

uniapp 安卓百度识图UTS原生插件

目 录

使用说明	1
使用方法	1.1
更新日志	1.2
插件方法	2
申请插件所需权限	2.1
初始化SDK	2.2
组合接口API	2.3
通用物体和场景识别	2.4
图像单主体检测	2.5
动物识别	2.6
植物识别	2.7
logo识别-检索	2.8
logo识别-入库	2.9
logo识别-删除	2.10
果蔬识别	2.11
图像内容理解-提交请求	2.12
图像内容理解-获取结果	2.13
图像多主体检测	2.14
菜品识别	2.15
地标识别	2.16
车辆检测	2.17
车型识别	2.18

使用方法

介绍

安卓百度识图UTS原生插件集成了百度图像识别API服务，功能包括通用物体和场景识别，动物识别，植物识别，果蔬识别，菜品识别，车辆识别等。使用插件必须先成为[百度开发者](#)申请应用，插件UTS开发，支持uniapp和uniapp x

联系作者

关注微信公众号可联系作者



插件地址

<https://ext.dcloud.net.cn/plugin?id=24216>

权限

1. android.permission.WRITE_EXTERNAL_STORAGE
2. android.permission.READ_EXTERNAL_STORAGE

用法

在需要使用插件的页面加载以下代码

使用方法

```
import * as module from "@uni_modules/leven-uts-baiduRecImage"
```

示例文件

请到插件首页通过 [使用 HBuilderX 导入示例项目](#)

更新日志

2025-07-05

首次发布

申请插件所需权限

方法名

requestPermissions

用法

- 用法如下：

```
module.requestPermissions(res => {
  console.log(res);
})
```

- 参数说明

无

回调

- 示例

```
{
  "data": {
    "grantedList": [
      "android.permission.READ_EXTERNAL_STORAGE",
      "android.permission.WRITE_EXTERNAL_STORAGE"
    ]
  },
  "message": "权限被拒绝",
  "code": -1
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象

参数名	参数类型	参数描述
data.grantedList	Array	申请的权限列表，被拒绝的时候会返回
code	Integer	返回类型，0.成功，其他：失败

初始化SDK

方法名

initSdk

在使用其他接口之前必须先初始化

用法

- 用法如下：

```
module.initSdk({
  //百度申请的应用API Key
  clientId: "",
  //百度申请的应用Secret Key
  clientSecret: ""
}, res => {
  console.log(res)
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
clientId	String	是	无	百度申请的应用API Key
clientSecret	String	是	无	百度申请的应用Secret Key

回调

- 示例

```
{
  "message": "",
  "data": { },
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

组合接口API

方法名

combination

提供一个API接口，同时调用多个模型服务。支持图像识别的多个接口：通用物体和场景识别、图像单主体检测、动物识别、植物识别、果蔬识别、自定义菜品识别-检索、菜品识别、红酒识别、货币识别、地标识别、图像多主体检测等12个模型服务的调用。可通过入参灵活指定需要调用的模型服务，传入图像，返回指定模型的识别结果。

用法

- 用法如下：

```
module.combination({  
  //识别的图片地址或本地文件路径  
  url: "https://img0.baidu.com/it/u=3036689871,1624291970&fm=253&app=138&f=JPE  
  // url: "/storage/emulated/0/DCIM/Pindd/chat/1731129361947.jpg",  
  //指定本次调用的模型服务，以字符串数组表示。元素含义如下：  
  //advanced_general：通用物体和场景识别  
  //object_detect：图像单主体检测  
  //multi_object_detect：图像多主体检测  
  //animal：动物识别  
  //plant：植物识别  
  //ingredient：果蔬识别  
  //dish_search：自定义菜品识别-检索  
  //dishes：菜品识别  
  //red_wine：红酒识别  
  //currency：货币识别  
  //landmark：地标识别  
  scenes: ['animal', 'plant'],  
  //对特定服务，支持的个性化参数，若不填则使用默认设置。jsonObject说明：  
  //key为要设置入参的服务类型，可取值同scenes字段，例如：  
  //advanced_general：通用物体和场景识别  
  //object_detect：图像单主体检测  
  //value为各模型服务个性化参数，详情请参照下文或百度AI开放平台官网  
  // sceneConf: {}  
}, res => {  
  console.log(res);  
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
url	String	是	无	识别的图片地址 或本地文件路径
scenes	Array[String]	是	无	指定本次调用的 模型服务，以字 符串数组表示。 元素含义请参考 下方说明
sceneConf	Object	否	无	对特定服务，支 持的个性化参 数，若不填则使 用默认设置。

scenes 参数说明：

advanced_general：通用物体和场景识别

object_detect：图像单主体检测

multi_object_detect：图像多主体检测

animal：动物识别

plant：植物识别

ingredient：果蔬识别

dish_search：自定义菜品识别-检索

dishes：菜品识别

red_wine：红酒识别

currency：货币识别

landmark：地标识别

回调

- 示例

```
{
  "data": {
    "log_id": 17516530906920116,
    "result": {
      "animal": {
        "log_id": 1941199877223095800,
        "result": [
          {
            "name": "伊犁马",
            "score": "0.104577"
          },
          {
```

```
        "name": "骡子",
        "score": "0.0987811"
    },
    {
        "name": "蒙古马",
        "score": "0.0941953"
    },
    {
        "name": "荷兰奶牛",
        "score": "0.0603844"
    },
    {
        "name": "和牛",
        "score": "0.0536077"
    },
    {
        "name": "夏洛来牛",
        "score": "0.0438796"
    }
]
},
"plant": {
    "log_id": 1941199876534733300,
    "result": [
        {
            "name": "非植物",
            "score": 0.7388127
        }
    ]
}
}
},
"message": "",
"code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.log_id	String	唯一的log id，用于问题定位。
data.result	Object	返回结果json串，其内包含要调用的各个模型服务的返回结果。

参数名	参数类型	参数描述
data.advanced_general	Object	「通用物体和场景识别」服务返回结果
data.object_detect	Object	「图像单主体检测」服务返回结果
data.multi_object_detect	Object	「图像多主体检测」服务返回结果
data.animal	Object	「动物识别」服务返回结果
data.plant	Object	「植物识别」服务返回结果
data.ingredient	Object	「果蔬识别」服务返回结果
data.dish_search	Object	「自定义菜品识别-检索」服务返回结果
data.dishes	Object	「菜品识别」服务返回结果
data.red_wine	Object	「红酒识别」服务返回结果
data.currency	Object	「货币识别」服务返回结果
data.landmark	Object	「地标识别」服务返回结果
code	Integer	返回类型，0.成功，其他：失败

通用物体和场景识别

方法名

advancedGeneral

该请求用于通用物体及场景识别，即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片中的多个物体及场景标签。

用法

- 用法如下：

