基于 cnn 的卡纳达语手写数字识别

徐斌杰 陶泽辉 嵇宇泰

摘要—本研究利用卷积神经网络（CNN）对 Kannada-MNIST 数据集进行了分类实验。通过数据预处理、数据增强、模型构建与训练、模型评估和错误分析，我们实现了对 Kannada 字符的自动识别。最终模型在验证集上取得了较高的分类准确率，并生成了预测结果的提交文件。

***Index Terms*—卷积神经网络，手写数字识别，卡纳达语，深度学习，数据增强，模型评估，Kannada-MNIST**

I. 引言

A. 研究背景

手写数字识别是计算机视觉和模式识别领域中的一个重要研究方向。它在邮政编码识别、银行支票处理、表单数据录入等实际应用中具有广泛的应用前景。近年来，随着深度学习技术的发展，尤其是卷积神经网络（Convolutional Neural Networks，CNN）的出现，手写数字识别的精度和效率得到了显著提升。传统的手写数字识别方法依赖于手工特征提取，而 CNN 通过多层卷积和池化操作，可以自动从数据中学习特征，减少了手工特征提取的复杂性和局限性。

虽然 CNN 在手写数字识别中表现出色，但现有的大部分研究和应用主要集中在拉丁字母和阿拉伯数字上，例如著名的 MNIST 数据集。然而，对于卡纳达语等印度区域语言的手写数字识别研究相对较少。卡纳达语是印度南部卡纳塔克邦的主要语言，其手写数字识别在地方文献数字化、教育资源数字化等方面有着重要的意义。然而，由于卡纳达语手写数字的形态复杂多样，加上数据集的稀缺，使得该领域的研究面临一定挑战。

B. 研究目标

本文旨在探讨基于卷积神经网络的卡纳达语手写数字识别方法。具体目标包括：

1. 构建一个适用于卡纳达语手写数字识别的卷积神经网络模型。

2. 通过实验验证所提出模型的有效性，并与现有方法进行对比分析。

3. 提出在手写数字识别中应用深度学习技术的建议和改进方向。

C. 论文结构

本文结构如下：第二部分为文献综述，介绍手写数字识别的发展历史、现有方法及卷积神经网络的应用；第三部分介绍研究中使用的数据集及其预处理方法；第四部分详细描述所采用的卷积神经网络模型架构及训练方法；第五部分为实验部分，包括实验设置、结果分析及与其他方法的对比；第六部分对实验结果进行讨论，分析模型的优缺点，并提出改进建议；第七部分总结本文的主要贡献，并展望未来的研究方向；第八部分列出参考文献，附录中提供代码实现及数据集链接。

II. 相关工作

在手写数字识别领域，研究人员已经提出并实施了许多方法。下面将简要回顾这些方法，重点介绍与卡纳达语手写数字识别相关的工作。

A. 传统方法

早期的手写数字识别方法主要依赖于手工特征提取和传统的机器学习算法。这些方法包括：

1. 特征提取：使用边缘检测、轮廓分析和形状描述符等技术从图像中提取特征。这些特征用于表示数字图像的几何和纹理信息。

2. 分类算法：使用支持向量机（SVM）、k 近邻（k-NN）和随机森林等传统分类算法对提取的特征进行分类。这些方法在一定程度上能够解决手写数字识别问题，但它们对特征提取的依赖使其在复杂多样的手写体情况下表现有限。

B. 基于深度学习的方法

近年来，随着深度学习的发展，卷积神经网络（CNN）在图像识别任务中表现出色。CNN 通过自动学习特征表示，减少了手工特征提取的复杂性，并显著提高了识别精度。具体的应用包括：

