pystiche-papers

Release 0.1.dev69+ga8a87f4

Philip Meier, Julian Bültemeier

Dec 13, 2022

CONTENTS

CHAPTER

ONE

INSTALLATION

CHAPTER

TWO

PACKAGE REFERENCE

2.1 pystiche_papers.data

2.1.1 Utilities

class pystiche_papers.data.utils.**FiniteCycleBatchSampler**(*data_source*, *num_batches*, *batch_size=1*)

2.2 pystiche_papers.utils

class pystiche_papers.utils.**HyperParameters**(***kwargs*)

2.3 Paper implementations

Unfortunately, the reference implementation of the original authors often deviates from what is described in the paper. To account for this, the `impl_params` flag is used. It defaults to `True` since the parameters in the implementation were probably used to generate the results in the paper.

In general, the deviations can be separated into two classes:

1. **Behavioral changes**: These changes often result by misconceptions of the author for how their used framework or library works internally. An example for this is the usage of the mean squared error (MSE) in the reference implementation whereas the squared error (SE) is reported in the paper. In some cases these changes also account for completely undocumented behavior.

   These changes have hard-coded behavior and cannot be adapted freely, but rather only be switched between both sets.

2. **Hyper-parameter changes**: In contrast to 1., changes of the hyper-parameters are not hard-coded and can be freely adapted. Both sets of parameters can be accessed with the respective `hyper_parameters` functions.

You can find information on both types of changes for each paper implementation in the respective "Behavioral changes" and "Hyper parameters" sections.

2.3.1 pystiche_papers.gatys_ecker_bethge_2016

Title	Image Style Transfer Using Convolutional Neural Networks
Authors	Leon A. Gatys, Alexander. S. Ecker, and Matthias Bethge
Citation	[GEB16]
Reference implementation	Repository / Archive
Variant	Image optimization
Content loss	`FeatureReconstructionLoss`
Style loss	`GramLoss`

Behavioral changes

See also:

Paper implementations

The following parts are affected:

- *`FeatureReconstructionLoss`*
- *`MultiLayerEncodingLoss`*
- *`multi_layer_encoder`*

Hyper parameters

See also:

Paper implementations

Empty cells mean, that the parameter is not defined in the paper or no default is set in the reference implementation of the original authors. In both cases the available value is used as default.

`content_loss()`

Parameter	impl_params	
	True	False
layer	"relu4_2"	"conv4_2"
score_weight	1e0	

`style_loss()`

Parameter	impl_params	
	True	False
layers | ("relu1_1", "relu2_1", "relu3_1", "relu4_1", "relu5_1")		("conv1_1", "conv2_1", "conv3_1", "conv4_1", "conv5_1")
layer_weights	(2.4e-04, 6.1e-05, 1.5e-05, 3.8e-06, 3.8e-06)[1]	"mean"
score_weight	1e3[2]	

nst()

Parameter	impl_params	
	True	False
image_size	512	
starting_point	"content"	"random"
num_steps	500	

API

pystiche_papers.gatys_ecker_bethge_2016.**images**()

Return type DownloadableImageCollection

class pystiche_papers.gatys_ecker_bethge_2016.**FeatureReconstructionLoss**(*encoder*, *impl_params=True*, ***feature_reconstruction_loss_kwargs*)

Feature reconstruction loss from [GEB16].

Parameters

- **encoder** (Encoder) – Encoder used to encode the input.
- **impl_params** (bool) – If False, calculate the score with the squared error (SE) instead of the mean squared error (MSE). Furthermore, use a score correction factor of 1/2.
- ****feature_reconstruction_loss_kwargs** – Additional parameters of a pystiche.loss.FeatureReconstructionLoss.

See also:

- pystiche.loss.FeatureReconstructionLoss

pystiche_papers.gatys_ecker_bethge_2016.**content_loss**(*impl_params=True*, *multi_layer_encoder=None*, *hyper_parameters=None*)

Content loss from [GEB16].

Parameters

- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **multi_layer_encoder** (Optional[MultiLayerEncoder]) – Pretrained multi-layer encoder. If omitted, *multi_layer_encoder()* is used.
- **hyper_parameters** (Optional[*HyperParameters*]) – Hyper parameters. If omitted, *hyper_parameters()* is used.

See also:

- *pystiche_papers.gatys_ecker_bethge_2016.FeatureReconstructionLoss*

[1] The layer_weights are computed by $1/n^2$ where n denotes the number of channels of a feature map from the corresponding layer in the *multi_layer_encoder()*.

[2] The paper also reports score_weight=1e-4 for some images.

Return type *FeatureReconstructionLoss*

class pystiche_papers.gatys_ecker_bethge_2016.**MultiLayerEncodingLoss**(*multi_layer_encoder*, *layers*, *encoding_loss_fn*, *impl_params=True*, ***multi_layer_encoding_op_kwargs*)

Multi-layer encoding loss from [GEB16].

Parameters

- **multi_layer_encoder** (MultiLayerEncoder) – Multi-layer encoder.
- **layers** (Sequence[str]) – Layers of the multi_layer_encoder that the children losses operate on.
- **encoding_loss_fn** (Callable[[Encoder, float], Loss]) – Callable that returns a children loss given a pystiche.enc.SingleLayerEncoder extracted from the multi_layer_encoder and its corresponding layer weight.
- **impl_params** (bool) – If False, use a score correction factor of 1/4.
- ****multi_layer_encoding_op_kwargs** – Additional parameters of a pystiche.loss.MultiLayerEncodingLoss.

See also:

- pystiche.loss.MultiLayerEncodingLoss

pystiche_papers.gatys_ecker_bethge_2016.**style_loss**(*impl_params=True*, *multi_layer_encoder=None*, *hyper_parameters=None*)

Style loss from [GEB16].

Parameters

- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **multi_layer_encoder** (Optional[MultiLayerEncoder]) – Pretrained multi-layer encoder. If omitted, *multi_layer_encoder()* is used.
- **hyper_parameters** (Optional[*HyperParameters*]) – Hyper parameters. If omitted, *hyper_parameters()* is used.

See also:

- *pystiche_papers.gatys_ecker_bethge_2016.MultiLayerEncodingLoss*

Return type *MultiLayerEncodingLoss*

pystiche_papers.gatys_ecker_bethge_2016.**perceptual_loss**(*impl_params=True*, *multi_layer_encoder=None*, *hyper_parameters=None*)

Perceptual loss from [GEB16].

