

本科生实训报告

实训单位：中软国际

组长：韩佳迅

组员：彭钰婷

组员：林语盈

组员：苑伟锋

专业：计算机科学与技术

年级：2020 级

2022 年 7 月 23 日

目录

一、 团队介绍	1
二、 系统需求分析	1
(一) 市场需求	1
(二) 发展趋势	1
三、 系统概要设计	1
四、 系统详细设计	2
(一) 车牌定位	2
1. 二值化	2
2. 闭操作	2
3. 矩形矫正	2
(二) 字符分割	3
1. 基于像素直方图的字符分割	3
2. 基于边缘特征的字符分割	4
(三) 字符识别	4
1. 模板匹配	4
2. 基础网络	4
3. 注意力机制	6
(四) 网页模块	8
五、 系统实现	9
(一) 车牌定位	9
(二) 字符分割	9
(三) 字符识别	10
(四) 网页模块	10
六、 系统测试	11
(一) 引言	11
(二) 测试需求分析	11
(三) 计划测试过程	12
七、 系统安装手册	12
八、 项目总结报告	12
(一) 个人总结	12
(二) 团队总结	12

一、 团队介绍

组长：韩佳迅

组员：彭钰婷、林语盈、苑伟锋

分工：林语盈主要负责 web 端的开发搭建，可视化用户界面；彭钰婷主要负责预处理和车牌定位，即从包含有车牌的图片中定位出车牌区域位置；韩佳迅主要负责字符分割和模板匹配，即从车牌区域中分割出单独的字符，并将字符与预先设置好的模板进行匹配识别；苑伟峰主要负责搭建 Resnet 神经网络进行字符识别，即从分割后的字符图像（中文、英文、数字）中识别出字符值。除此之外，每人都了解并掌握项目开发的所有流程。

二、 系统需求分析

（一） 市场需求

车牌识别系统技术为什么越来越受到人们的欢迎。首先，它可以通过优越的管理技术对停车场系统进行管理，还有对车辆的防盗防爆也能进行监控，可以指导交通汽车流量，对于高速道路超速可以智能化监管，各种缴费功能全部具备。车辆的牌照号码是身份的唯一标记，开车进入停车场停车必须要进行登记车牌号，还要进行缴纳车辆停车费用。管理者引入车牌识别系统系统，就可以准确的对进来出去的汽车牌号进行车牌识别系统，从而设定其他权限，车牌识别系统系统的最大优点是投入成本低，管理高效，还具有安全可靠。

当前，车牌识别技术已经广泛应用于停车管理、称重系统、静态交通车辆管理、公路治超、公路稽查、车辆调度、车辆检测等各种场合，对于维护交通安全和城市治安，防止交通堵塞，实现交通自动化管理有着现实的意义。

（二） 发展趋势

现如今车牌识别技术在应用方面已开始由交通领域走向了非交通领域，比如 4S 店、汽修店、汽车美容店、加油站、地磅、充电桩、工地等领域，这些复杂场景的识别特点、需要集成的应用功能与交通场景存在极大的不同，所以现有的许多适用于动态交通或者静态交通的车牌识别产品，在这些更加细分的复杂场景中使用存在诸多痛点。目前针对智慧车服、智慧工地、智慧加油站、充电桩、地磅等细分场景都已经出现了各个复杂场景专用的车牌识别一体机，针对场景功能需求设计了许多具备亮点的特色功能，如智慧加油站的防爆车牌识别一体机、充电桩防占位车牌识别一体机等。

可以预见，未来越来越多的细分场景都必须使用场景专用智能车牌识别一体机，才能让车牌识别技术为产业革命带来更多的发展方向，帮助企业实现大数据管理，从而完成行业改革进步。

三、 系统概要设计

要实现一个好的车牌检测系统，大致需要如下几个步骤，包括**车牌定位—字符分割—字符识别**；我们在此基础上引入 web 端，包括**用户端输入图片、模型处理、将结果可视化输出到用户端**。下面简要叙述设计思路：

首先，从用户的输入端获得车辆照片；然后对图像做初步的处理：降噪、边缘增强、调整对比度等，为车牌定位做准备。后续使用 HSV 定位技术将车牌从原始图像中提取出来。接下来对字符按照规则进行分割，然后对分割所得的每个字符进行分类。在字符识别过程中，我们采用了模板匹配和神经网络两种方法。在神经网络方向，我们基于 Tensorflow2 实现了基础网络架构

ResNet18, 并在此基础上添加了注意力机制, 分别将 Coordinate Attention 和 Spatial Attention 用于英文和中文的图像分类。最后, web 端输出车牌号码, 反馈给用户车牌识别的结果。

四、系统详细设计

(一) 车牌定位

使用 HSV 定位车牌位置, 主要有二值化、闭操作、矩形矫正三个步骤, 下面逐一介绍:

