# 数据平台模型SDK文档

## 卡证分类

### 卡证分类模型性能

目前卡证分类模型在测试环境中分类准确率达99.9%，模型会将输入的图像分为9个类别分别为：银行卡、驾驶证主页、驾驶证副页、身份证正面、身份证背面、行驶证主页、行驶证副页正面、行驶证副页背面与其他，各类别样图如下所示。



### 2、NTS-net简介

卡证分类模型采用NTS-net一种细粒度分类模型，该模型于2018年发表在ECCV，Learning to Navigate for Fine-grained Classiﬁcation中。

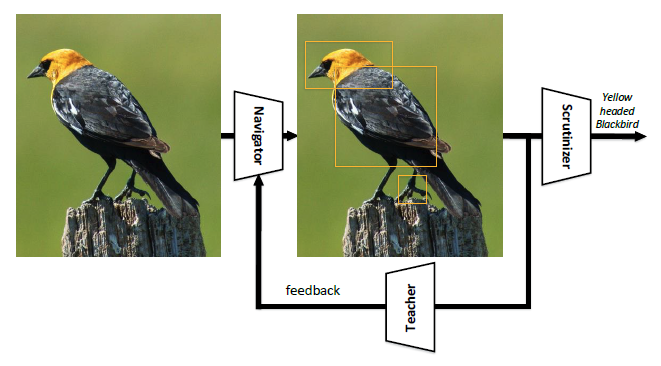
#### 细粒度分类与普通分类的区别

什么叫细粒度分类？举个例子，普通的分类任务为猫狗分类，即向模型输入一张图片，模型返回预测类型为猫还是狗。而细粒度分类任务为哈士奇与阿拉斯加分类，哈士奇与阿拉斯加同属于狗，且外形相似但属于不同类别。即细粒度分类任务需要解决的分类任务类间差距会更小且更具有挑战性。

#### NTS-net结构简介

NTS-net主要由三个模块组成，分别为Navigator Agent、Teacher Agent、Scrutinizer Agent。

Navigator Agent主要用于定位最具信息性的区域，如下图所示，当一张鸟的图片输入Navigator Agent之后，Navigator Agen会定位到鸟的头部、翅膀和脚部区域，认为这三个区域所包含的信息最能够将此种鸟类与其他鸟类区分开来。



Teacher Agent主要用于优化Navigator Agent的性能，使其能够更加精准的定位图片中最具信息性的区域。

Scrutinizer Agent 提取Navigator Agent给出的最具信息性区域的特征，并将该特征与全图特征相结合，最终使用这些特征信息预测出图像类别。

### 3、模型服务返回码

#### 正确返回：

|  |  |  |
| --- | --- | --- |
| Code | Message | 说明 |
| 200 | Success | 接口成功调用且成功返回预测结果 |

#### 错误码：

|  |  |  |
| --- | --- | --- |
| Code | Message | 说明 |
| 1000 | Unknown error | 接口异常 |

### 4、模型服务接口说明

|  |  |
| --- | --- |
| url | http://10.3.201.206:35600/classify/card\_9cls |
| 接口说明 | 8类卡证分类模型，额外增加others类别，当输入图片不属于8类中的一种时返回others；9个类别与返回值对应关系为：  0 银行卡(bank\_card)  1 驾驶证主页(driver\_license\_master)  2 驾驶证副页(driver\_license\_attachment)  3 身份证正面(id\_card\_front)  4 身份证背面(id\_card\_back)  5 行驶证主页(vehicle\_license\_master)  6 行驶证副页正面(vehicle\_license\_attachment\_front)  7 行驶证副页背面(vehicle\_license\_attachment\_back)  8 其他(others) |
| 入参 | img: 输入图像，base64 |
| 正确返回 | Pred中class为预测类别，score为该类别置信度 ，cost\_time为模型花费时间；  {"code": 200,  "message": "success",  "pred": {"class": 1,  "score": 0.9999971389770508,  "cost\_time": 0.9907121658325195}  } |

## 图像反欺诈

### 1 图像检索模型性能

目前对阿里提供的通用图像检索与自研的图像检索性能做了测试，结果如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模型名称 | 图像库大小 | 测试图数量 | top1准确率 | top3准确率 |
| 阿里通用图像搜索 | 49 | 19 | 10/19=0.526 | 13/19=0.684 |
| 自研 | 49 | 19 | 12/19=0.631 | 16/19=0.842 |

模型使用LSH局部敏感哈希算法，即使图片库中的图片数量达到千万级别也可以实现实时检索。在做图像检索时，首先需要将图片库输入模型，然后将待检索的图片输入模型，之后模型会从图像库中检索出与其最相似的图片，其检索过程如下图所示：



