广州華商學院
GUANGZHOU HUASHANG COLLEGE

专业综合实践课程论文

题　　目：　基于深度学习的图像识别系统：手势识别

学　　院：　人工智能学院

专　　业：　数据科学与大数据技术

年级班别：　21 本数据科学与大数据技术 1 班

学　　号：　421470142

学生姓名：　盛洁

摘要

本实验主要研究了手势识别图像分类器的设计与实现。通过手势识别技术收集并处理五种手势（数字 1、2、3、4、5），不同类别的手势图像数据集，进而设计并测试三款分类器以对手势图像进行分类。

在数据预处理阶段，通过摄像头录制手势照片的代表性图像帧，构建了一个包含至少 1000 张图像的数据集。随后，对数据集进行了清洗，包括去重、过滤和修复，通过 OpenCV 库将图像转换为灰度并调整尺寸，以确保数据集的准确性和可靠性。

在分类器设计与实现阶段，选择了卷积神经网络（CNN）我们选择了线性核函数，并设置了相应的参数。对于 CNN 分类器，我们采用前向传播定义基础网络并构建一个简单的卷积神经网络。

在分类结果比较阶段，我们使用了准确率、精确率、召回率和 F1 分数作为评估指标。实验结果表明，CNN 分类器在各项指标上均表现最优，准确率达到了 0.96，显示出了较高的分类准确率和泛化能力。

综上所述，本实验成功验证了手势识别图像数据集分类器的设计与实现过程，并证明了 CNN 分类器在手势图像分类任务中的优越性。未来工作可以进一步探索其他先进的分类算法和模型优化方法，以提高手势识别的准确率和效率，为手势识别技术的实际应用提供更多可能性。

关键词：Python、卷积神经网络、手势识别、模型训练

目录

1 绪论

1.1 项目背景与目的

手势识别是通过识别人类手势并结合相关算法实现对手势语义分类的一项议题[1]，例如，在智能家居、虚拟现实、游戏开发、安防监控等领域，手势识别技术能够为用户提供更直观、自然的交互方式,因此，手势识别具有重要研究意义。手势识别通常依赖于机器学习算法来对提取的特征进行分类和识别。深度学习技术的兴起为手势识别提供了更强大的工具，如卷积神经网络（CNN）等。使用 Python 编写手势分析程序，可以利用其丰富的库和框架快速实现手势识别的各种功能。

本实验主要研究了手势识别图像分类器的设计与实现。通过手势识别技术收集并处理五种手势（数字 1、2、3、4、5），不同类别的手势图像数据集，进而设计并测试分类器以对手势图像进行分类。以证明了 CNN 分类器在手势图像分类任务中的优越性。未来工作可以进一步探索其他先进的分类算法和模型优化方法，以提高手势识别的准确率和效率，为手势识别技术的实际应用提供更多可能性。同时培养学生对图像处理和机器学习的基本理解。 训练学生使用深度学习模型解决实际问题的能力。 提高学生的数据预处理、模型构建、评估和优化的技能。

2 数据预处理

2.1 数据集类别

数据选择了五种手势作为数据采集的类别，其中包括通过手势比数字 1、2、3、4、5。

2.2 数据集采集方法

使用手机拍摄五种手势以及网上搜索手势照片，每种手势至少 200 张图像，共计不少于 1000 张图像。

2.3 数据集清洗

对文件夹内的数据图像进行处理，首先检查图片是否为使用 OpenCV 库读取图像，并将其转换为灰度图像；其次将读取的图像调整为 image_size 指定的尺寸，并使用 append（label）函数将图像展平承一维数组。

其次人工检查图像质量，确保图片质量。删除模糊、光线不足或手势不明显的图像，保证数据集的清晰度和准确性。

最后对于部分存在轻微噪声或伪影的图像，使用图像处理软件进行修复，其中包括去噪、锐化、增加对比度等操作，提高数据集的可用性。

2.4 数据归类

将清洗后的图像按照五种手势，数字一到五按照类别进行归类，确保每个类别的图像数量相近、平均。

2.5 数据划分

使用 train_test_split 函数将数据集划分为训练集和测试集。使用 DataLoader 和 SubsetRandomSampler 创建训练和测试数据加载器，用于在训练和测试过程中批量加载数据。这个函数随机选择指定比例的数据作为测试集，其余的作为训练集。在代码中，test_size=0.2 参数指定了 20%的数据将被用作测试集，剩下的 80%作为训练集。random_state=42 参数确保了每次运行代码时，数据集的划分是一致的，这是为了结果的可重复性。搭建 CNN 模型，定义了一个 CNN 类继承自 nn.Module，构建了一个简单的卷积神经网络，包含三个卷积层、两个全连接层以及 ReLU 激活函数和最大池化层。

