第 5 章

车辆识别

车辆识别是智能交通系统的重要组成部分，也是交通管控、无人驾驶、疑犯追踪、行为分析等其他智能任务的基础。通过对马路摄像头抓拍的车辆图像进行数字图像处理与分析，确定目标车辆属于哪一种类别，并且确定车辆的颜色及车牌号。车辆识别目前已应用于很多领域，如拍照识车、违章停车监测、智能卡口、路况分析、智能定损以及疑犯追踪等[1]。

5.1 车辆识别介绍

目前，全国高速公路收费方式以人工收费为主，以 ETC 不停车收费为辅，另外还有部分移动支付（扫码）收费方式专用通道等。传统人工收费以及移动支付这些方式都有其缺点，例如容易发生车辆堵塞，费用计费误差、路径识别偏差，而且收费车道数量较多，收费广场占地面积大，成本较高[2]。

近年来，国家逐步出台各项政策迅速推进以 ETC 收费为主的模式部署，高速公路全面采用 ETC 自动收费系统的趋势越来越明显（见图 5-1）。例如 2018 年 5 月，交通运输部新闻发言人吴春耕表示，推动取消高速公路省界收费站，明确以 ETC 5.8G 为主流技术，实现高速公路不停车收费。2019 年两会期间政府工作报告表示，两年内基本取消全国高速公路省界收费站，实现不停车快捷收费，减少拥堵、便利群众。

但目前 ETC 收费的识别标识主要以计费卡等电子标签形式为主，目前还无法避免电子标签的违规挪用，必须辅以车牌识别的鉴权技术[3]。因此，车牌识别技术在 ETC 大规模推广的过程中，必然需要面临更大的挑战，例如车牌污损、天气光线影响拍照采样不清晰、车辆行驶路径与计费复杂等问题[4]。特别是针对无法避免出现的模糊车牌情况，如何进一步提升识别准确率，并且如何充分利用车牌精准识别技术实现更多应用模式，是目前需要解决的主要需求问题。

图 5-1 车辆通行展示

5.2 车辆识别的过程

人工智能车牌识别模型利用深度学习算法与深度学习一体机基础设施，对模型进行海量真实牌照数据的识别训练，实现对各种角度、光照、污损等条件下的车牌进行高效识别与结构化输出[1]。车牌识别的步骤如下。

5.2.1 前期预处理

针对雾霾、欠爆、噪声问题进行相应的图像增强处理，改善画面质量，如图 5-2 和图 5-3 所示。

图 5-2 前期处理（一）

图 5-3 前期处理（二）

5.2.2　车牌检测

使用深度学习轻量级网络设计思想，构建一个约 30 层的卷积网络，使用 5000 万个样本训练，得到的模型能在保证运行效率的同时，检测率大幅优于传统目标检测算法[5]，如图 5-4 所示。

