ALIBABA CLOUD

Alibaba Cloud

机器学习PAI

PAI-ModelHub公共模型仓库

文档版本：20201030

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
Courier字体	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.图像智能处理类模型 ---------- 05
2.自然语言处理（NLP）类模型 ---------- 10

1.图像智能处理类模型

PAI提供多种已经训练好的图像智能类模型供您使用，包括通用图像分类模型、通用图像检测模型、图像语义分割模型及图像实例分割模型。

通用图像分类模型

- 模型介绍

  基于ImageNet数据集训练的图像分类模型，该模型采用ResNet，详情请参见Deep Residual Learning for Image Recognition。

- 输入格式

  输入数据为JSON格式字符串，包含image字段，对应的value为图片内容的Base 64编码。

```
{
 "image": "图像文件内容的Base 64编码"
}
```

- 输出格式

  输出数据为JSON格式字符串，包含的字段如下表所示。

字段	描述	Shape	Type
class	类别ID	[]	INT32
class_name	类别名称	[]	STRING
class_probs	所有类别概率	[num_classes]	Dict[STRING, FLOAT]
request_id	请求的唯一标识	[]	STRING
success	请求是否成功	[]	BOOL
error_code	请求错误码	[]	INT
error_msg	请求错误信息	[]	STRING

  输出数据的示例如下。

```
{
"class": 3,
"class_name": "coho4",
"class_probs": {"coho1": 4.028851974258174e-10,
      "coho2": 0.48115724325180054,
      "coho3": 5.116515922054532e-07,
      "coho4": 0.5188422446937221},
 "request_id": "9ac294a4-f387-4c48-b640-d2c6d41fcbee",
 "success": true
}
```

- 测试数据

  下载通用图像分类模型的测试数据

通用图像检测模型

- 模型介绍

  模型采用Faster R-CNN，详情请参见Towards Real-Time Object Detection with Region Proposal Networks。训练数据集为COCO。

- 输入格式

  输入数据为JSON格式字符串，包含image字段，对应的value为图片内容的Base 64编码。

```
{
 "image": "图像文件内容的Base 64编码"
}
```

- 输出格式

  输出数据为JSON格式字符串，包含的字段如下表所示。

字段	描述	Shape	Type
detection_boxes	检测到的目标框[y1, x1, y2,x2]，其坐标顺序为[top, left, bottom, right]。	[num_detections, 4]	FLOAT
detection_scores	目标检测概率。	num_detections	FLOAT
detection_classes	目标区域类别ID。	num_detections	INT
detection_class_names	目标区域类别名称。	num_detections	STRING
request_id	请求的唯一标识。	[]	STRING
success	请求是否成功。	[]	BOOL
error_code	请求错误码。	[]	INT

字段	描述	Shape	Type
error_msg	请求错误信息。	[]	STRING

输出数据的示例如下。

```
{
 "detection_boxes": [[243.5308074951172, 197.69570922851562, 385.59625244140625, 247.7247772216797
], [292.1929931640625, 114.28043365478516, 571.2748413085938, 165.09771728515625]],
 "detection_scores": [0.9942291975021362, 0.9940272569656372],
 "detection_classes": [1, 1],
 "detection_classe_names": ["text", "text"],
 "request_id": "9ac294a4-f387-4c48-b640-d2c6d41fcbee",
 "success": true
}
```

- 测试数据

  下载通用图像检测模型的测试数据

图像语义分割模型

- 模型介绍

  模型采用DeepLab V3，详情请参见Rethinking Atrous Convolution for Semantic Image Segmentation。训练数据集为Pascal_Voc。

- 输入格式

  输入数据为JSON格式字符串，包含image字段，对应的value为图片内容的Base 64编码。

```
{
 "image": "图像文件内容的Base 64编码"
}
```

- 输出格式

  输出数据为JSON格式字符串，包含的字段如下表所示。

字段	描述	Shape	Type
probs	分割像素点概率	[output_height, output_width]	FLOAT
preds	分割像素类别ID	[output_height, output_widths]	INT
request_id	请求的唯一标识	[]	STRING
success	请求是否成功	[]	BOOL
error_code	请求错误码	[]	INT

字段	描述	Shape	Type
error_msg	请求错误信息	[]	STRING

输出数据的示例如下。

```
{
 "probs" : [[[0.8, 0.8], [0.6, 0.7]],[[0.8, 0.5], [0.4, 0.3]]],
 "preds" : [[1,1], [0, 0]],
  "request_id": "9ac294a4-f387-4c48-b640-d2c6d41fcbee",
  "success": true
}
```

- 测试数据

  下载图像语义分割模型的测试数据

图像实例分割模型

- 模型介绍

  模型采用Mask R-CNN，详情请参见Mask R-CNN。训练数据集为COCO。

- 输入格式

  输入数据为JSON格式字符串，包含image字段，对应的value为图片内容的Base 64编码。

```
{
 "image": "图像文件内容的Base 64编码"
}
```