3 模型构建

导入所需 Python 库，其中包括图像处理 OpenCV（cv2），NumPy，PyTorch 深度学习框架等数据加载和预处理工具，以及用于模型评估的 Scikit-learn 库。

3.1 CNN 原理

卷积神经网络（CNN）是一种深度学习模型，特别适用于处理图像数据。CNN 的原理是通过多层结构来学习图像的特征表示。它的核心是卷积层，该层使用滤波器（或称为卷积核）在图像上滑动以提取局部特征。每个滤波器负责识别图像中的特定特征，如边缘、纹理等。随后，激活函数（如 ReLU）引入非线性，使得网络能够学习更复杂的特征。CNN 通常包括池化层，用于降低特征维度，同时增加对图像位移的不变性。最后，全连接层将学习到的特征映射到最终的分类结果。CNN 通过反向传播和梯度下降算法进行训练，不断调整网络参数以最小化损失函数。这种分层特征学习机制使得 CNN 在图像分类任务中表现出色。

3.2 模型层结构

3.2.1 卷积层（Convolutional Layers）：

conv1：第一个卷积层，输入通道数为 1（灰度图像），输出通道数为 32，卷积核大小为 3x3，步长为 1；conv2：第二个卷积层，输入通道数为 32（`conv1`的输出），输出通道数为 64，卷积核大小为 3x3，步长为 1；conv3：第三个卷积层，输入通道数为 64（`conv2`的输出），输出通道数为 128，卷积核大小为 3x3，步长为 1。

3.2.2 激活函数（Activation Functions）：

relu1、relu2、relu3、relu4：ReLU（Rectified Linear Unit）激活函数，用于引入非线性，帮助网络学习复杂的模式。在卷积层之后使用，除了最后一个全连接层之前。

3.2.3 池化层（Pooling Layers）：

pool1、pool2、pool3：最大池化层，池化窗口大小为 2x2，用于降低特征图的空间维度，减少参数数量和计算量，同时保持特征的重要信息。

3.2.4 全连接层（Fully Connected Layers）：

fc1：第一个全连接层，输入特征数为 128*6*6（由前面的卷积和池化层确定），输出特征数为 128。fc2：第二个全连接层，输入特征数为 128（`fc1`的输出），输出特征数为 num_classes（类别数）。

3.2.5 Dropout 层：

dropout：Dropout 层，丢弃率为 0.5，用于防止过拟合，通过在训练过程中随机丢弃一部分神经元的输出来增加模型的泛化能力。

3.3 激活与优化选择

激活函数在卷积层之后使用，除了最后一个全连接层之前，用于引入非线性，帮助网络学习复杂的模式；损失函数中交叉熵损失（Cross-Entropy Loss）：criterion =nn.CrossEntropyLoss()，用于多分类问题，衡量模型输出的概率分布与真实标签的概率分布之间的差异；优化器 Adam：optimizer = optim.Adam(model.parameters())，一种自适应学习率的优化算法，结合了 RMSprop 和 Momentum 的优点，通常在训练深度学习模型时表现良好，不需要频繁调整学习率。

综上，CNN 模型通过组合卷积层、激活函数、池化层和全连接层来提取图像特征并进行分类。使用 ReLU 激活函数和交叉熵损失函数，以及 Adam 优化器来训练模型。

4 模型评估

4.1 评估指标：

在图像识别领域，准确率、召回率和 F1 分数是衡量模型性能的关键指标。准确率指的是模型预测正确的样本占总样本的比例，即（真正例 + 真负例）/ 总样本数。它反映了模型整体的预测准确性，但当数据集不平衡时，准确

