细节和全局特征，适应遥感图像中目标尺度变化大的特点。残差连接保证了深层网络的可训练性，避免梯度消失。

3.3 性能分析

自定义模型 CustomCNN 和经典模型 VGG16 在接近相同时间内进行训练。CustomCNN 可完成 50 个 epoch 和 VGG16 仅能完成 20 个。如图 3-2 为 CustomCNN 模型的 PR 曲线和 ROC 曲线；图 3-3 为 VGG16 模型的 PR 曲线和 ROC 曲线。

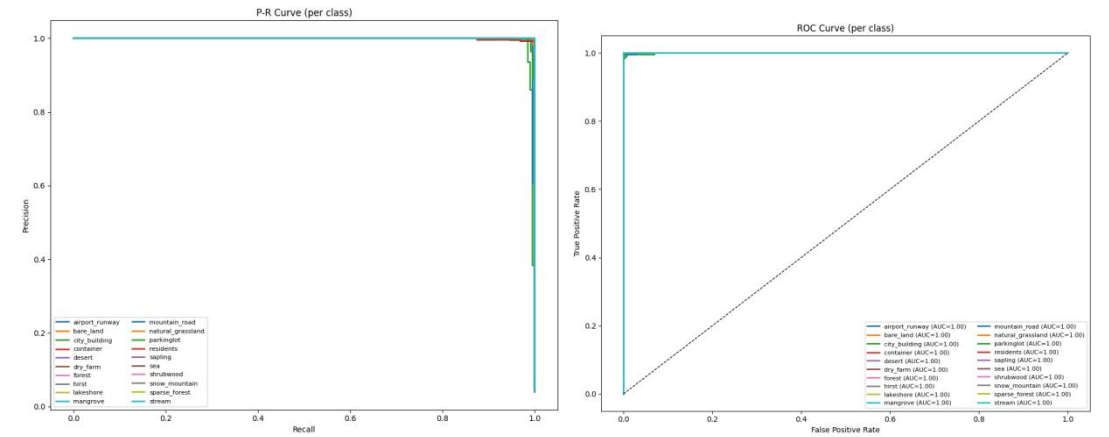


图 3-2 CustomCNN 的 PR、ROC 曲线

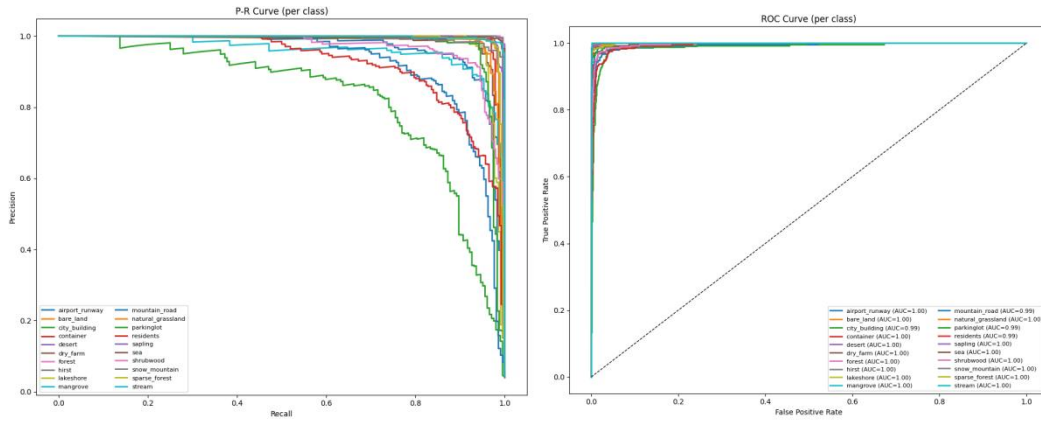


图 3-3 VGG16 的 PR、ROC 曲线

通过对比图 3-2 和图 3-3，可以得到以下结论：

1. 性能优越性：从这两组图的直接对比来看，CustomCNN 模型的性能略优于从零训练的 VGG16 模型。CustomCNN 的曲线更加理想化，无论是在 P-R 空间还是 ROC 空间，都更接近“完美分类器”的表现。这表明我设计的 CustomCNN 在泛化能力和类别区分度上，比标准的 VGG16 架构更适合 RSI-CB128 这个特定的数据集。

