

基于卷积神经网络的图片清晰度可见度识别分类

22122128, 孔馨怡

摘要: 本项目旨在基于卷积神经网络实现图片清晰度可见度识别分类任务, 其中采用了 VGG16 模型作为主要架构。通过对不同天气下路面环境图片数据集进行训练和测试, 评估模型在图像分类任务上的性能表现。关键技术包括 PyTorch Lightning 框架的深度学习、数据处理流程、LeNet-5 模型构建以及训练和测试方法的实现。

关键词: 卷积神经网络 CNN、VGG16、Pytorch、图像分类

1. 项目背景和意义

1.1 选题背景

当我们面对海量的图像数据时, 图像分类是一项至关重要的任务。传统的机器学习方法在处理图像分类时往往面临着维度灾难和特征提取的挑战, 而卷积神经网络 (CNN) 却以其出色的性能和广泛的应用而备受关注。

CNN 作为一种深度学习模型, 能够自动学习图像中的特征, 无需手动设计特征提取器。这种能力使得 CNN 在图像分类任务上表现出色。它通过一系列卷积层和池化层来提取图像的局部特征, 并通过全连接层将这些特征组合起来进行分类。这种分层特征提取的方式使得 CNN 能够有效地捕获图像中的空间结构和层次信息, 从而在图像分类任务中取得了巨大的成功。

在现实生活中, 面对各种环境条件下的图像数据, 特别是在行车过程中, 对路面的可见度的准确识别变得尤为重要。路面可见性受多种因素影响, 包括路灯亮度、天气、光照以及车辆自身灯光照射情况等。准确判断行车安全距离对驾驶者至关重要。因此, 利用深度学习技术, 特别是卷积神经网络 (CNN), 对图像进行智能处理, 以帮助准确理解当前的驾驶环境并推断出与路面情况相关的距离信息, 具有重要的实际意义。

本项目搜集了来自于不同摄像头拍摄、不同天气和光照条件下的图片。这些照片中包含了各种地标建筑物, 如房屋、天线等, 这些地标提供了重要的参考点, 可以帮助测量图像中的距离信息。通过利用这些地标, 可以对图像进行距离的估计, 并将图像根据不同的距离范围进行分类。为了更准确地对图像进行分类, 使用了能见度距离来衡量图像的清晰度和可见度。能见度距离是指在特定的天气和光照条件下, 能够清晰看到的最远距离。这个指标能够反映出图像的清晰度和可见度, 因此将其作为图像分类的标准。

1.2 项目数据解释

如下图 1 所示, 数据集中的图片所拍摄的建筑物标志与摄像头具有一定的距离, 在不同天气情况下, 可以看到的距离范围是不一样的, 清晰度也是不一样的, 用这个距离的指标来衡量图片清晰度, 可以对图片进行一个清晰度的分类, 间接地判断路面状况、天气情况等。



图 1. 摄像头拍摄图片可见度距离示例

能见度以 m 为单位，其图片可以分为白天、夜晚两种情况：

白天：能见度定义为能够清晰看清物体的轮廓和细节

晚上：能见度定义为能够清晰看清物体的轮廓

以下是能见度等级的划分：

能见度等级	距离范围
0	0 - 400 米
1	400 - 800 米
2	800 - 1200 米
3	1200 - 3000 米
4	3000 - 6000 米
5	6000 - 12000 米
6	12000 米以上

表 2. 能见度等级划分表

这里要说明的是：对于能见度具体数值的选择不是绝对精确的，可在标定物距离周围进行一定浮动，如距离 20km 的山峰清晰可见，能见度分类为【18——25km】之间的某个数值。

2. VGG16 模型

2.1 简介

VGGNet 是牛津大学计算机视觉组和 Google DeepMind 公司的研究员在 2014 年共同研发的深度卷积神经网络模型。在 ILSVRC2014 比赛的分类项目中，VGGNet 取得了第二名的成绩。VGGNet 的主要创新在于使用了非常小的卷积核（ 3×3 ）构建卷积神经网络的结构，从而能够取得较好的图像识别精度。特别是 VGG-16 和 VGG-19 这两个模型常被用来用来提取图像特征，为深度学习在图像分类领域的发展做出了重要贡献。

2.2 VGG16 网络结构具体介绍及特征提取原理

VGG16 通过深度堆叠的卷积层和池化层，逐渐提取图像的高级特征，并通过全连接层将这些特征映射到输出类别，从而实现图像识别任务：

卷积层（Convolutional Layers）：VGG16 包含 13 个卷积层，这些层采用了小尺寸的 3×3 卷积核和步长为 1 的卷积操作。通过堆叠多个卷积层，网络能够逐渐提取更加抽象和复杂的特征。每个卷积层后面都跟随着一个 ReLU 激活函数，以增加网络的非线性拟合能力。