1. LeNet-5：由 Yann LeCun 等人提出的 LeNet-5 是早期应用于手写数字识别的经典 CNN 模型。它在 MNIST 数据集上的成功验证了 CNN 在手写数字识别中的潜力。

2. AlexNet：AlexNet 的成功推动了深度学习在图像识别领域的发展。尽管其主要应用于 ImageNet 数据集，但其架构思想被广泛应用于其他图像识别任务，包括手写数字识别。

3. VGG 和 ResNet：这些更深层的网络进一步提高了图像识别的精度。VGG 通过增加网络深度和使用小卷积核来提高性能，而 ResNet 通过引入残差连接解决了深层网络的梯度消失问题。

C. 卡纳达语手写数字识别

与拉丁字符和阿拉伯数字相比，卡纳达语手写数字识别的研究相对较少。然而，近年来一些研究者开始关注这一领域，并取得了一些成果：

1. 数据集构建：一些研究者构建了卡纳达语手写数字的数据集，为进一步研究提供了基础。例如，UCI 机器学习库中包含了一个卡纳达语手写数字数据集，包含了从多个志愿者处收集的手写数字图像。

2. 传统与深度学习方法结合：一些研究采用了传统方法与深度学习方法相结合的策略。例如，通过预处理步骤提取手写数字的边缘或轮廓，然后使用 CNN 进行分类。这些方法在一定程度上提高了识别率。

3. 端到端的深度学习模型：近年来，更多的研究开始采用端到端的深度学习模型来处理卡纳达语手写数字识别问题。这些模型从输入的原始图像直接学习特征，并进行分类，简化了处理流程，并提高了识别精度。

D. 现有方法的对比分析

对现有的卡纳达语手写数字识别方法进行对比分析，可以发现：

1. 传统方法：尽管在一定程度上有效，但由于手工特征提取的局限性，其在复杂多样的手写体情况下表现较差。

2. 基于深度学习的方法：显著提高了识别精度，特别是端到端的 CNN 模型，通过自动学习特征表示，简化了处理流程，并在处理复杂手写体时表现出色。

3. 数据集的重要性：对于卡纳达语手写数字识别，数据集的质量和数量对模型性能有显著影响。未来的研究应关注构建更大规模、更高质量的卡纳达语手写数字数据集。

综上所述，基于 CNN 的手写数字识别方法在卡纳达语手写数字识别中显示出巨大潜力。本文将在现有研究的基础上，进一步探索和改进 CNN 模型，以提高卡纳达语手写数字的识别率。

III. 方法

在本研究中，我们采用了卷积神经网络（CNN）来实现卡纳达语手写数字识别。具体方法包括数据集的准备与预处理、CNN 模型的设计与训练，以及超参数的选择与优化。

A. 数据集准备与预处理

1. 数据集描述：本研究使用了一个包含卡纳达语手写数字的公开数据集。该数据集包含从多个志愿者处收集的手写数字图像，每个数字对应一个唯一的标签。其中训练数据有 60000 行和 785 列测试数据有 5000 行和 785 列。在给定的数据集中含有 735 列，其中，Label 包含要预测的标签（0-9）；Pixel 0 到 Pixel 783 是图像中的每一个像素点（即每一章图片包含 28*28 个像素点），同时，所有类别的分布均等，每一个数字都有 6000 个示例。

2. 数据预处理：对于大多数图像数据，像素值是 0 到 255 之间的整数。在神经网络中，使用较大的整数值输入可能会破坏或减慢学习过程。因此，最好对像素值进行归一化，以便每个像素值的值介于 0 到 1 之间。对此，可以通过将所有像素值除以最大像素值来实现：即除以 255。

```
X_train=X_train/255
test=test/255
```

此外，我们修改数据尺寸，使其满足神经网络的尺寸需求。将数据重塑为 60000 个高度 28、宽度 28 和 1 个通道的示例。对于标签数据，进行 one-hot 编码，以便模型更容易处理。

```
X_train=X_train.values.reshape(-1,28,28,1)
test=test.values.reshape(-1,28,28,1)

Y_train=to_categorical(Y_train)
```