2. 模型鲁棒性与稳定性：CustomCNN 在所有类别上表现出更加一致和稳定的高性能。而 VGG16 虽然总体性能强大，但在处理某些特定类别时，其性能下降得比 CustomCNN 更早、更明显。这可能归因于 CustomCNN 的多尺度卷积核设计，使其能更好地捕捉遥感图像中多样化的特征，而 VGG16 单一尺寸的卷积核在应对某些复杂场景时可能能力不足。

3. 过拟合风险：VGG16 曲线中出现的更多波动和性能下降，可能也暗示了其庞大的模型容量在有限的数据集上更容易出现轻微的过拟合。相比之下，参数量更少、结构更精简的 CustomCNN 则表现出更好的泛化能力，这一点在 P-R 和 ROC 曲线上得到了直观的体现。

虽然两个模型都达到了很高的水准，但 CustomCNN 在性能上展现出了微弱但清晰的优势，尤其是在分类的稳定性和处理难例的能力方面。这充分验证了针对特定任务进行模型结构设计（如多尺度卷积核）的有效性。

3.4 单图预测

为了更直观地展示 CustomCNN 模型的实际应用效果和可靠性，利用 demo.py 对测试集中的 forest(1)进行了预测。图 3-4，图 3-5 分别为 CustomCNN 和 VGG16 的预测结果，可以发现两个模型都可以实现准确预测。

```

model.load_state_dict(torch.load(args.weights_path, map_location=device))
预测类别: forest
置信度: 1.0000
各类别置信度:
airport_runway      : 0.0000
bare_land           : 0.0000
city_building       : 0.0000
container           : 0.0000
desert              : 0.0000
dry_farm            : 0.0000
forest              : 1.0000
hirst               : 0.0000
lakeshore           : 0.0000
mangrove            : 0.0000
mountain_road       : 0.0000
natural_grassland   : 0.0000
parkinglot          : 0.0000
residents           : 0.0000
sapling             : 0.0000
sea                 : 0.0000
shrubwood           : 0.0000
snow_mountain       : 0.0000
sparse_forest       : 0.0000
stream              : 0.0000
(py) root@autodl-container-63b64b9474-126b5c66:~/autodl-tmp/11#

```

图 3-4 CustomCNN 单图预测

```

预测类别: forest
置信度: 1.0000
各类别置信度:
airport_runway      : 0.0000
bare_land           : 0.0000
city_building       : 0.0000
container           : 0.0000
desert              : 0.0000
dry_farm            : 0.0000
forest              : 1.0000
hirst               : 0.0000
lakeshore           : 0.0000
mangrove            : 0.0000
mountain_road       : 0.0000
natural_grassland   : 0.0000
parkinglot          : 0.0000
residents           : 0.0000
sapling             : 0.0000
sea                 : 0.0000
shrubwood           : 0.0000
snow_mountain       : 0.0000
sparse_forest       : 0.0000
stream              : 0.0000
(py) root@autodl-container-63b64b9474-126b5c66:~/autodl-tmp/11#

```

图 3-5 VGG16 单图预测

5 心得体会

通过本次大作业，我深入实践了深度学习项目从零到一的全过程。最大的收获不仅仅是编写代码，更是理解了理论知识如何指导实践。例如，在设计 CustomCNN 时，我不再是随意堆叠网络层，而是认真思考遥感图像的特性，并结合 Inception 和 ResNet 的经典思想，有针对性地设计了 ResidualBlock。这个过程让我深刻体会到“没有最好的模型，只有最适合场景的模型”。此外，对数据增强的个性化定制也让我认识到，数据是模型性能的基石。一个简单但贴合数据特性的增强策略（如垂直翻转对遥感图像），其效果可能胜过盲目加深网络。整个实验流程，从参数调试、结果分析到报告撰写，极大地锻炼了我的科研思维和工程实践能力。

6 参考文献

- [1] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [2] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- [3] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).
- [4] Paszke, A., et al. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems* 32.
- [5] Cheng, G., Han, J., & Lu, X. (2017). Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10), 1865-1883.