Parameters

- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.

- **multi_layer_encoder** (Optional[MultiLayerEncoder]) – Pretrained multi-layer encoder. If omitted, *multi_layer_encoder()* is used.
- **hyper_parameters** (Optional[*HyperParameters*]) – Hyper parameters. If omitted, *hyper_parameters()* is used.

See also:

- *pystiche_papers.gatys_ecker_bethge_2016.content_loss()*
- *pystiche_papers.gatys_ecker_bethge_2016.style_loss()*

Return type PerceptualLoss

pystiche_papers.gatys_ecker_bethge_2016.**nst**(*content_image*, *style_image*, *impl_params=True*, *hyper_parameters=None*, *quiet=False*)

NST from [GEB16].

Parameters

- **content_image** (Tensor) – Content image for the NST.
- **style_image** (Tensor) – Style image for the NST.
- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **hyper_parameters** (Optional[*HyperParameters*]) – If omitted, *hyper_parameters()* is used.
- **quiet** (bool) – If True, no information is logged during the optimization. Defaults to False.

Return type Tensor

pystiche_papers.gatys_ecker_bethge_2016.**preprocessor**()

Return type Module

pystiche_papers.gatys_ecker_bethge_2016.**postprocessor**()

Return type Module

pystiche_papers.gatys_ecker_bethge_2016.**optimizer**(*input_image*)

Optimizer from [GEB16].

Parameters **input_image** (Tensor) – Image to be optimized.

Return type LBFGS

pystiche_papers.gatys_ecker_bethge_2016.**multi_layer_encoder**(*impl_params=True*)

Multi-layer encoder from [GEB16].

Parameters **impl_params** (bool) – If True, the MaxPool2d in the multi_layer_encoder are exchanged for AvgPool2d.

Return type MultiLayerEncoder

pystiche_papers.gatys_ecker_bethge_2016.**compute_layer_weights**(*layers*, *multi_layer_encoder=None*)

Return type `Tuple[float, ...]`

`pystiche_papers.gatys_ecker_bethge_2016.hyper_parameters(impl_params=True)`

Hyper parameters from [GEB16].

Return type *`HyperParameters`*

2.3.2 pystiche_papers.gatys_et_al_2017

Title	Controlling Perceptual Factors in Neural Style Transfer
Authors	Leon A. Gatys, Alexander. S. Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman
Citation	[GEB+17]
Reference implementation	Repository / Archive
Variant	Image optimization
Content loss	`FeatureReconstructionLoss`
Style loss	`GramLoss`

Behavioral changes

See also:

Paper implementations

The following parts are affected:

- *`MultiLayerEncodingLoss`*

Hyper parameters

`content_loss()`

Parameter	impl_params	
	True	False
layer	"relu4_2"	"conv4_2"
score_weight	1e0	

`style_loss()`

Parameter	impl_params	
	True	False
layers	("relu1_1", "relu2_1", "relu3_1", "relu4_1", "relu5_1")	("conv1_1", "conv2_1", "conv3_1", "conv4_1", "conv5_1")
layer_weights	(3.4e-04, 6.1e-05, 1.5e-05, 3.8e-06, 3.8e-06)[1]	
score_weight	1e3[?]	

[1] The values are reported in the supplementary material.

guided_style_loss()

Parameter	impl_params	
	True	False
layers	("relu1_1", "relu2_1", "relu3_1", "relu4_1", "relu5_1")	("conv1_1", "conv2_1", "conv3_1", "conv4_1", "conv5_1")
layer_weights	(2.4e-04, 6.1e-05, 1.5e-05, 3.8e-06, 3.8e-06)[?][2]	
region_weights		"sum"
score_weight	1e3[?]	

image_pyramid()

Parameter	impl_params	
	True	False
edge_sizes	(500, 1024)[3]	(512, 1024)
num_steps	[4]	(500, 200)

API

pystiche_papers.gatys_et_al_2017.**images**()

Return type DownloadableImageCollection

pystiche_papers.gatys_et_al_2017.**content_loss**(*multi_layer_encoder=None*, *hyper_parameters=None*)

Content_loss from [GEB+17].

Parameters

- **multi_layer_encoder** (Optional[MultiLayerEncoder]) – If omitted, *multi_layer_encoder()* is used.
- **hyper_parameters** (Optional[*HyperParameters*]) – If omitted, *hyper_parameters()* is used.

Return type FeatureReconstructionLoss

class pystiche_papers.gatys_et_al_2017.**MultiLayerEncodingLoss**(*multi_layer_encoder*, *layers*, *encoding_loss_fn*, *impl_params=True*, ***multi_layer_encoding_op_kwargs*)

Multi-layer encoding Loss from [GEB+17].

Parameters

- **multi_layer_encoder** (MultiLayerEncoder) – Multi-layer encoder.
- **layers** (Sequence[str]) – Layers of the multi_layer_encoder that the children losses operate on.

[2] The layer_weights are computed by $1/n^2$ where n denotes the number of channels of a feature map from the corresponding layer in the *multi_layer_encoder()*.

[3] The paper only reports the edge_size for the low resolution.

[4] The paper only reports the ratio. i.e. $500/200 = 2.5$ of num_steps.

- **get_encoding_op** – Callable that returns a children Loss given a pystiche.enc.SingleLayerEncoder extracted from the multi_layer_encoder and its corresponding layer weight.
- **impl_params** (bool) – If False, use a score correction factor of 1/4.
- ****multi_layer_encoding_op_kwargs** – Additional parameters of a pystiche.loss.MultiLayerEncodingLoss.

See also:

- pystiche.loss.MultiLayerEncodingLoss

pystiche_papers.gatys_et_al_2017.**style_loss**(*impl_params=True*, *multi_layer_encoder=None*, *hyper_parameters=None*)

Style_loss from [GEB+17].

Parameters

- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **multi_layer_encoder** (Optional[MultiLayerEncoder]) – If omitted, *multi_layer_encoder()* is used.
- **hyper_parameters** (Optional[*HyperParameters*]) – If omitted, *hyper_parameters()* is used.

See also:

- *pystiche_papers.gatys_et_al_2017.MultiLayerEncodingLoss*

Return type *MultiLayerEncodingLoss*

pystiche_papers.gatys_et_al_2017.**guided_style_loss**(*regions*, *impl_params=True*, *multi_layer_encoder=None*, *hyper_parameters=None*)

Guided style_loss from [GEB+17].

