

ModelArts

Data Preparation and Analysis

Issue 01
Date 2024-06-12



Copyright © Huawei Technologies Co., Ltd. 2024. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: <https://www.huawei.com>

Email: support@huawei.com

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process*. For details about this process, visit the following web page:

<https://www.huawei.com/en/psirt/vul-response-process>

For vulnerability information, enterprise customers can visit the following web page:

<https://securitybulletin.huawei.com/enterprise/en/security-advisory>

Contents

| | |
|--|-----------|
| 1 Introduction to Data Preparation..... | 1 |
| 2 Getting Started..... | 3 |
| 3 Creating a Dataset..... | 7 |
| 3.1 Dataset Overview..... | 7 |
| 3.2 Creating a Dataset..... | 9 |
| 3.3 Modifying a Dataset..... | 15 |
| 4 Importing Data..... | 17 |
| 4.1 Introduction to Data Importing..... | 17 |
| 4.2 Importing Data from OBS..... | 19 |
| 4.2.1 Introduction to Importing Data from OBS..... | 19 |
| 4.2.2 Importing Data from an OBS Path..... | 21 |
| 4.2.3 Specifications for Importing Data from an OBS Directory..... | 23 |
| 4.2.4 Importing a Manifest File..... | 28 |
| 4.2.5 Specifications for Importing a Manifest File..... | 30 |
| 4.3 Importing Data from Local Files..... | 40 |
| 5 Data Analysis and Preview..... | 42 |
| 5.1 Data Filtering..... | 42 |
| 5.2 Data Feature Analysis..... | 42 |
| 6 Labeling Data..... | 50 |
| 7 Publishing Data..... | 51 |
| 7.1 Introduction to Data Publishing..... | 51 |
| 7.2 Publishing a Data Version..... | 51 |
| 7.3 Managing Data Versions..... | 53 |
| 8 Exporting Data..... | 55 |
| 8.1 Introduction to Exporting Data..... | 55 |
| 8.2 Exporting Data to a New Dataset..... | 55 |
| 8.3 Exporting Data to OBS..... | 56 |

1 Introduction to Data Preparation

NOTE

Data management is being upgraded and is invisible to users who have not used data management.

The driving forces behind AI are computing power, algorithms, and data. Data quality affects model precision. Generally, a large amount of high-quality data is more likely to train a high-precision AI model. Models trained using normal data achieves 85% to 90% accuracy, while commercial applications have higher requirements. If you want to improve the model accuracy to 96% or even 99%, a large amount of high-quality data is required. In this case, the data must be more refined, scenario-based, and professional. The preparation of a large amount of high-quality data has become a challenging issue in AI development.

ModelArts is a one-stop AI development platform that supports AI lifecycle development, including data processing, algorithm development, model training, and model deployment. In addition, ModelArts provides that can be used to share data, algorithms, and models. ModelArts data management provides end-to-end data preparation, processing, and labeling.

ModelArts data management provides the following functions for you to obtain high-quality AI data:

- Data acquisition
 - Allows you to import data from OBS, MRS, DLI, and GaussDB(DWS).
 - Provides 18+ data augmentation operators to increase data volume for training.
- Improved data quality
 - Allows you to preview various formats of data including images, text, audios, and videos, helping you identify data quality.
 - Allows you to filter data by multiple search criteria, such as sample attributes and labeling information.
 - Provides 12+ labeling tools for refined, scenario-based, and professional data labeling.
 - Performs feature analysis based on samples and labeling results, helping you understand data quality.

- More efficient data preparation
 - Allows you to manage data by version for more efficient data management.
 - Provides capabilities such as interactive labeling for more efficient data labeling.
 - Enables team labeling and team labeling management for labeling a large amount of data.

2 Getting Started

This section uses preparing data for training an object detection model as an example to describe how to analyze and label sample data. During actual service development, you can select one or more data management functions to prepare data based on service requirements. The operation process is as follows:

- [Making Preparations](#)
- [Creating a Dataset](#)
- [Labeling Data](#)
- [Publishing Data](#)
- [Exporting Data](#)

 **NOTE**

Data management is being upgraded and is invisible to users who have not used data management.

Preparations

Before using data management of ModelArts, complete the following preparations:


When using data management, ModelArts needs to access dependent services such as OBS. Therefore, grant permissions on the **Global Configuration** page. For details, see [Configuring Agency Authorization \(Recommended\)](#).

Creating a Dataset

In this example, an OBS path is used as the input path to create a dataset. Perform the following operations to create an object detection dataset and import the data to the dataset:

- Step 1** Log in to the . In the navigation pane, choose **Data Management > Datasets**.
- Step 2** Click **Create**. On the **Create Dataset** page, create a dataset based on the data type and data labeling requirements.
 1. Set the basic information, the name and description of the dataset.

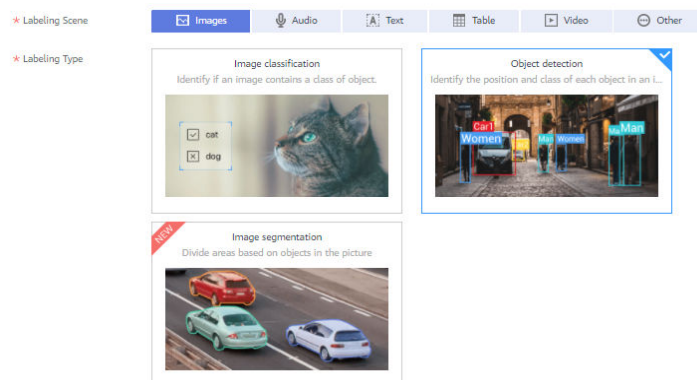
Figure 2-1 Basic information of a dataset



A form for entering dataset information. It has a field for 'Name' with the value 'dataset-144' and a green checkmark. Below it is a larger 'Description' text area. A '0/256' character count is visible at the bottom right of the description box.

2. Set labeling scene and type. In this example, choose **Images** and **Object detection**.

Figure 2-2 Dataset labeling scene and type



A screenshot of the labeling scene and type selection interface. At the top, there are tabs for 'Images', 'Audio', 'Text', 'Table', 'Video', and 'Other'. The 'Images' tab is selected. Below the tabs, there are three options for 'Labeling Type': 'Image classification' (with a cat image and checkboxes for 'cat' and 'dog'), 'Object detection' (with a street scene image and bounding boxes for 'Car', 'Woman', and 'Man'), and 'Image segmentation' (with a car image and a 'NEW' badge).

3. Select an OBS path as **Input Dataset Path**, and select another OBS path as **Output Dataset Path**.

Figure 2-3 Input and output dataset path



A screenshot of the input and output dataset path selection interface. It has two fields: 'Input Dataset Path' and 'Output Dataset Path', both with a placeholder 'Select an OBS path.' and a search icon. Below these is a 'Label Set' section with a text box 'Enter a label name.', a color selection dropdown, and an 'Add Label' button. At the bottom, there is a 'Team Labeling' toggle switch.

4. After setting the parameters, click **Create** in the lower right corner to create a dataset.

----End

Labeling Data

- Manual labeling
 - a. On the **Unlabeled** tab page, click an image. The system automatically directs you to the page for labeling the image.
 - b. On the toolbar of the labeling page, select a proper labeling tool. In this example, a rectangle is used for labeling.

Figure 2-4 Labeling tools



- c. Drag the mouse to select an object, enter a new label name in the displayed text box. If labels already exist, select one from the drop-down list box. Click **Add**.

- d. Click **Back to Data Labeling Preview** in the upper left part of the page to view the labeling information. In the dialog box that is displayed, click **Yes** to save the labeling settings. The selected image is automatically moved to the **Labeled** tab page. On the **Unlabeled** and **All** tab pages, the labeling information is updated along with the labeling process, including the added label names and the number of images for each label.

Publishing Data

ModelArts training management allows you to create training jobs using ModelArts datasets or files in an OBS directory. If a dataset is used as the data source of a training job, specify a dataset and version. Therefore, you must have published a dataset version. For details, see [Publishing a Data Version](#).

NOTE

Data that is from the same source and labeled in different batches are differentiated by version. This facilitates subsequent model building and development. You can select specified versions.

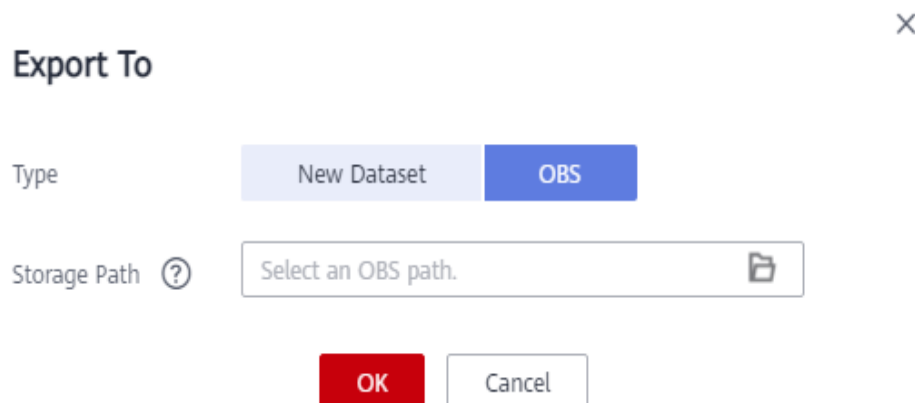
Exporting Data

ModelArts training management allows you to create training jobs using ModelArts datasets or files in an OBS directory. If you create a training job using an OBS directory, export the prepared data to OBS.

1. Export data to OBS.
 - a. On the dataset details page, select or filter the data to be exported, and click **Export** in the upper right corner.
 - b. Set **Type** to **OBS**, enter related information, and click **OK**.

Storage Path: path where the data to be exported is stored. You are advised not to save data to the input or output path of the current dataset.

Figure 2-5 Exporting to OBS



The screenshot shows a dialog box titled "Export To" with a close button (X) in the top right corner. Under the title, there are two buttons: "New Dataset" and "OBS". The "OBS" button is highlighted in blue, indicating it is selected. Below these buttons is a text input field labeled "Storage Path" with a question mark icon to its left and a file selection icon to its right. The text inside the field is "Select an OBS path.". At the bottom of the dialog, there are two buttons: "OK" (in red) and "Cancel" (in white).

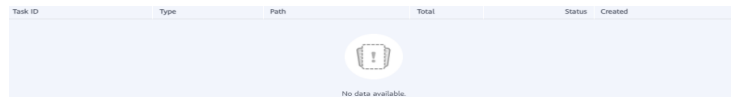
- c. After the data is exported, view it in the specified path.

2. View task history.

After exporting data, you can view the export task details in **Export History**.

- a. On the dataset details page, click **Export History** in the upper right corner.
- b. In the **View Task History** dialog box, view the export task history of the current dataset. You can view the task ID, creation time, export type, export path, total number of exported samples, and export status.

Figure 2-6 Export history



3 Creating a Dataset

Before using ModelArts to prepare data, create a dataset. Then, you can perform operations on the dataset, such as importing data, analyzing data, and labeling data.