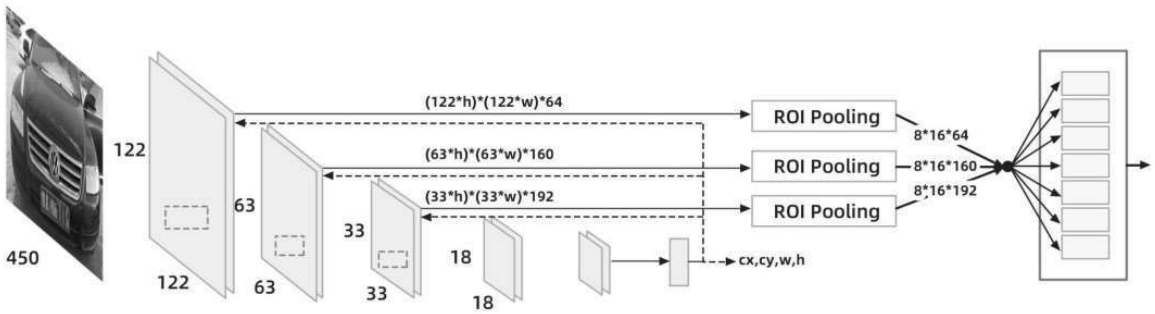

图 5-4　车牌检测（一）

5.2.3　车牌字符识别

使用深度学习轻量级网络设计思想，构建一个 25 层的卷积网络，使用 2 亿个样本训练，模型性能大幅优于 svm 等方法[6]，如图 5-5 和图 5-6 所示。

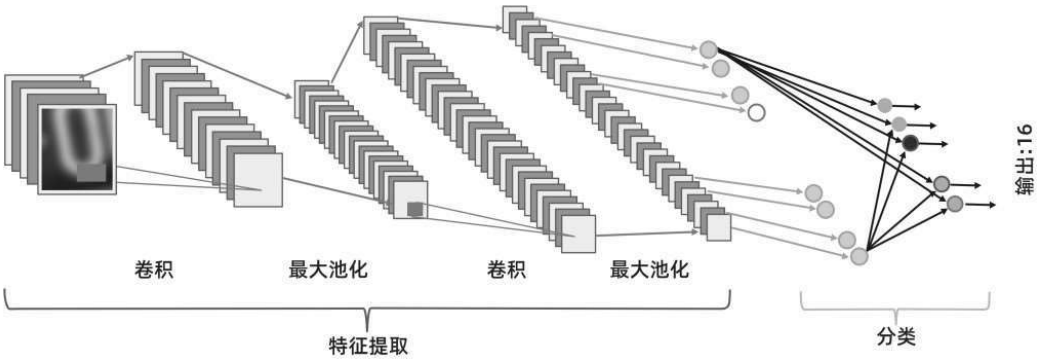

图 5-5　车牌检测（二）

（a）彩色原图

（b）灰度原图

（c）统一大小后的二值图

（d）去噪后的二值图

图 5-6　车牌字符识别

（e）字符区域图

（f）统一大小后的字符区域图

（g）细化并添加边界后的字符居中图

图 5-6　车牌字符识别（续）

5.2.4　车牌种类识别

能够识别的车牌种类有蓝牌、黄牌、双层黄牌、武警、警车、军车、新能源等。涵盖的地区包括大陆 31 个省份、直辖市及自治区，如图 5-7 所示。

图 5-7　车牌种类识别

5.2.5　结构化信息识别功能

基于深度学习的车辆识别算法，用于实现对高速公路监控视频、车辆卡口抓拍图片的二次识别，可精确识别车牌号码、车辆品牌、车辆子品牌、车辆年款、车辆颜色、车牌颜色、车辆类型、车牌类型等车辆细节信息，弥补前端卡口识别信息不足及识别不准的缺陷；支持车辆类型、车牌类型的细分，能够识别同一车型的不同年款，并输出年检标等特征图片，为基于大数据分析的深度应用提供更多有价值的车辆信息[6]。

车辆页面重点展示了视频中车辆所对应的车牌、车型、颜色、品牌、子品牌、时间、设备号和属性[3]，如图 5-8 所示。

图 5-8　车型识别

5.3　人工智能开放平台的前端运用

人工智能开放平台提供了车牌识别和车型识别两部分，车牌识别可以检测出图片中的车牌，并返回车牌边框坐标、车牌号码、车牌颜色等信息[7]。支持各种位置、白天及夜间的车牌识别。车型识别可以检测图片中车辆的具体车型，输出图片中车辆的类型、颜色等信息。

5.3.1　车牌识别接口调用

车牌识别技术通过对公路上摄像头拍摄的照片进行数字图像处理与分析，从复杂的背景中准确地提取出车牌区域，进而达到对汽车牌照的精确定位，并最终完成对车辆牌号、颜色等信息的识别。车牌识别技术在交通控制和监视中占有很重要的地位，具有广泛的应用前景，同时也成为车辆身份识别的主要手段[8]。

1．准备工作

首先，通过 vue-cli3 脚手架创建一个 vue 项目，创建完成后可以看到页面目录，如图 5-9 所示。

图 5-9　vue-cli3 脚手架创建

然后通过 npm run serve 启动服务，即可进行开发。

2．接口参数详解

所有 API 都包含两部分参数：一部分是公共请求头部分，另一部分是接口本身的业务参数。业务参数包括 parameter 参数和 requestbody 参数，其中 requestbody 参数使用 json 格式传输。