Parameters

- **regions** (Sequence[str]) – Regions of the input image to be stylized.
- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **multi_layer_encoder** (Optional[MultiLayerEncoder]) – If omitted, *multi_layer_encoder()* is used.
- **hyper_parameters** (Optional[*HyperParameters*]) – If omitted, *hyper_parameters()* is used.

See also:

- *pystiche_papers.gatys_et_al_2017.MultiLayerEncodingLoss*

Return type MultiRegionLoss

`pystiche_papers.gatys_et_al_2017.perceptual_loss(impl_params=True, multi_layer_encoder=None, hyper_parameters=None)`

Perceptual loss from [GEB+17].

Parameters

- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **multi_layer_encoder** (Optional[MultiLayerEncoder]) – If omitted, *multi_layer_encoder()* is used.
- **hyper_parameters** (Optional[*HyperParameters*]) – If omitted, *hyper_parameters()* is used.

Return type PerceptualLoss

`pystiche_papers.gatys_et_al_2017.guided_perceptual_loss(regions, impl_params=True, multi_layer_encoder=None, hyper_parameters=None)`

Guided perceptual loss from [GEB+17].

Parameters

- **regions** (Sequence[str]) – Regions of the input image to be stylized.
- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **multi_layer_encoder** (Optional[MultiLayerEncoder]) – If omitted, *multi_layer_encoder()* is used.
- **hyper_parameters** (Optional[*HyperParameters*]) – If omitted, *hyper_parameters()* is used.

Return type PerceptualLoss

`pystiche_papers.gatys_et_al_2017.nst(content_image, style_image, impl_params=True, hyper_parameters=None, quiet=False)`

NST from [GEB+17].

Parameters

- **content_image** (Tensor) – Content image for the NST.
- **style_image** (Tensor) – Style image for the NST.
- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **hyper_parameters** (Optional[*HyperParameters*]) – If omitted, *hyper_parameters()* is used.
- **quiet** (bool) – If True, not information is logged during the optimization. Defaults to False.

Return type Tensor

pystiche_papers.gatys_et_al_2017.**guided_nst**(*content_image*, *content_guides*, *style_images_and_guides*, *impl_params=True*, *hyper_parameters=None*, *quiet=False*)

Guided NST from [GEB+17].

Parameters

- **content_image** (Tensor) – Content image for the guided NST.
- **content_guides** (Dict[str, Tensor]) – Content image guides for the guided NST.
- **style_images_and_guides** (Dict[str, Tuple[Tensor, Tensor]]) – Dictionary with the style images and the corresponding guides for each region.
- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **hyper_parameters** (Optional[*HyperParameters*]) – If omitted, *hyper_parameters()* is used.
- **quiet** (bool) – If True, not information is logged during the optimization. Defaults to False.

Return type Tensor

pystiche_papers.gatys_et_al_2017.**image_pyramid**(*hyper_parameters=None*, ***image_pyramid_kwargs*)

Image pyramid from [GEB+17].

Parameters

- **hyper_parameters** (Optional[*HyperParameters*]) – If omitted, *hyper_parameters()* is used.
- ****image_pyramid_kwargs** – Additional parameters of a pystiche.pyramid.ImagePyramid.

See also:

- pystiche.pyramid.ImagePyramid

Return type ImagePyramid

pystiche_papers.gatys_et_al_2017.**preprocessor**()

Return type CaffePreprocessing

pystiche_papers.gatys_et_al_2017.**postprocessor**()

Return type CaffePostprocessing

pystiche_papers.gatys_et_al_2017.**multi_layer_encoder**()

Multi-layer encoder from [GEB+17].

Return type MultiLayerEncoder

pystiche_papers.gatys_et_al_2017.**optimizer**(*input_image*)

Optimizer from [GEB+17].

Parameters **input_image** (Tensor) – Image to be optimized.

Return type LBFGS

`pystiche_papers.gatys_et_al_2017.compute_layer_weights(layers, multi_layer_encoder=None)`

Return type `Tuple[float, ...]`

`pystiche_papers.gatys_et_al_2017.hyper_parameters(impl_params=True)`

Hyper parameters from [GEB+17].

Return type *HyperParameters*

2.3.3 pystiche_papers.johnson_alahi_li_2016

Title	Perceptual Losses for Real-Time Style Transfer and Super-Resolution
Authors	Justin Johnson, Alexandre Alahi, and Fei-Fei Li
Citation	[JAL16]
Reference implementation	Repository / Archive
Variant	Model optimization
Content loss	`FeatureReconstructionLoss`
Style loss	`GramLoss`
Regularization	`TotalVariationLoss`

Behavioral changes

See also:

Paper implementations

The following parts are affected:

- *content_transform()*
- *GramLoss*
- *TotalVariationLoss*
- *decoder()*
- *preprocessor()*
- *postprocessor()*
- *multi_layer_encoder()*

Hyper parameters

See also:

Paper implementations

content_loss()

Parameter	impl_params	
	True	False
layer	"relu2_2"	
score_weight		1e0

style_loss()

Parameter	impl_params	
	True	False
layers	("relu1_2", "relu2_2", "relu3_3", "relu4_3")	
layer_weights	"sum"	
score_weight		5e0

regularization()

Parameter	impl_params	
	True	False
score_weight		1e-6

content_transform()

Parameter	impl_params	
	True	False
image_size	(256, 256)	

style_transform()

Parameter	impl_params	
	True	False
edge_size	256	
edge	"long"	

`batch_sampler()`

Parameter	impl_params	
	True	False
num_batches	40000	
batch_size	4	

API

`pystiche_papers.johnson_alahi_li_2016.content_transform(impl_params=True, hyper_parameters=None)`

Content image transformation from [JAL16].

Parameters

- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.

  Additionally, if True, appends the *preprocessor()* as a last transformation step.

- **hyper_parameters** (Optional[*HyperParameters*]) – If omitted, *hyper_parameters()* is used.

Return type Sequential

`pystiche_papers.johnson_alahi_li_2016.style_transform(hyper_parameters=None)`

Style image transformation from [JAL16].

Parameters **hyper_parameters** (Optional[*HyperParameters*]) – If omitted, *hyper_parameters()* is used.