率可能会产生误导；召回率（也称为查全率）衡量的是模型识别出的正样本占所有实际正样本的比例，即真正例 / （真正例 + 假负例）。召回率关注的是模型捕捉到的正样本的完整性，对于需要尽可能多地识别出所有正样本的场景尤为重要;F1 分数是准确率和召回率的调和平均数,计算公式为 2 * (准确率 * 召回率) / (准确率 + 召回率)。F1 分数在准确率和召回率之间取得平衡，当两者都很高时，F1 分数也高，反之亦然。它特别适用于数据集不平衡的情况，因为它同时考虑了准确率和召回率。

这三个指标共同作用，提供了模型性能的全面视图。在实际应用中，根据具体需求，可能会更重视其中一个指标。例如，在医疗图像识别中，可能更重视召回率，因为漏诊的代价可能非常高。而在一些需要高准确度的场景，如安全监控，准确率可能更为关键。F1 分数则提供了一个综合考量的指标，帮助开发者在不同情况下做出平衡。

4.2 混淆矩阵

表 4.1 卷积神经网络混淆矩阵

Accuracy: 0.961352657004831				
	precision	recall	f1-score	support
gesture_1	0.96	0.98	0.97	44
gesture_2	0.95	0.93	0.94	41
gesture_3	0.95	0.95	0.95	39
gesture_4	0.97	1.00	0.99	36
gesture_5	0.98	0.96	0.97	47
accuracy			0.96	207
macro avg	0.96	0.96	0.96	207
weighted avg	0.96	0.96	0.96	207
Weighted F1 Score: 0.9612401450138349				

由表 4.2 可得模型总体准确率为 0.961352657004831。各类别（gesture_1 至

gesture_5）的精确率和召回率普遍较高，F1 分数均超过 0.94，显示出模型在不同类别上的均衡性能。特别是 gesture_4，精确率和召回率均为 1.00，表明在该类别上实现了完美分类。加权 F1 分数 0.961240 与宏观平均 F1 分数一致，强调了模型整体的高综合性能。样本数量方面，gesture_5 最多，为 47 个，但模型在各类别上均未因样本数量差异而出现性能偏差。

