
SMQTK-Detection

Release 0.20.1

Kitware, Inc.

Nov 28, 2023

CONTENTS

1	Installation	3
2	Detection Interfaces	5
2.1	Detection Element	5
2.2	ObjectDetector	7
2.3	ImageMatrixObjectDetector	8
2.4	DetectImageObjects	10
3	Detection Implementations	11
3.1	MemoryDetectionElement	11
4	Release Process and Notes	13
4.1	Steps of the SMQTK Release Process	13
4.2	Release Notes	13
5	Indices and tables	17
	Index	19

[GitHub](#)

Interfaces and support for black-box object detection.

INSTALLATION

Please reference the SMQTK-Core installation documentation as such documentation for this package is nearly identical. Of course, replace uses of smqtk-core with smqtk-detection.

DETECTION INTERFACES

Here we list and briefly describe the high level algorithm interfaces which SMQTK-Detection provides. Some implementations will require additional dependencies that cannot be packaged with SMQTK-Detection.

2.1 Detection Element

Data structure used by detector interfaces to communicate inference predictions.

class `smqtk_detection.interfaces.detection_element.DetectionElement`(*uuid: Hashable*)

Representation of a spatial detection.

classmethod `from_config`(*config_dict: Dict[Any, Any], uuid: Hashable, merge_default: bool = True*) → *DetectionElement*

Override of `smqtk.utils.configuration.Configurable.from_config()` with the added runtime argument `uuid`. See parent method documentation for details.

Parameters

- **config_dict** (*dict*) – JSON compliant dictionary encapsulating a configuration.
- **uuid** (*collections.abc.Hashable*) – UUID to assign to the produced `DetectionElement`.
- **merge_default** (*bool*) – Merge the given configuration on top of the default provided by `get_default_config`.

Returns

Constructed instance from the provided config.

Return type

DetectionElement

abstract `get_bbox()` → `AxisAlignedBoundingBox` | `None`

Returns

The spatial bounding box of this detection.

Return type

`smqtk.representation.AxisAlignedBoundingBox`

Raises

NoDetectionError – No detection `AxisAlignedBoundingBox` set yet.

abstract `get_classification()` → `ClassificationElement` | `None`

Returns

The classification element of this detection.

Return type

`smqtk.representation.ClassificationElement`

Raises

NoDetectionError – No detection `ClassificationElement` set yet or the element is empty.

classmethod `get_default_config()` → `Dict[str, Any]`

Generate and return a default configuration dictionary for this class. This will be primarily used for generating what the configuration dictionary would look like for this class without instantiating it.

By default, we observe what this class's constructor takes as arguments, turning those argument names into configuration dictionary keys. If any of those arguments have defaults, we will add those values into the configuration dictionary appropriately. The dictionary returned should only contain JSON compliant value types.

It is not be guaranteed that the configuration dictionary returned from this method is valid for construction of an instance of this class.

Returns

Default configuration dictionary for the class.

Return type

`dict`

```
>>> # noinspection PyUnresolvedReferences
>>> class SimpleConfig(Configurable):
...     def __init__(self, a=1, b='foo'):
...         self.a = a
...         self.b = b
...     def get_config(self):
...         return {'a': self.a, 'b': self.b}
>>> self = SimpleConfig()
>>> config = self.get_default_config()
>>> assert config == {'a': 1, 'b': 'foo'}
```

abstract `get_detection()` → `Tuple[AxisAlignedBoundingBox, ClassificationElement]`

Returns

The paired spatial bounding box and classification element of this detection.

Return type

`(smqtk.representation.AxisAlignedBoundingBox, smqtk.representation.ClassificationElement)`

Raises

NoDetectionError – No detection `AxisAlignedBoundingBox` and `ClassificationElement` set yet.

abstract `has_detection()` → `bool`

Returns

Whether or not this container currently contains a valid detection bounding box and classification element (must be non-zero).

Return type

`bool`

abstract set_detection(*bbox*: *AxisAlignedBoundingBox*, *classification_element*: *ClassificationElement*)
→ *DetectionElement*

Set a bounding box and classification element to this detection element.

Parameters

- **bbox** (*smqtk.representation.AxisAlignedBoundingBox*) – Spatial bounding box instance.
- **classification_element** (*smqtk.representation.ClassificationElement*) – The classification of this detection.