为了避免过度拟合问题，可以人为地扩展手写数字数据集。这个想法是通过小的转换来改变训练数据，以

重现某人书写数字时发生的变化（数据增强）。本次实验中，采用了以下数据增强方法：

- 随机将一些训练图像旋转 10 度
- 随机将一些训练图像缩放 10%
- 随机将图像在水平位置上平移宽度的 10%
- 随机将图像在垂直位置上平移高度的 10%

```
datagen = ImageDataGenerator(
    featurewise_center=False,
# 不将整个数据集的均值设置为0
    samplewise_center=False,
# 不将每个样本的均值设置为0
    featurewise_std_normalization=False,
# 不将整个数据集除以其标准差来归一化
    samplewise_std_normalization=False,
# 不将每个样本除以其标准差来归一化
    zca_whitening=False,  # 不应用ZCA白化
    rotation_range=10,
# 随机旋转图像，旋转范围为0到10度
    zoom_range=0.1,
# 随机缩放图像，缩放范围为10%
    width_shift_range=0.1,
# 随机水平平移图像，平移范围为总宽度的10%
    height_shift_range=0.1,
# 随机垂直平移图像，平移范围为总高度的10%
    horizontal_flip=False,
# 不随机水平翻转图像
    vertical_flip=False)
# 不随机垂直翻转图像
datagen.fit(X_train)
```

B. CNN 模型设计

我们构建了一个卷积神经网络模型，包含以下主要层次：

1. 第一卷积块:

· 两个卷积层 (Conv2D)，每个使用 32 个 5x5 的过滤器，padding 设为'Same'，激活函数为'relu'。

· 批归一化层 (BatchNormalization)，有助于加速训练和提高模型稳定性。

· 最大池化层 (MaxPool2D)，池化窗口为 2x2。

· Dropout 层，丢弃率为 0.25，用于防止过拟合。

2. 第二卷积块:

· 两个卷积层，使用 64 个 3x3 的过滤器，padding 设为'Same'，激活函数为'relu'。

· 批归一化层 (BatchNormalization)。

· 最大池化层，池化窗口为 2x2，步幅为 2x2。

· Dropout 层，丢弃率为 0.25。

3. 第三卷积块:

· 两个卷积层，使用 32 个 5x5 的过滤器，padding 设为'Same'，激活函数为'relu'。

· 批归一化层 (BatchNormalization)。

· 最大池化层，池化窗口为 2x2。

· Dropout 层，丢弃率为 0.25。

4. Flatten 层:

· 将多维特征图展平为一维向量，以便输入到全连接层中。

5. 全连接层:

· 全连接层 (Dense)，包含 256 个神经元，激活函数为'relu'。

· Dropout 层，丢弃率为 0.4，用于进一步减少过拟合风险。

6. 输出层:

· 输出层 (Dense)，包含 10 个神经元，对应于 10 个类别，激活函数为'softmax'，用于进行分类预测。

通过这些层的组合，我们的模型能够有效地提取图像特征并进行准确的分类。

```
model = Sequential([
    # 第一个卷积层
    Conv2D(96, (11, 11),
           strides=(4, 4),
           activation='relu',
           input_shape=(28, 28, 1),
           padding='same'),
           # 使用padding='same'适应28x28输入
    MaxPool2D(pool_size=(3, 3),
           strides=(2, 2),
           padding='same'),
    # 第二个卷积层
    Conv2D(256, (5, 5),
           activation='relu',
           padding='same'),
    MaxPool2D(pool_size=(3, 3),
           strides=(2, 2),
           padding='same'),
    # 第三个卷积层
    Conv2D(384, (3, 3),
           activation='relu',
           padding='same'),
    # 第四个卷积层
    Conv2D(384, (3, 3),
           activation='relu',
           padding='same'),
```

```
        # 第五个卷积层
        Conv2D(256, (3, 3),
               activation='relu',
               padding='same'),
        MaxPool2D(pool_size=(3, 3),
               strides=(2, 2),
               padding='same'),
        # Flatten层将多维特征图展平成一维向量
        Flatten(),
        # 全连接层
        Dense(4096, activation='relu'),
        Dropout(0.5),
        Dense(4096, activation='relu'),
        Dropout(0.5),
        # 输出层
        Dense(10, activation='softmax')
])
```

