

ModelArts

数据处理用户指南

文档版本 01
发布日期 2021-09-30



版权所有 © 华为技术有限公司 2021。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目 录

| | |
|---------------------------|----------|
| 1 数据处理简介..... | 1 |
| 2 使用预置的数据处理工具..... | 2 |
| 2.1 创建数据处理任务..... | 2 |
| 2.2 管理和查看数据处理任务..... | 4 |
| 3 数据处理预置算子说明..... | 6 |
| 3.1 数据校验..... | 6 |
| 3.2 数据清洗..... | 9 |
| 3.3 数据选择..... | 11 |
| 3.3.1 数据去重..... | 12 |
| 3.3.2 数据去冗余..... | 13 |
| 3.4 数据增强..... | 15 |
| 3.4.1 数据扩增..... | 15 |
| 3.4.2 数据生成..... | 20 |
| 3.4.3 数据域迁移..... | 22 |

1 数据处理简介

ModelArts平台提供的数据处理功能，基本目的是从大量的、杂乱无章的、难以理解的数据中抽取或者生成对某些特定的人们来说是有价值、有意义的数据。当数据采集和接入之后，数据一般是不能直接满足训练要求的。为了保障数据质量，以免对后续操作（如数据标注、模型训练等）带来负面影响，开发过程通常需要进行数据处理。

常见的数据处理类型有以下四种：

- **数据校验**：通常数据采集后需要进行校验，保证数据合法。
数据校验是指对数据可用性的基本判断和验证的过程。通常，我们采集的数据或多或少都会有很多格式问题，无法被进一步处理。以图像识别为例，用户经常会从网上找一些图片用于训练，但是其质量难以保证，有可能图片的名字、路径、后缀名都不满足训练算法的要求；图片也可能有部分损坏，造成无法解码、无法被算法处理的情况。因此，数据校验非常重要，可以帮助人工智能开发者提前发现数据问题，有效防止数据噪声造成的算法精度下降或者训练失败问题。
- **数据清洗**：数据清洗是指对数据进行去噪、纠错或补全的过程。
数据清洗是在数据校验的基础上，对数据进行一致性检查，处理一些无效值。例如在深度学习领域，可以根据用户输入的正样本和负样本，对数据进行清洗，保留用户想要的类别，去除用户不想要的类别。
- **数据选择**：数据选择一般是指从全量数据中选择数据子集的过程。
数据可以通过相似度或者深度学习算法进行选择。数据选择可以避免人工采集图片过程中引入的重复图片、相似图片等问题；在一批输入旧模型的推理数据中，通过内置规则的数据选择可以进一步提升旧模型精度。
- **数据增强**：
 - 数据扩增**通过简单的数据扩增例如缩放、裁剪、变换、合成等操作直接或间接的方式增加数据量。
 - 图像生成**应用相关深度学习模型，通过对原数据集进行学习，训练生成新的数据集的方式增加数据量。
 - 图像域迁移**应用相关深度学习模型，通过对原域和目标域数据集进行学习，训练生成原域向目标域迁移的数据。

2 使用预置的数据处理工具

2.1 创建数据处理任务

您可以创建一个数据处理任务，对已有的数据进行数据校验、数据清洗、数据选择或者数据增强操作。

前提条件

- 数据已准备完成：已经创建数据集或者已经将数据上传至OBS
- 确保您使用的OBS与ModelArts在同一区域

创建数据处理任务

1. 登录ModelArts管理控制台，在左侧的导航栏中选择“数据管理>数据处理”，进入“数据处理”页面。
2. 在“数据处理”页面，单击“创建”进入“创建数据处理”页面。
3. 在创建数据处理页面，填写相关算法参数。

- a. 填写基本信息。基本信息包括“名称”、“版本”和“描述”。其中“版本”信息由系统自动生成，按“V0001”、“V0002”规则命名，用户无法修改。

您可以根据实际情况填写“名称”和“描述”信息。

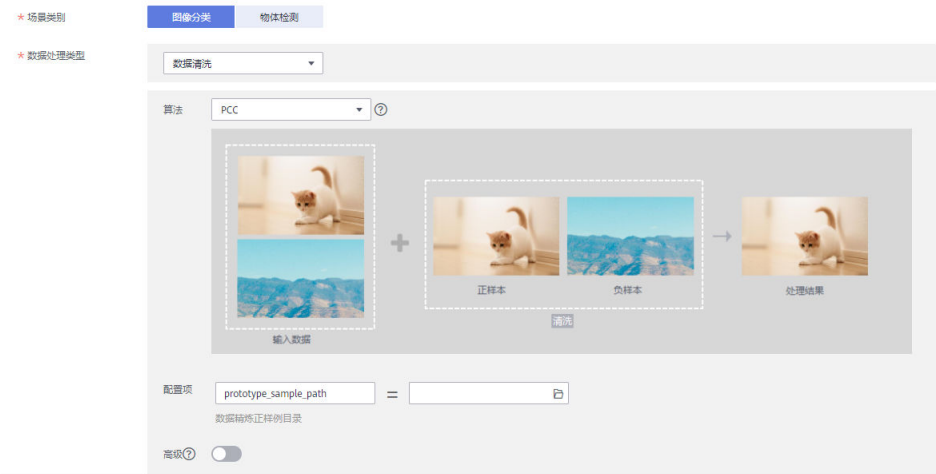
图 2-1 创建数据处理基本信息

The screenshot shows a form titled '创建数据处理任务' (Create Data Processing Task). The '基本信息' (Basic Information) section contains three fields: '名称' (Name) with the value 'PRE-test', '版本' (Version) with the value 'V0001' and a note '版本信息为自动生成' (Version information is auto-generated), and '描述' (Description) with a placeholder '请输入描述' (Please enter description). A character count '0/256' is visible at the bottom right of the description field.