3.1 Dataset Overview

NOTE

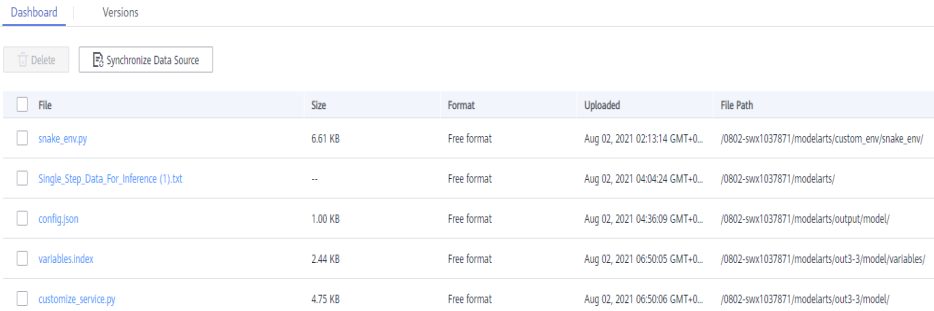
Data management is being upgraded and is invisible to users who have not used data management.

Dataset Types

ModelArts supports the following types of datasets:

- Images: in .jpg, .png, .jpeg, or .bmp format for image classification and object detection
- Audio: in .wav format for sound classification, speech labeling, and speech paragraph labeling
- Text: in .txt or .csv format for text classification, named entity recognition, and text triplet labeling
- Free format: allows data in any format. Labeling is not available for free format data. The free format applies if labeling is not required or needs to be customized. Select this format if your data is in multiple formats or your data is not in any of the preceding formats.

Figure 3-1 Example of a dataset in free format



The screenshot shows the 'Dashboard' view of a dataset. At the top, there are 'Delete' and 'Synchronize Data Source' buttons. Below is a table with columns: File, Size, Format, Uploaded, and File Path. The table lists several files, all in 'Free format'.

| File | Size | Format | Uploaded | File Path |
|---|---------|-------------|--------------------------------|---|
| <input type="checkbox"/> snake_env.py | 6.61 KB | Free format | Aug 02, 2021 02:13:14 GMT+0... | /0802-swxx1037871/modelarts/custom_env/snake_env/ |
| <input type="checkbox"/> Single_Step_Data_For_Inference (1).txt | ... | Free format | Aug 02, 2021 04:04:24 GMT+0... | /0802-swxx1037871/modelarts/ |
| <input type="checkbox"/> config.json | 1.00 KB | Free format | Aug 02, 2021 04:36:09 GMT+0... | /0802-swxx1037871/modelarts/output/model/ |
| <input type="checkbox"/> variables.index | 2.44 KB | Free format | Aug 02, 2021 06:50:05 GMT+0... | /0802-swxx1037871/modelarts/out3-3/model/variables/ |
| <input type="checkbox"/> customize_service.py | 4.75 KB | Free format | Aug 02, 2021 06:50:06 GMT+0... | /0802-swxx1037871/modelarts/out3-3/model/ |

Dataset Functions

Different types of datasets support different functions, such as auto labeling and team labeling. For details, see [Table 3-1](#).

Table 3-1 Functions supported by different types of datasets

| Data set Type | Labeling Type | Creating a Dataset | Importing Data | Exporting Data | Publishing a Dataset | Modifying a Dataset | Managing Dataset Versions | Auto Grouping | Data Features |
|---------------|---------------------------|--------------------|----------------|----------------|----------------------|---------------------|---------------------------|---------------|---------------|
| Image | Image classification | Supported | Supported | Supported | Supported | Supported | Supported | Supported | Supported |
| | Object detection | Supported | Supported | Supported | Supported | Supported | Supported | Supported | Supported |
| | Image segmentation | Supported | Supported | Supported | Supported | Supported | Supported | Supported | N/A |
| Audio | Sound classification | Supported | Supported | N/A | Supported | Supported | Supported | N/A | N/A |
| | Speech labeling | Supported | Supported | N/A | Supported | Supported | Supported | N/A | N/A |
| | Speech paragraph labeling | Supported | Supported | N/A | Supported | Supported | Supported | N/A | N/A |
| Text | Text classification | Supported | Supported | N/A | Supported | Supported | Supported | N/A | N/A |

| Data set Type | Labeling Type | Creating a Dataset | Importing Data | Exporting Data | Publishing a Dataset | Modifying a Dataset | Managing Dataset Versions | Auto Grouping | Data Features |
|---------------|--------------------------|--------------------|----------------|----------------|----------------------|---------------------|---------------------------|---------------|---------------|
| | Named entity recognition | Supported | Supported | N/A | Supported | Supported | Supported | N/A | N/A |
| | Text triplet | Supported | Supported | N/A | Supported | Supported | Supported | N/A | N/A |
| Video | Video labeling | Supported | Supported | N/A | Supported | Supported | Supported | N/A | N/A |
| Free format | Free format | Supported | N/A | - | Supported | Supported | Supported | N/A | N/A |
| Table | Table | Supported | Supported | N/A | Supported | Supported | Supported | N/A | N/A |

Specifications Restrictions

- The maximum numbers of samples and labels in a single text or audio database other than a table dataset are 1,000,000 and 10,000, respectively.
- The maximum size of a sample in a single text or audio database other than an image dataset is 5 GB.
- The maximum size of an image for object detection or image classification is 25 MB.
- The maximum size of a manifest file is 5 GB.
- The maximum size of a text file in a line is 100 KB.
- The maximum size of a labeling result file is 100 MB.

3.2 Creating a Dataset

Before using ModelArts to manage data, create a dataset. Then, you can perform operations on the dataset, such as labeling data, importing data, and publishing the dataset. This section describes how to create a dataset of the non-table type (image, audio, text, video, and free format) and table type.

NOTE

Data management is being upgraded and is invisible to users who have not used data management.

Prerequisites

- You have been authorized to access OBS. To do so, click the **Settings** page in the navigation pane of the ModelArts management console and add access authorization using an agency.
- OBS buckets and folders for storing data are available. In addition, the OBS buckets and ModelArts are in the same region. OBS parallel file systems are not supported. Select object storage.
- OBS buckets are not encrypted. ModelArts does not support encrypted OBS buckets. When creating an OBS bucket, do not enable bucket encryption.

Image, Audio, Text, and Free Format

1. Log in to the . In the navigation pane, choose **Data Management > Datasets**.

Figure 3-2 Dataset management page

| Name | Version | Labeling Progress | Created | Description | Operation |
|----------------------------------|---------|-------------------|---------------------------------|----------------------------------|--|
| ish-test2 15nBWVWQZV2CCosao | V005 | 72.73% (16/22) | Apr 03, 2020 11:41:24 GMT+08:00 | create from dataset xubo.cla... | Import Publish Labeling Export Delete More |
| ish-test1 1QZ3Koo9OBuP9BLFB8O | V002 | 100.00% (3/3) | Apr 03, 2020 11:32:00 GMT+08:00 | create from dataset xianao-te... | Import Publish Labeling Export Delete More |

NOTE

The number of datasets that can be created under an account in a region is limited. For details, see the number displayed on the **Dataset** page.

2. Click **Create**. On the **Create Dataset** page, create a dataset based on the data type and data labeling requirements.

Figure 3-3 Parameter settings

* Data Type: **Images** | Audio | Text | Video | Free format | Table
Supported formats: .jpg, .png, .jpeg, .bmp

* Data Source: **OBS** | AI Gallery | Local file

* Import Mode: **Path**

You can save the dataset file to be imported to the OBS path that you have permission to access. [Labeling file format](#)

* Import Path:

You can import up to 1000000 samples and 10000 labels.

* Labeling Status: **Unlabeled** | Labeled

* Output Dataset Path:

Path for storing output files such as labeled files. The path cannot be the same as the import path or subdirectory of the import path.

- **Name:** name of the dataset, which is customizable
- **Description:** details about the dataset
- **Data Type:** Select a data type based on your needs.
- **Data Source**
 - i. Importing data from OBS
If data is available in OBS, select **OBS** for **Data Source**, and configure other mandatory parameters. The labeling formats of the input data

vary depending on the dataset type. For details about the labeling formats supported by ModelArts, see [Introduction to Data Importing](#).

Figure 3-4 Importing data from OBS

ii. Importing data from a local path

If data is not stored in OBS and the required data cannot be downloaded from AI Gallery, ModelArts enables you to upload the data from a local path. Before uploading data, configure **Storage Path** and **Labeling Status**. Click **Upload data** to select the local file for uploading. Select a labeling format when the labeling status is **Labeled**. The labeling formats of the input data vary depending on the dataset type. For details about the labeling formats supported by ModelArts, see [Introduction to Data Importing](#).

- For more details about parameters, see [Table 3-2](#).

Table 3-2 Dataset parameters

| Parameter | Description |
|-------------|---|
| Import Path | <p>OBS path from which your data is to be imported. This path is used as the data storage path of the dataset.</p> <p>NOTE OBS parallel file systems are not supported. Select an OBS bucket.</p> <p>When you create a dataset, data in the OBS path will be imported to the dataset. If you modify data in OBS, the data in the dataset will be inconsistent with that in OBS. As a result, certain data may be unavailable. To modify data in a dataset, follow the operations provided in Import Mode or Importing Data from an OBS Path.</p> <p>If the numbers of samples and labels of the dataset exceed quotas, importing the samples and labels will fail.</p> |

| Parameter | Description |
|---------------------|---|
| Labeling Status | <p>Labeling status of the selected data, which can be Unlabeled or Labeled.</p> <p>If you select Labeled, specify a labeling format and ensure the data file complies with format specifications. Otherwise, the import may fail.</p> <p>Only image (object detection, image classification, and image segmentation), audio (sound classification), and text (text classification) labeling tasks support the import of labeled data.</p> |
| Output Dataset Path | <p>OBS path where your labeled data is stored.</p> <p>NOTE</p> <ul style="list-style-type: none"> • Ensure that your OBS path name contains letters, digits, and underscores (_) and does not contain special characters, such as ~'@#\$\$%^&*{}[];+=<>/ and spaces. • The dataset output path cannot be the same as the data input path or subdirectory of the data input path. • It is a good practice to select an empty directory as the dataset output path. • OBS parallel file systems are not supported. Select an OBS bucket. |

3. After setting the parameters, click **Submit**.

Table

1. Log in to the [ModelArts management console](#). In the navigation pane, choose **Data Management > Datasets**.

Figure 3-5 Dataset management page

| Name | Version | Labeling Progress | Created | Description | Operation |
|----------------------------------|---------|-------------------|---------------------------------|----------------------------------|--|
| ckh-bes12 5758XWVQD+Q2C0xao | V005 | 72.73% (16/22) | Apr 03, 2020 11:41:24 GMT+08:00 | create from dataset xubo_cg... | Import Publish Labeling Export Delete More |
| ckh-bes1 02Q3Xoo9S0B8PKBLFB80 | V002 | 100.00% (3/3) | Apr 03, 2020 11:32:00 GMT+08:00 | create from dataset xianao-te... | Import Publish Labeling Export Delete More |

NOTE

The number of datasets that can be created under an account in a region is limited. For details, see the number displayed on the **Dataset** page.

2. Click **Create**. On the **Create Dataset** page, create a table dataset based on the data type and data labeling requirements.

Figure 3-6 Parameters of a table dataset

The screenshot shows the configuration interface for a table dataset. Key elements include:

- Name:** A text input field containing "dataset-a108" with a green checkmark.
- Description:** An empty text input field with a character count of "0/256".
- Data Type:** A row of buttons: Images, Audio, Text, Video, Free format, and **Table** (highlighted with a red box).
- Data Source:** A row of buttons: **OBS** (selected), DWS, DLI, MRS, and Local file.
- File Path:** A dropdown menu with the text "Select an OBS path." and a folder icon.
- Contain Table Header:** A toggle switch that is currently turned on.
- Schema:** A section with a table for defining columns:

| Column Name | Type |
|----------------------|--------|
| <input type="text"/> | String |

 Below the table is an "Add Schema" button.
- Output Dataset Path:** A dropdown menu with the text "Select an OBS path." and a folder icon.

