MVTec 3D Data

Contents

MVTec 3D-AD Dataset (CC BY-NC-SA 4.0).

Description:

This script contains PyTorch Dataset, Dataloader and PyTorch Lightning DataModule for the MVTec 3D-AD dataset. If the dataset is not on the file system, the script downloads and extracts the dataset and create PyTorch data objects.

License:

MVTec 3D-AD dataset is released under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International

License (CC BY-NC-SA 4.0)(https://creativecommons.org/licenses/by-nc-sa/4.0/).

Reference:

- **Paul Bergmann, Xin Jin, David Sattlegger, Carsten Steger: The MVTec 3D-AD Dataset for Unsupervised 3D Anomaly**

  Detection and Localization in: Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP, 202-213, 2022, DOI: 10.5220/0010865000003124.

class **anomalib.data.depth.mvtec_3d.MVTec3D(*root*='./datasets/MVTec3D', *category*='bagel', *train_batch_size*=32, *eval_batch_size*=32, *num_workers*=8, *task*=TaskType.SEGMENTATION, *image_size*=None, *transform*=None, *train_transform*=None, *eval_transform*=None,**

***test_split_mode**=TestSplitMode.FROM_DIR, **test_split_ratio**=0.2, **val_split_mode**=ValSplitMode.SAME_AS_TEST, **val_split_ratio**=0.5, **seed**=None)*

Bases: `AnomalibDataModule`

MVTec Datamodule.

Parameters:

- **root** (*Path | str*) – Path to the root of the dataset Defaults to `"./datasets/MVTec3D"`.
- **category** (*str*) – Category of the MVTec dataset (e.g. “bottle” or “cable”). Defaults to `bagel`.
- **train_batch_size** (*int, optional*) – Training batch size. Defaults to `32`.
- **eval_batch_size** (*int, optional*) – Test batch size. Defaults to `32`.
- **num_workers** (*int, optional*) – Number of workers. Defaults to `8`.
- **task** (*TaskType*) – Task type, ‘classification’, ‘detection’ or ‘segmentation’ Defaults to `TaskType.SEGMENTATION`.
- **image_size** (*tuple[int, int], optional*) – Size to which input images should be resized. Defaults to `None`.
- **transform** (*Transform, optional*) – Transforms that should be applied to the input images. Defaults to `None`.
- **train_transform** (*Transform, optional*) – Transforms that should be applied to the input images during training. Defaults to `None`.
- **eval_transform** (*Transform, optional*) – Transforms that should be applied to the input images during evaluation. Defaults to `None`.
- **test_split_mode** (*TestSplitMode*) – Setting that determines how the testing subset is obtained. Defaults to `TestSplitMode.FROM_DIR`.
- **test_split_ratio** (*float*) – Fraction of images from the train set that will be reserved for testing. Defaults to `0.2`.
- **val_split_mode** (*ValSplitMode*) – Setting that determines how the validation subset is obtained. Defaults to `ValSplitMode.SAME_AS_TEST`.
- **val_split_ratio** (*float*) – Fraction of train or test images that will be reserved for validation. Defaults to `0.5`.
- **seed** (*int | None, optional*) – Seed which may be set to a fixed value for reproducibility. Defaults to `None`.

prepare_data()

Download the dataset if not available.

Return type:

None

class anomalib.data.depth.mvtec_3d.MVTec3DDataset(*task*, *root='./datasets/MVTec3D'*, *category='bagel'*, *transform=None*, *split=None*)

Bases: AnomalibDepthDataset

MVTec 3D dataset class.

Parameters:

- **task** (*TaskType*) – Task type, `classification`, `detection` or `segmentation`
- **root** (*Path* | *str*) – Path to the root of the dataset Defaults to `"./datasets/MVTec3D"`.
- **category** (*str*) – Sub-category of the dataset, e.g. 'bagel' Defaults to `"bagel"`.
- **transform** (*Transform, optional*) – Transforms that should be applied to the input images. Defaults to `None`.
- **split** (*str* | *Split* | *None*) – Split of the dataset, usually Split.TRAIN or Split.TEST Defaults to `None`.

anomalib.data.depth.mvtec_3d.make_mvtec_3d_dataset(*root*, *split=None*, *extensions=None*)

Create MVTec 3D-AD samples by parsing the MVTec AD data file structure.

The files are expected to follow this structure: - *path/to/dataset/split/category/image_filename.png* - *path/to/dataset/ground_truth/category/mask_filename.png*

This function creates a DataFrame to store the parsed information. The DataFrame follows this format:

	path	split	label	image_path	mask_path	label_index
0	datasets/ name	test	defect	filename.png	ground_truth/defect/ filename_mask.png	1

Parameters:

- **root** (*Path*) – Path to the dataset.
- **split** (*str | Split | None, optional*) – Dataset split (e.g., 'train' or 'test'). Defaults to `None`.
- **extensions** (*Sequence[str] | None, optional*) – List of file extensions to be included in the dataset. Defaults to `None`.

Examples

The following example shows how to get training samples from the MVTec 3D-AD 'bagel' category:

```
>>> from pathlib import Path
>>> root = Path('./MVTec3D')
>>> category = 'bagel'
>>> path = root / category
>>> print(path)
PosixPath('MVTec3D/bagel')
```

```
>>> samples = create_mvtec_3d_ad_samples(path, split='train')
>>> print(samples.head())
    path          split label image_path                          mask_path
    MVTec3D/bagel train good MVTec3D/bagel/train/good/rgb/105.png MVTec3D/bagel
    MVTec3D/bagel train good MVTec3D/bagel/train/good/rgb/017.png MVTec3D/bagel
```

Returns:

An output DataFrame containing the samples of the dataset.

Return type:

DataFrame

Previous
Folder 3D Data

Next
Data Utils