- b. 设置场景类别。场景类别当前支持“图像分类”和“物体检测”。
- c. 设置数据处理类型。数据处理类型支持“数据清洗”、“数据校验”、“数据选择”和“数据增强”。

针对不同的数据处理类型，您需要填写相应算子的设置参数，算子的详细参数参见[数据处理预置算子说明](#)。

图 2-2 设置场景类别和数据处理类型



- d. 设置输入与输出。需根据实际数据情况选择“数据集”或“OBS目录”。设置为“数据集”时，需填写“数据集名称”和“数据集版本”；设置为“OBS目录”时，需填写正确的OBS路径。

图 2-3 输入输出设置-数据集



图 2-4 输入输出设置-OBS 目录



e. 确认参数填写无误后，单击“创建”，完成数据处理任务的创建。

2.2 管理和查看数据处理任务

删除数据处理任务

当已有的数据处理任务不再使用时，您可以删除数据处理任务。

处于“完成”、“失败”、“已停止”、“运行失败”、“部署中”状态的训练作业，您可以单击操作列的“删除”，删除对应的数据处理任务。

查看数据处理任务详情

1. 登录ModelArts管理控制台，在左侧的导航栏中选择“数据管理>数据处理”，进入“数据处理”页面。
2. 在数据处理列表中，单击数据处理任务名称，进入数据处理任务的版本管理页面。您可以在该页面进行数据处理任务的“修改”与“删除”。

图 2-5 数据处理版本管理页面



3. 您可以在版本管理页面，通过切换页签查看“配置信息”、“日志”和“结果展示”。

图 2-6 日志页面

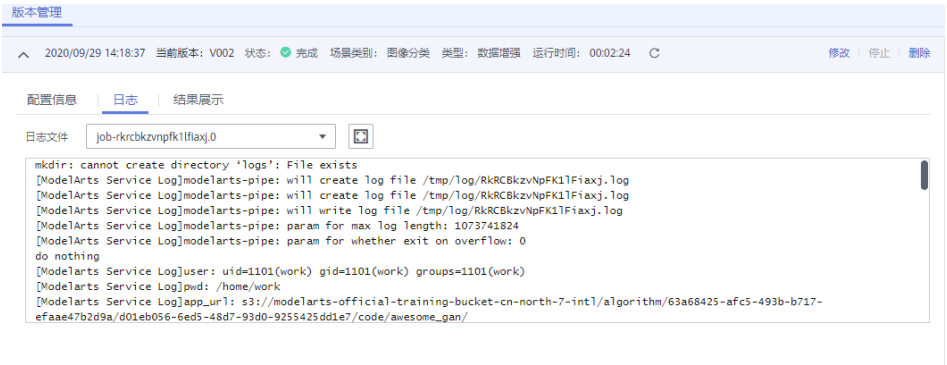
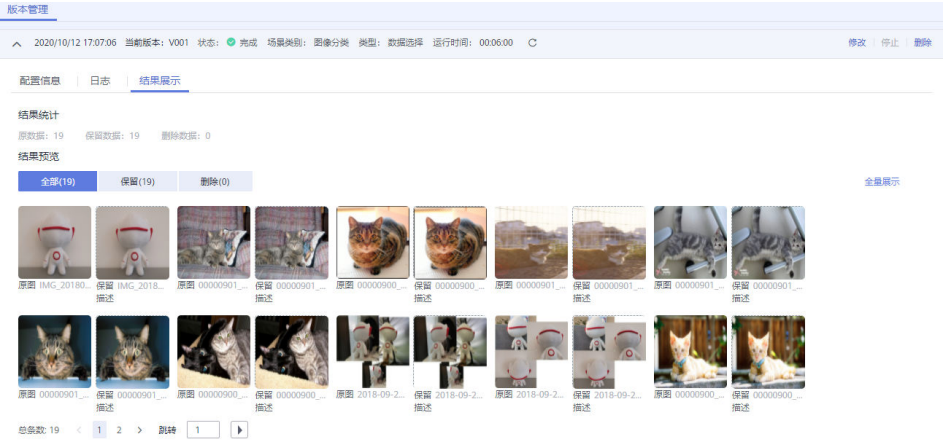


图 2-7 结果展示页面



3 数据处理预置算子说明

3.1 数据校验

MetaValidation 算子概述

ModelArts的数据校验通过MetaValidation算子实现。当前ModelArts支持jpg、jpeg、bmp、png四种图片格式。物体检测场景支持xml标注格式，不支持“非矩形框”标注。针对您提供的数据集，MetaValidation算子支持对图片和xml文件进行数据校验：

表 3-1 图片类数据校验

| 异常情况 | 处理方案 |
|---------------------------|-----------------|
| 图片本身损坏无法解码 | 过滤掉不能解码的图片 |
| 图片通道可能是1通道、2通道，不是常用的3通道 | 转换图片成RGB三通道 |
| 图片格式不在ModelArts支持的格式范围内 | 转换图片格式至jpg格式 |
| 图片后缀与实际格式不符，但格式在MA支持的格式内 | 后缀转换成与实际格式一致 |
| 图片后缀与实际格式不符，且格式不在MA支持的格式内 | 转换图片格式至jpg格式 |
| 图片分辨率过大 | 宽、高按指定大小同比例进行裁剪 |