1. 二值化

首先设置 HSV 中蓝色分量范围值, 最小为 100, 最大为 140。然后逐像素点进行遍历, 实现如下:

```
1 rows, cols = h_split.shape
2 for row in np.arange(rows):
3     for col in np.arange(cols):
4         H = h_split[row, col]
5         S = s_split[row, col]
6         V = v_split[row, col]
7         # if (在蓝色值域区间, 且满足 S 和 V 的一定条件):
8         # 颜色置为 255
```

这样就完成了二值化操作, 结果表现为蓝色背景的区域被置为 255, 其余区域被置为 0。

2. 闭操作

接下来执行闭操作, 其主要流程是先膨胀后腐蚀, 作用是填充物体内部细小空洞、连接邻近物体、平滑边界的同时并不明显改变面积。执行闭操作的流程如下:

```
1 kernel = cv.getStructuringElement(cv.MORPH_RECT, (10, 3))
2 morphology_image = cv.morphologyEx(binary_image, cv.MORPH_CLOSE, kernel)
```

闭操作执行后, 相邻区域连成一片, 图像噪声减少。

3. 矩形矫正

矩形矫正需要用到 util.py 文件中的函数。对于预选的所有图片, 首先通过旋转矫正倾斜的车牌区域, 然后将车牌图片调整成统一尺寸。下面详细介绍这两个函数:

旋转矫正 旋转矫正过程需要如下步骤:

1. 获取该等值线框对应的外接正交矩形 (长和宽分别与水平和竖直方向平行);
2. 判断矩形相对最低角点而言在哪一象限。如果宽度比高度小, 说明矩形相对于最低角点而言, 在第二象限; 否则相对于最低角点而言在第一象限;
3. 创建一个放大的图像, 以便存放之前图像旋转后的结果;

4. 将旋转前的图像拷贝到放大图像的中心位置；
5. 计算旋转中心；
6. 获取执行旋转所需的变换矩阵；
7. 执行旋转；
8. 截取与最初等值线框长、宽相同的部分。

需要注意的是，在获取执行旋转所需的变换矩阵时，需要调整角度为 $angle + 180$ ，这样旋转的结果才是正确的。

统一尺寸 设置车牌区域标准高度、宽度后，使用 cv2 库的 `resize` 函数即可调整尺寸，方便后续程序设计。

（二） 字符分割

主流的字符分割有两种方向，一是基于连通域（边缘特征）的字符分割；二是基于像素直方图的字符分割。

然而，就本项目而言，若单独使用基于连通域（边缘特征）的方法，易导致车牌中的部分噪声被错误分割；若单独使用基于像素直方图的方法，则分割的字符图片会出现边缘留白，并非贴合轮廓划分字符，这会导致模板匹配和神经网络分类不准确。因此，我们使用两种方法结合，取长补短。

首先，我们使用**基于像素直方图**的方法：对图片进行二值化处理，统计水平方向和竖直方向上各行各列的黑色像素的个数，根据像素的特点确定分割位置，完成字符分割。

其次，我们对上述分割后的字符图片进行**基于连通域（边缘特征）**的字符分割：通过形态学处理使得字符成为一个整体（主要针对汉字），通过边缘检测获取每一个字符的轮廓，即可实现字符分割。

1. 基于像素直方图的字符分割

1. 对车牌图像预处理：首先，将车牌大小进行 `resize` 成固定大小，然后进行高斯去噪和二值化操作，将字符与背景区分成黑白二值。

2. 分别设置两个列表 `white` 和 `black` 按列进行黑白像素点的统计，并记录所有列的黑白像素最大值 `black_max`、`white_max`。

3. 根据黑白像素最大值确定二值化后的车牌是黑底白字还是白底黑字。

4. 按列对二值化车牌图像的像素点进行遍历，先根据黑白像素的分布找到字符的左右轮廓，并进行截断，再继续按像素分布寻找下一个字符的轮廓。这部分代码如下：

```

1  def find_end(start):
2      end = start + 1
3      for m in range(start + 1, width - 1):
4          if (black[m] if arg else white[m]) > ( 0.95 * black_max if arg else 0.95 * white_max):
5              end = m
6              break
7      return end
8

```

```

9   n = 1
10  start = 1
11  end = 2
12  word = []
13  while n < width - 2:
14      n += 1
15      if (white[n] if arg else black[n]) > (0.05 * white_max if arg else 0.05 * black_max):
16          start = n
17          end = find_end(start)
18          n = end
19          if end - start > 5:
20              word_i = image[1:height, start:end]
21              word_i = cv.resize(word_i, (15, 30))
22              word.append(word_i)