Return type Module

`pystiche_papers.johnson_alahi_li_2016.images()`

Return type DownloadableImageCollection

`pystiche_papers.johnson_alahi_li_2016.dataset(root, impl_params=True, transform=None)`

Return type ImageFolderDataset

`pystiche_papers.johnson_alahi_li_2016.batch_sampler(data_source, hyper_parameters=None)`

Batch sampler from [JAL16].

Parameters

- **data_source** (Sized) – Dataset to sample from.
- **hyper_parameters** (Optional[*HyperParameters*]) – If omitted, *hyper_parameters()* is used.

Return type *FiniteCycleBatchSampler*

`pystiche_papers.johnson_alahi_li_2016.image_loader(dataset, hyper_parameters=None, pin_memory=True)`

Return type DataLoader

`pystiche_papers.johnson_alahi_li_2016.content_loss(impl_params=True, multi_layer_encoder=None, hyper_parameters=None)`

Content loss from [JAL16].

Parameters

- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **multi_layer_encoder** (Optional[MultiLayerEncoder]) – Pretrained MultiLayerEncoder. If omitted, the default *multi_layer_encoder()* is used.
- **hyper_parameters** (Optional[*HyperParameters*]) – If omitted, *hyper_parameters()* is used.

Return type FeatureReconstructionLoss

class `pystiche_papers.johnson_alahi_li_2016.GramLoss(encoder, impl_params=True, **gram_op_kwargs)`

Gram loss from [JAL16].

Parameters

- **encoder** (Encoder) – Encoder used to encode the input.
- **impl_params** (bool) – If True, normalize the Gram matrix additionally by the number of channels.
- ****gram_op_kwargs** – Additional parameters of a pystiche.loss.GramLoss.

See also:

- pystiche.loss.GramLoss

`pystiche_papers.johnson_alahi_li_2016.style_loss(impl_params=True, multi_layer_encoder=None, hyper_parameters=None)`

Style loss from [JAL16].

Parameters

- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **multi_layer_encoder** (Optional[MultiLayerEncoder]) – Pretrained MultiLayerEncoder. If omitted, the default *multi_layer_encoder()* is used.
- **hyper_parameters** (Optional[*HyperParameters*]) – If omitted, *hyper_parameters()* is used.

Return type MultiLayerEncodingLoss

class `pystiche_papers.johnson_alahi_li_2016.TotalVariationLoss(**total_variation_op_kwargs)`

Total variation loss from [LW16].

Parameters ****total_variation_op_kwargs** – Additional parameters of a pystiche.loss.TotalVariationLoss.

In contrast to pystiche.loss.TotalVariationLoss, the score is calculated with the squared error (SE) instead of the mean squared error (MSE).

See also:

- pystiche.loss.TotalVariationLoss

pystiche_papers.johnson_alahi_li_2016.**regularization**(*hyper_parameters=None*)

Regularization from [JAL16].

Parameters **hyper_parameters** (Optional[*HyperParameters*]) – If omitted, *hyper_parameters()* is used.

Return type *TotalVariationLoss*

pystiche_papers.johnson_alahi_li_2016.**perceptual_loss**(*impl_params=True*, *multi_layer_encoder=None*, *hyper_parameters=None*)

Perceptual loss comprising content and style loss as well as a regularization.

Parameters

- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **multi_layer_encoder** (Optional[MultiLayerEncoder]) – Pretrained MultiLayerEncoder. If omitted, the default *multi_layer_encoder()* is used.
- **hyper_parameters** (Optional[*HyperParameters*]) – If omitted, *hyper_parameters()* is used.

Return type PerceptualLoss

pystiche_papers.johnson_alahi_li_2016.**encoder**(*instance_norm=True*)

Encoder part of the *Transformer* from [JAL16] .

Parameters **instance_norm** (bool) – If True, use InstanceNorm2d rather than BatchNorm2d as described in the paper. In addition, the number of channels of the convolution layers is reduced by half.

Return type SequentialModule

pystiche_papers.johnson_alahi_li_2016.**decoder**(*impl_params=True*, *instance_norm=True*)

Decoder part of the *Transformer* from [JAL16].

Parameters

- **impl_params** (bool) – If True, the output of the is not externally pre-processed before being fed into the *perceptual_loss()*. Since this step is necessary to get meaningful encodings from the *multi_layer_encoder()*, the pre-processing transform has to be learned within the output layer of the decoder. To make this possible, `150 * tanh(input)` is used as activation in contrast to the `(tanh(input) + 1) / 2` given in the paper.
- **instance_norm** (bool) – If True, use InstanceNorm2d rather than BatchNorm2d as described in the paper. In addition, the number of channels of the convolution layers is reduced by half.

Return type SequentialModule

class pystiche_papers.johnson_alahi_li_2016.**Transformer**(*impl_params=True*, *instance_norm=True*, *init_weights=True*)

pystiche_papers.johnson_alahi_li_2016.**transformer**(*style=None*, *framework='pystiche'*, *impl_params=True*, *instance_norm=True*)

Pretrained transformer from [JAL16] .

Parameters

- **style** (Optional[str]) – Style the transformer was trained on. Can be one of styles given by *images()*. If omitted, the transformer is initialized with random weights according to the procedure used by the original authors.
- **framework** (str) – Framework that was used to train the the transformer. Can be one of "pystiche" (default) and "luatorch".
- **impl_params** (bool) – If True, use the parameters used in the reference implementation of the original authors rather than what is described in the paper.
- **instance_norm** (bool) – If True, use InstanceNorm2d rather than BatchNorm2d as described in the paper.

For framework == "pystiche" all combinations of parameters are available.

The weights for framework == "luatorch" were ported from the reference implementation (impl_params is True) of the original authors. See https://download.pystiche.org/models/LICENSE for licensing details. The following combinations of parameters are available:

style	instance_norm	
	True	False
"candy"	x	
"composition_vii"		x
"feathers"	x	
"la_muse"	x	x
"mosaic"	x	
"starry_night"		x
"the_scream"	x	
"the_wave"		x
"udnie"	x	

Return type *Transformer*

pystiche_papers.johnson_alahi_li_2016.**training**(*content_image_loader*, *style_image*, *impl_params=True*, *instance_norm=None*, *hyper_parameters=None*, *quiet=False*)

Training a transformer for the NST.

