

uniapp 安卓商米果蔬识别UTS版原生插件

目 录

使用说明	1
使用方法	1.1
更新日志	1.2
常见问题	1.3
插件方法	2
申请插件所需权限	2.1
初始化	2.2
备份模型	2.3
获取Roi区域	2.4
设置Roi区域	2.5
单个果蔬学习	2.6
果蔬批量学习	2.7
果蔬识别	2.8
根据商品SKU删除学习记忆	2.9
根据图片ID删除学习记忆	2.10
清空所有学习数据	2.11
获取版本号	2.12
释放资源	2.13

使用方法

介绍

安卓商米果蔬识别UTS版原生插件集成了商米果蔬识别的sdk并对商米果蔬识别sdk进行了封装，操作方便无需关心sdk内部处理。插件集成了初始化、果蔬学习、果蔬识别等重要功能。插件UTS开发，支持uniapp和uniapp x

注意事项

1. 关于果蔬识别的模型文件配置。插件已包含sdk需要使用的模型文件。模型文件会比较大(5.33M)，如果初始化时加载的模型文件为自定义路径模型文件，可以将插件中的模型文件删除。插件中的模型文件路径为：
`uni_modules\leven-uts-sunmiFruit\utssdk\app-android\assets` 下，删除后可以减小包的体积

联系作者

关注微信公众号可联系作者



插件地址

权限

1. android.permission.READ_PHONE_STATE
2. android.permission.WRITE_EXTERNAL_STORAGE
3. android.permission.READ_EXTERNAL_STORAGE

使用方法

用法

在需要使用插件的页面加载以下代码

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

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

2025-06-22

首次发布

常见问题

商米sdk常见问题

商米sdk常见问题请参考：<https://developer.sunmi.com/docs/zh-CN/cdixeghjk491/xcqmeghjk546>

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

方法名

init

用法

- 用法如下：

```
module.init({
  // 商米申请的应用appId
  appId: "",
  // 是否强制刷新
  isForceRefresh: false,
  // 模型文件路径配置类型，可选值assets和custom
  modelPathType: "assets",
  // 存放模型文件的文件夹名称，当模型文件路径配置类型为assets时可用
  modelConfigDirName: "fruit_config",
  // 模型文件路径，绝对路径，当模型文件路径配置类型为custom时必传
  path: "/storage/emulated/0/leven_sunmi_food/fruit_config/",
  // 是否强制将assets下模型文件复制到项目中，只针对非首次安装，默认：false
  // modelPathType为assets时有效，目的是为了初始化是不会修改已经设置过的参数
  // modelPathType为custom时会默认加载path下的模型文件，不会进行复制到项目中的操作，所以当
  // modelPathType为custom时该参数无效
  isForceCopyAssetsConfig: false
}, 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
appId	String	是	无	商米申请的应用id
isForceRefresh	Boolean	否	false	是否强制从网络读取token
modelPathType	String	否	assets	模型文件路径配置类型，可选值assets和custom

参数名	参数类型	是否必填	默认值	参数描述
modelConfigDir Name	String	否	fruit_config	存放模型文件的 文件夹名称，当 模型文件路径配 置类型为assets 时可用
path	String	否	无	模型文件路径， 绝对路径，当模 型文件路径配置 类型为custom时 必传
isForceCopyAss etsConfig	Boolean	否	false	是否强制将 assets下模型文 件复制到项目 中，只针对非首 次安装，默认： false， modelPathType 为assets时有 效，目的是为了 初始化是不会修 改已经设置过的 参数， modelPathType 为custom时会默 认加载path下的 模型文件，不会 进行复制到项目 中的操作，所以 当 modelPathType 为custom时该参 数无效

回调

- 示例

初始化

```
{
  "message": "",
  "data": {
    "license": "Nk1yM0d5bGtTN3VKUFNxdTJxVXBjN0cySUZxSGt5ckMvTytBK1ZJZlRmNmRLUE",
  },
  "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.license	String	授权码
code	Integer	返回类型，0.成功，其他：失败

备份模型

方法名

saveConfiguration

如果是自定义的配置文件不需要备份

用法

- 用法如下：

```
module.saveConfiguration({
  //需要备份到的目录，只支持备份初始化时assets的配置文件
  //如果是自定义的配置文件不需要备份
  savePath: "/storage/emulated/0/leven_sunmi_food/",
  //存放模型文件的文件夹名称，需要跟初始化时的参数一致
  modelConfigDirName: "fruit_config",
}, 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
savePath	String	是	无	需要备份到的目录，只支持备份初始化时assets的配置文件
modelConfigDirName	String	否	fruit_config	存放模型文件的文件夹名称，需要跟初始化时的参数一致

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

获取Roi区域

方法名

getRoi

由于不同设备需要设置不同的Roi区域，因此关于Roi区域参数的更改不会保存在food_config文件夹下的config.json中，未做持久化。因此，开发人员需在sdk初始化时设置参数。

用法

- 用法如下：

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

- 参数说明

无

回调

- 示例

```
{
  "message": "",
  "data": {
    "x2": -1,
    "y2": -1,
    "x1": -1,
    "y1": -1
  },
  "code": 0
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示

参数名	参数类型	参数描述
data	Object	数据对象
data.x1	Integer	roi区域第一个点横坐标
data.y1	Integer	roi区域第一个点纵坐标
data.x2	Integer	roi区域第二个点横坐标
data.y2	Integer	roi区域第二个点纵坐标
data.x3	Integer	roi区域第三个点横坐标
data.y3	Integer	roi区域第三个点纵坐标
data.x4	Integer	roi区域第四个点横坐标
data.y4	Integer	roi区域第四个点纵坐标
code	Integer	返回类型，0.成功，其他：失败