表 3-2 标注类文件数据校验

| 异常情况 | 处理方案 |
|------------------|---------|
| xml结构残缺，无法解析 | 过滤xml文件 |
| xml中没有标注“object” | 过滤xml文件 |

| 异常情况 | 处理方案 |
|----------------------------------|-------------------------------------|
| xml中没有矩形框“bndbox” | 过滤xml文件 |
| 某些标注“object”中没有矩形框“bndbox” | 过滤标注“object” |
| 图片经过裁剪后，xml文件中宽高不符 | 修改错误宽高参数为图片真实宽高 |
| xml中没有“width”、“height”字段 | 根据图片真实宽高补全xml中的“width”、“height”字段和值 |
| 图片经过裁剪后，xml中矩形框“bndbox”大小不符 | 按图片裁剪比例缩放xml文件中“bndbox”值 |
| xml中矩形框“bndbox”宽或高值过小，显示为一条线 | 矩形框宽或高差值小于2，移除当前“object” |
| xml中矩形框“bndbox”最小值大于最大值 | 移除当前“object” |
| 矩形框“bndbox”超出图片边界，且超出部分占框面积50%以上 | 移除当前“object” |
| 矩形框“bndbox”超出图片边界，但超出部分小于框面积50% | 矩形框“bndbox”拉回到图片边界 |

说明

数据校验过程不会改动原始数据，通过校验的图片或xml文件保存在指定的输出路径下。

参数说明

表 3-3 数据校验-MetaValidation 算子参数说明

| 参数名 | 是否必选 | 默认值 | 参数说明 |
|------------------|------|-----|---|
| image_max_width | 否 | -1 | 输入图片宽度最大值，若输入图片宽度超过设定值则按比例裁剪。单位为px。 默认值 -1 表示不做裁剪。 |
| image_max_height | 否 | -1 | 输入图片长度最大值，若输入图片长度超过设定值则按比例裁剪。单位为px。 默认值 -1 表示不做裁剪。 |

输入要求

算子输入分为两种，“数据集”或“OBS目录”。

- 选择“数据集”，请从下拉框中选择ModelArts中管理的数据集及其版本。要求数据集类型与您在任务中选择的场景类别一致。

- 选择“OBS目录”，存放结构又分两种情况，“仅包含图片”或“包含图片和标注信息”。
 - “仅包含图片”：当目录下全是图片时，支持jpg、jpeg、png、bmp格式，嵌套子目录的图片也将全部读入。
 - “包含图片和标注信息”：根据不同场景类型，结构不同。

图像分类场景，其目录结构如下所示。如下目录结构，仅支持单标签场景。

```
input_path/
--label1/
----1.jpg
--label2/
----2.jpg
--./
```

物体检测场景，其目录结构如下所示。支持jpg、jpeg、png、bmp格式的图片，xml为标准的PACAL VOC格式标注文件。

```
input_path/
--1.jpg
--1.xml
--2.jpg
--2.xml
...
```

输出说明

- **图像分类**

输出数据的目录结构如下所示。

```
output_path/
--Data/
----class1/ # 若输入数据有标注信息会一并输出，class1为标注类别
-----1.jpg
-----2_checked.jpg
----class2/
-----3.jpg
-----4_checked.jpg
----5_checked.jpg
--output.manifest
```

其中manifest文件内容示例如下所示。会给每一条数据加上一个校验属性
"property":{"@modelarts:data_checked":true}

```
{
  "id": "xss",
  "source": "obs://hard_example_path/Data/fc8e2688015d4a1784dcba44d840307_14_checked.jpg",
  "property": {
    "@modelarts:data_checked": true
  },
  "usage": "train",
  "annotation": [
    {
      "name": "Cat",
      "type": "modelarts/image_classification"
    }
  ]
}
```

- **物体检测**

在输出目录下，文件结构如下所示。

```
output_path/
--Data/
----1_checked.jpg
----1_checked.xml # 若输入数据在校验过程中经过了转换，文件名会加上'_checked'
----2.jpg         # 若输入数据未经过转换，则以原来的名字保存
----2.xml
--output.manifest
```

其中manifest文件内容示例如下所示。会给每一条数据加上一个校验属性
"property":{"@modelarts:data_checked":true}

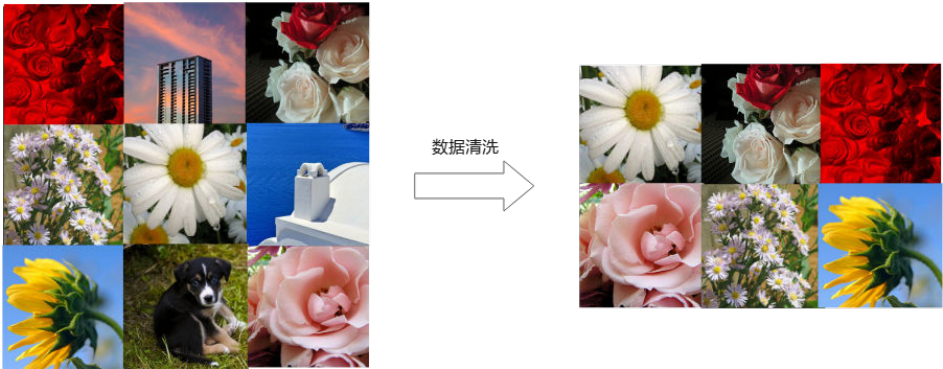
```
{
  "source": "obs://hard_example_path/Data/be462ea9c5abc09f_checked.jpg",
  "property": {
    "@modelarts:data_checked": true
  },
  "annotation": [
    {
      "annotation-loc": "obs://hard_example_path/Data/be462ea9c5abc09f_checked.xml",
      "type": "modelarts/object_detection",
      "annotation-format": "PASCAL VOC",
      "annotated-by": "modelarts/hard_example_algo"
    }
  ]
}
```