Parameters

- **content_image_loader** (DataLoader) – Content images used as input for the transformer.
- **style_image** (Union[str, Tensor]) – Style image on which the transformer should be trained. If str, the image is read from *images()*.
- **impl_params** (bool) – If True, uses the parameters used in the reference implementation of the original authors rather than what is described in the paper. For details see below.
- **instance_norm** (Optional[bool]) – If True, use InstanceNorm2d rather than BatchNorm2d as described in the paper. If omitted, defaults to impl_params.
- **hyper_parameters** (Optional[*HyperParameters*]) – If omitted, *hyper_parameters()* is used.
- **quiet** (bool) – If True, not information is logged during the optimization. Defaults to False.

If impl_params is True , an external preprocessing of the images is used.

Return type Module

pystiche_papers.johnson_alahi_li_2016.**stylization**(*input_image*, *transformer*, *impl_params=True*, *instance_norm=None*, *framework='pystiche'*)

Transforms an input image into a stylised version using the transfromer.

Parameters

- **input_image** (Tensor) – Image to be stylised.
- **transformer** (Union[Module, str]) – Pretrained transformer for style transfer or the `style` to load a pretrained transformer with *transformer()*.
- **impl_params** (bool) – If `True`, uses the parameters used in the reference implementation of the original authors rather than what is described in the paper. For details see below.
- **instance_norm** (Optional[bool]) – If `True`, use InstanceNorm2d rather than BatchNorm2d as described in the paper. If omitted, defaults to `impl_params`.
- **framework** (str) – Framework that was used to train the the transformer. Can be one of `"pystiche"` (default) and `"luatorch"`. This only has an effect, if if a pretrained `transformer` is loaded.

Return type Tensor

pystiche_papers.johnson_alahi_li_2016.**hyper_parameters**()

Hyper parameters from [JAL16].

Return type *HyperParameters*

pystiche_papers.johnson_alahi_li_2016.**preprocessor**(*impl_params=True*)

Preprocessor from [JAL16].

Parameters **impl_params** (bool) – If `True`, the input is preprocessed for models trained with the Caffe framework. If `False`, the preprocessor performs the identity operation.

See also:

- `pystiche.enc.CaffePreprocessing`

Return type Module

pystiche_papers.johnson_alahi_li_2016.**postprocessor**(*impl_params=True*)

Postprocessor from [JAL16].

Parameters **impl_params** (bool) – If `True`, the input is postprocessed from models trained with the Caffe framework. If `False`, the postprocessor performs the identity operation.

See also:

- `pystiche.enc.CaffePostprocessing`

Return type Module

pystiche_papers.johnson_alahi_li_2016.**multi_layer_encoder**(*impl_params=True*)

Multi-layer encoder from [JAL16].

Parameters **impl_params** (bool) – If `True`, the necessary preprocessing of the images is performed internally.

Return type VGGMultiLayerEncoder

`pystiche_papers.johnson_alahi_li_2016.optimizer(transformer)`

Optimizer from [JAL16].

Parameters **transformer** (Module) – Transformer to be optimized.

Return type Adam

2.3.4 pystiche_papers.li_wand_2016

Title	Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis
Authors	Chuan Li and Michael Wand
Citation	[LW16]
Reference implementation	Repository / Archive
Variant	Image optimization
Content loss	FeatureReconstructionLoss
Style loss	MRFLoss
Regularization	TotalVariationLoss

Behavioral changes

See also:

Paper implementations

The following parts are affected:

- *FeatureReconstructionLoss*
- *MRFLoss*
- *TotalVariationLoss*
- *target_transforms*

Hyper parameters

See also:

Paper implementations

content_loss()

Parameter	impl_params	
	True	False
layer	"relu4_1"	"relu_4_2"
score_weight	2e1	1e0

target_transforms()

Parameter	impl_params	
	True	False
num_scale_steps	0	3
scale_step_width	5e-2	
num_rotate_steps	0	2
rotate_step_width	7.5	

style_loss()

Parameter	impl_params	
	True	False
layers	("relu3_1", "relu4_1")	
layer_weights	"sum"	
patch_size	3	
stride	2	1
score_weight	1e-4	1e0

regularization()

Parameter	impl_params	
	True	False
score_weight	1e-3	

image_pyramid()

Parameter	impl_params	
	True	False
max_edge_size	384	
num_steps	100	200
num_levels	3	None[1]
min_edge_size	64	
edge	"long"	

[1] num_levels=None implies that the number of levels is automatically calculated depending on max_edge_size and min_edge_size. See pystiche.pyramid.OctaveImagePyramid for details.

API

pystiche_papers.li_wand_2016.**images**()

Return type DownloadableImageCollection

class pystiche_papers.li_wand_2016.**FeatureReconstructionLoss**(*encoder*, *impl_params=True*, ***feature_reconstruction_loss_kwargs*)

Feature reconstruction loss from [LW16].

Parameters

- **encoder** (Encoder) – Encoder used to encode the input.
- **impl_params** (bool) – If False, calculate the score with the squared error (SE) instead of the mean squared error (MSE).
- ****feature_reconstruction_loss_kwargs** – Additional parameters of a pystiche.loss.FeatureReconstructionLoss.

See also:

pystiche.loss.FeatureReconstructionLoss

pystiche_papers.li_wand_2016.**content_loss**(*impl_params=True*, *multi_layer_encoder=None*, *hyper_parameters=None*)

Content loss from [LW16].

Parameters

- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **multi_layer_encoder** (Optional[MultiLayerEncoder]) – Pretrained multi-layer encoder. If omitted, *multi_layer_encoder()* is used.
- **hyper_parameters** (Optional[*HyperParameters*]) – Hyper parameters. If omitted, *hyper_parameters()* is used.

See also:

pystiche_papers.li_wand_2016.FeatureReconstructionLoss

Return type *FeatureReconstructionLoss*

class pystiche_papers.li_wand_2016.**MRFLoss**(*encoder*, *patch_size*, *impl_params=True*, ***mrf_loss_kwargs*)

MRF loss from [LW16].

Parameters

- **encoder** (Encoder) – Encoder used to encode the input.
- **patch_size** (Union[int, Tuple[int, int]]) – Spatial size of the neural patches.
- **impl_params** (bool) – If True, normalize the gradient of the neural patches. If False, use a score correction factor of 1/2.
- ****mrf_loss_kwargs** – Additional parameters of a pystiche.loss.MRFLoss.

In contrast to `pystiche.loss.MRFLoss`, the score is calculated with the squared error (SE) instead of the mean squared error (MSE).