Raises

ValueError – No, or invalid, *AxisAlignedBoundingBox* or *ClassificationElement* was provided.

Returns

Self

Return type

DetectionElement

2.2 ObjectDetector

This interface defines a method to generate object detections (*smqtk_detection.interfaces.detection_element.DetectionElement*) over a given *smqtk_dataprovider.interfaces.data_element.DataElement*.

class *smqtk_detection.interfaces.object_detector.ObjectDetector*

Abstract interface to an object detection algorithm.

An object detection algorithm is one that can take in data and output zero or more detection elements, where each detection represents a spatial region in the data.

This high level interface only requires detection element returns (spatial bounding-boxes with associated classification elements).

detect_objects(*data_element*: ~*smqtk_dataprovider.interfaces.data_element.DataElement*, *de_factory*: ~*smqtk_detection.detection_element_factory.DetectionElementFactory* = <*smqtk_detection.detection_element_factory.DetectionElementFactory* object>, *ce_factory*: ~*smqtk_classifier.classification_element_factory.ClassificationElementFactory* = <*smqtk_classifier.classification_element_factory.ClassificationElementFactory* object>)
→ *Iterator*[*DetectionElement*] | *None*

Detect objects in the given data.

UUIDs of detections are based on the hash produced from the combination of:

- Detection bounding-box bounding coordinates
- Classification label set predicted for a bounding box.

Parameters

- **data_element** (*smqtk.representation.DataElement*) – Source data from which to detect objects within.
- **de_factory** (*smqtk.representation.DetectionElementFactory*) – Factory for generating *DetectionElement* instances. The default factory yields *MemoryClassificationElement* instances.

- **ce_factory** (*smqtk.representation.ClassificationElementFactory*) – Factory for generating *ClassificationElement* instances for detections. The default factory yields *MemoryClassificationElement* instances.

Raises

ValueError – Given data element content was not of a valid content type that this class reports as valid for object detection.

Returns

Iterator over result *DetectionElement* instances as generated by the given *DetectionElementFactory*, containing classification elements as generated by the given *ClassificationElementFactory*.

Return type

`collections.abc.Iterable[smqtk.representation.DetectionElement]`

2.3 ImageMatrixObjectDetector

class `smqtk_detection.interfaces.object_detector.ImageMatrixObjectDetector`(*image_reader: ImageReader*)

Class of object detectors that operate over the pixel matrix of an image.

This sub abstract class standardizes the use of an `smqtk.algorithms.ImageReader` algorithm to read an image file's pixels as well as determine which image formats are valid input elements. There is a special exception of *MatrixDataElement* types as they directly provide a matrix.

We define an alternate abstract method for implementing classes to define: `_detect_objects_matrix`. This method is given a numpy ndarray instance for the implementing class to utilize. The return requirements are the same as the `_detect_objects` method.

classmethod `from_config`(*config_dict: dict, merge_default: bool = True*) → *ImMatObDet*

Instantiate a new instance of this class given the configuration JSON-compliant dictionary encapsulating initialization arguments.

This method should not be called via super unless an instance of the class is desired.

Parameters

- **config_dict** (*dict*) – JSON compliant dictionary encapsulating a configuration.
- **merge_default** (*bool*) – Merge the given configuration on top of the default provided by `get_default_config`.

Returns

Constructed instance from the provided config.

Return type

ImageMatrixObjectDetector

abstract `get_config`() → *dict*

Return a JSON-compliant dictionary that could be passed to this class's `from_config` method to produce an instance with identical configuration.

In the most cases, this involves naming the keys of the dictionary based on the initialization argument names as if it were to be passed to the constructor via dictionary expansion. In some cases, where it doesn't make sense to store some object constructor parameters are expected to be supplied at as configuration values (i.e. must be supplied at runtime), this method's returned dictionary may leave those parameters out. In such cases, the object's `from_config` class-method would also take additional positional arguments to fill in for the parameters that this returned configuration lacks.

Returns

JSON type compliant configuration dictionary.

Return type

dict

classmethod `get_default_config()` → dict

Generate and return a default configuration dictionary for this class. This will be primarily used for generating what the configuration dictionary would look like for this class without instantiating it.