Below the Output Dataset Path field, there is a note: "Path for storing output files such as labeled files. The path cannot be the same as the import path or subdirectory of the import path."

- **Name:** name of the dataset, which is customizable
- **Description:** details about the dataset
- **Data Type:** Select a data type based on your needs.
- For more details about parameters, see [Table 3-3](#).

Table 3-3 Dataset parameters

| Parameter | Description |
|------------|--|
| Local file | Storage Path: Select an OBS path. |
| Schema | <p>Names and types of table columns, which must be the same as those of the imported data. Set the column name based on the imported data and select the column type. For details about the supported types, see Table 3-4.</p> <p>Click Add Schema to add a new record. When creating a dataset, you must specify a schema. Once created, the schema cannot be modified.</p> <p>When data is imported from OBS, the schema of the CSV file in the file path is automatically obtained. If the schemas of multiple CSV files are inconsistent, an error will be reported.</p> <p>NOTE After you select data from OBS, column names in Schema are automatically displayed, which is the first-row data of the table by default. To ensure the correct prediction code, you need to change column names in Schema to attr_1, attr_2, ..., and attr_n. attr_n is the last column, indicating the prediction column.</p> |

| Parameter | Description |
|---------------------|---|
| Output Dataset Path | <p>OBS path for storing table data. The data imported from the data source is stored in this path. The path cannot be the same as the file path in the OBS data source or subdirectories of the file path.</p> <p>After a table dataset is created, the following four directories are automatically generated in the storage path:</p> <ul style="list-style-type: none"> • annotation: version publishing directory. Each time a version is published, a subdirectory with the same name as the version is generated in this directory. • data: data storage directory. Imported data is stored in this directory. • logs: directory for storing logs. • temp: temporary working directory. |

Table 3-4 Schema data types

| Type | Description | Storage Space | Range |
|-----------|--|---------------|---|
| String | String type | N/A | N/A |
| Short | Signed integer | 2 bytes | -32768 to 32767 |
| Int | Signed integer | 4 bytes | -2147483648 to 2147483647 |
| Long | Signed integer | 8 bytes | -9223372036854775808 to 9223372036854775807 |
| Double | Double-precision floating point | 8 bytes | N/A |
| Float | Single-precision floating point | 4 bytes | N/A |
| Byte | Signed integer | 1 byte | -128 to 127 |
| Date | Date type in the format of "yyyy-MM-dd", for example, 2014-05-29 | N/A | N/A |
| Timestamp | Timestamp that represents date and time in the format of "yyyy-MM-dd HH:mm:ss" | N/A | N/A |

| Type | Description | Storage Space | Range |
|---------|--------------|---------------|------------|
| Boolean | Boolean type | 1 byte | TRUE/FALSE |

 NOTE

When using a CSV file, pay attention to the following:

- When the data type is set to **String**, the data in the double quotation marks is regarded as one record by default. Ensure the double quotation marks in the same row are closed. Otherwise, the data will be too large to display.
- If the number of columns in a row of the CSV file is different from that defined in the schema, the row will be ignored.

3. After setting the parameters, click **Submit**.

3.3 Modifying a Dataset

The basic information of a created dataset can be modified to keep pace with service changes.

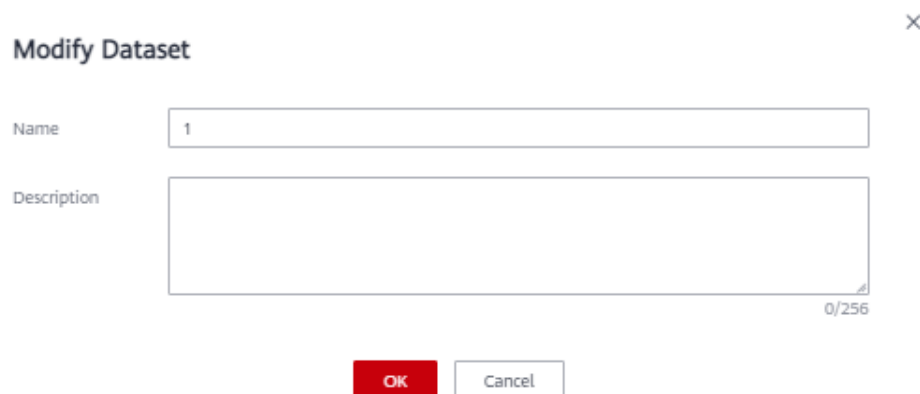
Prerequisites

A created dataset is available.

Modifying the Basic Information of a Dataset

1. Log in to the . In the navigation pane, choose **Data Management > Datasets**.
2. In the dataset list, choose **More > Modify** in the **Operation** column of the target dataset.
3. Modify the basic information by referring to [Table 3-5](#) and click **OK**.

Figure 3-7 Modify Dataset



Modify Dataset X

Name

Description

0/256

Table 3-5 Parameters

| Parameter | Description |
|-------------|--|
| Name | Name of a dataset, which must be 1 to 64 characters long and start with a letter. Only letters, digits, underscores (_), and hyphens (-) are allowed. The name must start with a letter. |
| Description | Brief description of the dataset. |

4 Importing Data

4.1 Introduction to Data Importing

After a dataset is created, you can import more data. ModelArts allows you to import data from different data sources.

- [Importing Data from OBS](#)
- [Importing Data from Local Files](#)

ModelArts AI Gallery provides a large number of built-in datasets, including file and table datasets. You can download and use the built-in datasets from AI Gallery. You can also import your data to ModelArts.

File Data Sources

You can import data by downloading built-in datasets from AI Gallery, or from OBS or a local file. After the import, the data from the import path is automatically synchronized to the data source path of the dataset.

- **OBS:** Import data from an OBS path or a manifest file.
- **Local file:** Import local data that has been uploaded to an OBS path.

Table Data Sources

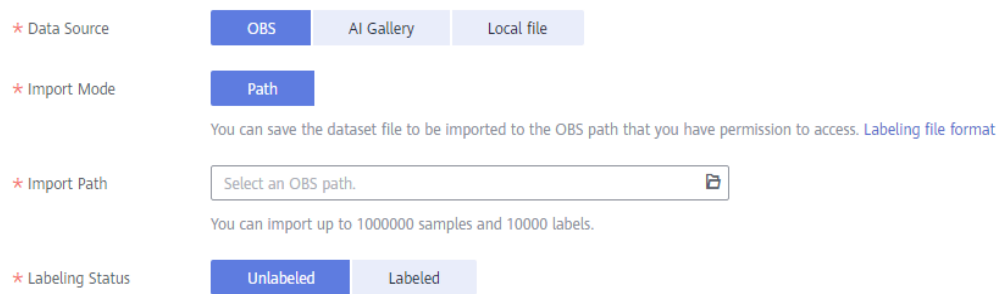
You can import data by downloading built-in datasets from AI Gallery, or from OBS, DWS, DLI, MRS, and local files.

Import Mode

There are five modes for importing data to a dataset.

- When you create a dataset, select an import path. The data is automatically synchronized from the import path.

Figure 4-1 Importing data when creating a dataset



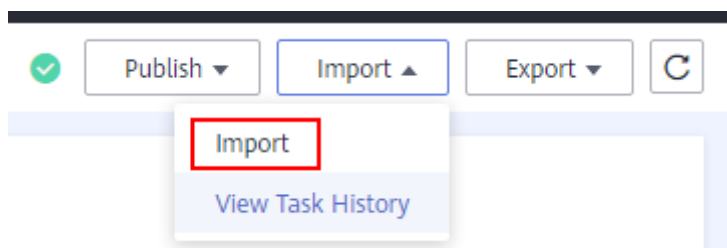
- After a dataset is created, click **Import** in the **Operation** column on the dataset list page.

Figure 4-2 Importing data on the dataset list page



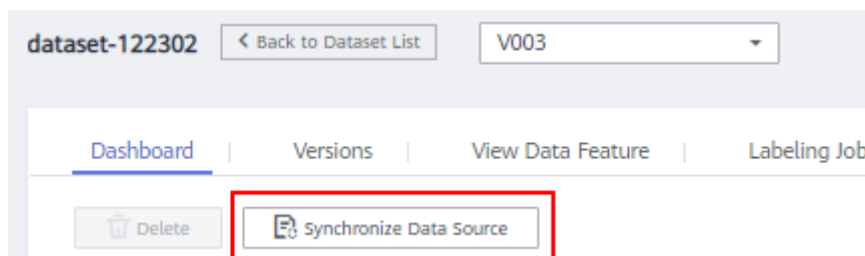
- On the dataset list page, click a dataset. On the dataset details page, choose **Import > Import**.

Figure 4-3 Importing data on the dataset details page



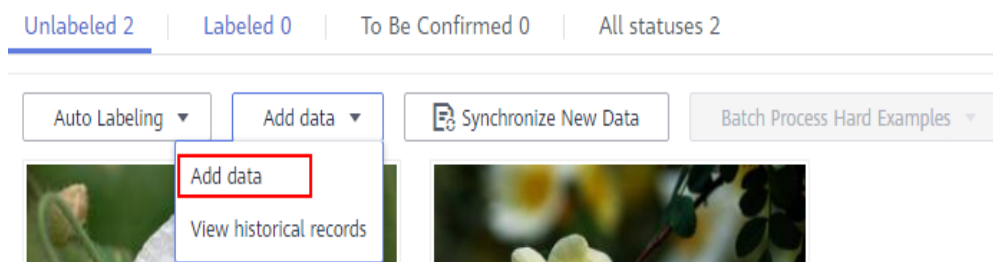
- On the dataset list page, click a dataset. On the dataset details page, click **Synchronize Data Source** to synchronize data from OBS.

Figure 4-4 Synchronizing data sources on the dataset details page



- Add data on the labeling job details page.

Figure 4-5 Adding data on the labeling job details page



4.2 Importing Data from OBS

4.2.1 Introduction to Importing Data from OBS

Import Modes

You can import data from OBS through an OBS path or a manifest file.

- **OBS path:** indicates that the dataset to be imported has been stored in an OBS path. In this case, select an OBS path that you can access. In addition, the directory structure in the OBS path must comply with the specifications. For details, see [Specifications for Importing Data from an OBS Directory](#). This import mode is available only for the following types of datasets: **Image classification**, **Object detection**, **Text classification**, **Table**, and **Sound classification**. For other types of datasets, data can be imported only through a manifest file.
- **Manifest file:** indicates that the dataset file is in the manifest format and the manifest file has been uploaded to OBS. The manifest file defines the mapping between labeling objects and content. For details about the specifications of manifest files, see [Specifications for Importing a Manifest File](#).

NOTE

Before importing an object detection dataset, ensure that the labeling range of the labeling file does not exceed the size of the original image. Otherwise, the import may fail.

