

7 结果

本章主要分为五个部分，分别展示了车牌检测和提取，字符分割，模型训练，字符识别的过程与结果以及软件页面。其中，车牌检测和提取，字符分割，软件页面三部分主要以图片的形式展示结果。模型训练，字符识别两部分主要以折线图，直方图等形式展示。

7.1 车牌提取

这一部分主要展示对用户拍下的图片进行预处理，轮廓初选，轮廓聚类，以及提取车牌的每一步结果。

7.1.1 预处理

图 7-1 为在住宅小区中用安卓手机拍摄的一张彩色原图，其中包括了树，车，车牌，车牌字符，影子等一系列元素。原图经灰度化处理后会得到如图 7-2 所示的灰度图。然后灰度图经过黑白顶帽处理，高斯模糊，自适应阈值化一系列处理后，得到如图 7-3 所示的阈值化后的图，可观察到该图中的所有像素点非黑即白，即将原图中的所有物体全部用黑白两色勾勒填充。



图 7-1 用户拍摄图



7-2 灰度图



图 7-3 阈值图

7.1.2 轮廓初步筛选

对图 7-3 中所有白色部分用绿色线条勾勒其轮廓，得到如图 7-4 所示的轮廓。之后经过初步筛选，去掉不可能是车牌字符的轮廓，剩下的轮廓如图 7-5 所示。

从图片中可以观察到，大部分或细碎或长宽比例不合格的轮廓被去掉，主要剩下了车牌字符，车标，以及少部分细碎的轮廓。而从 Android Studio 日志台结果截图 7-6 中，能看到图 7-4 和 7-5 中具体的轮廓个数被量化，图片 7-4 中共有 20271 个轮廓，图片 7-5 中却只剩下了 2043 个轮廓，是原图轮廓个数的 1/10。这一步骤大大减少了后续处理的计算量。