池化层 (Pooling Layers): VGG16 在卷积层之间插入了 5 个最大池化层, 这些池化层的作用是减小特征图的空间尺寸, 同时保留重要的特征。池化操作通常在每个特征图上执行, 将每个区域的像素值聚合成单个值, 以减少计算量并提高模型的鲁棒性。

全连接层 (Fully Connected Layers): VGG16 的全连接层部分包含 3 个全连接层, 用于将卷积和池化层提取的特征映射到输出类别。这些全连接层通过权重连接将输入特征映射到输出类别的概率分布。

激活函数 (Activation Function): 在每个卷积层和全连接层之后, 都会应用非线性激活函数 (如 ReLU), 以增加网络的表示能力和非线性拟合能力。

Softmax 输出层: 在 VGG16 的最后, 通常使用 Softmax 函数将网络输出转换为类别概率分布, 以便进行多类别分类。

2.3 VGG16 网络结构具体参数

下表提供了 VGG16 模型的网络层次结构、输入输出情况以及每个层次的主要参数。VGG16 共包含 16 个卷积层和 3 个全连接层。每个层次的输入、输出情况以及卷积/池化窗口大小和可训练参数数量都在表中列出:

网络层简介	输入	输出	(卷积/池化) 窗口	可训练参数
Conv1_1	224x224x3 RGB 图像	224x224x64 特征图	3x3 卷积核	1792
Conv1_2	224x224x64 特征图	224x224x64 特征图	3x3 卷积核	36928
MaxPool1	224x224x64 特征图	112x112x64 特征图	2x2 池化窗口	0
Conv2_1	112x112x64 特征图	112x112x128 特征图	3x3 卷积核	73856
Conv2_2	112x112x128 特征图	112x112x128 特征图	3x3 卷积核	147584
MaxPool2	112x112x128 特征图	56x56x128 特征图	2x2 池化窗口	0
Conv3_1	56x56x128 特征图	56x56x256 特征图	3x3 卷积核	295168
Conv3_2	56x56x256 特征图	56x56x256 特征图	3x3 卷积核	590080
Conv3_3	56x56x256 特征图	56x56x256 特征图	3x3 卷积核	590080
MaxPool3	56x56x256 特征图	28x28x256 特征图	2x2 池化窗口	0
Conv4_1	28x28x256 特征图	28x28x512 特征图	3x3 卷积核	1180160
Conv4_2	28x28x512 特征图	28x28x512 特征图	3x3 卷积核	2359808
Conv4_3	28x28x512 特征图	28x28x512 特征图	3x3 卷积核	2359808
MaxPool4	28x28x512 特征图	14x14x512 特征图	2x2 池化窗口	0
Conv5_1	14x14x512 特征图	14x14x512 特征图	3x3 卷积核	2359808
Conv5_2	14x14x512 特征图	14x14x512 特征图	3x3 卷积核	2359808
Conv5_3	14x14x512 特征图	14x14x512 特征图	3x3 卷积核	2359808
MaxPool5	14x14x512 特征图	7x7x512 特征图	2x2 池化窗口	0
Flatten	7x7x512 特征图	25088 维向量	None	0
FC6	25088 维向量	4096 维向量	None	102764544
FC7	4096 维向量	4096 维向量	None	16781312
Output	4096 维向量	1000 维向量 (类别数)	None	4097000

表 3. VGG16 网络层级及参数统计

VGG16 采用了多个卷积层和池化层, 以逐渐提取图像的抽象特征。每个卷积层后面跟随着 ReLU 激活函数, 以增加非线性拟合能力。最后, 全连接层将提取的特征映射到输出类别。VGG16 的结构设计旨在增加网络的深度和复杂度, 从而提高图像识别的性能和表现能力。

3. 核心技术实现

3.1 模型实现部分

这部分是一个基于 PyTorch Lightning 框架的深度学习项目的关键部分。

整个代码段构建了一个端到端的深度学习项目，包括数据加载、预处理、模型定义和训练过程，用于图像分类任务。

DataSet 定义：ImageCSVDataset 类是一个自定义的数据集类，用于加载存储在 CSV 文件中的图像数据集。它通过__init__方法读取 CSV 文件，并根据文件中的路径加载图像数据。如果指定了数据转换（transform），则在加载图像时进行转换。__len__方法返回数据集的长度，__getitem__方法用于获取特定索引位置的图像和标签。

DataModule 定义：FogDataModule 类是一个 PyTorch Lightning 的数据模块，用于组织数据加载和预处理的逻辑。在 setup 方法中，它根据给定的 CSV 文件路径创建了一个 ImageCSVDataset 实例，并将数据集划分为训练集和验证集。同时，它定义了训练、验证和测试数据加载器。

LightningModule 定义：VGG16 类是一个 PyTorch Lightning 的模型定义类，用于定义一个基于 VGG16 架构的深度学习模型。在 forward 方法中，它调用了 timm 库中预训练的 VGG16 模型，并对输出进行 softmax 处理。同时，它定义了训练、验证、测试和推理过程中的损失计算和指标计算逻辑，以及优化器的配置。