```

2. 基于边缘特征的字符分割

在基于像素图的分割时，我们只对左右轮廓根据像素分布进行了截断，这其中还包含一些噪声和边缘留白部分。因此，我们对上述分割后的图像一一进行轮廓提取。

首先，我们对字符图片进行 `resize` 成固定大小，然后再进行下面的操作：

1. 处理汉字：由于大部分汉字存在偏旁部首，而并非连通的一体（如“京”、“津”、“苏”等）。为了防止在轮廓处理时将汉字拆解成不同部分，我们先对汉字图像进行膨胀操作，使之膨胀为近似整体，再进行轮廓查找。剩余部分与下面对英文和数字的处理相同。
2. 处理英文和数字：使用 `cv2.findContours` 查找字符图像的轮廓。然后逐一筛选这些轮廓，计算其长宽高。若轮廓形状符合字符形状，则将其纳入最后的结果；否则舍弃。

(三) 字符识别

1. 模板匹配

1. 准备模板：设置字符列表，存放可能出现的中文、英文和数字字符。然后，按照列表索引来对应读取数据集下的模板图片。
2. 预处理：对读取到的模板图像二值化，并将待测图像进行 `resize`，使其与模板大小一致。
3. 对分割后的字符与模板逐一匹配：将分割后的字符分为三类，一是汉字（对应于第一个字符），二是英文（对应于第二个字符），三是英文或数字（对应其余字符）。使用 `cv.matchTemplate` 分别计算字符图像和模板的匹配得分，并取最高得分为匹配结果。

2. 基础网络

我们选择的基础网络是 ResNet18，让我们聚焦于神经网络局部：假设我们的原始输入为 \mathbf{x} ，而希望学出的理想映射为 $f(\mathbf{x})$ （作为上方激活函数的输入）。左图虚线框中的部分需要直接拟合出该映射 $f(\mathbf{x})$ ，而右图虚线框中的部分则需要拟合出残差映射 $f(\mathbf{x}) - \mathbf{x}$ 。残差映射在现实中往往更容易优化。以本节开头提到的恒等映射作为我们希望学出的理想映射 $f(\mathbf{x})$ ，我们只需将右图虚线框内上方的加权运算（如仿射）的权重和偏置参数设成 0，那么 $f(\mathbf{x})$ 即为恒等映射。实际中，当理想映射 $f(\mathbf{x})$ 极接近于恒等映射时，残差映射也易于捕捉恒等映射的细微波动。右图是

ResNet 的基础架构-残差块 (residual block)。在残差块中, 输入可通过跨层数据线路更快地向前传播。

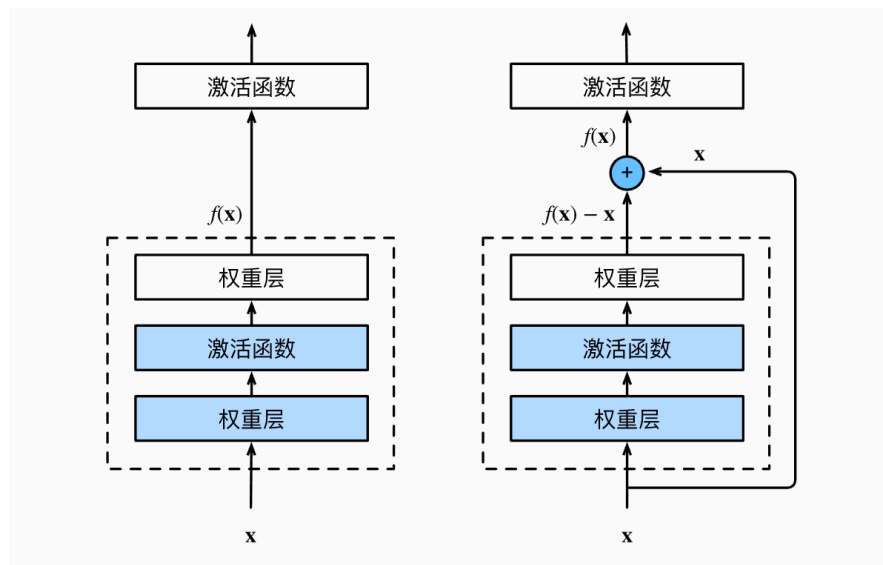


图 1: 正常块 (左) 残差块 (右) https://zh.d2l.ai/chapter_convolutional-modern/resnet.html

ResNet 沿用了 VGG 完整的 3×3 卷积层设计。残差块里首先有 2 个有相同输出通道数的 3×3 卷积层。每个卷积层后接一个批量规范化层和 ReLU 激活函数。然后通过跨层数据通路, 跳过这 2 个卷积运算, 将输入直接加在最后的 ReLU 激活函数前。这样的设计要求 2 个卷积层的输出与输入形状一样, 从而使它们可以相加。如果想改变通道数, 就需要引入一个额外的 1×1 卷积层来将输入变换成需要的形状后再做相加运算。本次实验中我们使用的基础网络是 ResNet18。

网络中共有 4 个模块, 每个模块有 4 个卷积层 (不包括恒等映射的 1×1 卷积层)。加上第一个 7×7 卷积层和最后一个全连接层, 共有 18 层。因此, 这种模型通常被称为 ResNet-18。