（1）HTTP 的 header 参数如表 5-1 所示。

表 5-1　HTTP 的 header 参数

参 数 名 称	类　　型	备　　注
appId		应用 ID
timestamp		时间戳/s
nonce		随机字符串
sign		接口请求签名，待计算

（2）业务参数如表 5-2 所示。

表 5-2　业务参数

参 数 名 称	类　　型	备　　注
img	string	图像数据，base64 编码后进行 urlencode 编码，要求 base64 编码和 urlencode 编码后大小不超过 4 MB。图片的 base64 编码是包含图片头的，如 data:image/jpg;base64，支持的图片格式有 jpg、bmp、png，最短边至少为 50 px，最长边最大为 4096 px

3．前端接口调用实现

1）页面展示

车牌识别页面展示如图 5-10 所示。

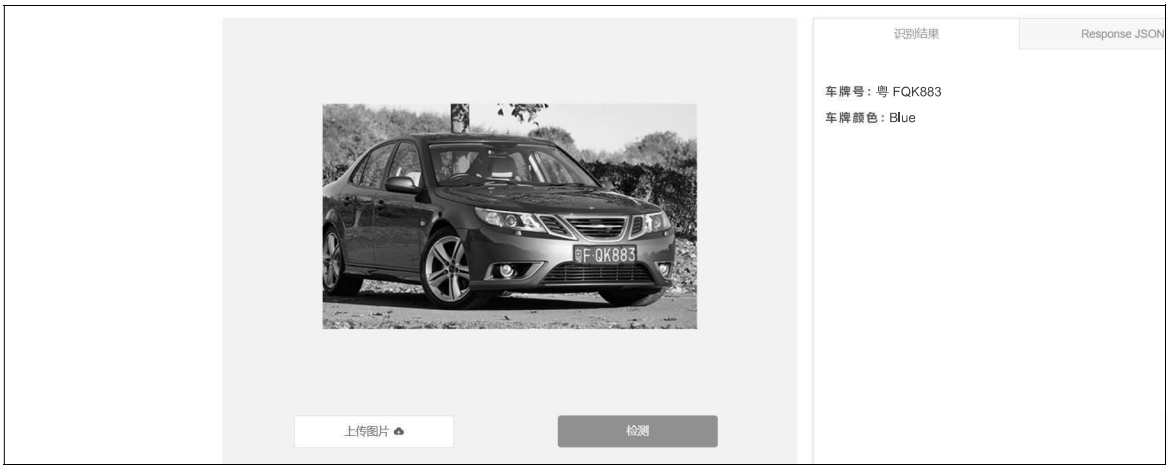

图 5-10　车牌识别页面展示

2）HTML 部分

搭建框架，将页面的大概架构展示出来。

```
<template>
<div class="main">
  <div class="demo_title">
    <h3>功能演示</h3>
    <p>车牌识别技术要求能够将运动中的汽车牌照从复杂背景中提取并识别出来</p>
    <p>通过车牌提取、图像预处理、特征提取、车牌字符识别等技术，识别车辆牌号信息</p>
  </div>
  <div class="demo_body">
    <div class="request">
      <div class="image-container">
        <div class="imageContainerResult">
          <div class="showScanFlag" v-show="showScanFlag"> </div>
          <img :src="request_img" ref="ref_request_img" class="requestImg">
          <canvas :width="canvasWidth" :height="canvasHeight" class="canvasDiv"
ref="imgcanvas"></canvas>
        </div>
      </div>
      <el-row>
        <el-col :span="10" :offset="3">
          <el-upload class="img-upload"
action="false" :show-file-list="false" :auto-upload="false"
ref="upload" :on-change="imgChange">
            <el-button plain>上传图片<i class="el-icon-upload el-icon--right"></i></el-button>
          </el-upload>
        </el-col>
        <el-col :span="10" :offset="1">
          <el-button type="primary" @click.native="detect()">检测</el-button>
        </el-col>
      </el-row>
      <div class="img_menu">
        <ul class="img_menu_ul">
          <li @click="imgMenuClick(item)"
class="img_menu_li" :class="menuId==item.id?'active':''" v-for="(item,index) in
imgMenu" :key="index">
            <img :src="item.src">
          </li>
        </ul>
      </div>
    </div>
    <div class="result">
      <el-tabs class="tabs" type="border-card" :stretch="true">
        <el-tab-pane label="识别结果">
          <div class="result_body1">
            <span class="span_style">车牌号：</span><span>{{plate}}</span><br><br>
            <span class="span_style">车牌颜色：</span><span>{{plate_color}}</span>
          </div>
        </el-tab-pane>
        <el-tab-pane label="Response JSON">
          <div class="result_body2">
```

```
                    <el-input type="textarea" v-model="textarea" :readonly="true"
resize="none" :rows="33">
                    </el-input>
                </div>
            </el-tab-pane>
        </el-tabs>
    </div>
  </div>
</div>
</template>
```