```
module.advancedGeneral({
  //识别的图片地址或本地文件路径
  // url: "https://nimg.ws.126.net/?url=http%3A%2F%2Fdingyue.ws.126.net%2F2025
  url: "/storage/emulated/0/DCIM/Pindd/chat/1731129361947.jpg",
  //用于控制返回结果是否带有百科信息，若不输入此参数，则默认不返回百科结果；若输入此参数，会根据
  baikeNum: 2
}, res => {
  console.log(res);
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
url	String	是	无	识别的图片地址或本地文件路径
baikeNum	Integer	否	无	用于控制返回结果是否带有百科信息，若不输入此参数，则默认不返回百科结果；若输入此参数，会根据输入的整数返回相应个数的百科信息

- ```
{
 "data": {
 "result_num": 5,
 "log_id": 1941202881408309800,
 "result": [
 {
 "root": "商品-电脑办公",
 "baike_info": {
 "description": "书本是指装订成册的著作，包括纸质书、绢、竹简、羊皮卷等。语出《所",
 "image_url": "https://bking.cdn.bcebos.com/pic/34fae6cd7b899e51ec7a0",
 "baike_url": "https://baike.baidu.com/item/%E4%B9%A6%E6%9C%AC/360626",
 },
 "keyword": "书本",
 "score": 0.957122
 },
 {
 "root": "非自然图像-书籍封面",
 "baike_info": {
 "description": "",
 "image_url": "",
 "baike_url": ""
 },
 "keyword": "书籍",
 "score": 0.717132
 },
 {
 "root": "非自然图像-文字图",
 "keyword": "防火宣传画",
 "score": 0.436656
 },
 {
 "root": "非自然图像-文字图",
 "keyword": "秩序册",
 "score": 0.205975
 },
 {
 "root": "非自然图像-文字图",
 "keyword": "建筑施工安全挂图",
 "score": 0.0095
 }
]
 },
 "message": ""
}
```

```
"code": 0
}
```

回调说明：

| 参数名                                | 参数类型          | 参数描述                            |
|------------------------------------|---------------|---------------------------------|
| message                            | String        | 消息提示                            |
| data                               | Object        | 数据对象                            |
| data.log_id                        | String        | 唯一的log id，用于问题定位                |
| data.result_num                    | Integer       | 返回结果数目，及result数组中的元素个数，最多返回5个结果 |
| data.result                        | Array[Object] | 标签结果数组                          |
| data.result.keyword                | String        | 图片中的物体或场景名称                     |
| data.result.score                  | float         | 置信度，0-1                         |
| data.result.root                   | String        | 识别结果的上层标签，有部分钱币、动漫、烟酒等tag无上层标签  |
| data.result.baike_info             | Object        | 对应识别结果的百科词条名称                   |
| data.result.baike_info.baike_url   | String        | 对应识别结果百度百科页面链接                  |
| data.result.baike_info.image_url   | String        | 对应识别结果百科图片链接                    |
| data.result.baike_info.description | String        | 对应识别结果百科内容描述                    |
| code                               | Integer       | 返回类型，0.成功，其他：失败                 |

# 图像单主体检测

## 方法名

objectDetect

用户向服务请求检测图像中的主体位置

## 用法

- 用法如下：

```
module.objectDetect({
 //识别的图片地址或本地文件路径
 // url: "https://img0.baidu.com/it/u=1178721097,709892898&fm=253&app=138&f=J
 url: "/storage/emulated/0/levenBaiduFace/1234.jpg",
 //如果检测主体是人，主体区域是否带上人脸部分，0-不带人脸区域，其他-带人脸区域，裁剪类需求推
 withFace: 1
}, res => {
 console.log(res)
})
```

- 参数说明

| 参数名      | 参数类型    | 是否必填 | 默认值 | 参数描述                                                                        |
|----------|---------|------|-----|-----------------------------------------------------------------------------|
| url      | String  | 是    | 无   | 识别的图片地址或本地文件路径                                                              |
| withFace | Integer | 否    | 无   | 如果检测主体是人，主体区域是否带上人脸部分，0-不带人脸区域，其他-带人脸区域，裁剪类需求推荐带人脸，检索/识别类需求推荐不带人脸。默认取1，带人脸。 |



# 回调

- 示例

```
{
 "data": {
 "log_id": 1941205095849767200,
 "result": {
 "left": 1,
 "height": 572,
 "width": 560,
 "top": 1
 }
 },
 "message": "",
 "code": 0
}
```

- 回调说明：

| 参数名                | 参数类型          | 参数描述                |
|--------------------|---------------|---------------------|
| message            | String        | 消息提示                |
| data               | Object        | 数据对象                |
| data.log_id        | String        | 唯一的log id，用于问题定位    |
| data.result        | Array[Object] | 标签结果数组              |
| data.result.left   | String        | 表示定位位置的长方形左上顶点的水平坐标 |
| data.result.top    | String        | 表示定位位置的长方形左上顶点的垂直坐标 |
| data.result.width  | String        | 表示定位位置的长方形的宽度，单位px  |
| data.result.height | String        | 表示定位位置的长方形的高度，单位px  |
| code               | Integer       | 返回类型，0.成功，其他：失败     |