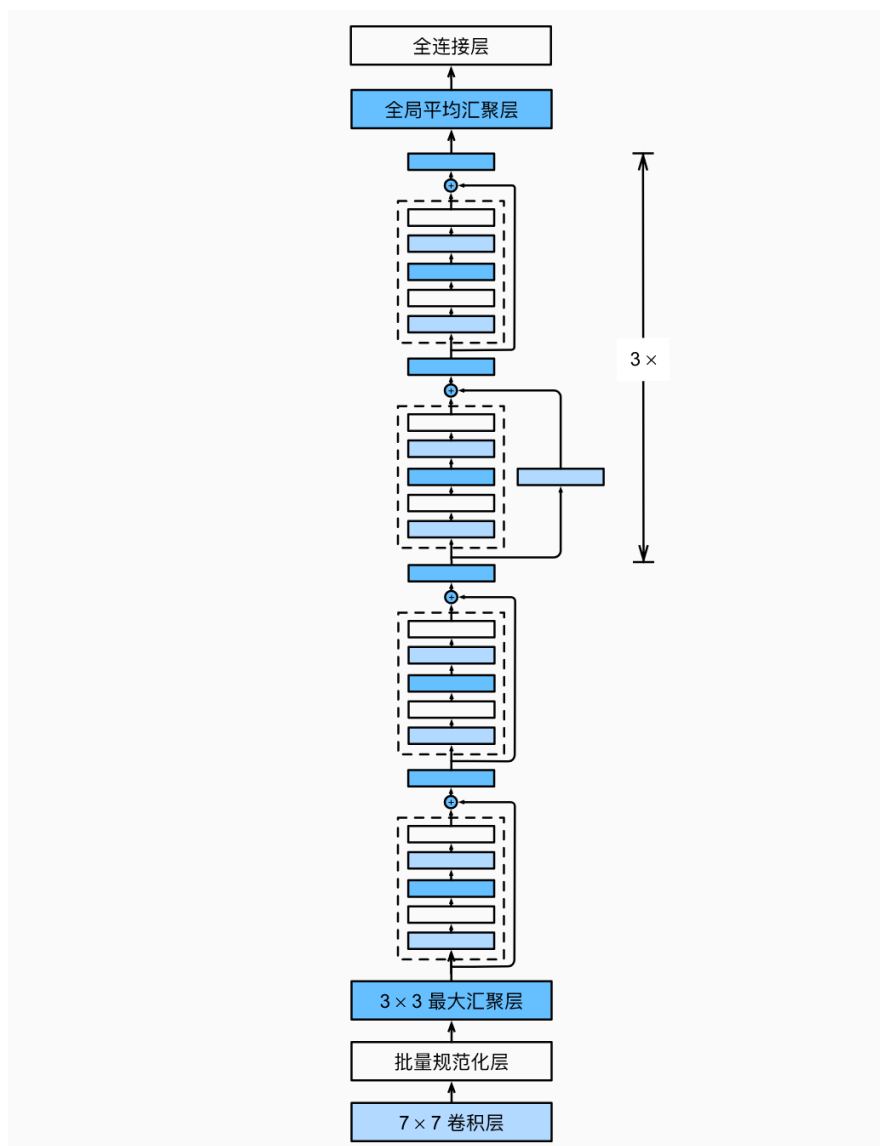


图 2: 网络结构示意图 https://zh.d2l.ai/chapter_convolutional-modern/resnet.html

3. 注意力机制

视觉注意力机制是人类视觉所特有的一种大脑信号处理机制，而深度学习中的注意力机制正是借鉴了人类视觉的注意力思维方式。一般来说，人类在观察外界环境时会迅速的扫描全景，然后根据大脑信号的处理快速的锁定重点关注的目标区域，最终形成注意力焦点。该机制可以帮助人类在有限的资源下，从大量无关背景区域中筛选出具有重要价值信息的目标区域，帮助人类更加高效的处理视觉信息。

注意力机制是机器学习中的一种数据处理方法，广泛应用在自然语言处理（NLP）、图像处理（CV）及语音识别等各种不同类型的机器学习任务中。根据注意力机制应用于域的不同，即注意力权重施加的方式和位置不同，将注意力机制分为空间域、通道域和混合域三种。空间域：将图片中的空间域信息做对应的变换，从而将关键得信息提取出来。对空间进行掩码的生成，进行打分，代表是 Spatial Attention Module [1]。通道域：类似于给每个通道一个权重，来代表该通道与关键信息的相关程度。对通道生成掩码，进行打分，代表是 Channel Attention Module、SENet [1]。混合域：即空间域和通道域的融合，代表是 BAM, CBAM [1]。