IV. 实验

A. 模型训练

在训练开始时，采用了较高的学习率，以获取较高学习率带来的快速计算优势。我们使用 Adam 优化器，学习率为 0.001，损失函数为多分类交叉熵。在训练过程中使用 ReduceLROnPlateau 回调函数，当验证准确率在连续 3 个 epoch 没有提升时，学习率减半。

经过上述步骤的准备，模型在训练时具有较高的准确率和较低的损失，模型在验证集上具有较好的效果

B. 模型评估

在评估模型时，分别用损失和准确率来衡量模型的效果

可以看到，随着训练轮次的增加，模型效果越来越好，最终趋于稳定

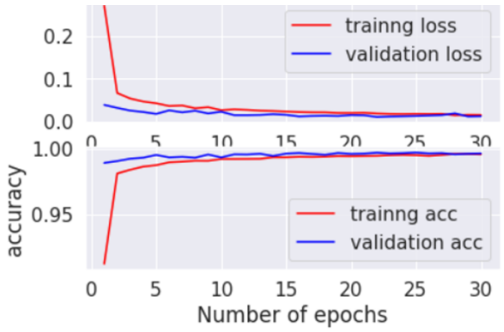

图 1. 训练曲线

通过混淆矩阵评估模型在验证集上的表现，训练好的模型在某些情况会较高的可能性出现错误：

- 将 0 识别为 1
- 将 3 识别为 7

	0	1	2	3	4	5	6	7	8	9
0	853	5	0	0	0	0	0	0	0	0
1	0	913	0	0	0	0	0	0	0	0
2	1	0	907	0	0	0	0	0	0	0
3	0	0	0	884	0	0	0	6	0	0
4	0	0	0	0	923	0	0	0	3	0
5	0	0	0	2	1	871	0	0	0	0
6	0	0	0	0	0	0	859	2	0	1
7	0	0	0	1	0	0	2	930	0	0
8	0	0	0	0	0	0	0	0	918	0
9	1	0	0	0	0	0	3	0	0	914

图 2. 混淆矩阵

混淆矩阵显示了真实标签与预测标签之间的对应关系。选取一些模型预测错误的样本进行可视化，分析这些错误的原因，以便进一步改进模型。

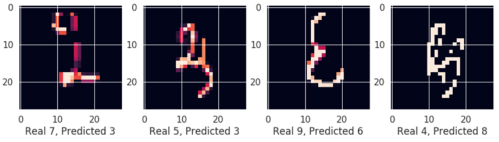

图 3. 可视化样例

C. 最终分数

Best Score: 0.97240

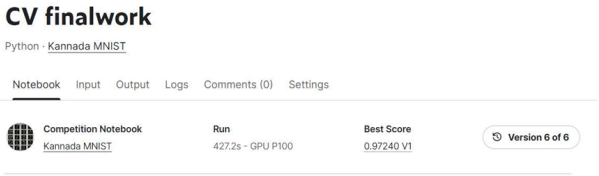

图 4. 最终分数

V. 总结

本研究使用卷积神经网络（CNN）对 Kannada-MNIST 数据集进行了分类实验，取得了显著的分类效果。通过数据预处理、模型训练和评估，证实了 CNN 在处理卡纳达语手写数字识别任务中的有效性。未来的研究可以进一步探索更复杂的模型架构，增加数据增强

策略，并延长训练时间，以进一步提升模型的准确率和泛化能力。

参考文献

[1] Shahules. *Indian way to learn CNN*. [Online]. Available: https://www.kaggle.com/code/shahules/indian-way-to-learn-cnn. [Accessed: 2019].

[2] V. U. Prabhu, "Kannada-MNIST: A new handwritten digits dataset for the Kannada language," arXiv:1908.01242 [cs.CV], Aug. 2019. [Online]. Available: https://doi.org/10.48550/arXiv.1908.01242.

VI. 代码链接

请访问 github 主页:
https://github.com/eric-522620/CV-finalwork

VII. 分工

徐斌杰：组长，构建神经网络模型并训练
陶泽辉：组员，信息整理与 ppt 制作
嵇宇泰：组员，撰写实验报告