

电表识别模型设计思路

总体流程

电表数字图 → 数据处理 → CNN 特征提取 → RNN 序列建模 → CTC 损失训练/解码输出 → 得到数字识别结果

字符与标签编码工具 (utils部分)

字符集按识别需求定义为"0123456789"，共 10 类 + 1 个 `blank`（用于 CTC）。
定义了以下几个函数：

- `encode_label`：将字符串标签（如 "123"）编码为数字索引 `[1,2,3]`。
- `decode_label`：将模型预测索引转回字符串（并去除重复/blank）。
- `ctc_beam_search`：实现 CTC 的 Beam Search 解码。
- `decode_output`：支持 `argmax` 或 `beam` 两种方式解码输出。
经过实际训练比较，我们发现此任务用 `argmax` 解码效果优于 `beam`

数据增强

定义了 `MeterDigitDataset` 类：

- 从 CSV 文件中读取图片路径和标签。设计了 train/test 模式切换，应用于不同的图像增强：`train` 模式对图像进行**随机旋转 ($\pm 2^\circ$) 和标准化处理**。`test` 模式仅进行**固定尺寸缩放与标准化**。
- 返回：图像张量、编码标签、标签长度、模型输入长度（图宽/16）等。

CRNN模型

选择融合 CNN + RNN 的架构，为数字序列图像建模：

- **CNN 部分**：采用 **4 层卷积块**（每层包括：`Conv2d` → `BatchNorm` → `ReLU` → `MaxPool`），逐层提取图像的空间特征；每次池化将图像高度减半，最终高度由原始输入 `img_height` 变为 `H/16`，宽度 `W` 则作为序列时间步展开；输出特征图尺寸为 `(B, 256, H/16, W)`。
- **压缩层**：将 CNN 输出的特征 `(C × H')` 通过 `Linear` 层映射至统一维度 `256`，形成时间序列特征：`(B, W, 256 × H') → (B, W, 256)`。这一步用于降低参数复杂度，提升后续 RNN 训练稳定性。
- **RNN (双向 LSTM) 部分**：从前后两个方向建模图像序列中各时间步的上下文依赖。
- **输出层**：线性映射为每个时间步上的分类结果（共 **11 类**：10 数字 + blank）。

训练策略

- 划分训练集和验证集（80% / 20%）。使用 `CTCLoss` 进行训练。
- 每轮训练：
 - 前向传播 + CTC Loss。
 - 后向传播 + 优化器更新。
 - 学习率调度器（`ReduceLROnPlateau`）根据验证集 loss 调整学习率。
- 验证阶段：用 `argmax` 解码，按前 5 位数字是否匹配计算准确率。
- 自动保存：验证集 loss 最小的模型和验证集准确率最高的模型。
定义了CTC专用的 `collate_fn()` 函数用于拼接不等长 label 序列，适配CTC的要求：合并图像、标签，记录每个样本的 label 长度和输入长度。