本次实验中我们选取了效果较好的 Coordinate Attention 和 Spatial Attention 加入到中文和英文的分类模型当中。

英文模型：Coordinate Attention

由于本次实验中字符分割得到的图片是单通道的，所以典型的通道注意力如 SENet [2] 等可能不起作用，所以我们更多地寻找适用于空间注意力的模型。因此针对英文模型，由于训练集图片宽高比为 1，我们使用 Coordinate Attention [3]。具体操作如下3：取输入张量 $X \in \mathbb{R}^{C \times H \times W}$ ，并在两个空间维 H 和 W 上应用平均池化，得到两个张量 $X' \in \mathbb{R}^{C \times H \times 1}$ and $X'' \in \mathbb{R}^{C \times 1 \times W}$ ，然后将这两个张量连接成 $X''' \in \mathbb{R}^{C \times 1 \times (H+W)}$ ，这两个张量随后通过 2D 卷积传递。它根据指定的还原比 r 减少从 C 到 $\frac{C}{r}$ 通道。然后是一个 normalization 层，然后是一个激活函数。最后，将张量分解为 $\hat{X} \in \mathbb{R}^{\frac{C}{r} \times 1 \times W}$ and $\tilde{X} \in \mathbb{R}^{\frac{C}{r} \times H \times 1}$ 。这两个张量分别通过两个 2D 卷积核，每个核都从 $\frac{C}{r}$ 增加到到 C 个通道，最后对得到的两个张量进行 sigmoid 激活。然后，将 attention maps 按元素顺序与原始输入张量 x 相乘。

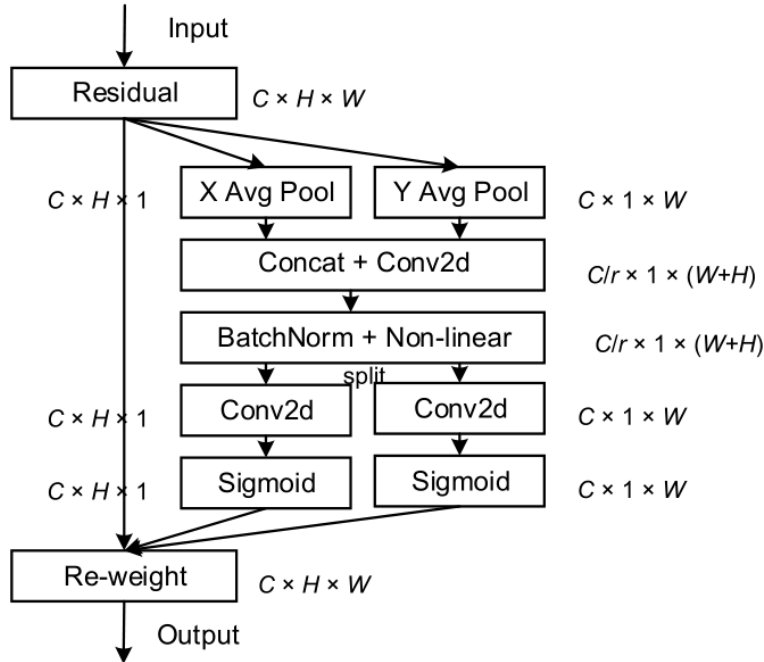


图 3: Hou Qibin, Daquan Zhou, and Jiashi Feng. "Coordinate attention for efficient mobile network design." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.

中文模型：Spatial Attention

由于训练集中文图片高宽比不是 1，所以大多由卷积组成的注意力机制应用起来比较困难，此处我们使用主要由池化操作组成的 Spatial Attention [4]。主要操作如下4：将输入的特征图 $F (H \times W \times C)$ 分别经过基于 width 和 height 的 global max pooling (全局最大池化) 和 global average pooling (全局平均池化)，得到两个 $1 \times 1 \times C$ 的特征图，接着，再将它们分别送入一个两层的神经网络 (MLP)，第一层神经元个数为 C/r (r 为减少率)，激活函数为 Relu，第二层神经元个数为 C ，这个两层的神经网络是共享的。而后，将 MLP 输出的特征进行基于 element-wise 的加和操作，再经过 sigmoid 激活操作，生成最终的 channel attention feature，即 M_c 。最后，将 M_c 和输入特征图 F 做 element-wise 乘法操作，生成 Spatial attention 模块需要的输入特征。

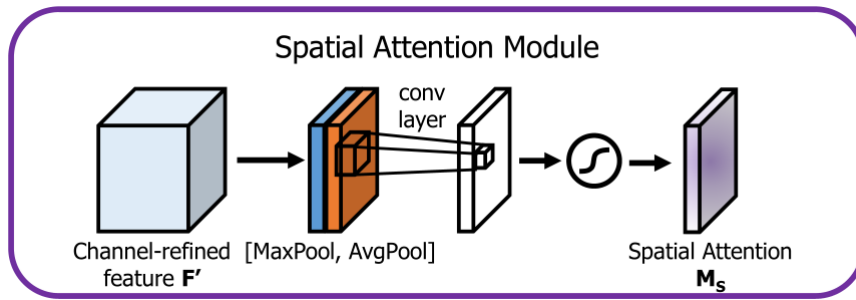


图 4: Woo, Sanghyun, et al. "Cbam: Convolutional block attention module." Proceedings of the European conference on computer vision (ECCV). 2018.