设置Roi区域

方法名

setRoi

用法

- 用法如下：

```
module.setRoi({
  x1: 100,
  y1: 100,
  x2: 100,
  y2: 100
}, 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
x1	Integer	否	无	roi区域第一个点横坐标
x2	Integer	否	无	roi区域第二个点横坐标
x3	Integer	否	无	roi区域第三个点横坐标
x4	Integer	否	无	roi区域第四个点横坐标
y1	Integer	否	无	roi区域第一个点纵坐标
y2	Integer	否	无	roi区域第二个点纵坐标
y3	Integer	否	无	roi区域第三个点纵坐标
y4	Integer	否	无	roi区域第四个点纵坐标

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

单个果蔬学习

方法名

imageLearn

用法

- 用法如下：

```
module.imageLearn({
  //图片路径或网络地址
  url: "/storage/emulated/0/food/eggplant/eggplant2.png",
  //商品标识
  skuID: "紫茄子"
}, 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	图片路径或网络地址
skuID	String	是	无	商品标识

回调

- 示例

```
{
  "message": "",
  "data": {
    "skuID": "紫茄子",
    "supNum": 0,
    "name": "",
    "imageID": "0ba742f7-0000-0000-2840-000000000000"
  },
}
```

```
"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.skuID	String	商品的唯一标识
data.imageID	String	学习后sdk生成的图片id
data.supNum	Integer	-
data.name	String	名称
code	Integer	返回类型，0.成功，其他：失败

果蔬批量学习

方法名

batchImageLearn

用法

- 用法如下：

```
module.batchImageLearn({
  list: []
}, 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
list	Array[Object]	是	无	批量学习列表
list.url	String	是	无	图片路径或网络地址
list.skuID	String	是	无	商品标识

回调

- 示例

```
{
  "message": "",
  "data": {
    "status": "learnProgress",
    "failed": 0,
    "food": {
      "skuID": "紫茄子",
      "url": "http://www.yeyuboke.com/uniplugin/food/eggplant/eggplant1.png",
      "imageID": "581fe07c-0000-0000-2d88-000000000000"
    },
  },
  "progress": 0.09090909,
```

```

    "count": 11,
    "success": 1
  },
  "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.url	String	当前注册的url
data.status	String	当前注册的状态，learnStart：开始学习，learnProgress：学习中，learnComplete：注册完成
data.food	Object	学习的菜品
data.food.skuID	String	唯一标识
data.food.url	String	学习图片的路径
data.food.imageID	String	sdk生成的图片id
data.failed	Integer	当前学习失败的数量
data.progress	Float	当前学习的进度
data.count	Integer	学习的所有数量
data.success	Integer	当前已成功学习的数量
code	Integer	返回类型，0.成功，其他：失败

果蔬识别

方法名

recognize

用法

- 用法如下：

```
module.recognize({
  //图片路径或网络地址
  // url: "/storage/emulated/0/food/red_apple/red_apple.png",
  url: "http://www.yeyuboke.com/uniplugin/food/eggplant/eggplant2.png",
  //top N 数量
  topN: 3
}, 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	图片路径或网络地址
topN	String	否	1	top N 数量

回调

- 示例

```
{
  "message": "",
  "data": {
    "productInfos": [
      {
        "skuCode": "紫茄子",
        "name": "",
        "score": 1
      }
    ]
  }
}
```

```
    ],
    "sessionId": "5346b07b-0000-0000-2840-000000000000",
    "scores": [
      1
    ],
    "skuCodes": [
      "紫茄子"
    ]
  },
  "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.scores	Array[float]	长度等于skuCodes，对应每个skuCode的置信度
data.skuCodes	Array[String]	商品标识码SKU ID
data.sessionID	String	sessionId
data.productInfos	Array[Object]	商品信息
data.productInfos.skuCode	String	商品标识码SKU ID
data.productInfos.name	String	名称
data.productInfos.score	float	置信度
code	Integer	返回类型，0.成功，其他：失败

根据商品SKU删除学习记忆

方法名

deleteBySkuID

用法

- 用法如下：

```
module.deleteBySkuID({
  skuID: "紫茄子"
}, 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
skuID	String	是	无	商品标识

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

根据图片ID删除学习记忆

方法名

deleteByImageID

用法

- 用法如下：

```
module.deleteByImageID({
  //商品学习时返回的imageID
  imageID: "f0d376d9-ffff-0fff-0955-000000000000"
}, 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
imageID	String	是	无	商品学习时返回的imageID

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

根据图片ID删除学习记忆

参数名	参数类型	参数描述
code	Integer	返回类型，0.成功，其他：失败

清空所有学习数据

方法名

clearAllProductFeature

用法

- 用法如下：

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

- 参数说明

无

回调

- 示例

```
{  
  "data": {},  
  "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
code	Integer	返回类型，0.成功，其他：失败

获取版本号

方法名

getVersion

用法

- 用法如下：

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

- 参数说明

无

回调

- 示例

```
{
  "message": "",
  "data": {
    "version": "v2.2.3@a2f64a6"
  },
  "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.version	String	版本号
code	Integer	返回类型，0.成功，其他：失败

释放资源

方法名

release

用法

- 用法如下：

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

- 参数说明

无

回调

- 示例

```
{
  "data": {},
  "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
code	Integer	返回类型，0.成功，其他：失败