# 动物识别

## 方法名

animal

该请求用于识别一张图片，即对于输入的一张图片（可正常解码，且长宽比较合适），输出动物识别结果。

## 用法

- 用法如下：

```
module.animal({
 //识别的图片地址或本地文件路径
 // url: "https://img0.baidu.com/it/u=2055817040,2686595757&fm=253&app=138&f=
 url: "/storage/emulated/0/Pictures/WeiXin/wx_camera_1736592085759.jpg",
 //返回预测得分top结果数，默认为6
 topNum: 3,
 //用于控制返回结果是否带有百科信息，若不输入此参数，则默认不返回百科结果；若输入此参数，会相
 baikeNum: 2
}, res => {
 console.log(res)
})
```

- 参数说明

| 参数名    | 参数类型    | 是否必填 | 默认值 | 参数描述                  |
|--------|---------|------|-----|-----------------------|
| url    | String  | 是    | 无   | 识别的图片地址<br>或本地文件路径    |
| topNum | Integer | 否    | 6   | 返回预测得分top<br>结果数，默认为6 |

| 参数名      | 参数类型    | 是否必填 | 默认值 | 参数描述                                                           |
|----------|---------|------|-----|----------------------------------------------------------------|
| baikeNum | Integer | 否    | 无   | 用于控制返回结果是否带有百科信息，若不输入此参数，则默认不返回百科结果；若输入此参数，会根据输入的整数返回相应个数的百科信息 |

## 回调

- 示例

```
{
 "data": {
 "log_id": 1941205686602509800,
 "result": [
 {
 "name": "布偶猫",
 "baike_info": {
 "description": "",
 "image_url": "",
 "baike_url": ""
 },
 "score": "0.763794"
 },
 {
 "name": "蓝眼白猫",
 "baike_info": {
 "description": "蓝眼白猫，祖先为安哥拉猫、波斯猫。起源于1880年代。习性温顺，耳",
 "image_url": "https://bking.cdn.bcebos.com/pic/42166d224f4a20a4e28e8",
 "baike_url": "https://baike.baidu.com/item/%E8%93%9D%E7%9C%BC%E7%99%"
 },
 "score": "0.0655258"
 },
 {
 "name": "家猫",
 "score": "0.0382168"
 }
]
 },
 "message": "",
```

```
"code": 0
}
```

回调说明：

| 参数名                                | 参数类型          | 参数描述             |
|------------------------------------|---------------|------------------|
| message                            | String        | 消息提示             |
| data                               | Object        | 数据对象             |
| data.log_id                        | String        | 唯一的log id，用于问题定位 |
| data.result                        | Array[Object] | 标签结果数组           |
| data.result.name                   | String        | 动物名称，示例：蒙古马      |
| data.result.score                  | float         | 置信度，0-1          |
| data.result.baike_info             | Object        | 对应识别结果的百科词条名称    |
| data.result.baike_info.baike_url   | String        | 对应识别结果百度百科页面链接   |
| data.result.baike_info.image_url   | String        | 对应识别结果百科图片链接     |
| data.result.baike_info.description | String        | 对应识别结果百科内容描述     |
| code                               | Integer       | 返回类型，0.成功，其他：失败  |

# 植物识别

## 方法名

plant

该请求用于识别一张图片，即对于输入的一张图片（可正常解码，且长宽比较合适），输出植物识别结果。

## 用法

- 用法如下：

```
module.plant({
 //识别的图片地址或本地文件路径
 // url: "https://img0.baidu.com/it/u=2055817040,2686595757&fm=253&app=138&f=
 url: "/storage/emulated/0/DCIM/Camera/IMG_20250323_153609.jpg",
 //用于控制返回结果是否带有百科信息，若不输入此参数，则默认不返回百科结果；若输入此参数，会根据输入
 baikeNum: 2
}, res => {
 console.log(res)
})
```

- 参数说明

| 参数名      | 参数类型    | 是否必填 | 默认值 | 参数描述                                                           |
|----------|---------|------|-----|----------------------------------------------------------------|
| url      | String  | 是    | 无   | 识别的图片地址或本地文件路径                                                 |
| baikeNum | Integer | 否    | 无   | 用于控制返回结果是否带有百科信息，若不输入此参数，则默认不返回百科结果；若输入此参数，会根据输入的整数返回相应个数的百科信息 |

## 回调

- 示例

```
{
 "data": {
 "log_id": 1941207333452726000,
 "result": [
 {
 "name": "樱花",
 "baike_info": {
 "description": "樱花 (Prunus subg. Cerasus sp.) 是蔷薇科、樱亚属植物。是...",
 "image_url": "https://bking.cdn.bcebos.com/pic/0df3d7ca7bcb0a46f21fc",
 "baike_url": "https://baike.baidu.com/item/%E6%A8%B1%E8%8A%B1/70387"
 },
 "score": 0.8441227
 },
 {
 "name": "东京樱花",
 "baike_info": {
 "description": "东京樱花 (学名: Prunus × yedoensis Matsum.) , 别名樱花、E...",
 "image_url": "https://bking.cdn.bcebos.com/pic/11385343fbf2b2119313d",
 "baike_url": "https://baike.baidu.com/item/%E4%B8%9C%E4%BA%AC%E6%A8%"
 },
 "score": 0.43085563
 },
 {
 "name": "紫叶李",
 "score": 0.13593236
 }
]
 },
 "message": "",
 "code": 0
}
```

- 回调说明：

| 参数名               | 参数类型          | 参数描述             |
|-------------------|---------------|------------------|
| message           | String        | 消息提示             |
| data              | Object        | 数据对象             |
| data.log_id       | String        | 唯一的log id，用于问题定位 |
| data.result       | Array[Object] | 标签结果数组           |
| data.result.name  | String        | 动物名称，示例：蒙古马      |
| data.result.score | float         | 置信度，0-1          |

| 参数名                                | 参数类型    | 参数描述            |
|------------------------------------|---------|-----------------|
| data.result.baike_info             | Object  | 对应识别结果的百科词条名称   |
| data.result.baike_info.baike_url   | String  | 对应识别结果百度百科页面链接  |
| data.result.baike_info.image_url   | String  | 对应识别结果百科图片链接    |
| data.result.baike_info.description | String  | 对应识别结果百科内容描述    |
| code                               | Integer | 返回类型，0.成功，其他：失败 |