4.3 数据可视化

4.3.1 混淆矩阵

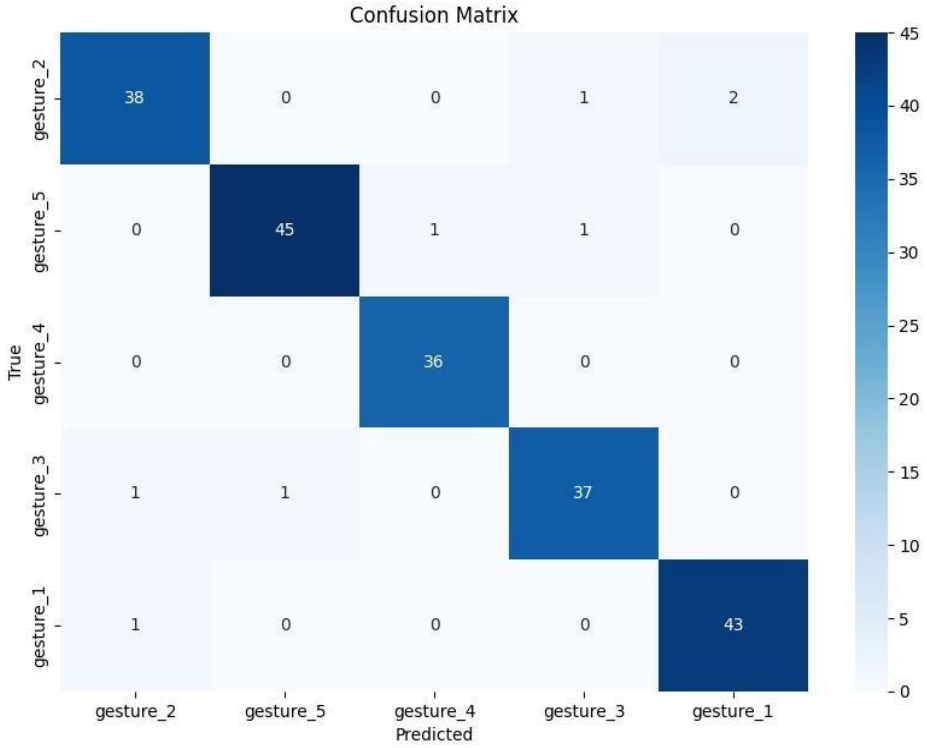

图 4.1 卷积神经网络混淆矩阵

由图 4.3 可得 gesture_1 的分类效果较差，大部分样本被错误地预测为 gesture_2 或 gesture_5。gesture_2 的分类效果较好，大部分样本被正确预测。gesture_3 存在一定程度的误分类，特别是被错误地预测为 gesture_2。

4.3.2 损失曲线可视化

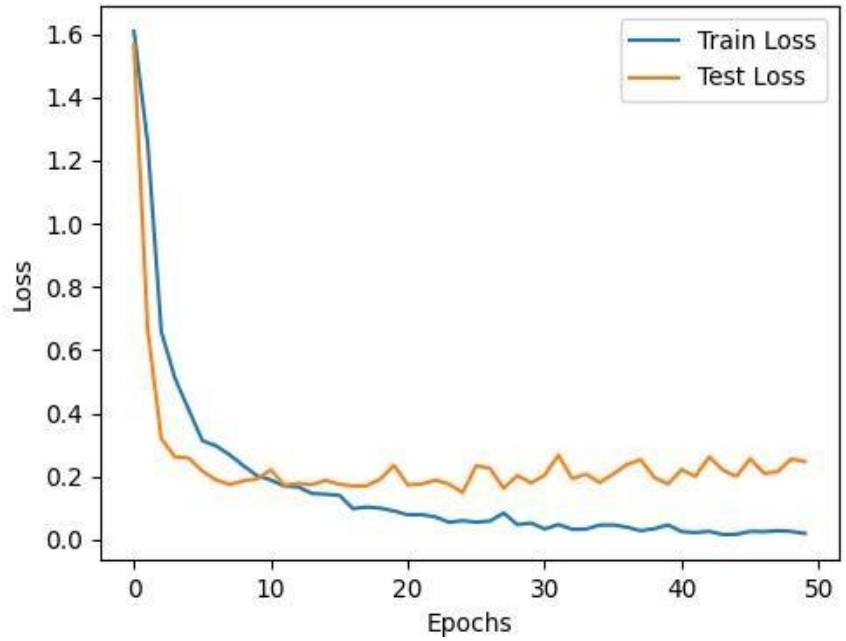

图 4.1CNN 损失曲线

由图 4.2 得，训练损失（Train Loss）和测试损失（Test Loss）都随着训练周期的增加而降低。这表明模型在训练集和测试集上的性能都在提升。损失曲线从大约 1.6 下降到 0.4 左右，显示了显著的下降趋势。