3.2 数据清洗

PCC 算子概述

ModelArts的数据清洗通过PCC算子实现。图像分类或者物体检测的数据集中可能存在非所需类别的图像，需要将这些图像去除掉，以免对标注、模型训练造成干扰。

图 3-1 PCC 算子效果



参数说明

表 3-4 数据清洗-PCC 算子参数说明

| 参数名 | 是否必选 | 默认值 | 参数说明 |
|-----------------------|------|------|---|
| prototype_sample_path | 是 | None | 数据清洗正样例目录。目录应存放正样例图片文件，算法将这些图片为正样例，对输入中的数据进行过滤，即保留与“prototype_sample_path”目录下图片相似度高的数据。 请输入一个真实存在的OBS目录，且目录下已包含提供的正样例图片，且以obs://开头。如： <i>obs://obs_bucket_name/folder_name</i> |

| 参数名 | 是否必选 | 默认值 | 参数说明 |
|-----------------------|------|------|---|
| criticism_sample_path | 否 | None | 数据清洗负样例目录。目录应存放负样例图片文件，算法将这些图片为负样例，对算法输入中的数据进行过滤，即保留与“criticism_sample_path”目录下图片相似度差距较大的数据。 建议该参数和“prototype_sample_path”配合使用，可以提高数据清洗的准确性。 请输入一个真实存在的OBS目录，且以obs://开头。如： <i>obs://obs_bucket_name/folder_name</i> |
| n_clusters | 否 | auto | 数据样本的种类数，默认值auto。您可以输入小于样本总数的整数或auto。auto表示使用正样本目录的图片个数作为数据样本的种类数。 |
| similarity_threshold | 否 | 0.9 | 相似度阈值。两张图片相似程度超过阈值时，判定为相似图片，反之按非相似图片处理。输入取值范围为0~1。 |
| embedding_distance | 否 | 0.2 | 样本特征间距。两张图片样本特征间距小于设定值，判定为相似图片，反之按非相似图片处理。输入取值范围为0~1。 |
| do_validation | 否 | True | 是否做数据校验，可填True或者False。表示数据清洗前需要做数据校验，否则只做数据清洗。 |

输入要求

算子输入分为两种，“数据集”或“OBS目录”。

- 选择“数据集”，请从下拉框中选择ModelArts中管理的数据集及其版本。要求数据集类型与您在本任务中选择的场景类别一致。
- 选择“OBS目录”，存放结构又分两种情况，“仅包含图片”或“包含图片和标注信息”。
 - “仅包含图片”：当目录下全是图片时，支持jpg、jpeg、png、bmp格式，嵌套子目录的图片也将全部读入。
 - “包含图片和标注信息”：根据不同场景类型，结构不同。

图像分类场景，其目录结构如下所示。如下目录结构，仅支持单标签场景。

```
input_path/
--label1/
----1.jpg
--label2/
----2.jpg
--./
```

物体检测场景，其目录结构如下所示。支持jpg、jpeg、png、bmp格式的图片，xml为标准的PACAL VOC格式标注文件。

```
input_path/
--1.jpg
```

```
--1.xml  
--2.jpg  
--2.xml  
...
```

输出说明

- **图像分类**

输出数据的目录结构如下所示。

```
output_path/  
--Data/  
----class1/ # 若输入数据有标注信息会一并输出，class1为标注类别  
-----1.jpg  
----class2/  
-----2.jpg  
----3.jpg  
--output.manifest
```

其中manifest文件内容示例如下所示。

```
{  
  "id": "xss",  
  "source": "obs://home/fc8e2688015d4a1784dcdba44d840307_14.jpg",  
  "usage": "train",  
  "annotation": [  
    {  
      "name": "Cat",  
      "type": "modelarts/image_classification"  
    }  
  ]  
}
```

- **物体检测**

输出数据的目录结构如下所示。

```
output_path/  
--Data/  
----1.jpg  
----1.xml # 若输入数据有标注信息会一并输出，xml为标注文件  
----2.jpg  
----3.jpg  
--output.manifest
```

其中manifest文件内容示例如下所示。

```
{  
  "source": "obs://fake/be462ea9c5abc09f.jpg",  
  "annotation": [  
    {  
      "annotation-loc": "obs://fake/be462ea9c5abc09f.xml",  
      "type": "modelarts/object_detection",  
      "annotation-format": "PASCAL VOC",  
      "annotated-by": "modelarts/hard_example_algo"  
    }  
  ]  
}
```

3.3 数据选择

3.3.1 数据去重

SimDeduplication 算子概述

- 可以依据用户设置的相似程度阈值完成图像去重处理。图像去重是图像数据处理常见的数据处理方法。图像重复指图像内容完全一样，或者有少量的尺度、位移、色彩、亮度变化，或者是添加了少量其他内容等。

图 3-2 SimDeduplication 效果图



表 3-5 高级参数说明

| 参数名 | 是否必选 | 默认值 | 参数说明 |
|----------------------|------|------|---|
| similarity_threshold | 否 | 0.9 | 相似程度阈值，两张图片间的相似度大于阈值时，其中一张会作为重复图片被过滤掉。取值范围为0~1。 |
| do_validation | 否 | True | 是否做数据校验，可填True或者False。表示数据去重前需要做数据校验，否则只做数据去重。 |

输入要求

算子输入分为两种，“数据集”或“OBS目录”。

- 选择“数据集”，请从下拉框中选择ModelArts中管理的数据集及其版本。要求数据集类型与您在任务中选择的场景类别一致。
- 选择“OBS目录”，存放结构又分两种情况，“仅包含图片”或“包含图片和标注信息”。

- “仅包含图片”：当目录下全是图片时，支持jpg、jpeg、png、bmp格式，嵌套子目录的图片也将全部读入。
- “包含图片和标注信息”：根据不同数据类型，结构不同。