# logo识别-检索

## 方法名

logo

具体可参考<https://ai.baidu.com/ai-doc/IMAGERECOGNITION/Ok3bcxc59>

## 用法

- 用法如下：

```
module.logo({
 //识别的图片地址或本地文件路径
 url: "https://qcloud.dpfile.com/pc/zF3oCvKmIZ8zAl2lTUoIJbaDh3NyBRjFhpos2H4_A",
 // url: "/storage/emulated/0/DCIM/Camera/IMG_20250323_153609.jpg",
 //是否只检索用户子库，true则只检索用户子库，false(默认)为检索底库+用户子库
 customLib: true
}, res => {
 console.log(res);
})
```

- 参数说明

| 参数名       | 参数类型    | 是否必填 | 默认值 | 参数描述                                       |
|-----------|---------|------|-----|--------------------------------------------|
| url       | String  | 是    | 无   | 识别的图片地址或本地文件路径                             |
| customLib | Integer | 否    | 无   | 是否只检索用户子库，true则只检索用户子库，false(默认)为检索底库+用户子库 |

## 回调

- 示例



```
{
 "data": {
 "log_id": 843411868,
 "result_num": 1,
 "result": [
 {
 "type": 0,
 "name": "科颜氏",
 "probability": 0.99998807907104,
 "location": {
 "width": 296,
 "top": 20,
 "height": 128,
 "left": 23
 }
 }
]
 },
 "message": "",
 "code": 0
}
```

回调说明：

| 参数名                        | 参数类型          | 参数描述                                                    |
|----------------------------|---------------|---------------------------------------------------------|
| message                    | String        | 消息提示                                                    |
| data                       | Object        | 数据对象                                                    |
| data.log_id                | String        | 唯一的log id，用于问题定位                                        |
| data.result_num            | String        | 识别结果数，标识返回结果数目                                          |
| data.result                | Array[Object] | 返回结果数组，每一项为一个识别出的logo                                   |
| data.result.name           | String        | 识别的品牌名称                                                 |
| data.result.probability    | float         | 分类结果置信度（0--1.0）                                         |
| data.result.type           | String        | type=0为1千种高优商标识别结果;type=1为2万类logo库的结果；其它type为自定义logo库结果 |
| data.result.location       | Object        | 位置信息（左起像素位置、上起像素位置、像素宽、像素高）                             |
| data.result.location .left | String        | 左起像素位置                                                  |

| 参数名                          | 参数类型    | 参数描述            |
|------------------------------|---------|-----------------|
| data.result.location .top    | String  | 上起像素位置          |
| data.result.location .width  | String  | 像素宽             |
| data.result.location .height | String  | 像素高             |
| code                         | Integer | 返回类型，0.成功，其他：失败 |

# logo识别-入库

## 方法名

logoAdd

具体可参考<https://ai.baidu.com/ai-doc/IMAGERECOGNITION/Ok3bcxc59>

## 用法

- 用法如下：

```
module.logoAdd({
 //识别的图片地址或本地文件路径
 url: "https://qcloud.dpfile.com/pc/zF3oCvKmIZ8zAl2lTUoIJbaDh3NybRjFhpos2H4_A",
 // url: "/storage/emulated/0/DCIM/Camera/IMG_20250323_153609.jpg",
 //此处需要传对应的品牌名称name字段，必须为json格式，检索时带回。name长度小于100B，示例{'
 brief: `{"name": "abc"}`
}, res => {
 console.log(res);
})
```

- 参数说明

| 参数名   | 参数类型   | 是否必填 | 默认值 | 参数描述                                                              |
|-------|--------|------|-----|-------------------------------------------------------------------|
| url   | String | 是    | 无   | 识别的图片地址或本地文件路径                                                    |
| brief | String | 否    | 无   | 此处需要传对应的品牌名称name字段，必须为json格式，检索时带回。name长度小于100B，示例{"name": "abc"} |

## 回调

- 示例

```
{
 "data": {
 "log_id": 972934653,
 "cont_sign": "217113248,419422301"
 },
 "message": "",
 "code": 0
}
```

- 回调说明：

| 参数名            | 参数类型    | 参数描述             |
|----------------|---------|------------------|
| message        | String  | 消息提示             |
| data           | Object  | 数据对象             |
| data.log_id    | String  | 唯一的log id，用于问题定位 |
| data.cont_sign | String  | 输入图片签名，可用于删除     |
| code           | Integer | 返回类型，0.成功，其他：失败  |

# logo识别-删除

## 方法名

logoDelete

具体可参考<https://ai.baidu.com/ai-doc/IMAGERECOGNITION/Ok3bcxc59>

## 用法

- 用法如下：

```
module.logoDelete({
 //识别的图片地址或本地文件路径和contSign二选一
 url: "https://qcloud.dpfile.com/pc/zF3oCvKmIZ8zAl2lTUoIJbaDh3NybRjFhpos2H4_A",
 // url: "/storage/emulated/0/DCIM/Camera/IMG_20250323_153609.jpg",
 //图片签名（和url二选一，url优先级更高）
 contSign: "123456"
}, res => {
 console.log(res);
})
```

- 参数说明

| 参数名      | 参数类型   | 是否必填 | 默认值 | 参数描述                   |
|----------|--------|------|-----|------------------------|
| url      | String | 是    | 无   | 识别的图片地址或本地文件路径         |
| contSign | String | 否    | 无   | 图片签名（和url二选一，url优先级更高） |

## 回调

- 示例

```
{
 "data": {
 "log_id": 972934653
 }
}
```

logoi识别-删除

```
 },
 "message": "",
 "code": 0
}
```

- 回调说明：

| 参数名         | 参数类型    | 参数描述             |
|-------------|---------|------------------|
| message     | String  | 消息提示             |
| data        | Object  | 数据对象             |
| data.log_id | String  | 唯一的log id，用于问题定位 |
| code        | Integer | 返回类型，0.成功，其他：失败  |