Table 4-1 Import modes supported by datasets

| Dataset Type | Labeling Type | From an OBS Path | From a Manifest File |
|--------------|----------------------|---|---|
| Images | Image classification | Supported You can import unlabeled or labeled data. Format specifications of labeled data: Image Classification | Supported You can import unlabeled or labeled data. Format specifications of labeled data: Image Classification |

| Dataset Type | Labeling Type | From an OBS Path | From a Manifest File |
|--------------|---------------------------|---|---|
| | Object detection | Supported You can import unlabeled or labeled data. Format specifications of labeled data: Object Detection | Supported You can import unlabeled or labeled data. Format specifications of labeled data: Object Detection |
| | Image segmentation | Supported You can import unlabeled or labeled data. | Supported You can import unlabeled or labeled data. |
| Audio | Sound classification | Supported You can import unlabeled or labeled data. Follow the format specifications described in Sound Classification . | Supported You can import unlabeled or labeled data. Format specifications of labeled data: Sound Classification |
| | Speech labeling | Supported You can import unlabeled data. | Supported You can import unlabeled or labeled data. Format specifications of labeled data: Speech Labeling |
| | Speech paragraph labeling | Supported You can import unlabeled data. | Supported You can import unlabeled or labeled data. Format specifications of labeled data: Speech Paragraph Labeling |
| Text | Text classification | Supported You can import unlabeled or labeled data. Format specifications of labeled data: Text Classification | Supported You can import unlabeled or labeled data. Format specifications of labeled data: Text Classification |
| | Named entity recognition | Supported You can import unlabeled data. | Supported You can import unlabeled or labeled data. Format specifications of labeled data: Named Entity Recognition |

| Dataset Type | Labeling Type | From an OBS Path | From a Manifest File |
|--------------|----------------|---|---|
| | Text triplet | Supported You can import unlabeled data. | Supported You can import unlabeled or labeled data. Format specifications of labeled data: Text Triplet |
| Video | Video labeling | Supported You can import unlabeled data. | Supported You can import unlabeled or labeled data. |
| Other | Free format | Supported You can import unlabeled data. | N/A |
| Table | Table | Supported Follow the format specifications described in Tables . | N/A |

4.2.2 Importing Data from an OBS Path

Prerequisites

- A dataset is available.
- The data to be imported is stored in OBS. The manifest file is stored in OBS.
- The OBS bucket and ModelArts are in the same region and you can operate the bucket.

Importing File Data from an OBS Path

The parameters on the GUI for data import vary according to the dataset type. The following uses a dataset of the image classification type as an example.

1. Log in to the . In the navigation pane, choose **Data Management > Datasets**.
2. Locate the row that contains the desired dataset and click **Import** in the **Operation** column. Alternatively, click the dataset name to go to the **Dashboard** tab page of the dataset, and click **Import** in the upper right corner.
3. In the **Import** dialog box, configure parameters as follows and click **OK**.
 - **Data Source: OBS**
 - **Import Mode: Path**
 - **Import Path:** OBS path for storing data

- **Labeling Status: Labeled**
 - **Advanced Feature Settings:** disabled by default
- Import by Tag** enables the system to automatically obtain the labels of the current dataset. Click **Add Label** to add a label. This parameter is optional. If **Import by Tag** is disabled, you can add or delete labels for imported data when labeling data.

Figure 4-6 Importing data from an OBS path

Import

* Data Source: OBS Local file

* Import Mode: Path manifest

You can save the dataset file to be imported to the OBS path that you have permission to access. [Labeling file format](#)

* Import Path:

* Labeling Status: Unlabeled Labeled

* Labeling Format:

The labeled object and labeled file must be stored in the same directory, and their names must be the same. Labeled files must be in the PASCAL_VOC format.

File storage structure

- dataset-import-example
 - IMG_20180919_114732.jpg
 - IMG_20180919_114732.xml
 - IMG_20180919_114745.jpg
 - IMG_20180919_114745.xml

After the data is imported, it will be automatically synchronized to the dataset. On the **Datasets** page, click the dataset name to view its details and create a labeling job to label the data.

Labeling Status of File Data

The labeling status can be **Unlabeled** or **Labeled**.

- **Unlabeled:** Only the labeling object (such as unlabeled images or texts) is imported.
- **Labeled:** Both the labeling object and content are imported. Labeling content importing is not supported for datasets in free format.

To ensure that the labeling content can be correctly read, you must store data in strict accordance with the specifications.

If **Import Mode** is set to **Path**, store the data to be imported according to the labeling file specifications. For details, see [Specifications for Importing Data from an OBS Directory](#).

If **Import Mode** is set to **manifest**, the manifest file specifications must be met.

 **NOTE**

- If the labeling status is set to **Labeled**, ensure that the folder or manifest file complies with the format specifications. Otherwise, the import may fail.
- After the import of labeled data, check whether the imported data is in the labeled state.

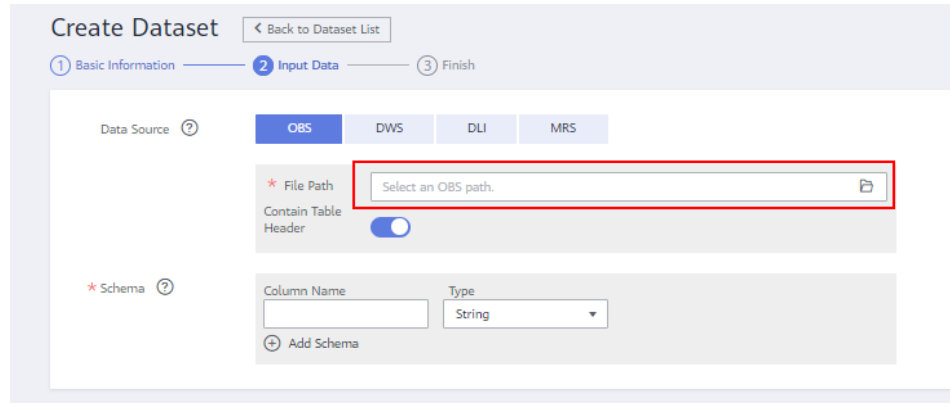
Importing a Table Dataset from OBS

ModelArts allows you to import table data (CSV files) from OBS.

Import description:

- The prerequisite for successful import is that the schema of the data source must be the same as that specified during dataset creation. The schema indicates column names and types of a table. Once specified during dataset creation, the values cannot be changed.
- When a CSV file is imported from OBS, the data type is not validated, but the number of columns must be the same as that in the schema of the dataset. If the data format is invalid, the data is set to null. For details, see [Table 3-4](#).
- You must select the directory where the CSV file is stored. The number of columns in the CSV file must be the same as that in the dataset schema. The schema of the CSV file can be automatically obtained.

```
dataset-import-example
table_import_1.csv
table_import_2.csv
table_import_3.csv
table_import_4.csv
```



4.2.3 Specifications for Importing Data from an OBS Directory

When importing data from OBS, the data storage directory and file name must comply with the ModelArts specifications.

Only the following labeling types of data can be imported by **Labeling Format**: image classification, object detection, image segmentation, text classification, and sound classification.

 **NOTE**

- To import data from an OBS directory, you must have the read permission on the OBS directory.
- The OBS buckets and ModelArts must be in the same region.

Image Classification

Data for image classification can be stored in two formats:

Format 1: ModelArts imageNet 1.0

- Images with the same label must be stored in the same directory, with the label name as the directory name. If there are multiple levels of directories, the last level is used as the label name.

In the following example, **Cat** and **Dog** are label names.

```
dataset-import-example
├── Cat
│   ├── 10.jpg
│   ├── 11.jpg
│   └── 12.jpg
└── Dog
    ├── 1.jpg
    ├── 2.jpg
    └── 3.jpg
```

Format 2: ModelArts image classification 1.0

- The image and labeled file must be stored in the same directory, with the content in the labeled file used as label names.

In the following example, **import-dir-1** and **import-dir-2** are the imported subdirectories:

```
dataset-import-example
├── import-dir-1
│   ├── 10.jpg
│   ├── 10.txt
│   ├── 11.jpg
│   ├── 11.txt
│   ├── 12.jpg
│   └── 12.txt
└── import-dir-2
    ├── 1.jpg
    ├── 1.txt
    ├── 2.jpg
    └── 2.txt
```

The following shows a label file for a single label, for example, the **1.txt** file:

```
Cat
```

The following shows a label file for multiple labels, for example, the **2.txt** file:

```
Cat
Dog
```

- Only images in JPG, JPEG, PNG, and BMP formats are supported. The size of a single image cannot exceed 5 MB, and the total size of all images uploaded at a time cannot exceed 8 MB.

Object Detection

Data for object detection can be stored in two formats:

Format 1: ModelArts PASCAL VOC 1.0

- The simple mode of object detection requires you to store labeled objects and your label files (in one-to-one relationship with the labeled objects) in the same directory. For example, if the name of the labeled object file is **IMG_20180919_114745.jpg**, the name of the label file must be **IMG_20180919_114745.xml**.

The label files must be in PASCAL VOC format. For details about the format, see [Table 4-7](#).

Example:

```
dataset-import-example
  IMG_20180919_114732.jpg
  IMG_20180919_114732.xml
  IMG_20180919_114745.jpg
  IMG_20180919_114745.xml
  IMG_20180919_114945.jpg
  IMG_20180919_114945.xml
```

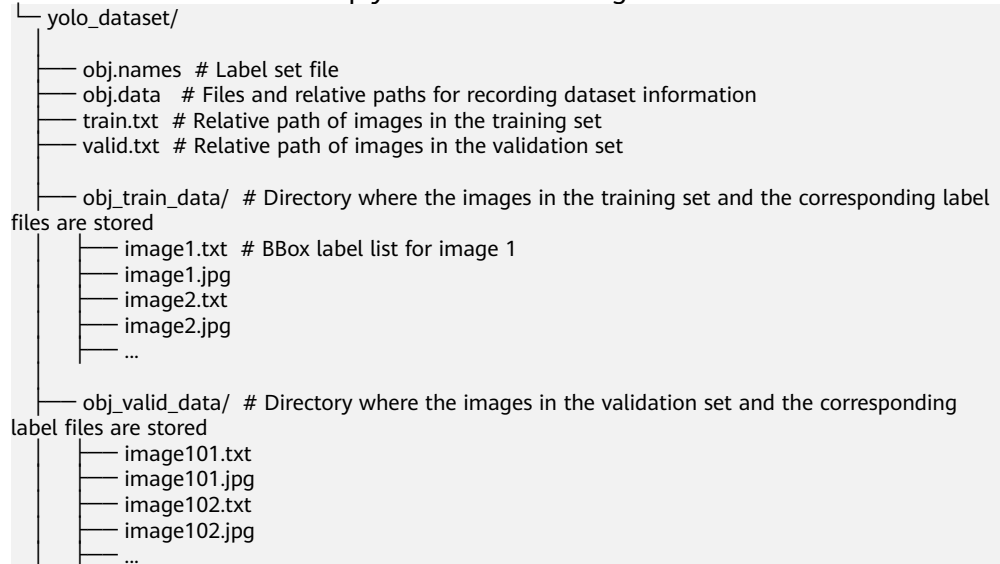
A label file example is as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<annotation>
  <folder>NA</folder>
  <filename>bike_1_1593531469339.png</filename>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>554</width>
    <height>606</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>Dog</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <occluded>0</occluded>
    <bndbox>
      <xmin>279</xmin>
      <ymin>52</ymin>
      <xmax>474</xmax>
      <ymax>278</ymax>
    </bndbox>
  </object>
  <object>
    <name>Cat</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <occluded>0</occluded>
    <bndbox>
      <xmin>279</xmin>
      <ymin>198</ymin>
      <xmax>456</xmax>
      <ymax>421</ymax>
    </bndbox>
  </object>
</annotation>
```

- Only images in JPG, JPEG, PNG, and BMP formats are supported. A single image cannot exceed 5 MB, and the total size of all images uploaded at a time cannot exceed 8 MB.

Format 2: YOLO

- A YOLO dataset must comply with the following structure:



A YOLO dataset supports only training sets and validation sets. If other sets are imported, they will be invalid in the YOLO dataset.

- **obj.data** contains the following content and at least one of the **train** and **valid** subsets must be contained. The file paths are relative paths.