- 输出格式

  输出数据为JSON格式字符串，包含的字段如下表所示。

字段	描述	Shape	Type
detection_boxes	检测到的目标框[y1, x1, y2,x2]，其坐标顺序为[top, left, bottom, right]。	[num_detections, 4]	FLOAT
detection_scores	目标检测概率。	num_detections	FLOAT
detection_classes	目标区域类别ID。	num_detections	INT
detection_class_names	目标区域类别名称。	num_detections	STRING
detection_masks	目标区域的Mask。	[num_detections, image_height, image_width]	BOOL
request_id	请求的唯一标识。	[]	STRING

字段	描述	Shape	Type
success	请求是否成功。	[]	BOOL
error_code	请求错误码。	[]	INT
error_msg	请求错误信息。	[]	STRING

输出数据的示例如下。

```
{
 "detection_boxes": [[243.5308074951172, 197.69570922851562, 385.59625244140625, 247.7247772216797
], [292.1929931640625, 114.28043365478516, 571.2748413085938, 165.09771728515625]],
 "detection_scores": [0.9942291975021362, 0.9940272569656372],
 "detection_classes": [1, 1],
 "detection_classe_names": ["text", "text"],
 "detection_masks": [[[1,1], [0, 0]], [[0,1], [1, 1]]],
  "request_id": "9ac294a4-f387-4c48-b640-d2c6d41fcbee",
  "success": true
}
```

- 测试数据

  下载图像实例分割模型的测试数据

2.自然语言处理（NLP）类模型

PAI提供多种已经训练好的自然语言处理类模型供您使用，例如BERT文本向量化模型。

自然语言处理NLP（Natural Language Processing）是人工智能和语言学领域的分支学科，能够挖掘自然语言文本蕴含的信息和知识。常见的应用包括：

- 文本分类，适用于新闻标签打标、情感分析、垃圾邮件识别及商品评价分类等场景。
- 文本匹配，适用于问答匹配、句子相似度匹配、自然语言推理及对话检索等场景。
- 序列标注，适用于命名实体识别NER及情感词抽取等场景。
- 特征提取，提取的文本特征可以在文本领域或结合其他领域（例如计算机视觉）进行后续操作。

PAI-ModelHub提供以上服务的部署流程，并提供BERT特征提取模型供您使用。

BERT文本向量化模型

- 模型介绍

  除了对BERT预训练完成的模型进行Finetune外，BERT生成的向量本身也很有价值。例如，将BERT作为一个特征提取器，输入一个文本序列，输出一个向量序列。还可以先对CLS输出的向量进行Dense，再将该向量作为整个句子的句向量。

  

  当用户给定一个句子S，该组件会自动将其分词为Subtoken形式S = [CLS, tok1, tok2, ..., tokN, SEP]，并给出以下三种类型结果供用户选择：

  - pool_output：对句子进行编码后的向量，即图中的C'。
  - first_token_output：即图中的C。
  - all_hidden_outputs：即图中的[C, T1, T2, ..., TN, TSEP]。

- 输入格式

  输入数据为JSON格式字符串，包含如下字段：

  - first_sequence：对应的value为第一个文本字符串。
  - second_sequence：对应的value为第二个文本字符串（可以为空）。
  - sequence_length：对应文本的截断长度，最大为512。
  - output_schema：返回的向量选择。如果是多选，则使用英文逗号（,）分隔。可选项分别为pool_output、first_token_output及all_hidden_outputs。

  ```
  {
    "first_sequence": "第一个文本字符串",
    "second_sequence": "第二个文本字符串（可以为空）",
    "sequence_length": 128
    "output_schema": "返回的向量选择"
  }
  ```

- 输出格式

  输出数据为JSON格式字符串，包含的字段如下表所示。

字段	描述	Shape	Type
pool_output	英文逗号（,）分隔的768维向量，表示对句子进行编码后的向量，即图中的 C'。	[]	STRING
first_token_output	英文逗号（,）分隔的768维向量，即图中的C。	[]	STRING
all_hidden_outputs	768*sequence_length维向量，其中向量由英文逗号（,）分隔，序列由英文分号（;）分隔，即图中的[C, T1, T2, ..., TN, TSEP]。	[]	STRING

- 测试数据

```
# 输入。
{
  "first_sequence": "双十一花呗提额在哪",
  "second_sequence": "",
  "sequence_length": 128,
  "output_schema": "pool_output,first_token_output,all_hidden_outputs"
}

# 输出。
{
  "pool_output": "0.999340713024,...,0.836870908737",
  "first_token_output": "0.789340713024,...,0.536870908737",
  "all_hidden_outputs": "0.999340713024,...,0.836870908737;...;0.899340713024,...,0.936870908737"
}
```