

DRAEM

Contents

- `Draem`
- `DraemModel`
- `DraemLoss`

DRÆM - A discriminatively trained reconstruction embedding for surface anomaly detection.

Paper <https://arxiv.org/abs/2108.07610>

class

`anomalib.models.image.draem.lightning_model.Draem(enable_sspcab=False, sspcab_lambda=0.1, anomaly_source_path=None, beta=(0.1, 1.0))`

Bases: `AnomalyModule`

DRÆM: A discriminatively trained reconstruction embedding for surface anomaly detection.

Parameters:

- **`enable_sspcab`** (*bool*) – Enable SSPCAB training. Defaults to `False`.
- **`sspcab_lambda`** (*float*) – SSPCAB loss weight. Defaults to `0.1`.
- **`anomaly_source_path`** (*str* | *None*) – Path to folder that contains the anomaly source images. Random noise will be used if left empty. Defaults to `None`.

`configure_optimizers()`

Configure the Adam optimizer.

Return type:

`Optimizer`

property `learning_type`: *LearningType*

Return the learning type of the model.

Returns:

Learning type of the model.

Return type:

LearningType

setup_sspcab()

Prepare the model for the SSPCAB training step by adding forward hooks for the SSPCAB layer activations.

Return type:

None

property `trainer_arguments`: *dict[str, Any]*

Return DRÆM-specific trainer arguments.

training_step(*batch*, **args*, ***kwargs*)

Perform the training step of DRAEM.

Feeds the original image and the simulated anomaly image through the network and computes the training loss.

Parameters:

- **batch** (*dict[str, str | torch.Tensor]*) – Batch containing image filename, image, label and mask
- **args** – Arguments.
- **kwargs** – Keyword arguments.

Return type:

Union[`Tensor`, Mapping[`str`, `Any`], None]

Returns:

Loss dictionary

`validation_step(batch, *args, **kwargs)`

Perform the validation step of DRAEM. The Softmax predictions of the anomalous class are used as anomaly map.

Parameters:

- **batch** (*dict[str, str | torch.Tensor]*) – Batch of input images
- **args** – Arguments.
- **kwargs** – Keyword arguments.

Return type:

`Union[Tensor, Mapping[str, Any], None]`

Returns:

Dictionary to which predicted anomaly maps have been added.

PyTorch model for the DRAEM model implementation.

`class anomalib.models.image.draem.torch_model.DraemModel(sspcab=False)`

Bases: `Module`

DRAEM PyTorch model consisting of the reconstructive and discriminative sub networks.

Parameters:

sspcab (*bool*) – Enable SSPCAB training. Defaults to `False`.

`forward(batch)`

Compute the reconstruction and anomaly mask from an input image.

Parameters:

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

Return type:

`Tensor | tuple[Tensor, Tensor]`

Returns:

Predicted confidence values of the anomaly mask. During training the reconstructed input images are returned as well.

Loss function for the DRAEM model implementation.

`class anomalib.models.image.draem.loss.DraemLoss`

Bases: `Module`

Overall loss function of the DRAEM model.

The total loss consists of the sum of the L2 loss and Focal loss between the reconstructed image and the input image, and the Structural Similarity loss between the predicted and GT anomaly masks.

`forward(input_image, reconstruction, anomaly_mask, prediction)`

Compute the loss over a batch for the DRAEM model.

Return type:

`Tensor`

< [Previous](#)
[DFM](#)

[Next](#) >
[DSR](#)