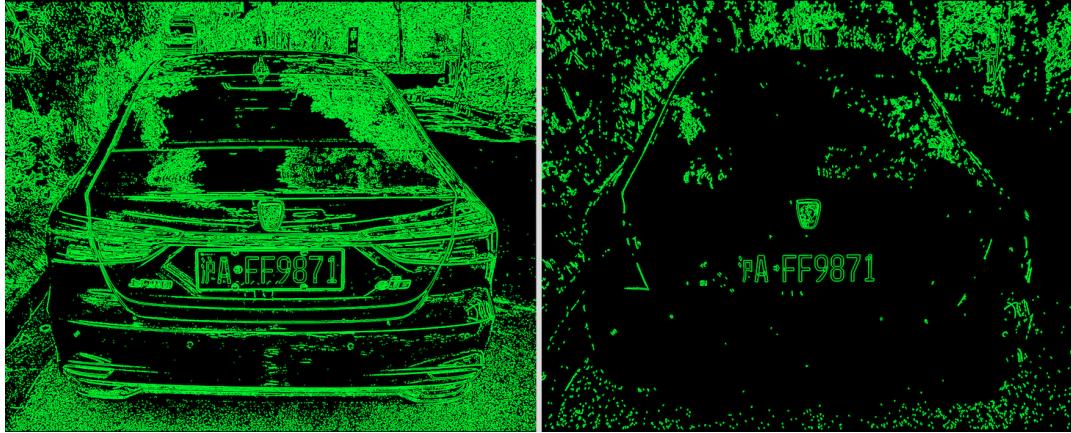


图 7-4 轮廓图

图 7-5 初步筛选轮廓图

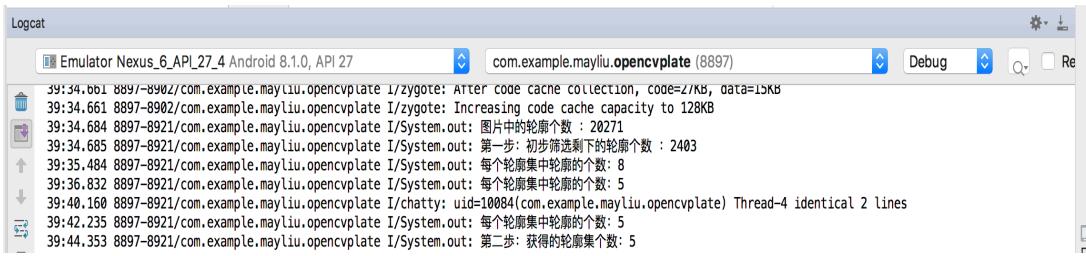


图 7-6 Android 日志台输出结果

7.1.3 轮廓聚类

对图 7-5 中的轮廓集根据轮廓的特征进行聚类分组，且只保留和显示组内轮廓数大与等于 5 的轮廓组，并将同一组的所有轮廓用同一随机颜色的线条勾勒，得到的结果如图 7-7 所示。从图中可以看到，一共剩下了 5 组轮廓，分别用红色，亮绿，深绿，浅绿以及橙色五个程序随机生成的颜色表示。同时从图 7-6 日志台的记录可以发现各轮廓集中轮廓的个数分别为 5, 5, 5, 5, 8。

然后对五组轮廓分别计算其外接矩形的位置和大小，并用红色线条勾勒出来，如图 7-8 所示。红色矩形代表可能是矩形车牌的区域。

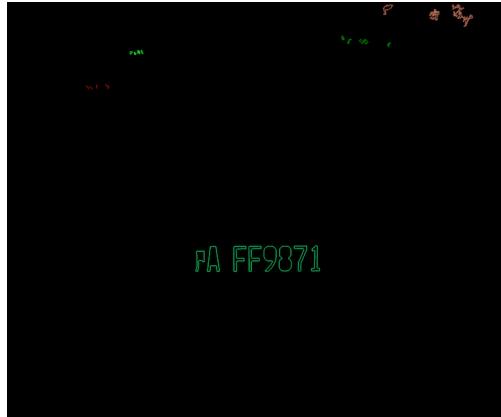


图 7-7 轮廓聚类图



图 7-8 可能的车牌矩形

7.1.4 提取车牌

该项目中假设轮廓数最多的一组为车牌字符组，所以对图 7-8 中个数为 8 的轮廓集的矩形区域在原图中进行提取，得到如下图 7-9 的车牌图片。同理，图 7-10 为从其他图片中提取的蓝色车牌图片，在此展示原图以便接下来观察蓝色和绿色牌照处理的不同。至此，车牌提取过程结束。



图 7-9 车牌图片 1 (绿色)



图 7-10 车牌图片 2 (蓝色)

7.2 字符分割

7.2.1 预处理与轮廓筛选

对图 7-9 的绿色车牌进行一系列包括灰化，黑白顶帽，高斯模糊，自适应阈值化的处理后，得到如图 7-11 所示的阈值化后的车牌。然后通过初步筛选，过滤掉一部分不可能是车牌的轮廓得到如图 7-12 所示的结果。通过观察可以发现车牌字符被完整地保留了下来，并。之后对保留下来的轮廓进行进一步的聚类，并保留和显示轮廓个数大于 5 的轮廓组，结果如 7-13 所示。通过图 7-13 可以看出经过聚类和筛选后，只剩下了车牌字符的轮廓，而且由于汉字的不连通性，“沪”字只保留了其中的一部分，对于这一缺失的处理结果会在下一节中进行展示。



图 7-11 阈值化车牌 1



图 7-12 轮廓初步筛选 1



图 7-13 轮廓聚类 1

对图 7-10 中的蓝色车牌进行 `inRange()` 和 `bitwise_not()` 操作，将图片中蓝色部分用黑色填充，其他填充白色，得到如 7-14 所示的阈值化图。然后同样进行轮廓筛选和聚类，结果如图 7-15 和 7-16 所示



图 7-14 阈值化车牌 2



图 7-15 轮廓初步筛选 2



图 7-16 轮廓聚类 2

7.2.2 分割车牌字符

程序对图 7-13 中的所有 8 个轮廓找到其对应的外接矩形，并在阈值化后的车牌图片中用绿色线条画出，结果如图 7-17 所示。从 7-17 图中可以看出除汉字外的所有车牌字符都准确地被矩形框所包围，可以进行下一步的分割工作。所以，我们