# 果蔬识别

## 方法名

ingredient

该请求用于识别果蔬类食材，即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片中的果蔬食材结果。

## 用法

- 用法如下：

```
module.ingredient({
 //识别的图片地址或本地文件路径
 url: "https://pics5.baidu.com/feed/6a600c338744ebf88d6a403c5ed7cb256059a753.",
 // url: "/storage/emulated/0/DCIM/Camera/IMG_20250323_153609.jpg",
 //返回预测得分top结果数，如果为空或小于等于0默认为5；如果大于20默认20
 topNum: 2
}, res => {
 console.log(res);
})
```

- 参数说明

| 参数名    | 参数类型    | 是否必填 | 默认值 | 参数描述                                   |
|--------|---------|------|-----|----------------------------------------|
| url    | String  | 是    | 无   | 识别的图片地址或本地文件路径                         |
| topNum | Integer | 否    | 5   | 返回预测得分top结果数，如果为空或小于等于0默认为5；如果大于20默认20 |

## 回调

- 示例

```
{
 "data": {
 "result_num": 2,
 "log_id": 1941211221829888500,
 "result": [
 {
 "name": "苹果",
 "score": 0.9983537
 },
 {
 "name": "非果蔬食材",
 "score": 0.001263225
 }
]
 },
 "message": "",
 "code": 0
}
```

- 回调说明：

| 参数名               | 参数类型          | 参数描述             |
|-------------------|---------------|------------------|
| message           | String        | 消息提示             |
| data              | Object        | 数据对象             |
| data.log_id       | String        | 唯一的log id，用于问题定位 |
| data.result_num   | Integer       | 返回结果数目           |
| data.result       | Array[Object] | 标签结果数组           |
| data.result.name  | String        | 动物名称，示例：蒙古马      |
| data.result.score | float         | 置信度，0-1          |
| code              | Integer       | 返回类型，0.成功，其他：失败  |



# 图像内容理解-提交请求

## 方法名

imageUnderstandingRequest

该请求用于图像内容理解，支持输入图片和提问信息，多维度识别与理解图片内容，包括人、物、行为、场景、文字等，输出对图片内容的描述。用于图片内容问答，图片打标签，图片物体识别等业务场景

## 用法

- 用法如下：

```
module.imageUnderstandingRequest({
 //识别的图片地址或本地文件路径
 url: "http://www.yeyuboke.com/svgga/image_default.jpg",
 // url: "/storage/emulated/0/DCIM/Camera/IMG_20250323_153609.jpg",
 //提问信息，如“这张图片里有什么？”、“图中的人物是谁，并进行简单介绍”限制 100 个字符之内
 question: "这张图片里有什么"
}, res => {
 console.log(res);
 if (res.code == 0) {
 let data = res.data;
 let result = data.result || {};
 this.taskId = result.task_id;
 }
})
```

- 参数说明

| 参数名      | 参数类型   | 是否必填 | 默认值 | 参数描述                                            |
|----------|--------|------|-----|-------------------------------------------------|
| url      | String | 是    | 无   | 识别的图片地址或本地文件路径                                  |
| question | String | 否    | 无   | 提问信息，如“这张图片里有什么？”、“图中的人物是谁，并进行简单介绍”限制 100 个字符之内 |

# 回调

- 示例

```
{
 "data": {
 "log_id": 1941212002376818200,
 "result": {
 "task_id": "1941212002376818081"
 }
 },
 "message": "",
 "code": 0
}
```

- 回调说明：

| 参数名                 | 参数类型          | 参数描述                             |
|---------------------|---------------|----------------------------------|
| message             | String        | 消息提示                             |
| data                | Object        | 数据对象                             |
| data.log_id         | String        | 唯一的log id，用于问题定位                 |
| data.result         | Array[Object] | 标签结果数组                           |
| data.result.task_id | String        | 该请求生成的task_id，后续使用该task_id获取识别结果 |
| code                | Integer       | 返回类型，0.成功，其他：失败                  |

# 图像内容理解-获取结果

## 方法名

imageUnderstandingResult

该请求用于图像内容理解，支持输入图片和提问信息，多维度识别与理解图片内容，包括人、物、行为、场景、文字等，输出对图片内容的描述。用于图片内容问答，图片打标签，图片物体识别等业务场景

## 用法

- 用法如下：

```
module.imageUnderstandingResult({
 taskId: this.taskId
}, res => {
 console.log(res);
})
```

- 参数说明

| 参数名    | 参数类型   | 是否必填 | 默认值 | 参数描述                 |
|--------|--------|------|-----|----------------------|
| taskId | String | 是    | 无   | 调用提交请求接口时，返回的task_id |

## 回调

- 示例

```
{
 "data": {
 "log_id": 1941212499595490300,
 "result": {
 "ret_msg": "success",
 "task_id": "1941212002376818081",
 "ret_code": 0,
 "description": "这张图片里有：\n\n1. 一棵大树，树叶呈现为金黄色，显示出秋天的特色。
 }
 },
}
```

```
"message": "",
"code": 0
}
```

回调说明：

| 参数名                     | 参数类型          | 参数描述                               |
|-------------------------|---------------|------------------------------------|
| message                 | String        | 消息提示                               |
| data                    | Object        | 数据对象                               |
| data.log_id             | String        | 唯一的log id，用于问题定位                   |
| data.result             | Array[Object] | 标签结果数组                             |
| data.result.task_id     | String        | 该请求生成的task_id                      |
| data.result.ret_code    | String        | 识别状态，0：处理成功；1：处理中                  |
| data.result.ret_msg     | String        | 识别状态信息，success：处理成功；processing：处理中 |
| data.result.description | String        | 针对输入的 question 问题，对图片内容进行分析后输出的答案  |
| code                    | Integer       | 返回类型，0.成功，其他：失败                    |

# 图像多主体检测

## 方法名

multiObjectDetect

检测出图片中多个主体，并给出位置、标签和置信得分。

## 用法

- 用法如下：

```
module.multiObjectDetect({
 //识别的图片地址或本地文件路径
 url: "https://pics5.baidu.com/feed/6a600c338744ebf88d6a403c5ed7cb256059a753.",
}, res => {
 console.log(res);
})
```

