

R-KDE

Contents

- `Rkde`
- `RkdeModel`
- `FeatureExtractor`
- `RegionExtractor`
- `RoiStage`

Region Based Anomaly Detection With Real-Time Training and Analysis.

<https://ieeexplore.ieee.org/abstract/document/8999287>

```
class anomalib.models.image.rkde.lightning_model.Rkde(roi_stage=RoiStage.RCNN,  
roi_score_threshold=0.001, min_box_size=25, iou_threshold=0.3,  
max_detections_per_image=100, n_pca_components=16,  
feature_scaling_method=FeatureScalingMethod.SCALE, max_training_points=40000)
```

Bases: `MemoryBankMixin`, `AnomalyModule`

Region Based Anomaly Detection With Real-Time Training and Analysis.

Parameters:

- **roi_stage** ([RoiStage](#), *optional*) – Processing stage from which rois are extracted. Defaults to `RoiStage.RCNN`.
- **roi_score_threshold** (*float, optional*) – Minimum confidence score for the region proposals. Defaults to `0.001`.
- **min_size** (*int, optional*) – Minimum size in pixels for the region proposals. Defaults to `25`.
- **iou_threshold** (*float, optional*) – Intersection-Over-Union threshold used during NMS. Defaults to `0.3`.
- **max_detections_per_image** (*int, optional*) – Maximum number of region proposals per image. Defaults to `100`.
- **n_pca_components** (*int, optional*) – Number of PCA components. Defaults to `16`.
- **feature_scaling_method** ([FeatureScalingMethod](#), *optional*) – Scaling method applied to features before passing to KDE. Options are *norm* (normalize to unit vector length) and *scale* (scale to max length observed in training). Defaults to `FeatureScalingMethod.SCALE`.
- **max_training_points** (*int, optional*) – Maximum number of training points to fit the KDE model. Defaults to `40000`.

static `configure_optimizers()`

RKDE doesn't require optimization, therefore returns no optimizers.

Return type:

`None`

`fit()`

Fit a KDE Model to the embedding collected from the training set.

Return type:

`None`

property `learning_type`: *LearningType*

Return the learning type of the model.

Returns:

Learning type of the model.

Return type:

LearningType

property `trainer_arguments: dict[str, Any]`

Return R-KDE trainer arguments.

Returns:

Arguments for the trainer.

Return type:

dict[str, Any]

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

Perform a training Step of RKDE. For each batch, features are extracted from the CNN.

Parameters:

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

Return type:

None

Returns:

Deep CNN features.

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

Perform a validation Step of RKde.

Similar to the training step, features are extracted from the CNN for each batch.

Parameters:

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

Return type:`Union[Tensor, Mapping[str, Any], None]`**Returns:**

Dictionary containing probability, prediction and ground truth values.

Torch model for region-based anomaly detection.

```
class anomalib.models.image.rkde.torch_model.RkdeModel(roi_stage=RoiStage.RCNN,  
roi_score_threshold=0.001, min_box_size=25, iou_threshold=0.3,  
max_detections_per_image=100, n_pca_components=16,  
feature_scaling_method=FeatureScalingMethod.SCALE, max_training_points=40000)
```

Bases: `Module`

Torch Model for the Region-based Anomaly Detection Model.

Parameters:

- **roi_stage** ([RoiStage](#), *optional*) – Processing stage from which rois are extracted. Defaults to `RoiStage.RCNN`.
- **roi_score_threshold** (*float, optional*) – Minimum confidence score for the region proposals. Defaults to `0.001`.
- **min_size** (*int, optional*) – Minimum size in pixels for the region proposals. Defaults to `25`.
- **iou_threshold** (*float, optional*) – Intersection-Over-Union threshold used during NMS. Defaults to `0.3`.
- **max_detections_per_image** (*int, optional*) – Maximum number of region proposals per image. Defaults to `100`.
- **n_pca_components** (*int, optional*) – Number of PCA components. Defaults to `16`.
- **feature_scaling_method** ([FeatureScalingMethod](#), *optional*) – Scaling method applied to features before passing to KDE. Options are *norm* (normalize to unit vector length) and *scale* (scale to max length observed in training). Defaults to `FeatureScalingMethod.SCALE`.
- **max_training_points** (*int, optional*) – Maximum number of training points to fit the KDE model. Defaults to `40000`.