要对汉字字符进行一系列的计算与处理，具体步骤详见，最后得到的结果如图 7-18 所示，汉字也被绿色的矩形框正确地包围。

在找到车牌上所有字符的外接矩形后，我们就可以确定所有字符在图片中的位置和大小，然后将其一一分割出来，得到如图 7-19 的 8 个字符的图片。至此，车牌字符分割过程结束。

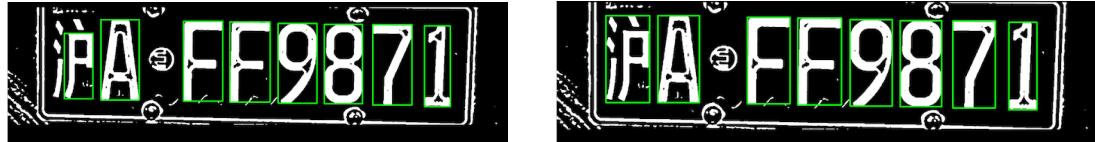


图 7-17 未对中文字符处理后的结果

图 7-18 对中文字符处理后的分割结果



图 7-19 分割结果

7.3 卷积神经网络的建立与训练

由于强大的 Tensorflow 包中提供了一种可以将用户所设计的神经网络和训练过程的可视化的工具—Tensorboard。通过编程，Tensorboard 可以将用户自己搭建的神经网络模型，训练中各个参数的变化以及准确率的变化等反映在对应的折线图，直方图以及流程图等中，如图 7-20 所示。这一部分主要基于模型生成的神经网络框图，各神经层权重集与偏差集变化的折线，直方图来进行结果分析与展示。可以从图 7-20 中看出，在这里设置曲线的平滑度为 0.852，使得折线图基本拟合成一条曲线，其中颜色较浅的一条线为真实值，较深的一条线是经过平滑后的曲线。在这一部分的结果展示中，主要展示的是数字+字母样本集的训练过程与结果。训练的迭代次数为 1000。