See also:

- `pystiche.loss.MRFLoss`
- *`pystiche_papers.li_wand_2016.extract_normalized_patches2d()`*

`pystiche_papers.li_wand_2016.`**`style_loss`**(*impl_params=True*, *multi_layer_encoder=None*, *hyper_parameters=None*)

Style loss from [LW16].

Parameters

- **`impl_params`** (`bool`) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **`multi_layer_encoder`** (`Optional`[`MultiLayerEncoder`]) – Pretrained multi-layer encoder. If omitted, *`multi_layer_encoder()`* is used.
- **`hyper_parameters`** (`Optional`[*`HyperParameters`*]) – Hyper parameters. If omitted, *`hyper_parameters()`* is used.

See also:

- *`pystiche_papers.li_wand_2016.MRFLoss`*

Return type `MultiLayerEncodingLoss`

`class` `pystiche_papers.li_wand_2016.`**`TotalVariationLoss`**(*impl_params=True*, ***total_variation_loss_kwargs*)

Total variation loss from [LW16].

Parameters

- **`impl_params`** (`bool`) – If `False`, use a score correction factor of 1/2.
- **`**total_variation_loss_kwargs`** – Additional parameters of a `pystiche.loss.TotalVariationLoss`.

In contrast to `pystiche.loss.TotalVariationLoss`, the the score is calculated with the squared error (SE) instead of the mean squared error (MSE).

See also:

- `pystiche.loss.TotalVariationLoss`

`pystiche_papers.li_wand_2016.`**`regularization`**(*impl_params=True*, *hyper_parameters=None*)

Regularization from [LW16].

Parameters

- **`impl_params`** (`bool`) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **`hyper_parameters`** (`Optional`[*`HyperParameters`*]) – Hyper parameters. If omitted, *`hyper_parameters()`* is used.

See also:

- *pystiche_papers.li_wand_2016.TotalVariationLoss*

Return type *TotalVariationLoss*

pystiche_papers.li_wand_2016.**perceptual_loss**(*impl_params=True*, *multi_layer_encoder=None*, *hyper_parameters=None*)

Perceptual loss from [LW16].

Parameters

- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **multi_layer_encoder** (Optional[MultiLayerEncoder]) – Pretrained multi-layer encoder. If omitted, *multi_layer_encoder()* is used.
- **hyper_parameters** (Optional[*HyperParameters*]) – Hyper parameters. If omitted, *hyper_parameters()* is used.

See also:

- *pystiche_papers.li_wand_2016.content_loss()*
- *pystiche_papers.li_wand_2016.style_loss()*
- *pystiche_papers.li_wand_2016.regularization()*

Return type PerceptualLoss

pystiche_papers.li_wand_2016.**nst**(*content_image*, *style_image*, *impl_params=True*, *hyper_parameters=None*, *quiet=False*)

NST from [LW16].

Parameters

- **content_image** (Tensor) – Content image for the NST.
- **style_image** (Tensor) – Style image for the NST.
- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **hyper_parameters** (Optional[*HyperParameters*]) – If omitted, *hyper_parameters()* is used.
- **quiet** (bool) – If True, not information is logged during the optimization. Defaults to False.

Return type Tensor

pystiche_papers.li_wand_2016.**image_pyramid**(*impl_params=True*, *hyper_parameters=None*, ***image_pyramid_kwargs*)

Image pyramid from [LW16].

Parameters

- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.

- **hyper_parameters** (Optional[*HyperParameters*]) – If omitted, *hyper_parameters()* is used.
- **image_pyramid_kwargs** (Any) – Additional options. See ImagePyramid for details.

See also:

- pystiche.pyramid.OctaveImagePyramid

Return type OctaveImagePyramid

pystiche_papers.li_wand_2016.**hyper_parameters**(*impl_params=True*)

Hyper parameters from [LW16].

Parameters **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.

Return type *HyperParameters*

pystiche_papers.li_wand_2016.**extract_normalized_patches2d**(*input*, *patch_size*, *stride*)

Extract 2-dimensional patches from the input with normalized gradient.

If stride >= patch_size, this behaves just like pystiche.extract_patches2d(). Otherwise, the gradient of the input is normalized such that every value is divided by the number of patches it appears in.

Examples

```
>>> import torch
>>> import pystiche
>>> input = torch.ones(1, 1, 4, 4).requires_grad_(True)
>>> target = torch.zeros(1, 1, 4, 4).detach()
>>> # without normalized gradient
>>> input_patches = pystiche.extract_patches2d(
...    input, patch_size=2, stride=1
... )
>>> target_patches = pystiche.extract_patches2d(
...     target, patch_size=2, stride=1
... )
>>> loss = 0.5 * torch.sum((input_patches - target_patches) ** 2.0)
>>> loss.backward()
>>> input.grad
tensor([[[[1., 2., 2., 1.],
          [2., 4., 4., 2.],
          [2., 4., 4., 2.],
          [1., 2., 2., 1.]]]])
```

```
>>> import torch
>>> import pystiche
>>> import pystiche_papers.li_wand_2016 as paper
>>> input = torch.ones(1, 1, 4, 4).requires_grad_(True)
>>> target = torch.zeros(1, 1, 4, 4).detach()
>>> # with normalized gradient
>>> input_patches = paper.extract_normalized_patches2d(
...    input, patch_size=2, stride=1
```

(continues on next page)

(continued from previous page)

```
... )
>>> target_patches = pystiche.extract_patches2d(
...     target, patch_size=2, stride=1
... )
>>> loss = 0.5 * torch.sum((input_patches - target_patches) ** 2.0)
>>> loss.backward()
>>> input.grad
tensor([[[[1., 1., 1., 1.],
          [1., 1., 1., 1.],
          [1., 1., 1., 1.],
          [1., 1., 1., 1.]]]])
```

Parameters

- **input** (Tensor) – Input tensor of shape $B \times C \times H \times W$
- **patch_size** (Union[int, Sequence[int]]) – Patch size
- **stride** (Union[int, Sequence[int]]) – Stride

Return type Tensor

pystiche_papers.li_wand_2016.**target_transforms**(*impl_params=True*, *hyper_parameters=None*)

MRF target transformations from [LW16].

Parameters

- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*. In addition, if True, every transformation comprises a valid crop after the rotation to avoid blank regions. Furthermore, the image is rescaled instead of the motif, resulting in multiple image sizes.
- **hyper_parameters** (Optional[*HyperParameters*]) – Hyper parameters. If omitted, *hyper_parameters()* is used.