图像分类，其目录结构如下所示。如下目录结构，仅支持单标签场景。

```
input_path/  
--label1/  
----1.jpg  
--label2/  
----2.jpg  
--../
```

物体检测，其目录结构如下所示。支持jpg、jpeg、png、bmp格式的图片，xml为标准的PACAL VOC格式标注文件。

```
input_path/  
--1.jpg  
--1.xml  
--2.jpg  
--2.xml  
...
```

输出说明

- **图像分类**

输出数据的目录结构如下所示。

```
output_path/  
--Data/  
----class1/ # 若输入数据有标注信息会一并输出，class1为标注类别  
-----1.jpg  
----class2/  
-----2.jpg  
-----3.jpg  
--output.manifest
```

其中manifest文件内容示例如下所示。

```
{  
  "id": "xss",  
  "source": "obs://home/fc8e2688015d4a1784dcdba44d840307_14.jpg",  
  "usage": "train",  
  "annotation": [  
    {  
      "name": "Cat",  
      "type": "modelarts/image_classification"  
    }  
  ]  
}
```

- **物体检测**

输出数据的目录结构如下所示。

```
output_path/  
--Data/  
----1.jpg  
----1.xml # 若输入数据有标注信息会一并输出，xml为标注文件  
----2.jpg  
----3.jpg  
--output.manifest
```

其中manifest文件内容示例如下所示。

```
{  
  "source": "obs://fake/be462ea9c5abc09f.jpg",  
  "annotation": [  
    {  
      "annotation-loc": "obs://fake/be462ea9c5abc09f.xml",  
      "type": "modelarts/object_detection",  
      "annotation-format": "PASCAL VOC",  
      "annotated-by": "modelarts/hard_example_algo"  
    }  
  ]  
}
```

3.3.2 数据去冗余

RRD 算子概述

可以依据用户设置的比例去除差异最大的数据。

图 3-3 RRD 效果图



表 3-6 高级参数说明

| 参数名 | 是否必选 | 默认值 | 参数说明 |
|---------------|------|------|---|
| sample_ratio | 否 | 0.9 | 数据留下的百分比。取值范围为0~1。例如0.9表示保留百分之90的原数据。 |
| n_clusters | auto | auto | 数据样本的种类数，默认为auto，即按照目录中图片个数取类别总数，可指定具体类别数，如 4 |
| do_validation | 否 | True | 是否做数据校验，可填True或者False。表示数据去冗余前需要做数据校验，否则只做数据去重。 |

输入要求

算子输入分为两种，“数据集”或“OBS目录”。

- 选择“数据集”，请从下拉框中选择ModelArts中管理的数据集及其版本。要求数据集类型与您在任务中选择的场景类别一致。
- 选择“OBS目录”，存放结构又分两种情况，“仅包含图片”或“包含图片和标注信息”。

- “仅包含图片”：当目录下全是图片时，支持jpg、jpeg、png、bmp格式，嵌套子目录的图片也将全部读入。
- “包含图片和标注信息”：根据不同数据类型，结构不同。

图像分类，其目录结构如下所示。如下目录结构，仅支持单标签场景。

```
input_path/  
--label1/  
----1.jpg  
--label2/  
----2.jpg  
--./
```

物体检测，其目录结构如下所示。支持jpg、jpeg、png、bmp格式的图片，xml为标准的PACAL VOC格式标注文件。

```
input_path/  
--1.jpg  
--1.xml  
--2.jpg  
--2.xml  
...
```

输出说明

- **图像分类**

输出数据的目录结构如下所示。

```
output_path/  
  --Data/  
    ----class1/ # 若输入数据有标注信息会一并输出，class1为标注类别  
      -----1.jpg  
    ----class2/  
      -----2.jpg  
      -----3.jpg  
  --output.manifest
```

其中manifest文件内容示例如下所示。

```
{  
  "id": "xss",  
  "source": "obs://home/fc8e2688015d4a1784dcba44d840307_14.jpg",  
  "usage": "train",  
  "annotation": [  
    {  
      "name": "Cat",  
      "type": "modelarts/image_classification"  
    }  
  ]  
}
```

- **物体检测**

输出数据的目录结构如下所示。

```
output_path/  
  --Data/  
    ----1.jpg  
    ----1.xml # 若输入数据有标注信息会一并输出，xml为标注文件  
    ----2.jpg  
    ----3.jpg  
  --output.manifest
```

其中manifest文件内容示例如下所示。

```
{  
  "source": "obs://fake/be462ea9c5abc09f.jpg",  
  "annotation": [  
    {  
      "annotation-loc": "obs://fake/be462ea9c5abc09f.xml",  
      "type": "modelarts/object_detection",  
      "annotation-format": "PASCAL VOC",  
      "annotated-by": "modelarts/hard_example_algo"  
    }  
  ]  
}
```

3.4 数据增强

3.4.1 数据扩增

数据扩增算子概述

数据扩增主要用于训练数据集不足或需要仿真的场景，能通过对已标注的数据集做变换操作来增加训练图片的数量，同时会生成相应的标签。在深度学习领域，增强有重要的意义，能提升模型的泛化能力，增加抗扰动的能力。数据扩增过程不会改动原始数据，扩增后的图片或xml文件保存在指定的输出路径下。

