Print to PDF ▸

Loggers

Contents

Load PyTorch Lightning Loggers.

class **anomalib.loggers.AnomalibCometLogger**(*api_key=None*, *save_dir=None*, *project_name=None*, *rest_api_key=None*, *experiment_name=None*, *experiment_key=None*, *offline=False*, *prefix=''*, ***kwargs*)

Bases: `ImageLoggerBase`, `CometLogger`

Logger for comet.

Adds interface for `add_image` in the logger rather than calling the experiment object.

> **Note**
>
> Same as the CometLogger provided by PyTorch Lightning and the doc string is reproduced below.

Track your parameters, metrics, source code and more using Comet.

Install it with pip:

```
pip install comet-ml
```

Comet requires either an API Key (online mode) or a local directory path (offline mode).

Parameters:

- **api_key** (`str` | `None`) – Required in online mode. API key, found on Comet.ml. If not given, this will be loaded from the environment variable COMET_API_KEY or ~/.comet.config if either exists. Defaults to `None`.
- **save_dir** (`str` | `None`) – Required in offline mode. The path for the directory to save local comet logs. If given, this also sets the directory for saving checkpoints. Defaults to `None`.
- **project_name** (`str` | `None`) – Optional. Send your experiment to a specific project. Otherwise will be sent to Uncategorized Experiments. If the project name does not already exist, Comet.ml will create a new project. Defaults to `None`.
- **rest_api_key** (`str` | `None`) – Optional. Rest API key found in Comet.ml settings. This is used to determine version number Defaults to `None`.
- **experiment_name** (`str` | `None`) – Optional. String representing the name for this particular experiment on Comet.ml. Defaults to `None`.
- **experiment_key** (`str` | `None`) – Optional. If set, restores from existing experiment. Defaults to `None`.
- **offline** (`bool`) – If api_key and save_dir are both given, this determines whether the experiment will be in online or offline mode. This is useful if you use save_dir to control the checkpoints directory and have a ~/.comet.config file but still want to run offline experiments. Defaults to `None`.
- **prefix** (`str`) – A string to put at the beginning of metric keys. Defaults to `""`.
- **kwargs** – Additional arguments like *workspace*, *log_code*, etc. used by `CometExperiment` can be passed as keyword arguments in this logger.

Raises:

- **ModuleNotFoundError** – If required Comet package is not installed on the device.
- **MisconfigurationException** – If neither `api_key` nor `save_dir` are passed as arguments.

Example

```
>>> from anomalib.loggers import AnomalibCometLogger
>>> from anomalib.engine import Engine
```

```
...
>>> comet_logger = AnomalibCometLogger()
>>> engine =  Engine(logger=comet_logger)
```

See also

- Comet Documentation

add_image(*image*, *name=None*, ***kwargs*)

Interface to add image to comet logger.

Parameters:

- **image** (*np.ndarray* | *Figure*) – Image to log.
- **name** (*str* | *None*) – The tag of the image Defaults to `None`.
- **kwargs** – Accepts only *global_step* (int). The step at which to log the image.

Return type:

`None`

class anomalib.loggers.AnomalibTensorBoardLogger(*save_dir*, *name='default'*, *version=None*, *log_graph=False*, *default_hp_metric=True*, *prefix=''*, ***kwargs*)

Bases: `ImageLoggerBase`, `TensorBoardLogger`

Logger for tensorboard.

Adds interface for *add_image* in the logger rather than calling the experiment object.

Note

Same as the Tensorboard Logger provided by PyTorch Lightning and the doc string is reproduced below.

Logs are saved to `os.path.join(save_dir, name, version)`. This is the default logger in Lightning, it comes preinstalled.

