Migrating from 0.* to 1.0

Contents

Overview

The 1.0 release of the Anomaly Detection Library (AnomalyLib) introduces several changes to the library. This guide provides an overview of the changes and how to migrate from 0.* to 1.0.

Installation

For installation instructions, refer to the installation guide.

Changes to the CLI

Upgrading the Configuration

There are several changes to the configuration of Anomalib. The configuration file has been updated to include new parameters and remove deprecated parameters. In addition, some parameters have been moved to different sections of the configuration.

Anomalib provides a python script to update the configuration file from 0.* to 1.0. To update the configuration file, run the following command:

```
python tools/upgrade/config.py \
    --input_config <path_to_0.*_config> \
    --output_config <path_to_1.0_config>
```

This script will ensure that the configuration file is updated to the 1.0 format.

In the following sections, we will discuss the changes to the configuration file in more detail.

Changes to the Configuration File

Data

The `data` section of the configuration file has been updated such that the args can be directly used to instantiate the data object. Below are the differences between the old and new configuration files highlighted in a markdown diff format.

```
-dataset:
+data:
-  name: mvtec
-  format: mvtec
+  class_path: anomalib.data.MVTec
+  init_args:
-  path: ./datasets/MVTec
+    root: ./datasets/MVTec
     category: bottle
     image_size: 256
     center_crop: null
     normalization: imagenet
     train_batch_size: 72
     eval_batch_size: 32
     num_workers: 8
     task: segmentation
     test_split_mode: from_dir # options: [from_dir, synthetic]
     test_split_ratio: 0.2 # fraction of train images held out testing (usage depen
     val_split_mode: same_as_test # options: [same_as_test, from_test, synthetic]
     val_split_ratio: 0.5 # fraction of train/test images held out for validation (
     seed: null
-  transform_config:
-    train: null
-    eval: null
+    transform_config_train: null
+    transform_config_eval: null
-  tiling:
-    apply: false
-    tile_size: null
```

```
-    stride: null
-    remove_border_count: 0
-    use_random_tiling: False
-    random_tile_count: 16+data:
```

Here is the summary of the changes to the configuration file:

- The `name` and `format keys` from the old configuration are absent in the new configuration, possibly integrated into the design of the class at `class_path`.
- Introduction of a `class_path` key in the new configuration specifies the Python class path for data handling.
- The structure has been streamlined in the new configuration, moving everything under `data` and `init_args` keys, simplifying the hierarchy.
- `transform_config` keys were split into `transform_config_train` and `transform_config_eval` to clearly separate training and evaluation configurations.
- The `tiling` section present in the old configuration has been completely removed in the new configuration. v1.0.0 does not support tiling. This feature will be added back in a future release.

Model

Similar to data configuration, the `model` section of the configuration file has been updated such that the args can be directly used to instantiate the model object. Below are the differences between the old and new configuration files highlighted in a markdown diff format.

```
 model:
-  name: patchcore
-  backbone: wide_resnet50_2
-  pre_trained: true
-  layers:
+  class_path: anomalib.models.Patchcore
+  init_args:
+    backbone: wide_resnet50_2
+    pre_trained: true
+    layers:
     - layer2
     - layer3
-  coreset_sampling_ratio: 0.1
-  num_neighbors: 9
+    coreset_sampling_ratio: 0.1
```

```
+    num_neighbors: 9
-  normalization_method: min_max # options: [null, min_max, cdf]
+normalization:
+  normalization_method: min_max
```

Here is the summary of the changes to the configuration file:

- Model Identification: Transition from `name` to `class_path` for specifying the model, indicating a more explicit reference to the model's implementation.
- Initialization Structure: Introduction of `init_args` to encapsulate model initialization parameters, suggesting a move towards a more structured and possibly dynamically loaded configuration system.
- Normalization Method: The `normalization_method` key is removed from the `model` section and moved to a separate `normalization` section in the new configuration.

Metrics

The `metrics` section of the configuration file has been updated such that the args can be directly used to instantiate the metrics object. Below are the differences between the old and new configuration files highlighted in a markdown diff format.

```
metrics:
  image:
    - F1Score
    - AUROC
  pixel:
     - F1Score
     - AUROC
   threshold:
-    method: adaptive #options: [adaptive, manual]
-    manual_image: null
-    manual_pixel: null
+    class_path: anomalib.metrics.F1AdaptiveThreshold
+    init_args:
+      default_value: 0.5
```

Here is the summary of the changes to the configuration file:

- Metric Identification: Transition from `method` to `class_path` for specifying the metric, indicating a more explicit reference to the metric's implementation.
- Initialization Structure: Introduction of `init_args` to encapsulate metric initialization

parameters, suggesting a move towards a more structured and possibly dynamically loaded configuration system.

- Threshold Method: The `method` key is removed from the `threshold` section and moved to a separate `class_path` section in the new configuration.

Previous
Anomalib in 15 Minutes

Next
Reference Guide