ModelArts提供以下数据扩增算子：

表 3-7 数据扩增算子介绍

| 算子 | 算子说明 | 高级 |
|-----------|----------------------------------|--|
| AddNoise | 添加噪声，模拟常见采集设备在采集图片过程中可能会产生的噪声。 | <ul style="list-style-type: none"> noise_type: 添加噪声的分布类型，Gauss为高斯噪声，Laplace为拉普拉斯噪声，Poisson是泊松噪声，Impulse是脉冲噪声，SaltAndPepper为椒盐噪声。默认值为Gauss loc: 噪声分布的均值，仅在Gauss和Laplace生效。默认值为0 scale: 噪声分布的标准差，仅在Gauss和Laplace生效。默认值为1 lam: 泊松分布的lambda系数，仅在Poisson有效。默认值为2 p: 对于每个像素点，出现脉冲噪声或椒盐噪声的概率，仅在Impulse和SaltAndPepper有效。默认值为0.01 do_validation: 数据扩增前是否做数据校验。默认值为True。 |
| Blur | 模糊，使用滤波器对图像进行滤波操作，有时用于模拟成像设备的成像。 | <ul style="list-style-type: none"> blur_type: 可选Gauss和Average两种模式，分别为高斯和均值滤波。默认值为Gauss do_validation: 数据扩增前是否做数据校验。默认值为True。 |
| Crop | 图片裁剪，随机裁剪图片的一部分作为新的图片。 | <ul style="list-style-type: none"> crop_percent_min: 各边裁剪占比的随机取值范围的最小值。默认值为0.0 crop_percent_max: 各边裁剪占比的随机取值范围的最大值。默认值为0.2 do_validation: 数据扩增前是否做数据校验。默认值为True。 |
| CutOut | 随机擦除，在深度学习中常用的方法，用于模拟物体被障碍物遮挡。 | do_validation: 数据扩增前是否做数据校验。默认值为True。 |
| Flip | 翻转，沿图片水平轴或竖直轴做翻转，是非常常见的增强方法。 | <ul style="list-style-type: none"> lr_ud: 选择翻转的方向，lr为水平翻转，ud为竖直翻转。默认值为lr flip_p: 做翻转操作的概率。默认值为1。 do_validation: 数据扩增前是否做数据校验。默认值为True。 |
| Grayscale | 图片灰度化，将三通道的彩色图像转换到三通道的灰度图像。 | do_validation: 数据扩增前是否做数据校验。默认值为True。 |

| 算子 | 算子说明 | 高级 |
|-----------------|------------------------------------|---|
| HistogramEqual | 直方图均衡化，多半是用于让图片的视觉效果更加好，在某些场景下会使用。 | do_validation: 数据扩增前是否做数据校验。默认值为True。 |
| LightArithmetic | 亮度增强，对亮度空间做线性增强操作。 | do_validation: 数据扩增前是否做数据校验。默认值为True。 |
| LightContrast | 亮度对比度增强，使用一定的非线性函数改变亮度空间的亮度值。 | <p>func: 默认值为gamma</p> <ul style="list-style-type: none"> gamma为常见方法伽马矫正，公式为 $255*((v/255)**gamma)$ sigmoid为函数为S型曲线，公式为 $255*1/(1+exp(gain*(cutoff-l_{ij}/255)))$ log为对数函数，公式为 $255*gain*log_2(1+v/255)$ linear为线性函数，公式为 $127 + alpha*(v-127)$ <p>do_validation: 数据扩增前是否做数据校验。默认值为True。</p> |
| MotionBlur | 运动模糊，模拟物体运动时产生的残影现象。 | do_validation: 数据扩增前是否做数据校验。默认值为True。 |
| Padding | 图片填充，在边缘添加黑色的边。 | <ul style="list-style-type: none"> px_top: 图像顶端增加的像素行数。默认值为1 px_right: 图像右侧增加的像素行数。默认值为1 px_left: 图像左侧增加的像素行数。默认值为1 px_bottom: 图像底侧增加的像素行数。默认值为1 do_validation: 数据扩增前是否做数据校验。默认值为True。 |
| Resize | 调整图片大小。 | <ul style="list-style-type: none"> height: 变换后的图片高度。默认值224 width: 变换后的图片宽度。默认值224 do_validation: 数据扩增前是否做数据校验。默认值为True。 |

| 算子 | 算子说明 | 高级 |
|------------|--|--|
| Rotate | 旋转，将图像围绕中心点旋转的操作，操作完成之后保持图片原本的形状不变，不足的部分用黑色填充。 | <ul style="list-style-type: none"> angle_min: 旋转角度随机取值范围的最小值，每张图片会从范围中随机取值作为自己的参数。默认值为90° angle_max: 旋转角度随机取值范围的最大值，每张图片会从范围中随机取值作为自己的参数。默认值为-90° do_validation: 数据扩增前是否做数据校验。默认值为True。 |
| Saturation | 色度饱和度增强，对图片的HSV中的H和S空间做线性的变化，改变图片的色度和饱和度。 | do_validation: 数据扩增前是否做数据校验。默认值为True。 |
| Scale | 图片缩放，将图片的长或宽随机缩放到一定倍数。 | <ul style="list-style-type: none"> scaleXY: 缩放方向，X为水平，Y为垂直。默认值为X scale_min: 缩放比例随机取值范围的最小值。默认为0.5 scale_max: 缩放比例随机取值范围的最大值。默认值为1.5 do_validation: 数据扩增前是否做数据校验。默认值为True。 |
| Sharpen | 图像锐化，用于将边缘清晰化，让物体边缘更加明显。 | do_validation: 数据扩增前是否做数据校验。默认值为True。 |
| Shear | 图片错切，一般用于图片的几何变换，通过线性函数将像素点进行映射。 | <ul style="list-style-type: none"> shearXY: 错切方向，X为水平，Y为竖直。默认值为X shear_min: 错切角度随机取值范围的最小值。默认值为-30 shear_max: 错切角度随机取值范围的最大值。默认值为30 do_validation: 数据扩增前是否做数据校验。默认值为True。 |
| Translate | 图片平移，将图片整体像X轴或Y轴平移，超出原图部分舍弃，丢失部分用黑色填充。 | <ul style="list-style-type: none"> translateXY: 平移的方向，X为水平，Y为竖直。默认值为X do_validation: 数据扩增前是否做数据校验。默认值为True。 |

| 算子 | 算子说明 | 高级 |
|---------|--------------|--|
| Weather | 添加天气，模拟天气效果。 | <p>weather_mode: 添加天气的模式，默认值为Rain。</p> <ul style="list-style-type: none"> • Rain: 下雨 • Fog: 雾 • Snow: 雪 • Clouds: 云 <p>do_validation: 数据扩增前是否做数据校验。默认值为True。</p> |