```
classes = 5 # Optional
names = <path/to/obj.names># For example, obj.names
train = <path/to/train.txt># For example, train.txt
valid = <path/to/valid.txt># Optional, for example, valid.txt
backup = backup/ # Optional
```

- The **obj.names** file records the label list. Each row label is used as the file index.

```
label1 # index of label 1: 0
label2 # index of label 2: 1
label3
...
```

- The file paths in **train.txt** and **valid.txt** are relative paths, and the file list must be in one-to-one relationship with the files in the directories. The file structures of the two files are as follows:

```
<path/to/image1.jpg># For example, obj_train_data/image.jpg
<path/to/image2.jpg># For example, obj_train_data/image.jpg
...
```

- The .txt files in the **obj_train_data/** and **obj_valid_data/** directories contain the BBox label information of the corresponding images. Each line indicates a BBox label.

```
# image1.txt:
# <label_index> <x_center> <y_center> <width> <height>
0 0.250000 0.400000 0.300000 0.400000
3 0.600000 0.400000 0.400000 0.266667
```

x_center, **y_center**, **width**, and **height** indicate the normalized parameters for the target bounding box: the x-coordinate and y-coordinate of the center point, width, and height.

- Only images in JPG, JPEG, PNG, and BMP formats are supported. A single image cannot exceed 5 MB, and the total size of all images uploaded at a time cannot exceed 8 MB.

Text Classification

txt and csv files can be imported for text classification, with the text encoding format of UTF-8 or GBK.

Labeled objects and labels for text classification can be stored in two formats:

- ModelArts text classification combine 1.0: The labeled objects and labels for text classification are in the same text file. You can specify a separator to separate the labeled objects and labels, as well as multiple labels.

For example, the following shows an example text file. The **Tab** key is used to separate the labeled objects from the labels.

```
It touches good and responds quickly. I don't know how it performs in the future. positive
Three months ago, I bought a very good phone and replaced my old one with it. It can operate longer
between charges. positive
Why does my phone heat up if I charge it for a while? The volume button stuck after being pressed
down. negative
It's a gift for Father's Day. The delivery is fast and I received it in 24 hours. I like the earphones
because the bass sounds feel good and they would not fall off. positive
```

- ModelArts text classification 1.0: The labeled objects and labels for text classification are text files, and correspond to each other based on the rows. For example, the first row in a label file indicates the label of the first row in the file of the labeled object.

For example, the content of the labeled object **COMMENTS_20180919_114745.txt** is as follows:

```
It touches good and responds quickly. I don't know how it performs in the future.
Three months ago, I bought a very good phone and replaced my old one with it. It can operate longer
between charges.
Why does my phone heat up if I charge it for a while? The volume button stuck after being pressed
down.
It's a gift for Father's Day. The delivery is fast and I received it in 24 hours. I like the earphones
because the bass sounds feel good and they would not fall off.
```

The content of the label file **COMMENTS_20180919_114745_result.txt** is as follows:

```
positive
negative
negative
positive
```

This data format requires you to store labeled objects and your label files (in one-to-one relationship with the labeled objects) in the same directory. For example, if the name of the labeled object file is **COMMENTS_20180919_114745.txt**, the name of the label file must be **COMMENTS_20180919_114745_result.txt**.

Example of data files:

```
dataset-import-example
COMMENTS_20180919_114732.txt
COMMENTS_20180919_114732_result.txt
COMMENTS_20180919_114745.txt
COMMENTS_20180919_114745_result.txt
COMMENTS_20180919_114945.txt
COMMENTS_20180919_114945_result.txt
```

Sound Classification

ModelArts audio classification dir 1.0: Sound files with the same label must be stored in the same directory, and the label name is the directory name.

Example:

```
dataset-import-example
├── Cat
│   ├── 10.wav
│   ├── 11.wav
│   └── 12.wav
└── Dog
    ├── 1.wav
    ├── 2.wav
    └── 3.wav
```

Tables

CSV files can be imported from OBS. Select the directory where the files are stored. The number of columns in the CSV file must be the same as that in the dataset schema. The schema of the CSV file can be automatically obtained.

```
dataset-import-example
├── table_import_1.csv
├── table_import_2.csv
├── table_import_3.csv
└── table_import_4.csv
```

4.2.4 Importing a Manifest File

Prerequisites

- You have created a dataset.
- You have stored the data to be imported in OBS. You have stored the manifest file in OBS.
- The OBS bucket and ModelArts are in the same region and you can operate the bucket.

Importing File Data from a Manifest File

The parameters for data import vary according to the dataset type. The following uses an image dataset as an example.

1. Log in to the . In the navigation pane, choose **Data Management > Datasets**.
2. Locate the row that contains the desired dataset and click **Import** in the **Operation** column. Alternatively, you can click the dataset name to go to the **Dashboard** tab page of the dataset, and click **Import** in the upper right corner.
3. In the **Import** dialog box, set the parameters as follows and click **OK**.
 - **Data Source:** OBS
 - **Import Mode:** manifest
 - **Manifest File:** OBS path for storing the manifest file
 - **Labeling Status:** Labeled

- **Advanced Feature Settings:** disabled by default
 - Import by Tag** The system automatically obtains the labels of the dataset. You can click **Add Label** to add a label. This parameter is optional. If **Import by Tag** is disabled, you can add or delete labels for imported data when labeling data.
 - Import Only Hard Examples:** If this parameter is selected, only the **hard** attribute data of the manifest file is imported.

Figure 4-7 Importing a manifest file

Import

* Data Source: OBS (selected), Local file

* Import Mode: Path, manifest (selected)

The manifest file needs to define the mapping between labeling objects and content. [Manifest file specifications and examples](#)

* Manifest File: Select the directory where the manifest file resides. [Folder icon]

* Labeling Status: Unlabeled, Labeled (selected)

Advanced Feature Settings: Import Only Hard Examples (disabled), Import by Tag (disabled)

OK Cancel

After the data is imported, it will be automatically synchronized to the dataset. On the **Datasets** page, click the dataset name to view its details and create a labeling job to label the data.

Labeling Status of File Data

The labeling status can be **Unlabeled** or **Labeled**.

- **Unlabeled:** Only the labeling object (such as unlabeled images or texts) is imported.
- **Labeled:** Both the labeling object and content are imported. Labeling content importing is not supported for datasets in free format.

To ensure that the labeling content can be correctly read, you must store data in strict accordance with the specifications.

If **Import Mode** is set to **Path**, store the data to be imported according to the labeling file specifications.

If **Import Mode** is set to **manifest**, the manifest file specifications must be met. For details, see [Specifications for Importing a Manifest File](#).

 NOTE

If the labeling status is set to **Labeled**, ensure that the folder or manifest file complies with the format specifications. Otherwise, the import may fail.

4.2.5 Specifications for Importing a Manifest File

The manifest file defines the mapping between labeled objects and content. The manifest file import mode means that the manifest file is used for dataset import. The manifest file can be imported from OBS. When importing a manifest file from OBS, ensure that you have the permissions to access the directory where the manifest file is stored.

 NOTE

There are many requirements on the manifest file compilation. Import new data from OBS. Generally, manifest file import is used for data migration of ModelArts in different regions or using different accounts. If you have labeled data in a region using ModelArts, you can obtain the manifest file of the published dataset from the output path. Then you can import the dataset using the manifest file to ModelArts of other regions or accounts. The imported data carries the labeling information and does not need to be labeled again, improving development efficiency.

The manifest file that contains information about the original file and labeling can be used in labeling, training, and inference scenarios. The manifest file that contains only information about the original file can be used in inference scenarios or used to generate an unlabeled dataset. The manifest file must meet the following requirements:

- The manifest file uses the UTF-8 encoding format.
- The manifest file uses the JSON Lines format (jsonlines.org). A line contains one JSON object.

```
{"source": "/path/to/image1.jpg", "annotation": ... }  
{"source": "/path/to/image2.jpg", "annotation": ... }  
{"source": "/path/to/image3.jpg", "annotation": ... }
```

In the preceding example, the manifest file contains multiple lines of JSON object.

- The manifest file can be generated by you, third-party tools, or ModelArts Data Labeling. The file name can be any valid file name. To facilitate the internal use of the ModelArts system, the file name generated by the ModelArts data labeling function consists of the following strings: **DatasetName-VersionName.manifest**. For example, **animal-v201901231130304123.manifest**.

Image Classification

```
{  
  "source": "s3://path/to/image1.jpg",  
  "usage": "TRAIN",  
  "id": "0162005993f8065ef47eefb59d1e4970",  
  "annotation": [  
    {  
      "type": "modelarts/image_classification",  
      "name": "cat",  
      "property": {  
        "color": "white",  
        "kind": "Persian cat"  
      }  
    }  
  ],  
}
```

```

    "annotated-by":"human",
    "creation-time":"2019-01-23 11:30:30"
  },
  {
    "type": "modelarts/image_classification",
    "name":"animal",
    "annotated-by":"modelarts/active-learning",
    "confidence": 0.8,
    "creation-time":"2019-01-23 11:30:30"
  }
],
"inference-loc":"/path/to/inference-output"
}

```

Table 4-2 Parameters

| Parameter | Mandatory | Description |
|---------------|-----------|---|
| source | Yes | URI of an object to be labeled. For details about data source types and examples, see Table 4-3 . |
| usage | No | By default, the parameter value is left blank. Possible values are as follows: <ul style="list-style-type: none"> ● TRAIN: The object is used for training. ● EVAL: The object is used for evaluation. ● TEST: The object is used for testing. ● INFERENCE: The object is used for inference. If the parameter value is left blank, you decide how to use the object. |
| id | No | Sample ID exported from the system. You do not need to set this parameter when importing the sample. |
| annotation | No | If the parameter value is left blank, the object is not labeled. The value of annotation consists of an object list. For details about the parameters, see Table 4-4 . |
| inference-loc | No | This parameter is available when the file is generated by the inference service, indicating the location of the inference result file. |

Table 4-3 Data source types

| Type | Example |
|---------|--|
| OBS | "source":"s3://path-to-jpg" |
| Content | "source":"content://I love machine learning" |

Table 4-4 annotation objects

| Parameter | Mandatory | Description |
|---------------|-----------|---|
| type | Yes | Label type. Possible values are as follows: <ul style="list-style-type: none"> ● image_classification: image classification ● text_classification: text classification ● text_entity: named entity recognition ● object_detection: object detection ● audio_classification: sound classification ● audio_content: speech labeling ● audio_segmentation: speech paragraph labeling |
| name | Yes/No | This parameter is mandatory for the classification type but optional for other types. This example uses the image classification type. |
| id | Yes/No | Label ID. This parameter is mandatory for triplets but optional for other types. The entity label ID of a triplet is in E+number format, for example, E1 and E2 . The relationship label ID of a triplet is in R+number format, for example, R1 and R2 . |
| property | No | Labeling property. In this example, the cat has two properties: color and kind. |
| annotated-by | No | The default value is human , indicating manual labeling. <ul style="list-style-type: none"> ● human |
| creation-time | No | Time when the labeling job was created. It is the time when labeling information was written, not the time when the manifest file was generated. |
| confidence | No | Confidence score of machine labeling. The value ranges from 0 to 1. |

Text Classification

```
{
  "source": "content://I like this product ",
  "id": "XGDVGS",
  "annotation": [
    {
      "type": "modelarts/text_classification",
      "name": "positive",
      "annotated-by": "human",
      "creation-time": "2019-01-23 11:30:30"
    }
  ]
}
```

The **content** parameter indicates the text to be labeled. The other parameters are the same as those described in [Image Classification](#). For details, see [Table 4-2](#).

Named Entity Recognition