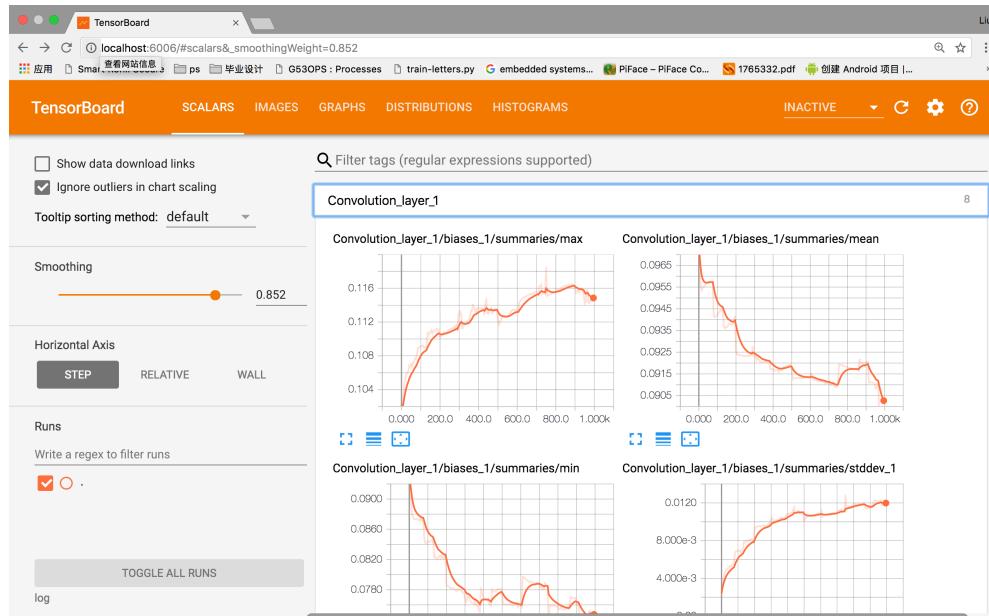


图 7-20 Tensorboard 可视化页面

7.3.1 损失率与准确率

图 7-21 损失率折线图与图 7-22 准确率折线图是用来观察模型在 1000 次迭代训练中损失率与准确率的变化过程。

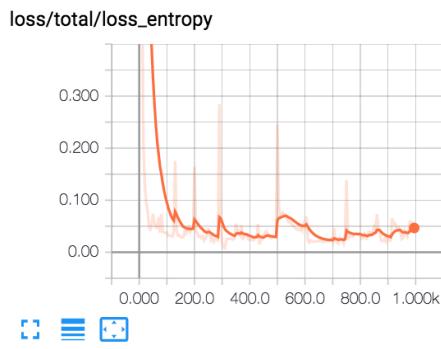


图 7-21 损失熵折线图

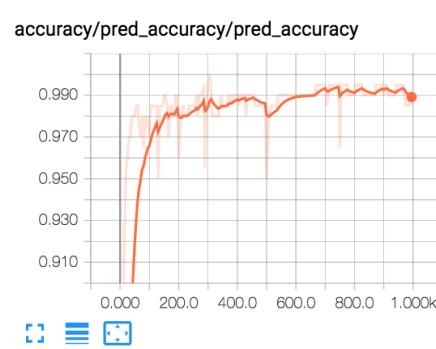


图 7-22 准确率折线图

图 7-21 的竖轴代表交叉熵 (loss_entropy) , 即预测值概率分布与实际值概率分布之间的距离, 图 7-17 的数轴代表预测的准确率, 即测试集中图片的分辨率取率, 两者的横轴均代表训练迭代的次数。从图 7-21 中可以观察到, 在 0 到 200 次迭代间, 交叉熵极速下降, 而在 100 次迭代后, 交叉熵基本稳定在 0.05 左右。同样从图 7-22 中可以发现, 在 0 到 200 次迭代间准确率快速上升到 0.980, 之后的 800 此迭代准确率基本稳定在 0.98 和 0.99 之间, 达到了一个很高的准确率。这一结果代表在前 200 次迭代中, 模型在迅速学习, 不断改变各层的权重和偏置变量, 并在 200 次迭代后的准确率基本稳定在了一个较高的水平, 生成了一个预测结果较为精准的模型。

7.3.2 权重集与偏置集

这一部分主要观察每一层的权重集与偏置集中的权重值与偏置值的变化。因为篇幅的原因, 这里主要介绍第一层卷积层权重集与偏置集的折线变化曲线和直方图。

图 7-23 中的四个折线图分别表示第一层卷积层权重集中权重的最大值, 平均值, 最小值, 以及标准差的变化的折线图从图 7-19 中我们可以看出权重集中最大权重值基本呈上升趋势, 而最小值逐渐成下降趋势, 平均值逐渐下降, 而方差逐渐上升接近于 0.1。从中我们看到权重集优化的全过程。