- 参数说明

| 参数名 | 参数类型   | 是否必填 | 默认值 | 参数描述           |
|-----|--------|------|-----|----------------|
| url | String | 是    | 无   | 识别的图片地址或本地文件路径 |

## 回调

- 示例

```
{
 "data": {
 "log_id": 1941213342475198700,
 "result": [
 {
 "location": {
 "left": 238,
 "height": 624,
 "width": 372,
 "top": 2
 }
 }
]
 }
}
```

```

 "name": "文化娱乐",
 "score": 0.1265733
 },
 {
 "location": {
 "left": 249,
 "height": 249,
 "width": 224,
 "top": 511
 },
 "name": "果蔬生鲜",
 "score": 0.6271499
 },
 {
 "location": {
 "left": 10,
 "height": 269,
 "width": 252,
 "top": 147
 },
 "name": "果蔬生鲜",
 "score": 0.81888455
 },
 {
 "location": {
 "left": 1,
 "height": 245,
 "width": 204,
 "top": 398
 },
 "name": "果蔬生鲜",
 "score": 0.61972374
 },
 {
 "location": {
 "left": 471,
 "height": 285,
 "width": 166,
 "top": 537
 },
 "name": "果蔬生鲜",
 "score": 0.21421935
 }
]
},
"message": "",
"code": 0
}

```

- 回调说明：

| 参数名                          | 参数类型          | 参数描述                |
|------------------------------|---------------|---------------------|
| message                      | String        | 消息提示                |
| data                         | Object        | 数据对象                |
| data.log_id                  | String        | 唯一的log id，用于问题定位    |
| data.result                  | Array[Object] | 标签结果数组              |
| data.result.name             | String        | 图片标签                |
| data.result.score            | float         | 置信度，0-1             |
| data.result.location         | Object        | 图中目标主体的检测框位置信息      |
| data.result.location .left   | String        | 表示定位位置的长方形左上顶点的水平坐标 |
| data.result.location .top    | String        | 表示定位位置的长方形左上顶点的垂直坐标 |
| data.result.location .width  | String        | 表示定位位置的长方形的宽度，单位px  |
| data.result.location .height | String        | 表示定位位置的长方形的高度，单位px  |
| code                         | Integer       | 返回类型，0.成功，其他：失败     |

# 菜品识别

## 方法名

dish

该请求用于菜品识别。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片的菜品名称、卡路里信息、置信度。

## 用法

- 用法如下：

```
module.dish({
 //识别的图片地址或本地文件路径
 url: "https://qcloud.dpfile.com/pc/iXLZPENj7u0exYjeZsl-uaGGbpK4pnEiIXgDFLoK0
 //返回结果top n,默认5
 topNum: 3,
 //默认0.95，可以通过该参数调节识别效果，降低非菜识别率
 filterThreshold: 0.9,
 //用于控制返回结果是否带有百科信息，若不输入此参数，则默认不返回百科结果；若输入此参数，会相
 baikeNum: 2
}, res => {
 console.log(res);
})
```

- 参数说明

| 参数名             | 参数类型   | 是否必填 | 默认值 | 参数描述                         |
|-----------------|--------|------|-----|------------------------------|
| url             | String | 是    | 无   | 识别的图片地址或本地文件路径               |
| topNum          | String | 否    | 无   | 返回结果top n,默认5                |
| filterThreshold | String | 否    | 无   | 默认0.95，可以通过该参数调节识别效果，降低非菜识别率 |



| 参数名      | 参数类型   | 是否必填 | 默认值 | 参数描述                                                           |
|----------|--------|------|-----|----------------------------------------------------------------|
| baikeNum | String | 否    | 无   | 用于控制返回结果是否带有百科信息，若不输入此参数，则默认不返回百科结果；若输入此参数，会根据输入的整数返回相应个数的百科信息 |

## 回调

- 示例

```
{
 "data": {
 "result_num": 3,
 "log_id": 1941214581455441700,
 "result": [
 {
 "calorie": "226",
 "name": "石锅酱汤",
 "has_calorie": true,
 "baike_info": {
 "description": "石锅酱汤是一个以高汤，豆腐，土豆为主要材料的菜品，口味咸鲜。",
 "image_url": "https://bking.cdn.bcebos.com/pic/f2deb48f8c5494ee5716e",
 "baike_url": "https://baike.baidu.com/item/%E7%9F%B3%E9%94%85%E9%85%",
 },
 "probability": "0.334537"
 },
 {
 "calorie": "14",
 "name": "泡菜汤",
 "has_calorie": true,
 "baike_info": {
 "description": "泡菜汤属于韩国料理，由猪肉、白菜，口味是甜辣。含有丰富的粗纤维",
 "image_url": "https://bking.cdn.bcebos.com/pic/9f2f070828381f30e9243",
 "baike_url": "https://baike.baidu.com/item/%E6%B3%A1%E8%8F%9C%E6%B1%",
 },
 "probability": "0.192817"
 },
 {
 "calorie": "98.98",
```

```
 "name": "番茄炖牛腩",
 "has_calorie": true,
 "probability": "0.120097"
 }
],
 "message": "",
 "code": 0
}
```

回调说明：

| 参数名                                | 参数类型          | 参数描述              |
|------------------------------------|---------------|-------------------|
| message                            | String        | 消息提示              |
| data                               | Object        | 数据对象              |
| data.log_id                        | String        | 唯一的log id，用于问题定位  |
| data.result_num                    | Integer       | 返回结果数目            |
| data.result                        | Array[Object] | 菜品识别结果数组          |
| data.result.name                   | String        | 菜名，示例：鱼香肉丝        |
| data.result.calorie                | float         | 卡路里，每100g的卡路里含量   |
| data.result.probability            | float         | 识别结果中每一行的置信度值，0-1 |
| data.result.baike_info             | Object        | 对应识别结果的百科词条名称     |
| data.result.baike_info.baike_url   | String        | 对应识别结果百度百科页面链接    |
| data.result.baike_info.image_url   | String        | 对应识别结果百科图片链接      |
| data.result.baike_info.description | String        | 对应识别结果百科内容描述      |
| code                               | Integer       | 返回类型，0.成功，其他：失败   |