```
{
  "source": "content://Michael Jordan is the most famous basketball player in the world.",
  "usage": "TRAIN",
  "annotation": [
    {
      "type": "modelarts/text_entity",
      "name": "Person",
      "property": {
        "@modelarts:start_index": 0,
        "@modelarts:end_index": 14
      },
      "annotated-by": "human",
      "creation-time": "2019-01-23 11:30:30"
    },
    {
      "type": "modelarts/text_entity",
      "name": "Category",
      "property": {
        "@modelarts:start_index": 34,
        "@modelarts:end_index": 44
      },
      "annotated-by": "human",
      "creation-time": "2019-01-23 11:30:30"
    }
  ]
}
```

The parameters such as **source**, **usage**, and **annotation** are the same as those described in [Image Classification](#). For details, see [Table 4-2](#).

[Table 4-5](#) describes the property parameters. For example, if you want to extract **Michael** from "**source**": "**content**://**Michael Jordan**", the value of **start_index** is **0** and that of **end_index** is **7**.

Table 4-5 property parameters

| Parameter | Data type | Description |
|------------------------|-----------|---|
| @modelarts:start_index | Integer | Start position of the text. The value starts from 0, including the characters specified by start_index . |
| @modelarts:end_index | Integer | End position of the text, excluding the characters specified by end_index . |

Text Triplet

```
{
  "source": "content://\"Three Body\" is a series of long science fiction novels created by Liu Cix.",
  "usage": "TRAIN",
  "annotation": [
    {
      "type": "modelarts/text_entity",
      "name": "Person",
      "id": "E1",
      "property": {
        "@modelarts:start_index": 67,
        "@modelarts:end_index": 74
      },
      "annotated-by": "human",
    }
  ]
}
```

```

"creation-time":"2019-01-23 11:30:30"
},
{
  "type":"modelarts/text_entity",
  "name":"Book",
  "id":"E2",
  "property":{
    "@modelarts:start_index":0,
    "@modelarts:end_index":12
  },
  "annotated-by":"human",
  "creation-time":"2019-01-23 11:30:30"
},
{
  "type":"modelarts/text_triplet",
  "name":"Author",
  "id":"R1",
  "property":{
    "@modelarts:from":"E1",
    "@modelarts:to":"E2"
  },
  "annotated-by":"human",
  "creation-time":"2019-01-23 11:30:30"
},
{
  "type":"modelarts/text_triplet",
  "name":"Works",
  "id":"R2",
  "property":{
    "@modelarts:from":"E2",
    "@modelarts:to":"E1"
  },
  "annotated-by":"human",
  "creation-time":"2019-01-23 11:30:30"
}
]

```

The parameters such as **source**, **usage**, and **annotation** are the same as those described in [Image Classification](#). For details, see [Table 4-2](#).

[Table 5 property parameters](#) describes the **property** parameters.

@modelarts:start_index and **@modelarts:end_index** are the same as those of named entity recognition. For example, when **source** is set to **content://"Three Body" is a series of long science fiction novels created by Liu Cix., Liu Cix** is an entity person, **Three Body** is an entity book, the person is the author of the book, and the book is works of the person.

Table 4-6 property parameters

| Parameter | Data type | Description |
|------------------------|-----------|---|
| @modelarts:start_index | Integer | Start position of the triplet entities. The value starts from 0, including the characters specified by start_index . |
| @modelarts:end_index | Integer | End position of the triplet entities, excluding the characters specified by end_index . |
| @modelarts:from | String | Start entity ID of the triplet relationship |

| Parameter | Data type | Description |
|---------------|-----------|--|
| @modelarts:to | String | Entity ID pointed to in the triplet relationship |

Object Detection

```
{
  "source": "s3://path/to/image1.jpg",
  "usage": "TRAIN",
  "annotation": [
    {
      "type": "modelarts/object_detection",
      "annotation-loc": "s3://path/to/annotation1.xml",
      "annotation-format": "PASCAL VOC",
      "annotated-by": "human",
      "creation-time": "2019-01-23 11:30:30"
    }
  ]
}
```

- The parameters such as **source**, **usage**, and **annotation** are the same as those described in [Image Classification](#). For details, see [Table 4-2](#).
- **annotation-loc** indicates the path for saving the label file. This parameter is mandatory for object detection and image segmentation but optional for other labeling types.
- **annotation-format** indicates the format of the label file. This parameter is optional. The default value is **PASCAL VOC**. Only **PASCAL VOC** is supported.

Table 4-7 PASCAL VOC format parameters

| Parameter | Mandatory | Description |
|-----------|-----------|--|
| folder | Yes | Directory where the data source is located |
| filename | Yes | Name of the file to be labeled |
| size | Yes | Image pixel <ul style="list-style-type: none"> • width: image width. This parameter is mandatory. • height: image height. This parameter is mandatory. • depth: number of image channels. This parameter is mandatory. |
| segmented | Yes | Segmented or not |

| Parameter | Mandatory | Description |
|-----------|-----------|---|
| object | Yes | <p>Object detection information. Multiple object{} functions are generated for multiple objects.</p> <ul style="list-style-type: none"> • name: type of the labeled content. This parameter is mandatory. • pose: shooting angle of the labeled content. This parameter is mandatory. • truncated: whether the labeled content is truncated (0 indicates that the content is not truncated). This parameter is mandatory. • occluded: whether the labeled content is occluded (0 indicates that the content is not occluded). This parameter is mandatory. • difficult: whether the labeled object is difficult to identify (0 indicates that the object is easy to identify). This parameter is mandatory. • confidence: confidence score of the labeled object. The value ranges from 0 to 1. This parameter is optional. • bndbox: bounding box type. This parameter is mandatory. For details about the possible values, see Table 4-8. |

Table 4-8 Bounding box types

| Parameter | Shape | Labeling information |
|-----------|-----------|--|
| point | Point | <p>Coordinates of a point</p> <pre><x>100<x> <y>100<y></pre> |
| line | Line | <p>Coordinates of points</p> <pre><x1>100<x1> <y1>100<y1> <x2>200<x2> <y2>200<y2></pre> |
| bndbox | Rectangle | <p>Coordinates of the upper left and lower right points</p> <pre><xmin>100<xmin> <ymin>100<ymin> <xmax>200<xmax> <ymin>200<ymin></pre> |

| Parameter | Shape | Labeling information |
|-----------|---------|--|
| polygon | Polygon | Coordinates of points <x1>100<x1> <y1>100<y1> <x2>200<x2> <y2>100<y2> <x3>250<x3> <y3>150<y3> <x4>200<x4> <y4>200<y4> <x5>100<x5> <y5>200<y5> <x6>50<x6> <y6>150<y6> |
| circle | Circle | Center coordinates and radius <cx>100<cx> <cy>100<cy> <r>50<r> |

Example:

```

<annotation>
  <folder>test_data</folder>
  <filename>260730932.jpg</filename>
  <size>
    <width>767</width>
    <height>959</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>point</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <occluded>0</occluded>
    <difficult>0</difficult>
    <point>
      <x1>456</x1>
      <y1>596</y1>
    </point>
  </object>
  <object>
    <name>line</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <occluded>0</occluded>
    <difficult>0</difficult>
    <line>
      <x1>133</x1>
      <y1>651</y1>
      <x2>229</x2>
      <y2>561</y2>
    </line>
  </object>

```

```
<object>
  <name>bag</name>
  <pose>Unspecified</pose>
  <truncated>0</truncated>
  <occluded>0</occluded>
  <difficult>0</difficult>
  <bndbox>
    <xmin>108</xmin>
    <ymin>101</ymin>
    <xmax>251</xmax>
    <ymax>238</ymax>
  </bndbox>
</object>
<object>
  <name>boots</name>
  <pose>Unspecified</pose>
  <truncated>0</truncated>
  <occluded>0</occluded>
  <difficult>0</difficult>

  <polygon>
    <x1>373</x1>
    <y1>264</y1>
    <x2>500</x2>
    <y2>198</y2>
    <x3>437</x3>
    <y3>76</y3>
    <x4>310</x4>
    <y4>142</y4>
  </polygon>
</object>
<object>
  <name>circle</name>
  <pose>Unspecified</pose>
  <truncated>0</truncated>
  <occluded>0</occluded>
  <difficult>0</difficult>
  <circle>
    <cx>405</cx>
    <cy>170</cy>
    <r>100</r>
  </circle>
</object>
</annotation>
```

Sound Classification

```
{
  "source":
  "s3://path/to/pets.wav",
  "annotation": [
    {
      "type": "modelarts/audio_classification",
      "name": "cat",
      "annotated-by": "human",
      "creation-time": "2019-01-23 11:30:30"
    }
  ]
}
```

The parameters such as **source**, **usage**, and **annotation** are the same as those described in [Image Classification](#). For details, see [Table 4-2](#).

Speech Labeling

```
{
  "source": "s3://path/to/audio1.wav",
  "annotation": [
```

```
{
  "type":"modelarts/audio_content",
  "property":{
    "@modelarts:content":"Today is a good day."
  },
  "annotated-by":"human",
  "creation-time":"2019-01-23 11:30:30"
}
]
```

- The parameters such as **source**, **usage**, and **annotation** are the same as those described in [Image Classification](#). For details, see [Table 4-2](#).
- The **@modelarts:content** parameter in **property** indicates speech content. The data type is **String**.

Speech Paragraph Labeling