### 2 模型简介

模型主要由两个部分构成，分别为特征提取与特征查询。在特征提取阶段会使用Resnet50抽取2048维的特征向量，之后LSH算法会将这2048维特征向量通过哈希映射到对应的桶中。由于在高维数据空间中的两个相邻的数据被映射到低维数据空间中后，将会有很大的概率任然相邻，所以两张相似的图像经过处理之后其最终的特征在很大概率上会位于同一个桶中。在图像检索阶段，待检索的图像也会使用Resnet50抽取2048维特征，然后通过哈希可找到待检索的桶。通过与桶中的特征一一对比即可找到与其最相似的那一张图，当图像库越大时其分桶数量也会越多，所以即使当图像库的图片数量达到千万级别，每个桶中的图像数量依然基本保持不变，所以检索时间也基本不会变化。

### 3、模型服务返回码

#### 正确返回：

|  |  |  |
| --- | --- | --- |
| Code | Message | 说明 |
| 200 | Success | 接口成功调用且成功返回预测结果 |

#### 错误码：

|  |  |  |
| --- | --- | --- |
| Code | Message | 说明 |
| 1000 | Unknown error | 接口异常 |
| 1100 | Error, can't extract ndarray from xxx.jpg's base64; | 不能从base64值中提取出图像信息 |
| 1101 | Error, the type of xxx.jpg's info is int not str; | 图像名称变量类不为str |
| 1102 | Error, invalid image, max(image.size) / min(image.size) > 5; | 无效图像，图像长宽比过大； |

### 4、模型服务接口说明

#### 增加图像特征

|  |  |
| --- | --- |
| url | http://10.3.201.204:35600/retrieval/add\_feature |
| 接口说明 | 调用该接口，首先会用Resnet50抽取图像特征，并将图像特征存入数据库中；调用后返回成功插入数据库的图片名称以及数据库中的图片id和异常的图片名称。 |
| 入参 | 以json形式传值,img\_name为图像名称，其中base64为图片用base64编码之后的数据;  info为该图像额外添加的说明信息等，需要以字符串形式传入；  {  "img\_name": "24.jpg",  "base64": "/9j/4AAQSkZJRgABAQA...",  "info": "24.jpg"  } |
| 正确返回 | Id 为图像特征在数据库中的id；  img\_name 为图像名称；  {  "message": "Success",  "data": {  "id": 264,  "img\_name": "24.jpg"  },  "code": 200  } |

#### 图像特征局部检索

|  |  |
| --- | --- |
| url | http://10.3.201.204:35600/retrieval/retrieval |
| 接口说明 | 调用该接口首先会根据传入的lib\_id\_list 构建待查询的图像特征库，然后找出top\_n 个最相似的图像特征。 |
| 入参 | img\_name 图像名称；  base64 图像base64值；  top\_n 寻找top\_n个最相似的图像；  lib\_id\_list 待查询的图像特征id；  {  "img\_name": "0.jpg",  "base64": "/9j/4AAQSkZJRgA...",  "top\_n": 3,  "lib\_id\_list": [163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180]  } |
| 正确返回 | img\_name 图像名称；  match\_list 匹配信息；  id 所匹配特征在数据库中的id；  score 置信度；  {  "message": "Success",  "data": {  "img\_name": "9.jpg",  "match\_list": [{  "id": 260,  "score": 0.617622125057031  }, {  "id": 240,  "score": 0.6512413239008998  }, {  "id": 252,  "score": 0.6595219688235  }]  },  "code": 200  } |

#### 图像特征全局检索

|  |  |
| --- | --- |
| url | http://10.3.201.204:35600/retrieval/global\_retrieval |
| 接口说明 | 调用该接口会提取图像特征，并与全体特征库中特征进行比对，找出top\_n 个最相似的图像特征。 |
| 入参 | img\_name 图像名称；  base64 图像base64值；  top\_n 寻找top\_n个最相似的图像；  {  "img\_name": "9.jpg",  "base64": "/9j/4AAQSkZ...",  "top\_n": 3  } |
| 正确返回 | img\_name 图像名称；  match\_list 匹配信息；  id 所匹配特征在数据库中的id；  score 置信度；  {  "message": "Success",  "data": {  "img\_name": "9.jpg",  "match\_list": [{  "id": 260,  "score": 0.617622125057031  }, {  "id": 240,  "score": 0.6512413239008998  }, {  "id": 252,  "score": 0.6595219688235  }]  },  "code": 200  } |