输入要求

算子输入分为两种，“数据集”或“OBS目录”。

- 选择“数据集”，请从下拉框中选择ModelArts中管理的数据集及其版本。要求数据集类型与您在任务中选择的场景类别一致。
- 选择“OBS目录”，存放结构支持“包含图片和标注信息”模式。

“包含图片和标注信息”，根据不同场景类型，结构不同。

图像分类场景，其目录结构如下所示。如下目录结构，仅支持单标签场景。

```
input_path/
  --label1/
    ----1.jpg
  --label2/
    ----2.jpg
  --../
```

物体检测场景，其目录结构如下所示。支持jpg、jpeg、png、bmp格式的图片，xml为标准的PACAL VOC格式标注文件。

```
input_path/
  --1.jpg
  --1.xml
  --2.jpg
  --2.xml
  ...
```

输出说明

由于算法中有些操作将会舍弃一些数据，输出文件夹里可能不包含全量数据集。例如，“Rotate”会舍弃标注框超出原始图片边界的图片。

输出目录结构如下所示。其中“Data”文件夹用于存放新生成的图片和标注信息，“manifest”文件存储文件夹中图片的结构，可直接导入到数据管理的数据集中。

```
|----data_url
|  |----Data
|  |  |----xxx.jpg
|  |  |----xxx.xml(xxx.txt)
|  |----output.manifest
```

其中manifest文件内容示例如下所示。

```
{
  "id": "xss",
  "source": "obs://home/fc8e2688015d4a1784dcba44d840307_14.jpg",
```

```
"usage": "train",
"annotation": [
  {
    "name": "Cat",
    "type": "modelarts/image_classification"
  }
]
```

3.4.2 数据生成

数据生成技术简介

图像生成利用Gan网络依据已知的数据集生成新的数据集。Gan是一个包含生成器和判别器的网络，生成器从潜在空间中随机取样作为输入，其输出结果需要尽量模仿训练集中的真实样本。判别器的输入则为真实样本或生成网络的输出，其目的是将生成网络的输出从真实样本中尽可能分辨出来。而生成网络则要尽可能地欺骗判别网络。两个网络相互对抗、不断调整参数，最终目的是使判别网络无法判断生成网络的输出结果是否真实。训练中获得的生成器网络可用于生成与输入图片相似的图片，用作新的数据集参与训练。基于Gan网络生成新的数据集不会生成相应的标签。图像生成过程不会改动原始数据，新生成的图片或xml文件保存在指定的输出路径下。

StyleGan 算子概述

基于StyleGan2用于在数据集较小的情形下，随机生成相似图像。StyleGAN提出了一个新的生成器结构，能够控制所生成图像的高层级属性(high-level attributes)，如发型、雀斑等；并且生成的图像在一些评价标准上得分更好。而本算法又增加了数据增强算法，可以在较少样本的情况下也能生成较好的新样本，但是样本数尽量在70张以上，样本太少生成出来的新图像不会有太多的样式。

图 3-4 StyleGan 算子

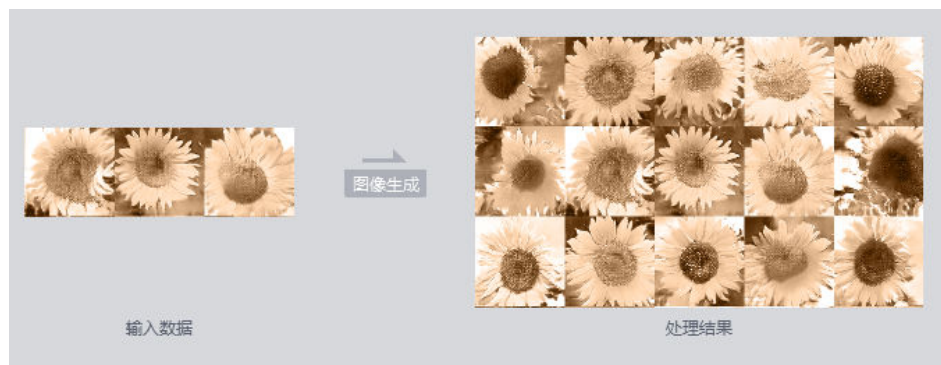


表 3-8 StyleGan 算子高级参数

| 参数名 | 默认值 | 参数说明 |
|------------|-----|----------------------------|
| resolution | 256 | 生成正方形图像的高宽，大小需要是2的次方。 |
| batch-size | 8 | 批量训练样本个数。 |
| total-kimg | 300 | 总共训练的图像数量为total_kimg*1000。 |

| 参数名 | 默认值 | 参数说明 |
|---------------|-------|---|
| generate_num | 300 | 生成的图像数量，如果是多个类的，则为每类生成的数量。 |
| predict | False | 是否进行推理预测，默认为False。如果设置True，需要在resume参数设置已经训练完成的模型的obs路径。 |
| resume | empty | 如果predict设置为True，需要填写Tensorflow模型文件的obs路径用于推理预测。当前仅支持“.pb”格式的模型。示例：obs://xxx/xxxx.pb。 默认值为empty。 |
| do_validation | True | 是否做数据校验，默认为True，表示数据生成前需要做数据校验，否则只做数据生成。 |