(四) 网页模块

对于网页展示模块，首先应实现文件上传功能，并实现图片展示功能。具体实现过程为，使用 file 控件选择文件，通过 ajax 技术和服务器端交互，单击预测按钮后，调用 test() 函数 (如下所示) 对其进行处理。将图片提交到服务器后，服务器端调用模型并对图片进行处理，并返回车牌识别结果。

网页前端为 html 文件，指定网页布局。使用 css 对网页进行进一步装饰。使用 javascript 控制前端表示。

使用 python flask 绑定路由并提供服务。

```

1  function test() {
2      var fileobj = $("#file0")[0].files[0];
3      console.log(fileobj);
4      var form = new FormData();
5      form.append("file", fileobj);
6      var out='';
7      var flower='';
8      $.ajax({
9          type: 'POST',
10         url: "predict",
11         data: form,
12         async: false,           //同步执行
13         processData: false, // 告诉 jquery 要传输 data 对象
14         contentType: false, //告诉 jquery 不需要增加请求头对于 contentType 的设置
15         success: function (arg) {
16             console.log(arg)
17             out = arg.result;
18         },error:function(){
19             console.log(" 后台处理错误");
20         }
21     });
22
23     out.forEach(e=>{

```

```

24         flower+=`<div style="border-bottom: 1px solid #CCCCCC;line-height: 60px;font-size:16px;
25         });
26
27         document.getElementById("out").innerHTML=flower;
28
29     }

```

main.py 中的处理如下所示。

```

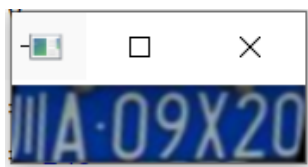
1  @app.route("/predict", methods=["POST"])
2  def predict():
3      image_path = "./images/"+request.files["file"].filename
4      print(image_path)
5      #request.files["file"].save(image_path)
6      info = get_prediction(image_path)
7      return jsonify(info)

```

五、 系统实现

(一) 车牌定位

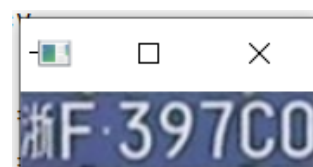
对 20220712-Src-opencv_plate_locate-images 中的 plate1-plate6 进行定位, 部分结果如图6所示。



(a) plate1 定位结果



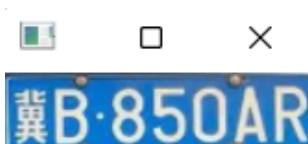
(b) plate2 定位结果



(c) plate6 定位结果

图 5: 车牌定位效果

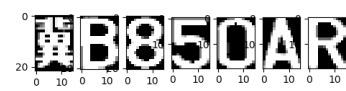
(二) 字符分割



(a) 车牌图像



(b) 二值化图像



(c) 分割后图像

图 6: 字符分割效果

(三) 字符识别

我们基于 TensorFlow2 实现了基础网络架构 ResNet18, 并在此基础上添加了 Coordinate Attention 和 Spatial Attention 分别用于英文和中国的图像分类。我们使用 Adam 作为优化器, 指定 CategoricalCrossentropy 为损失函数, 使用 categorical_accuracy 为分类指标, 调整 batch_size 的大小为 32, 经过 30 轮训练, 我们在英文和中文的验证集上的分类正确率均超过了 99%。