```
{
  "source":"s3://path/to/audio1.wav",
  "usage":"TRAIN",
  "annotation":[
    {
      "type":"modelarts/audio_segmentation",
      "property":{
        "@modelarts:start_time":"00:01:10.123",
        "@modelarts:end_time":"00:01:15.456",
        "@modelarts:source":"Tom",
        "@modelarts:content":"How are you?"
      },
      "annotated-by":"human",
      "creation-time":"2019-01-23 11:30:30"
    },
    {
      "type":"modelarts/audio_segmentation",
      "property":{
        "@modelarts:start_time":"00:01:22.754",
        "@modelarts:end_time":"00:01:24.145",
        "@modelarts:source":"Jerry",
        "@modelarts:content":"I'm fine, thank you."
      },
      "annotated-by":"human",
      "creation-time":"2019-01-23 11:30:30"
    }
  ]
}
```

- The parameters such as **source**, **usage**, and **annotation** are the same as those described in [Image Classification](#). For details, see [Table 4-2](#).
- [Table 4-9](#) describes the **property** parameters.

Table 4-9 property parameters

| Parameter | Data type | Description |
|-----------------------|-----------|---|
| @modelarts:start_time | String | Start time of the sound. The format is hh:mm:ss.SSS . hh indicates the hour, mm indicates the minute, ss indicates the second, and SSS indicates the millisecond. |

| Parameter | Data type | Description |
|---------------------|-----------|---|
| @modelarts:end_time | String | End time of the sound. The format is hh:mm:ss.SSS . hh indicates the hour, mm indicates the minute, ss indicates the second, and SSS indicates the millisecond. |
| @modelarts:source | String | Sound source |
| @modelarts:content | String | Sound content |

4.3 Importing Data from Local Files

Prerequisites

- You have created a dataset.
- You have created an OBS bucket. The OBS bucket and ModelArts are in the same region and you can operate the bucket.

Import Operation

Both file and table data can be uploaded from local files. The data uploaded from local files should be stored in an OBS directory. You must have created an OBS bucket.

In a single batch upload, a maximum of 100 files can be uploaded at a time, and the total size of the files cannot exceed 5 GB.

The parameters on the GUI for data import vary according to the dataset type. The following uses a dataset of the image classification type as an example.

1. Log in to the . In the navigation pane, choose **Data Management > Datasets**.
2. Locate the row that contains the desired dataset and click **Import** in the **Operation** column.

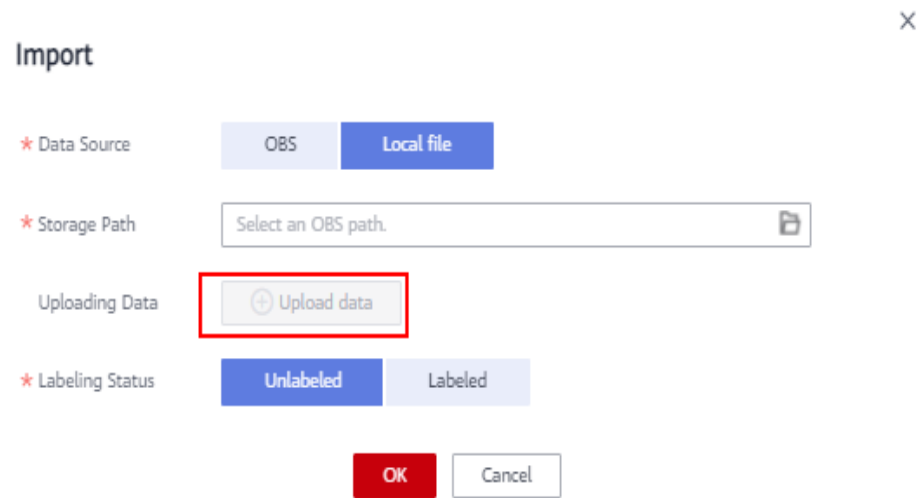
Figure 4-8 Importing data

| Name | Version | Labeling Progress | Created | Description | Operation |
|---|---------|-------------------|---------------------------------|-------------|--|
| def-dataset-image-20200519-001 ydf14i20Mbj256RCKEB | V003 | 75.00% (12/16) | May 19, 2020 15:56:12 GMT+08:00 | -- | Import Publish Labeling Export Delete More |

Alternatively, you can click the dataset name to go to the **Dashboard** tab page of the dataset, and click **Import** in the upper right corner.

3. In the **Import** dialog box, set the parameters as follows and click **OK**.
 - **Data Source:** Local file
 - **Storage Path:** Select an OBS path.
 - **Uploading Data:** Click **Upload data**, upload local data, and click **OK**.

Figure 4-9 Importing data from local files



5 Data Analysis and Preview

Generally, the quality of raw data cannot meet training requirements, for example, invalid or duplicate data exists. To help you improve data quality, ModelArts provides the following capabilities:

- **Data Filtering**: enables you to filter data based on sample attributes and auto grouping results.
- **Data Feature Analysis**: analyzes data features or labeling results, such as the brightness and bounding box distribution, helping you analyze data balance and improve the model training effect.

5.1 Data Filtering

On the **Dashboard** tab page of the dataset, the summary of the dataset is displayed by default. In the upper right corner of the page, click **Label**. The dataset details page is displayed, showing all data in the dataset by default. On the **All**, **Unlabeled**, or **Labeled** tab page, you can add filter criteria in the filter criteria area to quickly filter the data you want to view.

The following filter criteria are supported. You can set one or more filter criteria.

- **Label**: Select **All** or one or more labels you specified.
- **Sample Creation Time**: Select **Within 1 month**, **Within 1 day**, or **Custom** to customize a time range.
- **File Name** or **Path**: Filter files by file name or file storage path.
- **Labeled By**: Select the name of the user who labeled the image.

5.2 Data Feature Analysis

Images or target bounding boxes are analyzed based on image features, such as blurs and brightness to draw visualized curves to help process datasets.

You can also select multiple versions of a dataset to view their curves for comparison and analysis.

Background

- Data feature analysis is only available for image datasets of the image classification and object detection types.
- Data feature analysis is only available for the published datasets. The published dataset versions in **Default** format support data feature analysis.
- A data scope for feature analysis varies depending on the dataset type.
 - In a dataset of the object detection type, if the number of labeled samples is 0, the **View Data Feature** tab page is unavailable and data features are not displayed after a version is published. After the images are labeled and the version is published, the data features of the labeled images are displayed.
 - In a dataset of the image classification type, if the number of labeled samples is 0, the **View Data Feature** tab page is unavailable and data features are not displayed after a version is published. After the images are labeled and the version is published, the data features of all images are displayed.
- The analysis result is valid only when the number of images in a dataset reaches a certain level. Generally, more than 1,000 images are required.
- Image classification supports the following data feature metrics: **Resolution, Aspect Ratio, Brightness, Saturation, Blur Score, and Colorfulness** Object detection supports all data feature metrics. [Supported Data Feature Metrics](#) provides all data feature metrics supported by ModelArts.

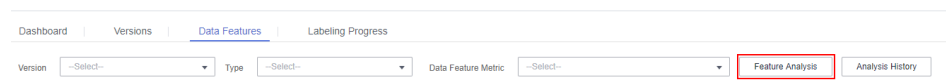
Data Feature Analysis

1. Log in to the . In the navigation pane, choose **Data Management > Datasets**.
2. Select a dataset and click **Data Features** in the **Operation** column. The **Data Features** tab page of the dataset page is displayed.

You can also click a dataset name to go to the dataset page and click the **Data Features** tab.

3. By default, feature analysis is not started for published datasets. You need to manually start feature analysis tasks for each dataset version. On the **Data Features** tab page, click **Feature Analysis**.

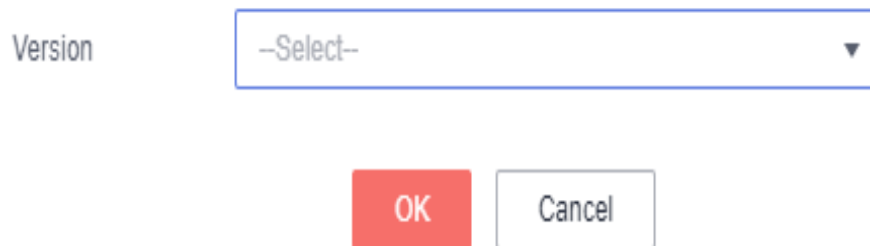
Figure 5-1 Feature Analysis



4. In the dialog box that is displayed, configure the dataset version for feature analysis and click **OK** to start analysis.

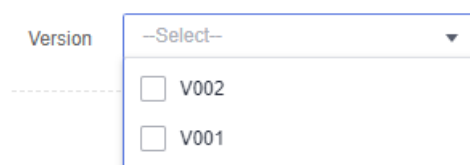
Version: Select a published version of the dataset.

Figure 5-2 Starting a data feature analysis task



5. After a data feature analysis task is started, it takes a certain period of time to complete, depending on the data volume. If the selected version is displayed in the **Version** drop-down list and can be selected, the analysis is complete.

Figure 5-3 Selecting a version for which feature analysis has been performed



6. View the data feature analysis result.

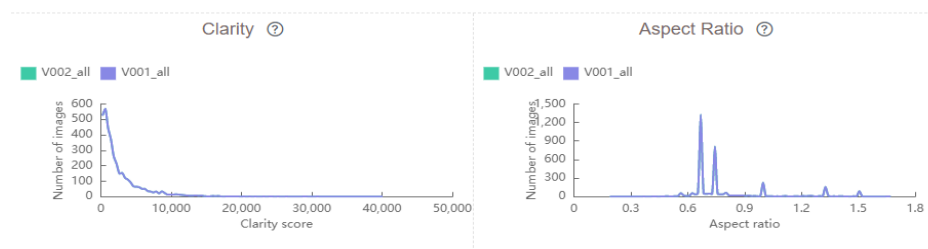
Version: Select the version to be compared from the drop-down list. You can also select only one version.

Type: Select the type to be analyzed. The value can be **all**, **train**, **eval**, or **inference**.

Data Feature Metric: Select metrics to be displayed from the drop-down list. For details, see [Supported Data Feature Metrics](#).

Then, the selected version and metrics are displayed on the page, as shown in [Figure 5-4](#). The displayed chart helps you understand data distribution for better data processing.

Figure 5-4 Data feature analysis



7. View historical records of the analysis task.

After data feature analysis is complete, you can click **Task History** on the right of the **Data Features** tab page to view historical analysis tasks and their statuses in the dialog box that is displayed.

Figure 5-5 Viewing the task history

View Task History

| Dataset Vers... | Task ID | Created | Duration(hh:... | Status |
|-----------------|---------------|------------------|-----------------|------------|
| V002 | Rdnjwum33T... | Apr 03, 2020 ... | 00:01:50 | Successful |
| V001 | gpw2hG6D6... | Apr 02, 2020 ... | 00:02:23 | Successful |

Supported Data Feature Metrics

Table 5-1 Data feature metrics

| Metric | Description | Explanation |
|--------------|---|--|
| Resolution | Image resolution. An area value is used as a statistical value. | Metric analysis results are used to check whether there is an offset point. If an offset point exists, you can resize or delete the offset point. |
| Aspect Ratio | An aspect ratio is a proportional relationship between an image's width and height. | The chart of the metric is in normal distribution, which is generally used to compare the difference between the training set and the dataset used in the real scenario. |

| Metric | Description | Explanation |
|------------------------|---|---|
| Brightness | Brightness is the perception elicited by the luminance of a visual target. A larger value indicates better image brightness. | The chart of the metric is in normal distribution. You can determine whether the brightness of the entire dataset is high or low based on the distribution center. You can adjust the brightness based on your application scenario. For example, if the application scenario is night, the brightness should be lower. |
| Saturation | Color saturation of an image. A larger value indicates that the entire image color is easier to distinguish. | The chart of the metric is in normal distribution, which is generally used to compare the difference between the training set and the dataset used in the real scenario. |
| Blur Score Clarity | Image clarity, which is calculated using the Laplace operator. A larger value indicates clearer edges and higher clarity. | You can determine whether the clarity meets the requirements based on the application scenario. For example, if data is collected from HD cameras, the clarity must be higher. You can sharpen or blur the dataset and add noises to adjust the clarity. |
| Colorfulness | Horizontal coordinate: Colorfulness of an image. A larger value indicates richer colors. Vertical coordinate: Number of images | Colorfulness on the visual sense, which is generally used to compare the difference between the training set and the dataset used in the real scenario. |
| Bounding Box Number | Horizontal coordinate: Number of bounding boxes in an image Vertical coordinate: Number of images | It is difficult for a model to detect a large number of bounding boxes in an image. Therefore, more images containing many bounding boxes are required for training. |

| Metric | Description | Explanation |
|---|--|---|
| <p>Std of Bounding Boxes Area Per Image</p> <p>Standard Deviation of Bounding Boxes Per Image</p> | <p>Horizontal coordinate: Standard deviation of bounding boxes in an image. If an image has only one bounding box, the standard deviation is 0. A larger standard deviation indicates higher bounding box size variation in an image.</p> <p>Vertical coordinate: Number of images</p> | <p>It is difficult for a model to detect a large number of bounding boxes with different sizes in an image. You can add data for training based on scenarios or delete data if such scenarios do not exist.</p> |
| <p>Aspect Ratio of Bounding Boxes</p> | <p>Horizontal coordinate: Aspect ratio of the target bounding boxes</p> <p>Vertical coordinate: Number of bounding boxes in all images</p> | <p>The chart of the metric is generally in Poisson distribution, which is closely related to application scenarios. This metric is mainly used to compare the differences between the training set and the validation set. For example, if the training set is a rectangle, the result will be significantly affected if the validation set is close to a square.</p> |
| <p>Area Ratio of Bounding Boxes</p> | <p>Horizontal coordinate: Area ratio of the target bounding boxes, that is, the ratio of the bounding box area to the entire image area. A larger value indicates a higher ratio of the object in the image.</p> <p>Vertical coordinate: Number of bounding boxes in all images</p> | <p>The metric is used to determine the distribution of anchors used in the model. If the target bounding box is large, set the anchor to a large value.</p> |

| Metric | Description | Explanation |
|--|---|--|
| Marginalization Value of Bounding Boxes | <p>Horizontal coordinate: Marginalization degree, that is, the ratio of the distance between the center point of the target bounding box and the center point of the image to the total distance of the image. A larger value indicates that the object is closer to the edge. (The total distance of an image is the distance from the intersection point of a ray (that starts from the center point of the image and passes through the center point of the bounding box) and the image border to the center point of the image.)</p> <p>Vertical coordinate: Number of bounding boxes in all images</p> | Generally, the chart of the metric is in normal distribution. The metric is used to determine whether an object is at the edge of an image. If a part of an object is at the edge of an image, you can add a dataset or do not label the object. |
| Overlap Score of Bounding Boxes Overlap Score of Bounding Boxes | <p>Horizontal coordinate: Overlap degree, that is, the part of a single bounding box overlapped by other bounding boxes. The value ranges from 0 to 1. A larger value indicates that more parts are overlapped by other bounding boxes.</p> <p>Vertical coordinate: Number of bounding boxes in all images</p> | The metric is used to determine the overlapping degree of objects to be detected. Overlapped objects are difficult to detect. You can add a dataset or do not label some objects based on your needs. |
| Brightness of Bounding Boxes Brightness of Bounding Boxes | <p>Horizontal coordinate: Brightness of the image in the target bounding box. A larger value indicates brighter image.</p> <p>Vertical coordinate: Number of bounding boxes in all images</p> | Generally, the chart of the metric is in normal distribution. The metric is used to determine the brightness of an object to be detected. In some special scenarios, the brightness of an object is low and may not meet the requirements. |

| Metric | Description | Explanation |
|---|---|---|
| Blur Score of Bounding Boxes Clarity of Bounding Boxes | Horizontal coordinate: Clarity of the image in the target bounding box. A larger value indicates higher image clarity. Vertical coordinate: Number of bounding boxes in all images | The metric is used to determine whether the object to be detected is blurred. For example, a moving object may become blurred during collection and its data needs to be collected again. |

6 Labeling Data

Model training requires a large amount of labeled data. Therefore, before training a model, label data. You can create a manual labeling job labeled by one person or by a group of persons (team labeling), or enable auto labeling to quickly label images. You can also modify existing labels, or delete them and re-label.

- Manual labeling: allows you to manually label data.

For details about data labeling, see .

7 Publishing Data

7.1 Introduction to Data Publishing

ModelArts distinguishes data of the same source according to versions processed or labeled at different time, which facilitates the selection of dataset versions for subsequent model building and development.

About Dataset Versions

- For a newly created dataset (before publishing), there is no dataset version information. The dataset must be published before being used for model development or training.
- The default naming rules of dataset versions are V001 and V002 in ascending order. You can customize the version number during publishing.
- You can set any version to the current version. Then the details of the version are displayed on the dataset details page.
- You can obtain the dataset in the manifest file format corresponding to each dataset version based on the value of **Storage Path**. The dataset can be used when you import data or filter hard examples.
- The version of a table dataset cannot be changed.

7.2 Publishing a Data Version

1. Log in to the . In the navigation pane, choose **Data Management > Datasets**.
2. Locate the row containing the target dataset and click **Publish** in the **Operation** column. Alternatively, click the dataset name to go to the **Dashboard** tab page of the dataset, and click **Publish** in the upper right corner.
3. In the displayed dialog box, set the parameters and click **OK**.

Figure 7-1 Publishing a dataset version

Publish New Version

* Version

⚠ Before you enable splitting, ensure each label has at least five labeled samples. Ensure there are at least two multi-label samples, if any. For details, go to the Dashboard tab page.

* Labeling Type Image cl... Object d... Image se... Free for...

Description

0/256

Table 7-1 Parameters for publishing a dataset

| Parameter | Description |
|-----------|---|
| Version | The naming rules of V001 and V002 in ascending order are used by default. A version name can be customized. Only letters, digits, hyphens (-), and underscores (_) are allowed. |
| Format | Only table datasets support version format setting. Available values are CSV and CarbonData . NOTE If the exported CSV file contains any command starting with =, +, -, or @, ModelArts automatically adds the Tab setting and escapes the double quotation marks (") for security purposes. |

| Parameter | Description |
|--------------|--|
| Splitting | <p>Only image classification, object detection, text classification, and sound classification datasets support data splitting.</p> <p>By default, this function is disabled. After this function is enabled, set the training and validation ratios.</p> <p>Enter a value ranging from 0 to 1 for Training Set Ratio. After the training set ratio is set, the validation set ratio is determined. The sum of the training set ratio and the validation set ratio is 1.</p> <p>NOTE To ensure the model accuracy, you are advised to set the training set ratio to 0.8 or 0.9.</p> <p>The training set ratio is the ratio of sample data used for model training. The validation set ratio is the ratio of the sample data used for model validation. The training and validation ratios affect the performance of training templates.</p> |
| Description | Description of the current dataset version. |
| Hard Example | <p>Only image classification and object detection datasets support hard example attributes.</p> <p>By default, this function is disabled. After this function is enabled, information such as the hard example attributes of the dataset are written to the corresponding manifest file.</p> |

Directory Structure of Dataset Versions

Datasets are managed based on OBS directories. After a new version is published, the directory is generated based on the new version in the output dataset path.

Take an image classification dataset as an example. After the dataset is published, the directory structure of related files generated in OBS is as follows:

```
|-- user-specified-output-path
    |-- DatasetName-datasetId
        |-- annotation
            |-- VersionName1
                |-- VersionName1.manifest
            |-- VersionName2
            ...
        |-- ...
```

7.3 Managing Data Versions

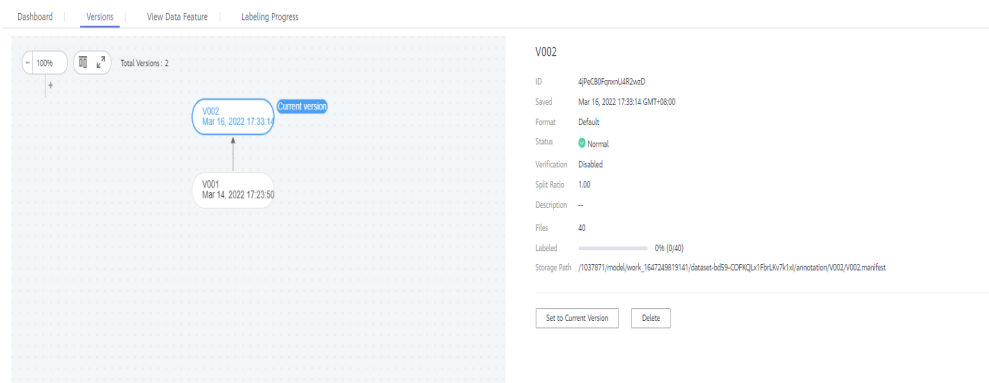
During data preparation, you can publish data into multiple versions for dataset management. You can view version updates, set the current version, and delete versions.

Viewing Dataset Version Updates

1. Log in to the . In the navigation pane, choose **Data Management > Datasets**.
2. In the dataset list, choose **More > Manage Version** in the **Operation** column. The **Manage Version** tab page is displayed.

You can view basic information about the dataset, and view the versions and publish time on the left.

Figure 7-2 Viewing dataset versions



Setting to Current Version

1. Log in to the . In the navigation pane, choose **Data Management > Datasets**.
2. In the dataset list, choose **More > Manage Version** in the **Operation** column. The **Manage Version** tab page is displayed.
3. On the **Manage Version** tab page, select the desired dataset version, and click **Set to Current Version** in the basic information area on the right. After the setting is complete, **Current version** is displayed to the right of the version name.

NOTE

Only the version in **Normal** status can be set to the current version.

Deleting a Dataset Version

1. Log in to the . In the navigation pane, choose **Data Management > Datasets**.
2. In the dataset list, choose **More > Manage Version** in the **Operation** column. The **Manage Version** tab page is displayed.
3. Locate the row that contains the target version, and click **Delete** in the **Operation** column. In the dialog box that is displayed, click **OK**.

NOTE

Deleting a dataset version does not remove the original data. Data and its labeling information are still stored in the OBS directory. However, this affects version management. Exercise caution when performing this operation.

8 Exporting Data

8.1 Introduction to Exporting Data

You can select data or filter data based on the filter criteria in a dataset and export to a new dataset or the specified OBS path. The historical export records can be viewed in task history.

Only datasets of image classification, object detection, and image segmentation types can be exported.

- For image classification datasets, only the label files in TXT format can be exported.
- For object detection datasets, only XML label files in Pascal VOC format can be exported.

8.2 Exporting Data to a New Dataset

1. Log in to the . In the navigation pane, choose **Data Management > Datasets**.
2. In the dataset list, select an image dataset and click the dataset name to go to the **Dashboard** tab page of the dataset.
3. Click **Export** in the upper right corner. In the displayed **Export To** dialog box, enter the related information and click **OK**.

Type: New Dataset.

Name: name of the new dataset

Storage Path: input path of the new dataset, that is, the OBS path where the data to be exported is stored

Output Path: output path of the new dataset, that is, the output path after labeling is complete. The output path cannot be the same as the storage path, and the output path cannot be a subdirectory of the storage path.

4. After the data is exported, view it in the specified path. After the data is exported, you can view the new dataset in the dataset list.
5. On the **Dashboard** tab page, click **Export History** in the upper right corner. In the displayed dialog box, view the task history of the dataset.

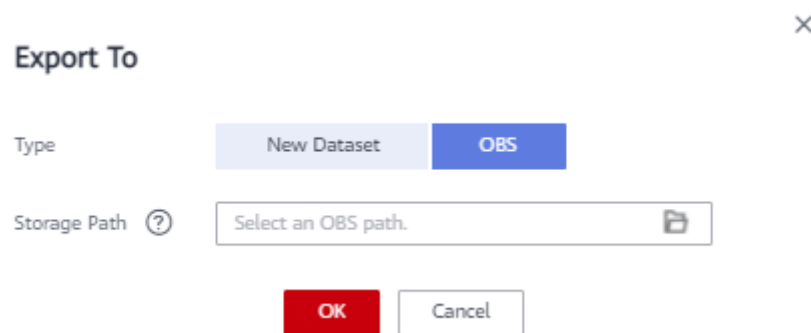
8.3 Exporting Data to OBS

1. Log in to the . In the navigation pane, choose **Data Management > Datasets**.
2. In the dataset list, select an image dataset and click the dataset name to go to the **Dashboard** tab page of the dataset.
3. Click **Export** in the upper right corner. In the displayed **Export To** dialog box, enter the related information and click **OK**.

Type: OBS.

Storage Path: path where the data to be exported is stored. You are advised not to save data to the input or output path of the current dataset.

Figure 8-1 Exporting data to OBS



4. After the data is exported, view it in the specified path.
5. On the **Dashboard** tab page, click **Export History** in the upper right corner. In the displayed dialog box, view the task history of the dataset.