3）js 部分

接下来，需要引用接口地址，引用方法如下（在 script 中引用）。

```
export function plate(params) {
  console.log("canshu", params);
  return request({
    url: 'http://ai.cstor.cn/api/plate',
    method: 'post',
    headers: {
      'Content-Type': 'Application/json',
    },
    data: params
  })
}
```

这里的 plate 就是接口文档中的车辆接口地址，代码如下。

```
export function plate(params) {
  return request({
    url: 'http://ai.cstor.cn/api/plate',
    method: 'post',
    headers: {
      'Content-Type': 'Application/json',
    },
    data: params
  })
}
```

单击图片上传，触发 imgChange 事件。

```
//上传图片
    imgChange(file) {
      this.plate = ''
      this.plate_color = ''
      this.textarea = ''
      let reader = new FileReader()
      let img = event.target.files[0]
      //创建对象
      let img2 = new Image()
      reader.onload = e => {
```

```
            //改变图片的 src
            this.request_img = reader.result
            img2.src = reader.result
        }
        if (img) {
            reader.readAsDataURL(img)
        }
        img2.onload = e => {
            this.present = this.AutoSize(img2, 1000, 1000)
        }
        let type = img.type; //文件的类型，判断是否是图片
        if (this.imgData.accept.indexOf(type) == -1) {
            this.$message({
                message: '上传图片只能是 JPG/PNG/JPEG 格式!',
                offset: 100
            })
            return
        }
        let form = new FormData()
        form.append('file', img, img.name)
        this.imageFile = form
    },
```

进行图片检测，detect 事件。

```
//开始检测
    detect() {
        this.showScanFlag = true
        if (this.imageFile === null) {
            let canvas = this.imagetoCanvas(this.$refs.ref_request_img)
            let blobimg = null
            const _this = this
            canvas.toBlob(function(blob) {
                blobimg = blob
                var reader = new FileReader()
                reader.readAsArrayBuffer(blobimg)
                reader.onloadend = function(e) {
                    //改变图片的 src
                    this.imgurl = reader.result;
                    let fd = new FormData()
                    let the_file = new Blob([e.target.result], {
                        type: "image/jpg"
                    })
                    fd.append("file", the_file, "images.jpg")
                    _this.imageFile = fd
                    setTimeout(() => {
                        _this.http_plate()
                    }, 1000)
                }
            }, "image/jpeg", 0.95)
        } else {
```

```
            setTimeout(() => {
              this.http_plate()
            }, 1000)
          }
},
```

http_plate 接口部分展示如下。

```
http_plate() {
      plate({
        img: this.request_img
      }).then(res => {
        this.showScanFlag = false
        if (res.code === 0) {
          if (res.data === null || res.data.length == 0) {
            this.plate = ''
            this.plate_color = ''
          } else {
            let data = res.data
            data.map((item, index) => {
              this.plate = this.plate + data[index].plate + '    '
              this.plate_color = this.plate_color + data[index].color + '    '
            })
            this.initCanvas(data)
          }
        }
        this.textarea = this.getFormatData(JSON.stringify(res))
      })
    },
```

