### 1. ****模块概述****

该模块是接在从图片中截取车牌后面的。也就是说，在这个模块运行之前，车牌已经被从车辆图像中单独提取出来。输入到这个模型的将是已经截取出来的车牌图像，模型会对其进行进一步的处理，并直接输出车牌字符。

### 2. ****端到端模型 (End-to-End)****

这个车牌识别模型是一个典型的**端到端**（End-to-End）模型。与传统的车牌识别方法不同，传统方法通常会将任务分为多个步骤，包括：

* 车牌检测
* 字符分割
* 字符识别

而端到端模型将这些步骤合并为一个整体，即模型直接接受输入图像并直接输出识别结果，无需进行显式的字符分割过程。换句话说，模型在训练过程中已经学会了如何从整张车牌图像中提取并识别字符，这样可以大大简化处理流程，提高效率。

我们小组原本计划采用先字符分割，再字符识别的方式，但是最终发现这种方法不仅正确率比较低，而且速度上过于缓慢。综合比较下，我们采用了这种端到端识别的方法。

### 3. ****LPRNet****

我们使用的具体模型是**LPRNet**，这种模型已被多篇研究论文提及，并因其高效的性能而得到广泛认可。LPRNet本质上是一种轻量级的**端到端车牌识别**模型，能够在不需要复杂的中间步骤（如字符分割）的情况下，直接从车牌图像中提取字符。

#### LPRNet的特点：

* **端到端训练**：LPRNet将车牌检测、字符分割和字符识别的任务合并到一个统一的网络中，简化了整个识别过程。
* **轻量化设计**：相较于传统的车牌识别网络，LPRNet的设计更加简洁，参数量较少，因此在硬件要求较低的设备上也能高效运行。
* **高效性**：尽管模型较为轻量化，LPRNet依然能够提供相对较好的识别准确率，特别是在处理速度上有明显的优势，适用于实时识别场景。

### 4. ****轻量化与准确率的权衡****

LPRNet是一个非常轻量化的模型，这意味着在很多情况下，它的准确率可能会稍低于更复杂、更重的模型。这是因为更复杂的模型通常具有更多的参数和更强的特征提取能力，但这也意味着它们对计算资源的需求更高。因此，在应用场景中，LPRNet的轻量化设计提供了一个**速度**与**准确率**的平衡：

* **速度**：LPRNet的速度较快，能够支持实时车牌识别，特别适用于需要快速反应的场合。
* **准确率**：虽然准确率可能略逊色于一些重型模型，但对于大多数应用场景，它的准确度仍然能够满足要求。

### 5. ****模型的工作流程****

* **输入**：输入是从车牌图像中裁剪出来的车牌部分。图像首先会被调整为适当的尺寸并进行归一化处理，以确保模型能够正确处理。
* **特征提取**：模型通过多个卷积层提取车牌的特征，捕捉车牌上字符的结构信息。
* **字符识别**：模型会直接输出一系列字符标签，这些字符标签对应车牌号上的实际字符。
* **后处理**：通过对输出的字符序列进行解码和去除重复字符等步骤，得到最终的车牌号。

### 6. 网络结构

该结构是基于卷积神经网络（CNN）设计的，主要包括以下几个部分：

6.1 **输入层**

* **输入图像大小**: 车牌图像大小通常为 94x24 像素，且该图像大小对每个模型配置是固定的。
* 输入图像是三通道的（RGB），表示为 3 x 94 x 24。

#### 6.2 ****Backbone 部分****

这个部分负责从输入图像中提取特征，包含多个卷积层、池化层和激活层。以下是网络中的主要层次：

**卷积层 + ReLU 激活函数**:

* + 第一层是一个 3x3 卷积层，输出 64 个特征图。
  + 然后使用 BatchNorm 和 ReLU 激活函数进一步处理。

**最大池化层**: 用于减少空间维度，增强特征。

* + 使用 MaxPool3d，池化核为 (1, 3, 3)，步长为 (1, 1, 1)。

small\_basic\_block: 这是一个自定义的小型卷积块，由一系列卷积层组成，目的是进一步提取车牌图像的特征。

* + 每个 small\_basic\_block 先进行 1x1 卷积来调整通道数，再进行不同尺寸的卷积（如 (3, 1) 和 (1, 3)）来处理空间特征，最后再使用 1x1 卷积来恢复通道数。

small\_basic\_block **层重复**:

* + 在模型中，我们可以看到有多个这样的卷积块（如 ch\_in=64, ch\_out=128，ch\_in=64, ch\_out=256 等），这些块帮助网络更好地捕捉车牌图像中的各种空间模式。

**池化层**:

* + 在某些层后，模型使用 MaxPool3d 或 AvgPool2d 来减少特征图的空间维度，以降低计算量，并增强对车牌字符的全局信息理解。

**Dropout 层**:

* + 为了防止过拟合，网络使用了 Dropout 层，这有助于在训练过程中随机丢弃部分神经元，从而提高网络的泛化能力。

#### 6.3 ****全局上下文建模（Global Context）****

该部分是 LPRNet 的特色之一，目的是通过对不同层特征的操作（如平均池化、平方和均值计算等）来建模车牌的全局上下文信息。这种信息有助于增强车牌字符的识别效果。

**平方和均值化**: 该步骤对每个特征图的平方进行平均，然后标准化，这样可以增强网络对全局特征的关注。

**特征拼接**: 将不同层的全局特征进行拼接，形成一个新的特征向量。

#### 6.4 ****输出层****

**卷积层**: 最后一层通过卷积层将特征映射到输出字符类别上。具体地，Conv2d 使得输出的通道数与字符类别数相等。

* + 例如，如果车牌号的最大长度为 8 个字符，并且字符集包含 66 个字符（包括省份码和数字），那么输出的通道数就是 66。

**平均池化**: 使用平均池化（torch.mean(x, dim=2)）来降低维度，输出最终的字符预测结果。

### 训练与测试

对于本次模型的训练，我们选取的数据集是CBLPRD330k，我们选取了50000张图片进行训练。训练集验证集的比例为7：3，训练20个epoch后，

当下的测试结果为，对于随机选取的2000张新图片，对于任意车牌，完全正确的概率为43%， 对于任意字符，完全正确的概率约为80%

Reference:

LPRNet原论文：[[1806.10447v1] LPRNet: License Plate Recognition via Deep Neural Networks](https://arxiv.org/abs/1806.10447v1)

数据集：<https://github.com/SunlifeV/CBLPRD-330k>

参考实现方式：[sirius-ai/LPRNet\_Pytorch: Pytorch Implementation For LPRNet, A High Performance And Lightweight License Plate Recognition Framework.](https://github.com/sirius-ai/LPRNet_Pytorch)