By default, we observe what this class's constructor takes as arguments, turning those argument names into configuration dictionary keys. If any of those arguments have defaults, we will add those values into the configuration dictionary appropriately. The dictionary returned should only contain JSON compliant value types.

It is not guaranteed that the configuration dictionary returned from this method is valid for construction of an instance of this class.

Returns

Default configuration dictionary for the class.

Return type

dict

is_valid_element(*data_element: DataElement*) → bool

Check if the given DataElement instance reports a content type that matches one of the MIME types reported by `valid_content_types`.

This override uses our stored ImageReader algorithm instance to define what DataElement instances are valid.

Parameters

data_element (*smqtk.representation.DataElement*) – Data element instance to check.

Returns

True if the given element has a valid content type as reported by `valid_content_types`, and False if not.

Return type

bool

valid_content_types() → Set[str]

Returns

A set valid MIME types that are “valid” within the implementing class’ context.

Return type

set[str]

2.4 DetectImageObjects

class smqtk_detection.interfaces.detect_image_objects.**DetectImageObjects**

Algorithm that generates object bounding boxes and classification maps for a set of input image matrices as `numpy.ndarray` type arrays.

abstract detect_objects(*img_iter: Iterable[ndarray]*) →
Iterable[Iterable[Tuple[AxisAlignedBoundingBox, Dict[Hashable, float]]]]

Generate paired bounding boxes and classification maps for detected objects in the given set of images.

Parameters

img_iter – Iterable of input images as numpy arrays.

Returns

Iterable of sets of paired bounding boxes and classification maps. Each set is the collection of detections for the corresponding input image.

DETECTION IMPLEMENTATIONS

Here we list and briefly describe the high level algorithm implementations which SMQTK-Detection provides.

3.1 MemoryDetectionElement

In-memory backend of the DetectionElement representation interface.

RELEASE PROCESS AND NOTES

4.1 Steps of the SMQTK Release Process

Please reference the SMQTK-Core release process documentation as that same process is applicable here, of course replacing uses of `smqtk-core` with `smqtk-detection`.

4.2 Release Notes

4.2.1 v0.15.0

This is the initial release of `smqtk-detection`, spinning off from v0.14.0 of the monolithic `SMQTK` library.

Updates / New Features

CI

- Added workflow for CI on GitHub.

Misc.

- Updated various type annotations for type-checking compliance.
- Updated `.gitignore` away from the old monorepo version.
- Added `readthedocs` conf files
- Updated to use now publicly available `smqtk-detection` package from PYPI.

Fixes

4.2.2 v0.16.0

Updates / New Features

Dependencies

- Remove `six` packages as dependencies and fix ramifications.
- Update `smqtk-*` dependency packages to latest patched versions.

Fixes

4.2.3 v0.17.0

Updates / New Features

Features

- Created *DetectImageObjects* interface and accompanying example implementation, *RandomDetector* to handle image matrices directly.

Fixes

Dependencies

- Update pinned version of Pillow used for development to address a security vulnerability.

4.2.4 v0.18.0

This minor release primarily adds two new optional plugins for *DetectImageObjects*: PyTorch-based ResNetFRCNN and CenterNetVisdrone. These may be usable after installing the “torch” and “centernet” extras, respectively. These implementations currently require an internet connection to be able to initially download pretrained models.

Updates / New Features

CI

- Add workflow to inherit the smqtk-core publish workflow.

Features

- Added *ResNetFRCNN* implementation of *DetectImageObjects* that relies on, optional dependencies *torch* and *torchvision*.
- Added optional *CenterNetVisdrone* implementation of *DetectImageObjects*. Source for the model used was taken from: <https://github.com/GNAYUOHZ/centernet-visdrone>
- Moved, and added to, interface convenience exposure to the package root module.

Miscellaneous

- Added a wrapper script to pull the versioning/changelog update helper from smqtk-core to use here without duplication.

Fixes

4.2.5 v0.18.1

This patch release addresses an issue with the centernet plugin where image preprocessing was not happening on an appropriate per-batch cadence which led to more RAM being consumed than intended for large input sequences.

Fixes

CI

- Also run CI unittests for PRs targeting branches that match the *release** glob.

Implementations

- Fix CenterNetVisdrone batched operation and class labels

4.2.6 v0.19.0

This minor release includes refinements to the *DetectImageObjects* plugins, including a change to return semantic class labels instead of just integers, reducing the need for the caller to know intrinsics about the plugin/model in order to use it.