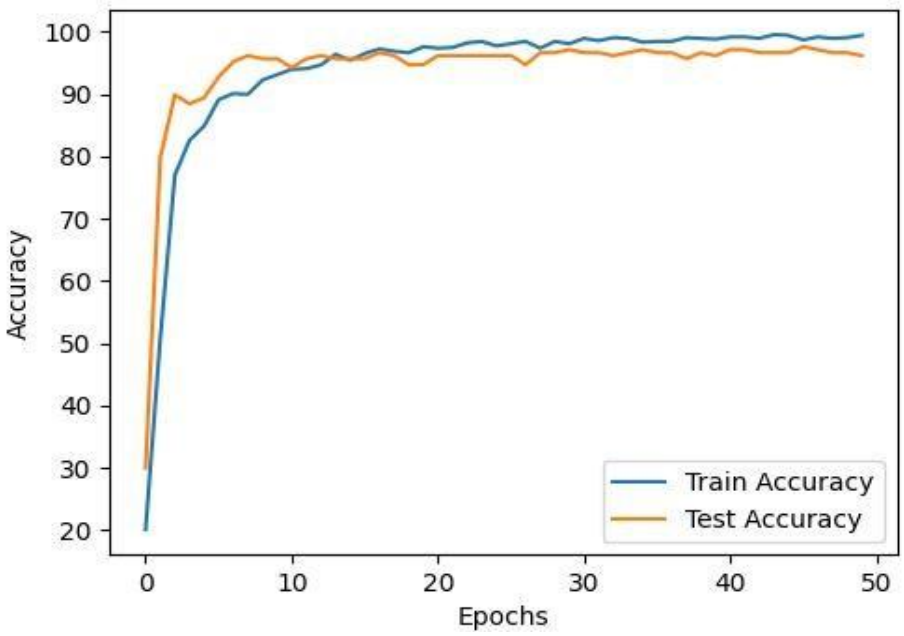

图 4.2 CNN 训练准确率曲线

由图 4.6 得，训练准确率（Train Accuracy）和测试准确率（Test Accuracy）

都随着训练周期的增加而提高。准确率从 0%开始，逐渐增加，训练准确率最终可能接近或超过 90%，测试准确率可能略低于训练准确率，模型在训练集上的拟合度较高。

5 结果分析与优化

5.1 结果分析：

根据本次结果比较可得，CNN 是最佳选择。其高准确率和 F1 分数证明了其在图像分类任务上的有效性。

其优点包括：强大的特征提取能力：CNN 能够自动学习图像的层次化特征，无需手动设计特征提取器。参数共享：通过卷积层的参数共享减少了模型的参数数量，降低了过拟合的风险。平移不变性：由于卷积操作的特性，CNN 具有一定程度的平移不变性，即对图像中物体位置的小幅度变化不敏感。

但同时存在计算资源需求高：深度 CNN 需要大量的计算资源，特别是在训练阶段。对小数据集的泛化能力有限：CNN 在小数据集上可能会过拟合，因为它们有大量的参数需要训练。解释性差：CNN 通常被视为“黑箱”模型，很难解释模型的决策过程。对输入数据的依赖性：CNN 对输入数据的质量和格式非常敏感，需要大量的数据预处理工作。调参困难：选择合适的网络架构、超参数和优化器可能需要大量的实验和经验等问题

CNN 模型在特定类别上的表现差异包括：类别复杂性：对于具有复杂纹理或结构的类别，CNN 可能需要更深的网络和更多的训练数据来准确分类。类别相似性：对于外观相似的类别，CNN 可能难以区分，因为它们之间的特征差异可能非常细微。数据不平衡：在类别不平衡的数据集中，CNN 可能会偏向于多数类，导致少数类的识别性能下降。类别的样本数量：对于样本数量较少的类别，CNN 可

能无法学习到足够的特征，导致性能下降。类别的多样性：在类别多样性高的数据集中，CNN 可能需要更多的数据来捕捉不同类别之间的差异。

5.2模型优化

考虑进一步探索深度学习模型，特别是 CNN 的潜力，通过调整网络结构或使用预训练模型来进一步提升性能。可以通过以下几种方式进行模型优化：数据增强：通过旋转、缩放、裁剪等方法增加数据多样性，减少过拟合，提高模型泛化能力。网络架构调整：根据任务需求选择合适的网络深度和宽度，可能包括添加或减少层数，调整卷积核大小等。超参数调优：包括学习率、批大小、优化器选择等，通过实验找到最佳配置。正则化技术：使用 Dropout、Batch Normalization 等技术减少过拟合，提高模型稳定性。激活函数选择：根据需要选择 ReLU、Leaky ReLU 等激活函数，以改善非线性和梯度传播。损失函数和优化器：根据问题类型选择合适的损失函数，如交叉熵损失，以及优化器如 Adam 或 SGD。迁移学习：利用预训练模型作为特征提取器，微调顶层以适应新任务，尤其在数据较少时有效。模型剪枝和量化：减少模型大小和计算量，提高推理速度，适用于部署到资源受限的设备。在实际应用中，应根据特定场景的需求选择合适的分类器；机器学习模型的优化是一个持续的过程，定期回顾和更新模型以适应新的数据和需求[2]。

6 参考文献

[1]袁帅,吕佳琪.基于 Copula 贝叶斯分类器与改进 YOLOv5 网络的手势识别研究[J].科技资讯,2023,21(20):26-29.

[2] 张政.基于深度学习的手势检测和识别研究[D].贵州大学,2021.DOI:10.27047/d.cnki.ggudu.2021.000948.