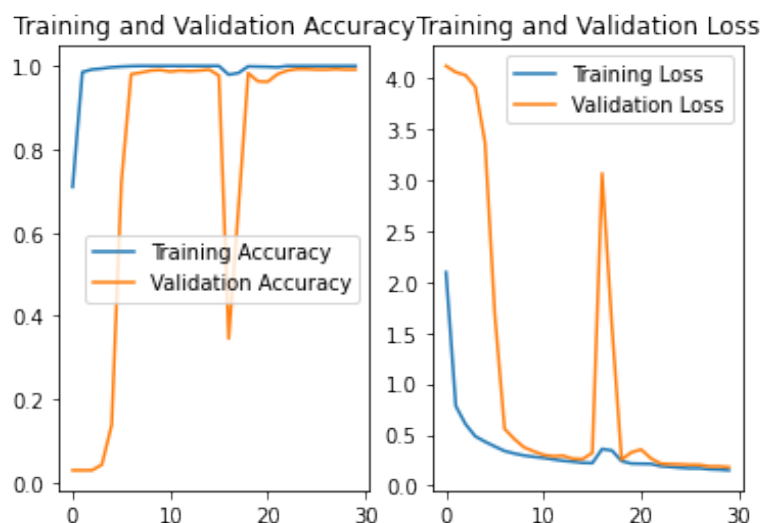


图 7: 英文模型 loss 与 acc 曲线

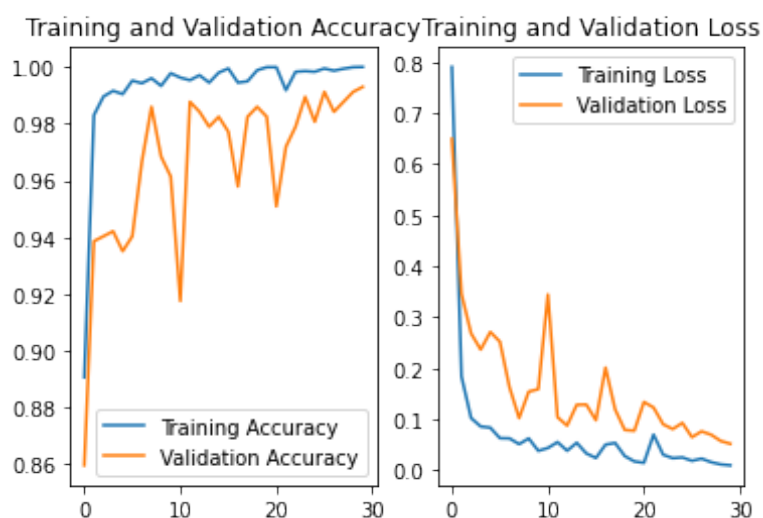


图 8: 中文模型 loss 与 acc 曲线

(四) 网页模块

网页前端为 html 文件, 指定了网页布局, 使用 javascript 与控制前端表示, 使用 flask 绑定路由并提供服务。实现了上传文件, 显示图片, 显示车牌识别结果的功能。