通过调用 initCanvas 重新调整上传的图片大小以及框选识别出来的车牌信息，代码如下。

```
//图片自动比例缩放
      AutoSize(Img, maxWidth, maxHeight) {
        let preWidth = Img.width
        let proHeight = Img.height
        var image = new Image()
        //原图片原始地址（用于获取原图片的真实宽高，当<img>标签指定了宽、高时不受影响）
        image.src = Img.src
        //当图片比图片框小时，不做任何改变
        if (image.width < maxWidth && image.height < maxHeight) {
          Img.width = image.width
          Img.height = image.height
        } else //原图片宽高比例大于图片框宽高比例时，则以框的宽为标准缩放，反之以框的高为标准缩放
        {
          if (maxWidth / maxHeight <= image.width / image.height) //原图片宽高比例大于图片框宽高比例
          {
            Img.width = maxWidth; //以框的宽度为标准
            Img.height = maxWidth * (image.height / image.width)
```

```
            } else { //原图片宽高比例小于图片框宽高比例
                Img.width = maxHeight * (image.width / image.height)
                Img.height = maxHeight //以框的高度为标准
            }
        }
        this.canvasWidth = Img.width
        this.canvasHeight = Img.height
        let result = Img.width / preWidth
        return result
    },
```

4．测试

在浏览器中访问 http://localhost:8080/#/，页面返回结果如图 5-11 所示，接口返回参数和 Data 参数分别如表 5-3 和表 5-4 所示。

图 5-11　车牌识别测试示例

表 5-3　接口返回参数

返回值名称	类　　型	描　　述
code	int	返回结果，0 表示成功，非 0 表示对应错误号
msg	int	返回描述
data	object	返回的数据见下文

表 5-4　Data 参数

返回值名称	类　　型	描　　述
x	int	车牌 x 坐标（以图片左上角为原点）
y	int	车牌 y 坐标（以图片左上角为原点）
width	int	车牌宽度
height	int	车牌高度
color	string	车牌颜色
plate	string	车牌号码

返回示例如下。

```
{
   "code": 0,
   "msg": "成功",
   "data": [{
      "width": 75,
      "height": 21,
      "x": 314,
      "y": 177,
      "color": "blue",
      "plate": "粤 FQK883"
   }]
}
```

5.3.2 车型识别接口调用

车型识别技术是智能交通系统中极为重要的一环，目前主要应用在 3 个方面：一是公路收费系统，它不仅能提高公路收费系统的工作效率，还能有效监督收费站的工作人员在工作中是否存在违规收费行为；二是车辆管理系统，可以快速识别出车辆套牌等违法行为，在一定程度上减少交通事故隐患；三是公安侦缉系统，能够协助公安部门对交通肇事逃逸事件或其他犯罪活动进行调查。

1．接口参数说明

（1）接口描述。针对上传的某一张图片，识别该图像中车辆的颜色及车型。

（2）请求说明。

HTTP 方法如下。

```
POST
```

请求 URL 如下。

```
http://ai.cstor.cn/api/vehicle_type
```

（3）HTTP 的 header 参数如表 5-5 所示。

表 5-5 HTTP 的 header 参数

参 数 名 称	类 型	备 注
appId		应用 ID
timestamp		时间戳/s
nonce		随机字符串
sign		接口请求签名，待计算

（4）业务参数如表 5-6 所示。

表 5-6 业务参数

参 数 名 称	类 型	备 注
img	string	图像数据，base64 编码后进行 urlencode 编码，要求 base64 编码和 urlencode 编码后大小不超过 4 MB。图片的 base64 编码是包含图片头的，如 data:image/jpg;base64，支持的图片格式有 jpg、bmp、png，最短边至少为 50 px，最长边最大为 4096 px