Example

```
>>> from anomalib.engine import Engine
>>> from anomalib.loggers import AnomalibTensorBoardLogger
...
>>> logger = AnomalibTensorBoardLogger("tb_logs", name="my_model")
>>> engine =  Engine(logger=logger)
```

Parameters:

- **save_dir** (*str*) – Save directory
- **name** (*str* | *None*) – Experiment name. Defaults to `'default'`. If it is the empty string then no per-experiment subdirectory is used. Default: `'default'`.
- **version** (*int* | *str* | *None*) – Experiment version. If version is not specified the logger inspects the save directory for existing versions, then automatically assigns the next available version. If it is a string then it is used as the run-specific subdirectory name, otherwise `'version_${version}'` is used. Defaults to `None`
- **log_graph** (*bool*) – Adds the computational graph to tensorboard. This requires that the user has defined the *self.example_input_array* attribute in their model. Defaults to `False`.
- **default_hp_metric** (*bool*) – Enables a placeholder metric with key `hp_metric` when `log_hyperparams` is called without a metric (otherwise calls to log_hyperparams without a metric are ignored). Defaults to `True`.
- **prefix** (*str*) – A string to put at the beginning of metric keys. Defaults to `''`.
- ****kwargs** – Additional arguments like *comment*, *filename_suffix*, etc. used by `SummaryWriter` can be passed as keyword arguments in this logger.

add_image(*image*, *name=None*, ***kwargs*)

Interface to add image to tensorboard logger.

Parameters:

- **image** (*np.ndarray* | *Figure*) – Image to log
- **name** (*str* | *None*) – The tag of the image Defaults to `None`.
- **kwargs** – Accepts only *global_step* (int). The step at which to log the image.

Return type:

`None`

class **anomalib.loggers.AnomalibWandbLogger**(*name=None*, *save_dir=None*, *offline=False*, *id=None*, *anonymous=None*, *version=None*, *project=None*, *log_model=False*, *experiment=None*, *prefix=''*, ***kwargs*)

Bases: `ImageLoggerBase`, `WandbLogger`

Logger for wandb.

Adds interface for *add_image* in the logger rather than calling the experiment object.

> **Note**
>
> Same as the wandb Logger provided by PyTorch Lightning and the doc string is reproduced below.

Log using Weights and Biases.

Install it with pip:

```
$ pip install wandb
```

Parameters:

- **name** (`str` | `None`) – Display name for the run. Defaults to `None`.
- **save_dir** (`str` | `None`) – Path where data is saved (wandb dir by default). Defaults to `None`.
- **offline** (`bool` | `None`) – Run offline (data can be streamed later to wandb servers). Defaults to `False`.
- **id** (`str` | `None`) – Sets the version, mainly used to resume a previous run. Defaults to `None`.
- **anonymous** (`bool` | `None`) – Enables or explicitly disables anonymous logging. Defaults to `None`.
- **version** (`str` | `None`) – Same as id. Defaults to `None`.
- **project** (`str` | `None`) – The name of the project to which this run will belong. Defaults to `None`.
- **log_model** (`str` | `bool`) – Save checkpoints in wandb dir to upload on W&B servers. Defaults to `False`.
- **experiment** (`type`[`Run`] | `type`[`RunDisabled`] | `None`) – WandB experiment object. Automatically set when creating a run. Defaults to `None`.
- **prefix** (`str` | `None`) – A string to put at the beginning of metric keys. Defaults to `''`.
- ****kwargs** – Arguments passed to `wandb.init()` like *entity*, *group*, *tags*, etc.

Raises:

- **ImportError** – If required WandB package is not installed on the device.
- **MisconfigurationException** – If both `log_model` and ``offline``is set to ``True``.

Example

```
>>> from anomalib.loggers import AnomalibWandbLogger
>>> from anomalib.engine import Engine
...
>>> wandb_logger = AnomalibWandbLogger()
>>> engine =  Engine(logger=wandb_logger)
```

> Note
>
> When logging manually through *wandb.log* or *trainer.logger.experiment.log*, make sure to use *commit=False* so the logging step does not increase.

> See also
>
> - Tutorial on how to use W&B with PyTorch Lightning
> - W&B Documentation

`add_image(image, name=None, **kwargs)`

Interface to add image to wandb logger.

Parameters:

- **image** (*np.ndarray* | *Figure*) – Image to log
- **name** (*str* | *None*) – The tag of the image Defaults to `None`.
- **kwargs** – Additional arguments to *wandb.Image*

Return type:

`None`

`save()`

Upload images to wandb server. :rtype: `None`

> Note
>
> There is a limit on the number of images that can be logged together to the *wandb* server.

Previous
Metrics

Next
Callbacks