# 地标识别

## 方法名

landmark

该请求用于识别地标，即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片中的地标识别结果。

## 用法

- 用法如下：

```
module.landmark({
 //识别的图片地址或本地文件路径
 url: "https://qcloud.dpfile.com/pc/hZbEa4gmnSZKfeaCFFlswGhWPaaR6UmCqxbbmak2n",
 res => {
 console.log(res);
 })
```

- 参数说明

| 参数名 | 参数类型   | 是否必填 | 默认值 | 参数描述           |
|-----|--------|------|-----|----------------|
| url | String | 是    | 无   | 识别的图片地址或本地文件路径 |

## 回调

- 示例

```
{
 "data": {
 "log_id": 1941215494209726200,
 "result": {
 "landmark": "广州塔"
 }
 },
 "message": ""
```

```
"code": 0
}
```

- 回调说明：

| 参数名                  | 参数类型          | 参数描述             |
|----------------------|---------------|------------------|
| message              | String        | 消息提示             |
| data                 | Object        | 数据对象             |
| data.log_id          | String        | 唯一的log id，用于问题定位 |
| data.result          | Array[Object] | 菜品识别结果数组         |
| data.result.landmark | String        | 地标名称，无法识别则返回空字符串 |
| code                 | Integer       | 返回类型，0.成功，其他：失败  |

# 车辆检测

## 方法名

vehicleDetect

传入单帧图像，检测图片中所有机动车辆，返回每辆车的类型和坐标位置，可识别小汽车、卡车、巴士、摩托车、三轮车5类车辆，并对每类车辆分别计数，同时可定位小汽车、卡车、巴士的车牌位置，支持指定矩形区域的车辆检测与数量统计。  
当前主要适用于普通监控场景，如道路、停车场等，无人机高空拍摄的图片，因车辆目标较小，识别效果可能欠佳

## 用法

- 用法如下：

```
module.vehicleDetect({
 //识别的图片地址或本地文件路径
 url: "https://img2.baidu.com/it/u=3814656217,2948212349&fm=253&app=138&f=JPE
 //只统计该矩形区域内的车辆数，缺省时为全图统计。
 //逗号分隔，如‘x1,y1,x2,y2,x3,y3...xn,yn’，按顺序依次给出每个顶点的x、y坐标（默认尾点
 //服务会做范围（顶点左边需在图像范围内）及个数校验（数组长度必须为偶数，且为4个顶点）；目前
 area: ""
}, res => {
 console.log(res);
})
```

- 参数说明

| 参数名 | 参数类型   | 是否必填 | 默认值 | 参数描述           |
|-----|--------|------|-----|----------------|
| url | String | 是    | 无   | 识别的图片地址或本地文件路径 |

| 参数名  | 参数类型   | 是否必填 | 默认值 | 参数描述                                                                                                                                                                                         |
|------|--------|------|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| area | String | 否    | 无   | 只统计该矩形区域内的车辆数，缺省时为全图统计。逗号分隔，如 'x1,y1,x2,y2,x3,y3...xn,yn'，按顺序依次给出每个顶点的x、y坐标（默认尾点和首点相连），形成闭合矩形区域。服务会做范围（顶点左边需在图像范围内）及个数校验（数组长度必须为偶数，且为4个顶点）；目前只支持单个矩形区域。坐标取值不能超过图像宽度和高度，比如1280的宽度，坐标值最大到1279。 |

## 回调

- 示例

```
{
 "data": {
 "vehicle_info": [
 {
 "type": "car",
 "location": {
 "left": 321,
 "height": 192,
 "width": 247,
```

```
 "top": 446
 },
 "probability": 0.9792143
 },
 {
 "type": "truck",
 "location": {
 "left": 73,
 "height": 282,
 "width": 242,
 "top": 2
 },
 "probability": 0.9749928
 },
 {
 "type": "carplate",
 "location": {
 "left": 143,
 "height": 11,
 "width": 33,
 "top": 239
 },
 "probability": 0.9625285
 },
 {
 "type": "car",
 "location": {
 "left": 817,
 "height": 86,
 "width": 119,
 "top": 1
 },
 "probability": 0.97677976
 },
 {
 "type": "carplate",
 "location": {
 "left": 855,
 "height": 9,
 "width": 24,
 "top": 61
 },
 "probability": 0.7743159
 },
 {
 "type": "car",
 "location": {
 "left": 384,
 "height": 133,
 "width": 171,
```

```
 "top": 175
 },
 "probability": 0.97093385
 },
 {
 "type": "carplate",
 "location": {
 "left": 996,
 "height": 7,
 "width": 22,
 "top": 18
 },
 "probability": 0.6030412
 },
 {
 "type": "car",
 "location": {
 "left": 916,
 "height": 122,
 "width": 136,
 "top": 63
 },
 "probability": 0.95606184
 },
 {
 "type": "carplate",
 "location": {
 "left": 965,
 "height": 8,
 "width": 32,
 "top": 145
 },
 "probability": 0.5563764
 },
 {
 "type": "car",
 "location": {
 "left": 680,
 "height": 135,
 "width": 152,
 "top": 98
 },
 "probability": 0.94669205
 },
 {
 "type": "carplate",
 "location": {
 "left": 319,
 "height": 8,
 "width": 34,
```



```
 "top": 402
 },
 "probability": 0.18455535
 },
 {
 "type": "car",
 "location": {
 "left": 244,
 "height": 168,
 "width": 171,
 "top": 278
 },
 "probability": 0.9186057
 },
 {
 "type": "car",
 "location": {
 "left": 551,
 "height": 198,
 "width": 181,
 "top": 215
 },
 "probability": 0.86463094
 },
 {
 "type": "car",
 "location": {
 "left": 963,
 "height": 40,
 "width": 106,
 "top": 0
 },
 "probability": 0.85787076
 },
 {
 "type": "car",
 "location": {
 "left": 790,
 "height": 29,
 "width": 80,
 "top": 0
 },
 "probability": 0.16999255
 }
],
"vehicle_num": {
 "tricycle": 0,
 "motorbike": 0,
 "car": 9,
 "truck": 1,
```

```

 "carplate": 5,
 "bus": 0
 },
 "log_id": 1941216264856520200
},
"message": "",
"code": 0
}