2．前端接口调用实现

1）页面展示

车型识别页面展示如图 5-12 所示。

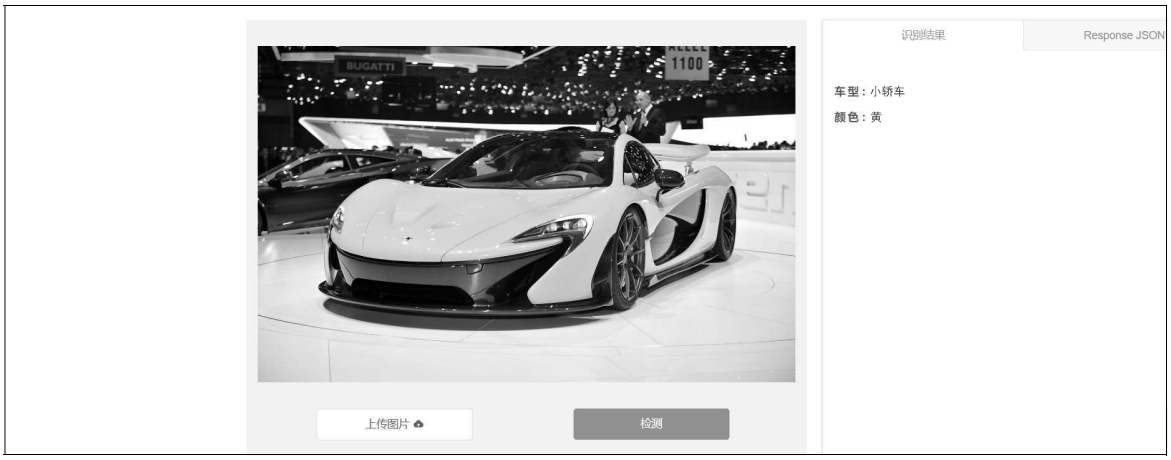

图 5-12　车型识别页面展示

2）HTML 部分

搭建框架，将页面的大概架构展示出来。

```
<div class="main">
  <div class="demo_title">
    <h3>功能演示</h3>
    <p>车型识别技术要求能够将运动中的车辆从复杂背景中提取并识别出来</p>
    <p>通过一系列算法，识别车辆类型、颜色等信息</p>
  </div>
  <div class="demo_body">
    <div class="request">
      <div class="image-container">
        <div class="imageContainerResult">
          <div class="showScanFlag" v-show="showScanFlag"> </div>
          <img :src="request_img" ref="ref_request_img" class="requestImg">
        </div>
      </div>
      <el-row>
        <el-col :span="10" :offset="3">
          <el-upload class="img-upload"
action="false" :show-file-list="false" :auto-upload="false"
ref="upload" :on-change="imgChange">
            <el-button plain>上传图片<i class="el-icon-upload el-icon--right"></i></el-button>
          </el-upload>
        </el-col>
        <el-col :span="10" :offset="1">
          <el-button type="primary" @click.native="detect()">检测</el-button>
        </el-col>
      </el-row>
      <div class="img_menu">
        <ul class="img_menu_ul">
```

```
            <li @click="imgMenuClick(item)"
class="img_menu_li" :class="menuId==item.id?'active':"" v-for="(item,index) in
imgMenu" :key="index">
              <img :src="item.src">
            </li>
          </ul>
        </div>
      </div>
      <div class="result">
        <el-tabs class="tabs" type="border-card" :stretch="true">
          <el-tab-pane label="识别结果">
            <div class="result_body1">
              <span class="span_style">车型：
</span><span>{{vehicle_type}}</span><br><br>
              <span class="span_style">颜色：</span><span>{{color}}</span>
            </div>
          </el-tab-pane>
          <el-tab-pane label="Response JSON">
            <div class="result_body2">
              <el-input type="textarea" v-model="textarea" :readonly="true"
resize="none" :rows="33">
              </el-input>
            </div>
          </el-tab-pane>
        </el-tabs>
      </div>
    </div>
</div>
```

3）js 部分

参照车牌管理的方法，需要调用车型的接口，代码如下。

```
export function vehicleType(params) {
  return request({
    method: 'post',
    url: 'http://ai.cstor.cn/api/vehicle_type/file',
    data: params
  }).then(res => {
      return res;
  });
}
```

其他 js 上传图片和检测图片的代码模块与车牌识别相同，这里不再一一讲解，可参考 5.3.1 节车牌识别前端接口的调用方法。

3．测试

在浏览器中访问 http://localhost:8080/#/，页面返回结果如图 5-13 所示，接口返回参数和 Data 参数分别如表 5-7 和表 5-8 所示。