See also:

- pystiche.loss.MRFLoss.scale_and_rotate_transforms()

Return type Sequence[Module]

pystiche_papers.li_wand_2016.**preprocessor**()

Preprocessor from [LW16].

Return type CaffePreprocessing

pystiche_papers.li_wand_2016.**postprocessor**()

Postprocessor from [LW16].

Return type CaffePostprocessing

pystiche_papers.li_wand_2016.**multi_layer_encoder**()

Multi-layer encoder from [LW16].

Return type MultiLayerEncoder

pystiche_papers.li_wand_2016.**optimizer**(*input_image*)

Optimizer from [LW16].

Parameters `input_image` (`Tensor`) – Image to be optimized.

Return type `LBFGS`

2.3.5 pystiche_papers.ulyanov_et_al_2016

Title	Texture Networks: Feed-forward Synthesis of Textures and Stylized Images
Authors	Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Viktor S. Lempitsky
Citation	[ULVL16] / [UVL17]
Reference implementation	Repository / Archive
Variant	Model optimization
Content loss	`FeatureReconstructionLoss`
Style loss	`GramLoss`

Instance norm

The authors published an improved version [UVL17] of their initial paper [ULVL16] with only a single but significant change: they developed `InstanceNorm2d` and used it as drop-in replacement for `BatchNorm2d` layers. To account for this we provide `instance_norm` flag, which defaults to `True`.

The original authors also use the same repository for both implementations and only differentiate between them with the branches:

Paper	Branch
[ULVL16]	`texture_nets_v1`
[UVL17]	`master`

Behavioral changes

See also:

Paper implementations

The following parts are affected:

- *content_transform()*,
- *GramLoss()*,
- `ConvBlock()`,
- `level()`,
- `Transformer()`.

Hyper parameters

See also:

Paper implementations

Although there are four possible combinations for `impl_params` and `instance_norm` only three are listed below. Since the hyper-parameters in both papers are equal, both combinations with `impl_params=False` are equal and thus not reported.

`content_loss()`

Parameter	impl_params / instance_norm		
	True / True	True / False	False
layer	"relu4_2"		
score_weight	1e0	6e-1	1e0

`style_loss()`

Parameter	impl_params		
	True / True	True / False	False
layers	("relu1_1", "relu2_1", "relu3_1", "relu4_1")	("relu1_1", "relu2_1", "relu3_1", "relu4_1", "relu5_1")	
layer_weight	"sum"		
score_weight	1e0	1e3	1e0

`content_transform()`

Parameter	Value
edge_size	256

`style_transform()`

Parameter	impl_params / instance_norm		
	True / True	True / False	False
edge_size	256		
edge	"long"		
interpolation_mode	"bicubic"	"bilinear"	

batch_sampler()

Parameter	impl_params / instance_norm		
	True / True	True / False	False
num_batches	2_000	300	200
batch_size	1	4	16

optimizer()

Parameter	impl_params / instance_norm		
	True / True	True / False	False
lr	1e-3	1e0	

lr_scheduler()

Parameter	impl_params / instance_norm		
	True / True	True / False	False
lr_decay	0.8		0.7
delay	0		4

Miscellaneous

Parameter	impl_params / instance_norm		
	True / True	True / False	False
num_epochs	25	10	

API

pystiche_papers.ulyanov_et_al_2016.**content_transform**(*impl_params=True*, *instance_norm=True*, *hyper_parameters=None*)

Content transform from [ULVL16][UVL17].

Parameters

- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **instance_norm** (bool) – Switch the behavior and hyper-parameters between both publications of the original authors. For details see *here*.
- **hyper_parameters** (Optional[*HyperParameters*]) – Hyper parameters. If omitted, *hyper_parameters()* is used.

Return type Sequential

pystiche_papers.ulyanov_et_al_2016.**style_transform**(*impl_params=True*, *instance_norm=True*, *hyper_parameters=None*)

Style transform from [ULVL16][UVL17].

Parameters

- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **instance_norm** (bool) – Switch the behavior and hyper-parameters between both publications of the original authors. For details see *here*.
- **hyper_parameters** (Optional[*HyperParameters*]) – Hyper parameters. If omitted, *hyper_parameters()* is used.

Return type Module

pystiche_papers.ulyanov_et_al_2016.**images**()

Images from [ULVL16][UVL17].

Return type DownloadableImageCollection

pystiche_papers.ulyanov_et_al_2016.**dataset**(*root*, *impl_params=True*, *instance_norm=True*, *transform=None*, *hyper_parameters=None*)

Return type Sized

pystiche_papers.ulyanov_et_al_2016.**image_loader**(*dataset*, *impl_params=True*, *instance_norm=True*, *num_workers=0*, *pin_memory=True*, *hyper_parameters=None*)

Return type DataLoader

pystiche_papers.ulyanov_et_al_2016.**content_loss**(*impl_params=True*, *instance_norm=True*, *multi_layer_encoder=None*, *hyper_parameters=None*)

Content loss from [ULVL16][UVL17].

Parameters

- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **instance_norm** (bool) – Switch the behavior and hyper-parameters between both publications of the original authors. For details see *here*.
- **multi_layer_encoder** (Optional[MultiLayerEncoder]) – Pretrained MultiLayerEncoder. If omitted, *multi_layer_encoder()* is used.
- **hyper_parameters** (Optional[*HyperParameters*]) – Hyper parameters. If omitted, *hyper_parameters()* is used.

See also:

- pystiche.loss.FeatureReconstructionLoss

Return type FeatureReconstructionLoss

class pystiche_papers.ulyanov_et_al_2016.**GramLoss**(*encoder*, *impl_params=True*, ***gram_op_kwargs*)

Gram loss from [ULVL16][UVL17].

Parameters

- **encoder** (Encoder) – Encoder used to encode the input.
- **impl_params** (bool) – If True, normalize the score twice by the batch size.
- ****gram_op_kwargs** – Additional parameters of a pystiche.loss.GramLoss.

See also:

- pystiche.loss.GramLoss

pystiche_papers.ulyanov_et_al_2016.**style_loss**(*impl_params=True*, *instance_norm=True*, *multi_layer_encoder=None*, *hyper_parameters=None*)

Style loss from [ULVL16][UVL17].