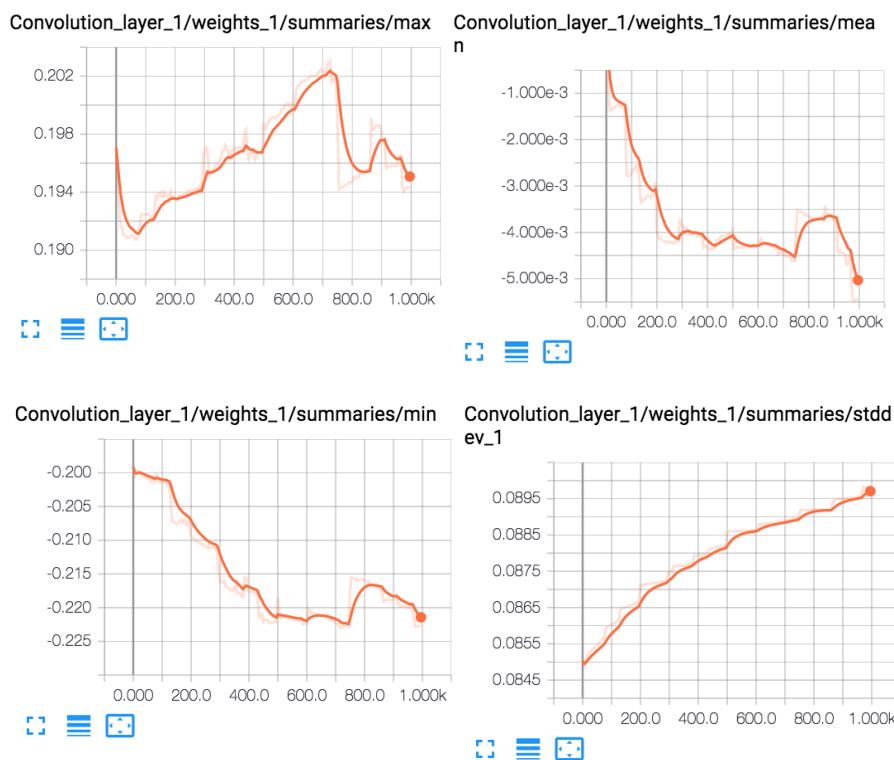


图 7-23 权重集变化折线图

从直方图 7-24 中我们能观察到偏置集中偏置值与权重集中权重值的分布情况。其中 x 轴表示权重值，y 轴表示该权重值出现的次数，z 轴表示训练的迭代次数。我们可以观察到权重值的最大值基本稳定在 0.2 左右，最小值基本稳定在 -0.2 左右。虽然权重值一直在变化，但是权重值和出现次数始终成正太分布，正如一开始从截断正态分布中随机选取值来初始化权重集。而对于偏置集，一开始都设定为常数 0.1，但随着迭代次数的增加，所取值基本均匀地分布在 0.07 到 0.12 之间，发生了很大的变化。

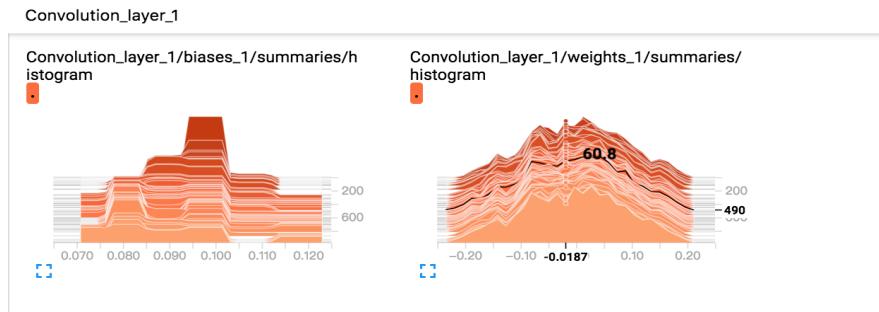


图 7-24 权重与偏差集直方图

7.4 字符识别

图 7-25 所示为对 7-9 中绿色车牌中字符的识别结果，其中每一行显示的汉字、字母和数字为系统识别的最终结果，逗号后的数字表示对该预测结果的可信率。可以看出，其中第 6 个字符“8”被错误地识别成“D”且可信率只有 73%。而其他 7 个字符均被正确地识别出来，且可信率都为 90% 以上，并有五个字符的识别可信率达到 100%。

```
05-14 19:44:47.127 10492-10828/com.example.mayliu.opencvplate I/System.out: nullTensorFlowProvince: 沪, 1.000000
05-14 19:44:47.127 10492-10828/com.example.mayliu.opencvplate I/System.out: TensorFlowLetter: A, 0.915650
05-14 19:44:47.127 10492-10828/com.example.mayliu.opencvplate I/System.out: TensorFlowDigits: F, 1.000000
05-14 19:44:47.127 10492-10828/com.example.mayliu.opencvplate I/System.out: TensorFlowDigits: F, 1.000000
05-14 19:44:47.127 10492-10828/com.example.mayliu.opencvplate I/System.out: TensorFlowDigits: 9, 1.000000
05-14 19:44:47.127 10492-10828/com.example.mayliu.opencvplate I/System.out: TensorFlowDigits: D, 0.730685
05-14 19:44:47.127 10492-10828/com.example.mayliu.opencvplate I/System.out: TensorFlowDigits: 7, 1.000000
05-14 19:44:47.127 10492-10828/com.example.mayliu.opencvplate I/System.out: TensorFlowDigits: 1, 0.999999
05-14 19:44:47.127 10492-10828/com.example.mayliu.opencvplate I/System.out: 沪AFF9D71
```