```

- 回调说明：

| 参数名                               | 参数类型          | 参数描述                                             |
|-----------------------------------|---------------|--------------------------------------------------|
| message                           | String        | 消息提示                                             |
| data                              | Object        | 数据对象                                             |
| data.log_id                       | String        | 唯一的log id，用于问题定位                                 |
| data.vehicle_num                  | Object        | 检测到的车辆数目                                         |
| data.vehicle_num.car              | Integer       | 小汽车数量                                            |
| data.vehicle_num.truck            | Integer       | 卡车数量                                             |
| data.vehicle_num.bus              | Integer       | 巴士数量                                             |
| data.vehicle_num.motorbike        | Integer       | 摩托车数量                                            |
| data.vehicle_num.tricycle         | Integer       | 三轮车数量                                            |
| data.vehicle_num.carplate         | Integer       | 车牌的数量，小汽车、卡车、巴士才能检测到车牌                           |
| data.vehicle_info                 | Array[Object] | 每个框的具体信息                                         |
| data.vehicle_info.type            | String        | 目标物体类型，car、truck、bus、motorbike、tricycle、carplate |
| data.vehicle_info.probability     | float         | 置信度分数，取值0-1之间，越接近1说明识别准确的概率越大                    |
| data.vehicle_info.location        | Object        | 检测到的目标坐标位置                                       |
| data.vehicle_info.location .left  | String        | 目标检测框左坐标                                         |
| data.vehicle_info.location .top   | String        | 目标检测框顶坐标                                         |
| data.vehicle_info.location .width | String        | 目标检测框宽度                                          |

| 参数名                                   | 参数类型    | 参数描述            |
|---------------------------------------|---------|-----------------|
| data.vehicle_info.location<br>.height | String  | 目标检测框高度         |
| code                                  | Integer | 返回类型，0.成功，其他：失败 |

# 车型识别

## 方法名

car

识别图片中车辆的具体车型，可识别常见的3000+款车型（小汽车为主），输出车辆的型号、颜色、年份、位置信息；支持返回对应识别结果的百度百科词条信息，包含词条名称、百科页面链接、百科图片链接、百科内容简介。  
注：当前只支持单主体识别，若图片中有多个车辆，则识别目标最大的车辆。

## 用法

- 用法如下：

```
module.car({
 //识别的图片地址或本地文件路径
 url: "https://inews.gtimg.com/om_bt/0l5skcEuIqTp0C2HthSgM5K6P2zqYkNSUumIz057",
 //返回结果top n,默认5
 topNum: 3,
 //用于控制返回结果是否带有百科信息，若不输入此参数，则默认不返回百科结果；若输入此参数，会根据
 baikeNum: 2,
 //返回车辆的品牌信息，默认不返回，可选值包括,true：返回品牌,false：不返回品牌
 outputBrand: true
}, res => {
 console.log(res);
})
```

- 参数说明

| 参数名    | 参数类型   | 是否必填 | 默认值 | 参数描述           |
|--------|--------|------|-----|----------------|
| url    | String | 是    | 无   | 识别的图片地址或本地文件路径 |
| topNum | String | 否    | 无   | 返回结果top n,默认5  |

| 参数名         | 参数类型   | 是否必填 | 默认值 | 参数描述                                                           |
|-------------|--------|------|-----|----------------------------------------------------------------|
| baikeNum    | String | 否    | 无   | 用于控制返回结果是否带有百科信息，若不输入此参数，则默认不返回百科结果；若输入此参数，会根据输入的整数返回相应个数的百科信息 |
| outputBrand | String | 否    | 无   | 返回车辆的品牌信息，默认不返回，可选值包括,true：返回品牌,false：不返回品牌                    |

## 回调

- 示例

```
{
 "data": {
 "location_result": {
 "left": 120,
 "height": 571,
 "width": 657,
 "top": 427
 },
 "log_id": 1941218801237266700,
 "brand": "宝马",
 "color_result": "蓝色",
 "result": [
 {
 "name": "宝马5系",
 "year": "2024",
 "score": 0.9252591133
 }
]
 },
 "message": ""
}
```

```
"code": 0
}
```

- 回调说明：

| 参数名                                | 参数类型          | 参数描述                                             |
|------------------------------------|---------------|--------------------------------------------------|
| message                            | String        | 消息提示                                             |
| data                               | Object        | 数据对象                                             |
| data.log_id                        | String        | 唯一的log id，用于问题定位                                 |
| data.color_result                  | String        | 车身颜色。共11种颜色，分别为白色、黑色、灰色、香槟色、黄色、红色、绿色、紫色、橙色、棕色、蓝色 |
| data.brand                         | String        | 车型品牌，实例：宝马；当output_brand=true时返回                 |
| data.location_result               | Object        | 车辆在图片中的位置信息                                      |
| data.location_result.width         | String        | 车辆区域的宽度                                          |
| data.location_result.height        | String        | 车辆区域的高度                                          |
| data.location_result.left          | String        | 车辆区域离左边界的距离                                      |
| data.location_result.top           | String        | 车辆区域离上边界的距离                                      |
| data.result                        | Array[Object] | 车型识别结果数组                                         |
| data.result.name                   | String        | 车型名称，示例：宝马x6                                     |
| data.result.score                  | String        | 置信度，取值0-1，示例：0.5321                              |
| data.result.year                   | String        | 年份                                               |
| data.result.baike_info             | Object        | 对应识别结果的百科词条名称                                    |
| data.result.baike_info.baike_url   | String        | 对应识别结果百度百科页面链接                                   |
| data.result.baike_info.image_url   | String        | 对应识别结果百科图片链接                                     |
| data.result.baike_info.description | String        | 对应识别结果百科内容描述                                     |
| code                               | Integer       | 返回类型，0.成功，其他：失败                                  |