3.2 训练部分

这部分主要在定义用于训练模型的函数。它接收一系列参数，包括批量大小（batch_size）、CSV 文件路径（csv_file）、模型权重保存目录（ckpt_dir）、模型权重保存名称（ckpt_name）、分类类别数（num_class）和训练时的最大 epoch 数（epoches）。然后，根据这些参数初始化了回调函数，配置了训练器，创建了模型和数据模块，并执行了模型的训练过程。最后返回测试结果和模型权重路径。

这一部分的代码与模型实现部分紧密相关，通过调用前面定义的模型和数据模块来完成整个训练过程。

4. 实验结果及分析

4.1 实验平台与工具选择

本文实验环境参数如下：

主机配置：

Windows11，机带 RAM16GB，处理器型号 12th Gen Intel(R) Core(TM) i7-12700H 2.30 GHz

编译环境：Python 3.10.2（PyCharm 2023.2.5（Community Edition））

本文的代码实现主要调用了 torch 库。

4.2 实验数据集

本文实验数据集各属性如下：数据集共包含 6 个不同摄像头拍摄的 2400 张图像，是在 2023 年拍摄。此数据集下包含 7 个目录，分别为 0-6，每个目录下有不同数量的图像，每个目录表示一个分类类别。所有的图像是以 JPG 格式存储，彩色图。对每一个目录下的图像，这些图像是在不同的光照、不同的天气状态下采集的。

4.3 实验结果及分析

训练过程中，我们采用了早停策略和模型检查点机制，以提高模型的训练效率和稳定性。我们共进行了 10 个 Epoch 的训练，每个 Epoch 的训练时间大约为 11 分钟。在训练过程中，我们使用了验证集对模型进行验证，最终在验证集上达到了约 86.2% 的准确率，并且验证集上的损失为 0.299。

同时，模型训练完成后，我们保存了训练得到的权重文件，并对测试集进行了测试，得到了如下结果：（如图 4）

测试集准确率（test_acc）：约 90.19%

测试集损失（test_loss）：约 0.238

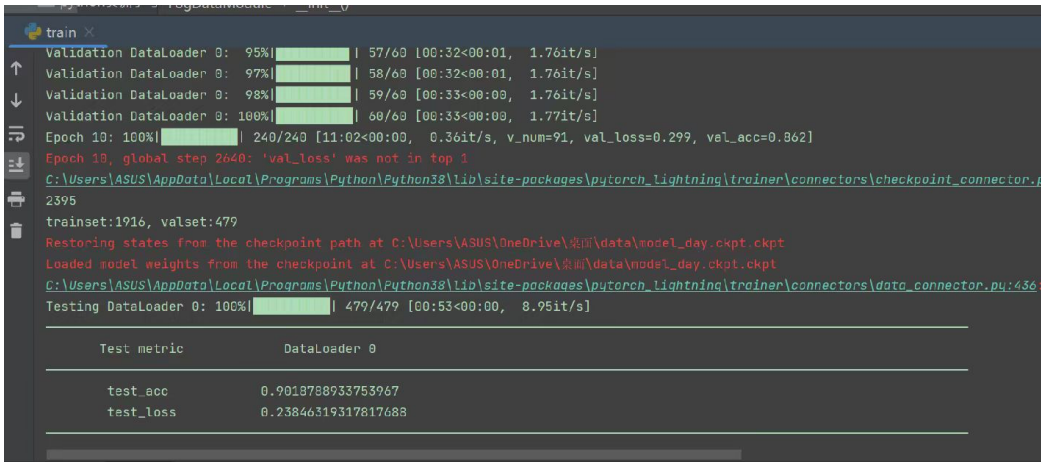


图 4. 实验结果输出

最后也可获得任意一张图片根据模型预测的可见度分类（距离范围），并打印标签在图片上，如下图 5 所示：



图 5. 实验结果输出一能见度分类输出

参考文献

[1] 杜圣杰,贾晓芬,黄友税.面向 CNN 模型图像分类任务的高效激活函数设计 U.红外与激光工程,2021,12(8):11-12.

[2] 彭斌,白静,李文静,郑虎,马向宇.面向图像分类的视觉 Transformer 研究进展.计算机科学与探索 2024,18(02),320-344

[3] 咎楠楠 (1984-). 基于全局 CNN 与局部 LSTM 的国画图像分类算法. DOI : 10.20033/j.1003-7241.(2024)04-0115-03

[4] 李伟,孙云娟.基于深度学习的 CNN 手写体数字识别.洛阳理工学院学报(自然科学版) 2024,34(01),56-60+66

[5] Yann LeCun, Leon Bottou, Yoshua Bengio, and Partrick Haffner. Gradient-Based Learning Applied to Document Recognition