图 7-25 识别结果 1

图 7-26 所示为对 7-10 中蓝色车牌中字符的识别结果，可以发现所有字符全部识别正确，且所有字符的可信率均达到 100%，这是一种最理想的识别结果。

```
05-14 18:39:51.867 7310-9655/com.example.mayliu.opencvplate I/System.out: nullTensorFlowProvince: 沪, 1.000000
05-14 18:39:51.867 7310-9655/com.example.mayliu.opencvplate I/System.out: TensorFlowLetter: D, 1.000000
05-14 18:39:51.867 7310-9655/com.example.mayliu.opencvplate I/System.out: TensorFlowDigits: 2, 1.000000
05-14 18:39:51.867 7310-9655/com.example.mayliu.opencvplate I/System.out: TensorFlowDigits: 5, 1.000000
05-14 18:39:51.867 7310-9655/com.example.mayliu.opencvplate I/System.out: TensorFlowDigits: 2, 1.000000
05-14 18:39:51.867 7310-9655/com.example.mayliu.opencvplate I/System.out: TensorFlowDigits: 4, 1.000000
05-14 18:39:51.867 7310-9655/com.example.mayliu.opencvplate I/System.out: TensorFlowDigits: 3, 1.000000
05-14 18:39:51.867 7310-9655/com.example.mayliu.opencvplate I/System.out: 沪D25243
```

图 7-26 识别结果 2

7.5 软件页面展示

当用户开启该软件时，会进入如图 7-27 所示的拍照页面，当点击拍照按钮后，会进入如图 7-28 所示的确定页面。



图 7-27 拍照页面

图 7-28 确认页面

当用户点击确认后，将进入到如 7-29 或 7-30 所示的识别页面，这个页面展示了用户所拍摄的照片。且对于 7-29，由于牌照为绿色，所以点击识别绿牌进行识别。而对于 7-30 点击识别蓝牌按钮对图中蓝色牌照进行识别。最终识别结果如图 7-31 和 7-32 所示。



图 7-29 识别页面 1



图 7-30 识别页面 2



图 7-31 识别页面 1



图 7-32 识别页面 2

之后用户向右滑动页面，将看到如图 7-33 所示的显示页面，在这个页面将以列表的形式来展示用户拍摄的照片以及识别的结果。当用户再向右滑动时，将显示如图 7-34 所示的拍照页面，用户点击相机图片，将会进入到拍照界面再次拍照。



图 7-33 展示页面

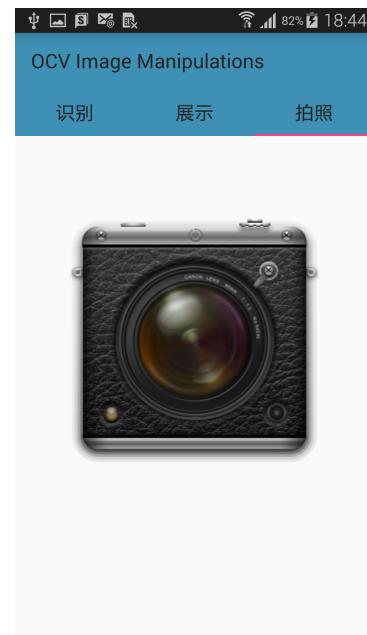


图 7-34 拍照页面

7.6 本章小结

本章主要呈现并分析了第三至六章操做的结果。首先通过图片分别展示了原图片经过车牌检测和提取，字符分割，字符识别等操作后的结果，并通过折线图展示了神经网络模型的训练结果与过程。在章末部分展示了整个软件的页面布局和使用方法。