数据输入

算子输入分为两种，“数据集”或“OBS目录”。

- 选择“数据集”，请从下拉框中选择ModelArts中管理的数据集及其版本。要求数据集类型与您在本任务中选择的场景类别一致。
- 选择“OBS目录”，图像生成算子不需要标注信息，输入支持单层级或双层级目录，存放结构支持“单层级”或“双层级”模式。

单层级目录结构如下所示：

```
image_folder----0001.jpg
-----0002.jpg
-----0003.jpg
...
-----1000.jpg
```

双层级目录结构如下所示：

```
image_folder----sub_folder_1----0001.jpg
-----0002.jpg
-----0003.jpg
...
-----0500.jpg
----sub_folder_2----0001.jpg
-----0002.jpg
-----0003.jpg
...
-----0500.jpg
...
----sub_folder_100----0001.jpg
-----0002.jpg
-----0003.jpg
...
-----0500.jpg
```

输出说明

输出目录的结构如下所示。其中“model”文件夹存放用于推理的“frozen pb”模型，“samples”文件夹存放训练过程中输出图像，“Data”文件夹存放训练模型生成的图像。


```
train_url----model----CYcleGan_epoch_10.pb
              ----CYcleGan_epoch_20.pb
              ...
              ----CYcleGan_epoch_1000.pb
----samples----0000_0.jpg
              ----0000_1.jpg
              ...
              ----0100_15.jpg
----Data----CYcleGan_0_0.jpg
              ----CYcleGan_0_1.jpg
              ...
              ----CYcleGan_16_8.jpg
----output_0.manifest
```

其中manifest文件内容示例如下所示。

```
{
  "id": "xss",
  "source": "obs://home/fc8e2688015d4a1784dcba44d840307_14.jpg",
  "usage": "train",
  "annotation": [
    {
      "name": "Cat",
      "type": "modelarts/image_classification"
    }
  ]
}
```

3.4.3 数据域迁移

CycleGan 算子概述

基于CycleGAN用于生成域迁移的图像，即将一类图片转换成另一类图片，把X空间中的样本转换成Y空间中的样本。CycleGAN可以利用非成对数据进行训练。模型训练时运行支持两个输入，分别代表数据的原域和目标域，在训练结束时会生成所有原域向目标域迁移的图像。

图 3-5 CycleGan 算子



表 3-9 CycleGan 算子高级参数

| 参数名 | 默认值 | 参数说明 |
|---------------|------|--|
| do_validation | True | 是否做数据校验，默认为True，表示数据生成前需要做数据校验，否则只做数据生成。 |
| image_channel | 3 | 生成图像的通道数。 |

| 参数名 | 默认值 | 参数说明 |
|-----------------|--------|---|
| image_height | 256 | 图像相关参数：生成图像的高，大小需要是2的次方。 |
| image_width | 256 | 图像相关参数：生成图像的宽，大小需要是2的次方 |
| batch_size | 1 | 训练相关参数：批量训练样本个数。 |
| max_epoch | 100 | 训练相关参数：训练遍历数据集次数。 |
| g_learning_rate | 0.0001 | 训练相关参数：生成器训练学习率。 |
| d_learning_rate | 0.0001 | 训练相关参数：判别器训练学习率。 |
| log_frequency | 5 | 训练相关参数：日志打印频率（按step计数）。 |
| save_frequency | 5 | 训练相关参数：模型保存频率（按epoch计数）。 |
| predict | False | 是否进行推理预测，默认为False。如果设置True，需要在resume参数设置已经训练完成的模型的obs路径。 |
| resume | empty | 如果predict设置为True，需要填写Tensorflow模型文件的obs路径用于推理预测。当前仅支持“.pb”格式的模型。示例：obs://xxx/xxxx.pb。 默认值为empty。 |

数据输入

算子输入分为两种，“数据集”或“OBS目录”。

- 选择“数据集”，请从下拉框中选择ModelArts中管理的数据集及其版本。要求数据集类型与您在任务中选择的场景类别一致。
- 选择“OBS目录”，图像生成算子不需要标注信息，输入支持单层级或双层级目录，存放结构支持“单层级”或“双层级”模式。

单层级目录结构如下所示：

```
image_folder----0001.jpg
-----0002.jpg
-----0003.jpg
...
-----1000.jpg
```

双层级目录结构如下所示：

```
image_folder----sub_folder_1----0001.jpg
-----0002.jpg
-----0003.jpg
...
-----0500.jpg
----sub_folder_2----0001.jpg
-----0002.jpg
-----0003.jpg
```

```
...
----0500.jpg
...
----sub_folder_100----0001.jpg
      ----0002.jpg
      ----0003.jpg
...
      ----0500.jpg
```

输出说明

输出目录的结构如下所示。其中“model”文件夹存放用于推理的“frozen pb”模型，“samples”文件夹存放训练过程中输出图像，“Data”文件夹存放训练模型生成的图像。

```
train_url----model----CYcleGan_epoch_10.pb
      ----CYcleGan_epoch_20.pb
      ...
      ----CYcleGan_epoch_1000.pb
----samples----0000_0.jpg
      ----0000_1.jpg
      ...
      ----0100_15.jpg
----Data----CYcleGan_0_0.jpg
      ----CYcleGan_0_1.jpg
      ...
      ----CYcleGan_16_8.jpg
----output_0.manifest
```

其中manifest文件内容示例如下所示。

```
{
  "id": "xss",
  "source": "obs://home/fc8e2688015d4a1784dcba44d840307_14.jpg",
  "usage": "train",
  "annotation": [
    {
      "name": "Cat",
      "type": "modelarts/image_classification"
    }
  ]
}
```