`fit(embeddings)`

Fit the model using a set of collected embeddings.

Parameters:

embeddings (*torch.Tensor*) – Input embeddings to fit the model.

Return type:

`bool`

Returns:

Boolean confirming whether the training is successful.

forward(*batch*)

Prediction by normality model.

Parameters:

batch (*torch.Tensor*) – Input images.

Returns:

The extracted features (when in training mode),
or the predicted rois and corresponding anomaly scores.

Return type:

Tensor | tuple[torch.Tensor, torch.Tensor]

Region-based Anomaly Detection with Real Time Training and Analysis.

Feature Extractor.

class `anomalib.models.image.rkde.feature_extractor.FeatureExtractor`

Bases: `Module`

Feature Extractor module for Region-based anomaly detection.

forward(*batch*, *rois*)

Perform a forward pass of the feature extractor.

Parameters:

- **batch** (*torch.Tensor*) – Batch of input images of shape [B, C, H, W].
- **rois** (*torch.Tensor*) – torch.Tensor of shape [N, 5] describing the regions-of-interest in the batch.

Returns:

`torch.Tensor` containing a 4096-dimensional feature vector for every RoI location.

Return type:

`Tensor`

Region-based Anomaly Detection with Real Time Training and Analysis.

Region Extractor.

class

```
anomalib.models.image.rkde.region_extractor.RegionExtractor(stage=RoiStage.RCNN,  
score_threshold=0.001, min_size=25, iou_threshold=0.3,  
max_detections_per_image=100)
```

Bases: `Module`

Extracts regions from the image.

Parameters:

- **stage** ([RoiStage](#), *optional*) – Processing stage from which rois are extracted. Defaults to `RoiStage.RCNN`.
- **score_threshold** (*float, optional*) – Minimum confidence score for the region proposals. Defaults to `0.001`.
- **min_size** (*int, optional*) – Minimum size in pixels for the region proposals. Defaults to `25`.
- **iou_threshold** (*float, optional*) – Intersection-Over-Union threshold used during NMS. Defaults to `0.3`.
- **max_detections_per_image** (*int, optional*) – Maximum number of region proposals per image. Defaults to `100`.

forward(batch)

Forward pass of the model.

Parameters:

batch (`torch.Tensor`) – Batch of input images of shape [B, C, H, W].

Raises:

ValueError – When `stage` is not one of `rcnn` or `rpn`.

Returns:

Predicted regions, tensor of shape [N, 5] where N is the number of predicted regions in the batch,

and where each row describes the index of the image in the batch and the 4 bounding box coordinates.

Return type:

Tensor

`post_process_box_predictions(pred_boxes, pred_scores)`

Post-processes the box predictions.

The post-processing consists of removing small boxes, applying nms, and keeping only the k boxes with the highest confidence score.

Parameters:

- **pred_boxes** (*torch.Tensor*) – Box predictions of shape (N, 4).
- **pred_scores** (*torch.Tensor*) – torch.Tensor of shape () with a confidence score for each box prediction.

Returns:

Post-processed box predictions of shape (N, 4).

Return type:

list[torch.Tensor]

```
class anomalib.models.image.rkde.region_extractor.RoiStage(value, names=None, *, module=None, qualname=None, type=None, start=1, boundary=None)
```

Bases: `str`, `Enum`

Processing stage from which rois are extracted.

[< Previous](#)
[Reverse Distillation](#)

[Next >](#)
[STFPM](#)