Synthetic Data Utils

Contents

Utilities to generate synthetic data.

anomalib.data.utils.generators.random_2d_perlin(*shape*, *res*, *fade=<function <lambda>>*)

Returns a random 2d perlin noise array.

Parameters:

- **shape** (*tuple*) – Shape of the 2d map.
- **res** (*tuple[int | torch.Tensor, int | torch.Tensor]*) – Tuple of scales for perlin noise for height and width dimension.
- **fade** (*_type_, optional*) – Function used for fading the resulting 2d map. Defaults to equation 6*t**5-15*t**4+10*t**3.

Returns:

Random 2d-array/tensor generated using perlin noise.

Return type:

np.ndarray | torch.Tensor

Augmenter module to generates out-of-distribution samples for the DRAEM implementation.

class anomalib.data.utils.augmenter.**Augmenter**(***anomaly_source_path***=*None*, ***p_anomalous***=*0.5*, ***beta***=*(0.2, 1.0)*)

Bases: `object`

Class that generates noisy augmentations of input images.

Parameters:

- **anomaly_source_path** (*str* | *None*) – Path to a folder of images that will be used as source of the anomalous
- **specified** (*noise. If not*) –
- **instead.** (*random noise will be used*) –
- **p_anomalous** (*float*) – Probability that the anomalous perturbation will be applied to a given image.
- **beta** (*float*) – Parameter that determines the opacity of the noise mask.

augment_batch(***batch***)

Generate anomalous augmentations for a batch of input images.

Parameters:

batch (*torch.Tensor*) – Batch of input images

Return type:

`tuple`[`Tensor`, `Tensor`]

Returns:

- Augmented image to which anomalous perturbations have been added.
- Ground truth masks corresponding to the anomalous perturbations.

generate_perturbation(***height***, ***width***, ***anomaly_source_path***=*None*)

Generate an image containing a random anomalous perturbation using a source image.

Parameters:

- **height** (*int*) – height of the generated image.
- **width** (`int`) – (int): width of the generated image.
- **anomaly_source_path** (*Path* | *str* | *None*) – Path to an image file. If not provided, random noise will be used
- **instead.** –

Return type:

`tuple`[`ndarray`, `ndarray`]

Returns:

Image containing a random anomalous perturbation, and the corresponding ground truth anomaly mask.

`rand_augmenter()`

Select 3 random transforms that will be applied to the anomaly source images.

Return type:

`Sequential`

Returns:

A selection of 3 transforms.

`anomalib.data.utils.augmenter.nextpow2(value)`

Return the smallest power of 2 greater than or equal to the input value.

Return type:

`int`

Dataset that generates synthetic anomalies.

This dataset can be used when there is a lack of real anomalous data.

class `anomalib.data.utils.synthetic.SyntheticAnomalyDataset(task, transform, source_samples)`

Bases: `AnomalibDataset`

Dataset which reads synthetically generated anomalous images from a temporary folder.

Parameters:

- **task** (*str*) – Task type, either “classification” or “segmentation”.
- **transform** (*A.Compose*) – Transform object describing the transforms that are applied to the inputs.
- **source_samples** (*DataFrame*) – Normal samples to which the anomalous augmentations will be applied.

***classmethod* from_dataset(*dataset*)**

Create a synthetic anomaly dataset from an existing dataset of normal images.

Parameters:

dataset (*AnomalibDataset*) – Dataset consisting of only normal images that will be converrted to a synthetic anomalous dataset with a 50/50 normal anomalous split.

Return type:

SyntheticAnomalyDataset

anomalib.data.utils.synthetic.make_synthetic_dataset(*source_samples*, *image_dir*, *mask_dir*, *anomalous_ratio=0.5*)

Convert a set of normal samples into a mixed set of normal and synthetic anomalous samples.

The synthetic images will be saved to the file system in the specified root directory under <root>/images. For the synthetic anomalous images, the masks will be saved under <root>/ground_truth.

Parameters:

- **source_samples** (*DataFrame*) – Normal images that will be used as source for the synthetic anomalous images.
- **image_dir** (*Path*) – Directory to which the synthetic anomalous image files will be written.
- **mask_dir** (*Path*) – Directory to which the ground truth anomaly masks will be written.
- **anomalous_ratio** (*float*) – Fraction of source samples that will be converted into anomalous samples.

Return type:

`DataFrame`

Previous
Tiling

Next
Models