对于上述功能, 实现了两个版本的界面, 如图9和图10所示。

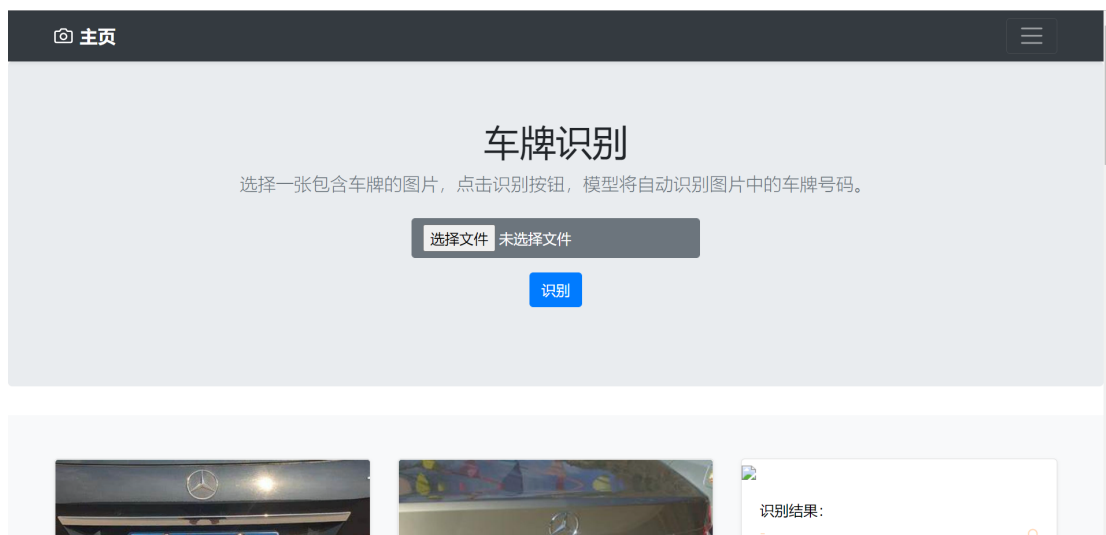


图 9: web 界面 1

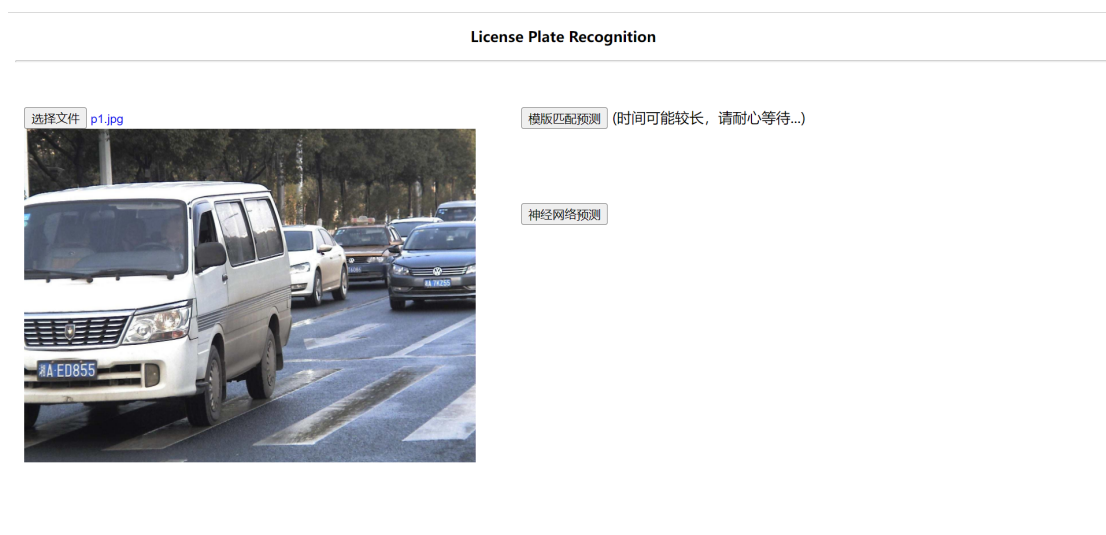


图 10: web 界面 2

六、 系统测试

(一) 引言

对整个项目进行了黑盒测试和联调，在完成网页功能的基础上，对一些特殊情况进行处理。

(二) 测试需求分析

首先，用户的需求是，输入一张从文件中上传的图片，通过模型输出车牌结果。具体的需求即显示图片、显示文字，并连接好后端的模型。第二，对于概要设计，总体项目包括网页前端的 HTML 文件，后端与模型的连接及模型端车牌定位、字符分割、字符识别的具体算法实现。第三，对于详细设计，均使用 Python 进行实现，前端使用 HTML 和 JavaScript 编写代码，后端

使用 Python flask 进行连接，模型端在字符分割和车牌定位部分使用 open CV 进行基于规则的图像处理，在字符识别部分使用 TensorFlow 进行实现深度网络的实现，具体如上一节所示。

（三） 计划测试过程

针对以上需求，首先，进行单元测试。对 web 单元、图像处理单元、字符分割单元、模板匹配字符识别单元、深度学习字符识别单元单独进行测试，保证每一部分的输入输出结果正确。接下来，进行集成测试，将各个单元连接起来并调试实现上述功能。最后，进行系统测试，对于整体进行优化，并处理一些异常情况，完成项目。

七、 系统安装手册

使用 python3.8.10
运行指令安装依赖环境

```
1 pip install -r requirements.txt
```

切换到项目主路径输入

```
1 python main.py
```

八、 项目总结报告

（一） 个人总结

林语盈：通过这个项目，我了解了 web 开发的基本流程和相关知识，复习了一些图像处理方法和机器学习算法，也对整个项目周期和整个实现过程有了更深入的了解。

关于 web 开发。在此项目之前，我们小组成员都不了解 web 的相关知识，也没有进行过类似的开发工作。因为我已经对机器视觉和相关深度学习算法较熟悉，于是想尝试学习一下新的技术。在学习和项目实现过程中，也遇到了一些困难，在较短的时间内尽量进行解决，最终基本实现了预期功能，但整体的项目架构还有待进一步改善。

关于模型。实际上，更好的模型方法即，车牌检测和车牌识别分别均使用深度学习方法，并尽可能统一数据集进行联调。在车牌检测部分可以使用 yolo 等模型，在车牌识别部分可以使用 LPRNet 等模型，效果均优于我们的方法，数据集可以使用 CCPD。对于目前的国内车牌识别而言，使用参数量类似的深度神经网络是完全足以表征的，无论是否增加字符分割这一步骤，效果都远好于基于规则的方法。但在项目实际实现过程中，由于设备原因与时间所限，我们的项目成员创新性的使用了自己设计的增加注意力机制的 resnet 实现字符识别，使用基于图像处理和基于规则的方法进行图像预处理和车牌定位和分割，故未使用效果更好的方法，这也是未来进一步的改进方向。

关于工程项目。与科研不同，工程项目更关注于整个项目的完整性、实用性和团队分工与协作，这是我体会最深的一点，这次实训过程让我收获颇丰。

（二） 团队总结

通过本次项目，团队成员均学习并掌握了项目开发流程。本次实训对于本科生的项目经验培养具有很大帮助。在此真诚感谢学校提供的实训机会，感谢中软国际的各位老师悉心教导，感谢各位队友的团结协作！

参考文献

- [1] Meng-Hao Guo, Tian-Xing Xu, Jiang-Jiang Liu, Zheng-Ning Liu, Peng-Tao Jiang, Tai-Jiang Mu, Song-Hai Zhang, Ralph R Martin, Ming-Ming Cheng, and Shi-Min Hu. Attention mechanisms in computer vision: A survey. *Computational Visual Media*, pages 1–38, 2022.
- [2] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.
- [3] Qibin Hou, Daquan Zhou, and Jiashi Feng. Coordinate attention for efficient mobile network design. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13713–13722, 2021.
- [4] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.