Parameters

- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **instance_norm** (bool) – Switch the behavior and hyper-parameters between both publications of the original authors. For details see *here*.
- **multi_layer_encoder** (Optional[MultiLayerEncoder]) – Pretrained MultiLayerEncoder. If omitted, *multi_layer_encoder()* is used.
- **hyper_parameters** (Optional[*HyperParameters*]) – Hyper parameters. If omitted, *hyper_parameters()* is used.

See also:

- *pystiche_papers.ulyanov_et_al_2016.GramLoss*

Return type MultiLayerEncodingLoss

pystiche_papers.ulyanov_et_al_2016.**perceptual_loss**(*impl_params=True*, *instance_norm=True*, *multi_layer_encoder=None*, *hyper_parameters=None*)

Perceptual loss from [ULVL16][UVL17].

Parameters

- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **instance_norm** (bool) – Switch the behavior and hyper-parameters between both publications of the original authors. For details see *here*.
- **multi_layer_encoder** (Optional[MultiLayerEncoder]) – Pretrained MultiLayerEncoder. If omitted, *multi_layer_encoder()* is used.
- **hyper_parameters** (Optional[*HyperParameters*]) – Hyper parameters. If omitted, *hyper_parameters()* is used.

See also:

- *pystiche_papers.ulyanov_et_al_2016.content_loss()*
- *pystiche_papers.ulyanov_et_al_2016.style_loss()*

Return type PerceptualLoss

`pystiche_papers.ulyanov_et_al_2016.transformer(style=None, impl_params=True, instance_norm=True, levels=6)`

Transformer from [ULVL16][UVL17].

Parameters

- **style** (Optional[str]) – Style the transformer was trained on. Can be one of styles given by *images()*. If omitted, the transformer is initialized with random weights.
- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **instance_norm** (bool) – Switch the behavior and hyper-parameters between both publications of the original authors. For details see *here*.
- **levels** (int) – Number of levels in the transformer. Defaults to 6.

Return type Transformer

`pystiche_papers.ulyanov_et_al_2016.training(content_image_loader, style, impl_params=True, instance_norm=True, hyper_parameters=None, quiet=False)`

Training a transformer for the NST.

Parameters

- **content_image_loader** (DataLoader) – Content images used as input for the transformer.
- **style** (Union[str, Tensor]) – Style image on which the transformer should be trained. If the input is str, the style image is read from *images()*.
- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **instance_norm** (bool) – Switch the behavior and hyper-parameters between both publications of the original authors. For details see *here*.
- **hyper_parameters** (Optional[*HyperParameters*]) – Hyper parameters. If omitted, *hyper_parameters()* is used.
- **quiet** (bool) – If True, not information is logged during the optimization. Defaults to False.

Return type Module

`pystiche_papers.ulyanov_et_al_2016.stylization(input_image, transformer, impl_params=True, instance_norm=False, hyper_parameters=None)`

Transforms an input image into a stylised version using the transformer.

Parameters

- **input_image** (Tensor) – Image to be stylised.
- **transformer** (Union[Module, str]) – Pretrained transformer for style transfer or string to load a pretrained transformer.
- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.

- **instance_norm** (bool) – Switch the behavior and hyper-parameters between both publications of the original authors. For details see *here*.
- **hyper_parameters** (Optional[*HyperParameters*]) – Hyper parameters. If omitted, *hyper_parameters()* is used.

Return type Tensor

pystiche_papers.ulyanov_et_al_2016.**hyper_parameters**(*impl_params=True*, *instance_norm=True*)

Hyper parameters from [ULVL16][UVL17].

Return type *HyperParameters*

pystiche_papers.ulyanov_et_al_2016.**multi_layer_encoder**()

Multi-layer encoder from [ULVL16][UVL17].

Return type VGGMultiLayerEncoder

pystiche_papers.ulyanov_et_al_2016.**preprocessor**()

Return type CaffePreprocessing

pystiche_papers.ulyanov_et_al_2016.**postprocessor**()

Return type CaffePostprocessing

pystiche_papers.ulyanov_et_al_2016.**optimizer**(*transformer*, *impl_params=True*, *instance_norm=True*, *hyper_parameters=None*)

Optimizer from [ULVL16][UVL17].

Parameters

- **transformer** (Module) – Transformer to be optimized.
- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **instance_norm** (bool) – Switch the behavior and hyper-parameters between both publications of the original authors. For details see *here*.
- **hyper_parameters** (Optional[*HyperParameters*]) – Hyper parameters. If omitted, *hyper_parameters()* is used.

Return type Adam

pystiche_papers.ulyanov_et_al_2016.**lr_scheduler**(*optimizer*, *impl_params=True*, *instance_norm=True*, *hyper_parameters=None*)

Learning rate scheduler from [ULVL16][UVL17].

Parameters

- **optimizer** (Optimizer) – Wrapped optimizer.
- **impl_params** (bool) – Switch the behavior and hyper-parameters between the reference implementation of the original authors and what is described in the paper. For details see *here*.
- **instance_norm** (bool) – Switch the behavior and hyper-parameters between both publications of the original authors. For details see *here*.
- **hyper_parameters** (Optional[*HyperParameters*]) – Hyper parameters. If omitted, *hyper_parameters()* is used.

Return type `ExponentialLR`

CHAPTER

THREE

LITERATURE REFERENCE

BIBLIOGRAPHY

[GEB+17] Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Controlling perceptual factors in neural style transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017. arXiv:1611.07865, doi:10.1109/CVPR.2017.397.

[GEB16] Leon A. Gatys, Alexander. S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016. arXiv:1508.06576, doi:10.1109/CVPR.2016.265.

[JAL16] Justin Johnson, Alexandre Alahi, and Fei-Fei Li. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*. 2016. arXiv:1603.08155, doi:10.1007/978-3-319-46475-6_43.

[LW16] Chuan Li and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016. arXiv:1601.04589, doi:10.1109/CVPR.2016.272.

[ULVL16] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Viktor S. Lempitsky. Texture networks: feed-forward synthesis of textures and stylized images. In *International Conference on Machine Learning (ICML)*. 2016. URL: http://proceedings.mlr.press/v48/ulyanov16.html, arXiv:1603.03417.

[UVL17] Dmitry Ulyanov, Andrea Vedaldi, and Viktor S. Lempitsky. Improved texture networks: maximizing quality and diversity in feed-forward stylization and texture synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017. arXiv:1701.02096, doi:10.1109/CVPR.2017.437.

PYTHON MODULE INDEX

p

INDEX

L

M

N

O

P

R

S

T