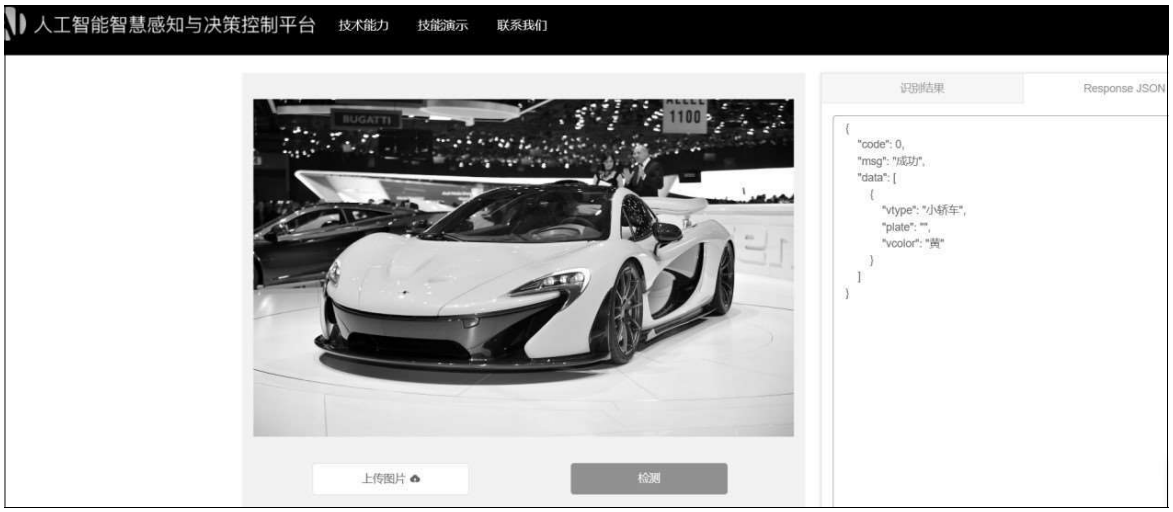

图 5-13　车型识别测试示例

表 5-7　接口返回参数

返回值名称	类　　型	描　　述
code	int	返回结果，0 表示成功，非 0 表示对应错误号
msg	int	返回描述
data	object	返回的数据见下文

表 5-8　Data 参数

返回值名称	类　　型	描　　述
vtype	string	车型
vcolor	string	车辆颜色

返回示例如下。

```
{
        "code": 0,
        "msg": "成功",
        "data": [
            {
                "vtype": "皮卡",
                "plate": "粤 FQK883",
                "vcolor": "灰"
            }
        ]
    }
```

习题

1．什么是车辆识别？
2．车辆识别的过程是怎样的？
3．上传一张车辆图片，如何调用接口文档将识别的车辆信息展示出来？
4．上传一张车型图片，如何调用接口文档展示车辆的车型信息？

参考文献

[1] 徐彩云．图像识别技术研究综述[J]．电脑知识与技术，2013（10）：2446-2447.
[2] 高晶辉．基于计算机网络技术的车辆识别技术的研究[J]．煤炭技术，2011，30（2）：184-186.
[3] 刘建伟，刘媛，罗雄麟．深度学习研究进展[J]．计算机应用研究，2014，31（7）：1921-1930.
[4] 陈钱，柏连发，张保民．红外图像直方图双向均衡技术研究[J]．红外与毫米波学报，2003（6）：428-430.
[5] 焦李成，杨淑媛，刘芳，等．神经网络七十年：回顾与展望[J]．计算机学报，2016，39（8）：1697-1716.
[6] 邓天民，邵毅明，崔建江．一种车型识别算法及其应用[C]//中国仪器仪表学会．计算机技术与应用进展：全国第 17 届计算机科学与技术应用（CACIS）学术会议论文集（上册）．合肥：中国科学技术大学出版社，2006：471-474.
[7] 百度 AI 开放平台．https://ai.baidu.com/.
[8] 郭磊，李克强，王建强，等．一种基于特征的车辆检测方法[J]．汽车工程，2006，28（11）：1031-1035.