See below for additional updates and fixes.

Updates / New Features

CI

- Updated CI unittests workflow to include codecov reporting. Reduced CodeCov report submission by skipping this step on scheduled runs.

Documentation

- Updated CONTRIBUTING.md to reference smqtk-core's CONTRIBUTING.md file.

Detect Image Objects

- Updated the *ResNetFRCNN* to return as its class labels the label strings instead of integers, reducing the burden of users from having to repeatedly find and allocate the appropriate int-to-label map.
- Update *CenterNetVisdrone* to lazy load its model on first inference.
- Update *CenterNetVisdrone* to return VisDrone class labels instead of integers.

Unit Tests

- Added unit tests for the *CenterNetVisdrone* plugin.

Utilities

- Remove duplicate implementation of *AxisAlignedBoundingBox*, using the tested version that comes from the *smqtk_image_io* package.

Fixes

Detect Image Object

- Fixed batched operation memory usage in *ResNetFRCNN* by loading only current batch into computation device memory. Previously all images were loaded at once.
- Fixed device mapping when loading certain background architectures for *CenterNetVisdrone*.

Dependency Versions

- Updated the developer dependency and locked version of *ipython* to address a security vulnerability.
- Removed *jedi* = “^0.17.2” requirement since recent *ipython* = “^7.17.3” update appropriately addresses the dependency.

4.2.7 v0.20.0

This minor release updates the minimum supported python to *python* = “^3.8”, addresses dependency vulnerabilities, and updates typing to conform with current mypy and pytest standards.

Updates / New Features

Python

- New minimum supported python changed to *python* = “^3.8”.

Dependencies

- Updated python minimum requirement to 3.8 (up from 3.6). This involved a number of updates and bifurcations of abstract requirements, an update to pinned versions for development/CI, and expansion of CI to cover python versions 3.10 and 3.11 (latest current release).

Fixes

Docs

- Fix erroneous references to previous monorepo.
- Fixed `sphinx_server.py` to reference correct directories.

Dependency versions

- Updated the locked versions of dependencies to reflect new minimum support *python* = “^3.8”.

4.2.8 v0.20.1

Updates / New Features

Fixes

- Fixed missing readthedocs url in `pyproject.toml`.

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

INDEX

D

`detect_objects()` (*smqtk_detection.interfaces.detect_image_objects.DetectImageObjects* method), 10

`detect_objects()` (*smqtk_detection.interfaces.object_detector.ObjectDetector* method), 7

`DetectImageObjects` (class in *smqtk_detection.interfaces.object_detector*), 7

`DetectionElement` (class in *smqtk_detection.interfaces.detection_element*), 5

F

`from_config()` (*smqtk_detection.interfaces.detection_element.DetectionElement* class method), 5

`from_config()` (*smqtk_detection.interfaces.object_detector.ImageMatrixObjectDetector* class method), 8

G

`get_bbox()` (*smqtk_detection.interfaces.detection_element.DetectionElement* method), 5

`get_classification()` (*smqtk_detection.interfaces.detection_element.DetectionElement* method), 5

`get_config()` (*smqtk_detection.interfaces.object_detector.ImageMatrixObjectDetector* method), 8

`get_default_config()` (*smqtk_detection.interfaces.detection_element.DetectionElement* class method), 6

`get_default_config()` (*smqtk_detection.interfaces.object_detector.ImageMatrixObjectDetector* class method), 9

`get_detection()` (*smqtk_detection.interfaces.detection_element.DetectionElement* method), 6

H

`has_detection()` (*smqtk_detection.interfaces.detection_element.DetectionElement* method), 6

I

`ImageMatrixObjectDetector` (class in *smqtk_detection.interfaces.object_detector*), 8

`is_valid_element()` (*smqtk_detection.interfaces.object_detector.ImageMatrixObjectDetector* method), 9

O

`ObjectDetector` (class in *smqtk_detection.interfaces.object_detector*), 7

S

`set_detection()` (*smqtk_detection.interfaces.detection_element.DetectionElement* method), 6

V

`valid_content_types()`

(*smqtk_detection.interfaces.object_detector.ImageMatrixObjectDetector* method), 9