
HIRISE_api

unknown

Aug 29, 2022

CONTENTS

1	HIRISE_api	3
1.1	HIRISE_api.hirise	3
1.2	HIRISE_api.models	4
1.3	HIRISE_api.preprocessing	8
1.4	HIRISE_api.tests	11
	Python Module Index	13

HIRISE Api tool

HIRISE_api

HIRISE_API

Modules

HIRISE_api.hirise

HIRISE_api.models

HIRISE_api.preprocessing

HIRISE_api.tests

1.1 HIRISE_api.hirise

Modules

HIRISE_api.hirise.Hirise_Image

HIRISE_api.hirise.Image_Client

HIRISE_api.hirise.utils

1.1.1 HIRISE_api.hirise.Hirise_Image

Classes

<code>HiriseImage(file_name)</code>	Class that creates an HIRISE image object that has specific attributes including latitude longitude
-------------------------------------	---

1.1.2 HIRISE_api.hirise.Image_Client

Classes

ImageClient()

1.1.3 HIRISE_api.hirise.utils

Functions

LBL_parser(label_url)	Function that parses the .LBL file in NASA's Planetary Data System
append_float_data_without_strip(param, ...)	Function that validates floating point data without stripping the last characters
downloadRange(start_range, end, step)	
file_parameters_list()	Function that returns file parameters list
get_website_data(base_url, page_key[, sub_key])	Function that assistes in wescaping the NASA website
image_map_parameters_list()	Function that returns image mapping parameters list
other_parameters_list()	Function that returns scaling factor, offset, center filter wavelength parameters list
timing_parameters_list()	Function that returns timing parameters list
validate_append_float_data(param, list_of_params)	Function that validates floating point data
viewing_parameters_list()	Function that returns viewing parameters list

1.2 HIRISE_api.models

Modules

HIRISE_api.models.Affinity_Propagation

HIRISE_api.models.Agglomerative_Clustering

HIRISE_api.models.BIRCH

HIRISE_api.models.DBSCAN

HIRISE_api.models.Ensemble_Models

HIRISE_api.models.HDBSCAN

HIRISE_api.models.KMeans

HIRISE_api.models.Mean_Shift

HIRISE_api.models.OPTICS

HIRISE_api.models.metrics

HIRISE_api.models.utils

1.2.1 HIRISE_api.models.Affinity_Propagation

Functions

<code>affinity_propagation_analysis(...[, plot, ...])</code>	Function that uses as input the encoded image samples and clusters the data using affinity propagation.
--	---

1.2.2 HIRISE_api.models.Agglomerative_Clustering

Functions

<code>agglomerative_clustering_analysis(...[, ...])</code>	Function that uses as input the encoded image samples and clusters the data using agglomerative clustering.
--	---

1.2.3 HIRISE_api.models.BIRCH

Functions

<code>BIRCH_analysis(encoded_samples, ...[, plot, ...])</code>	Function that uses as input the encoded image samples and clusters the data using affinity propagation.
--	---

1.2.4 HIRISE_api.models.DBSCAN

Functions

<code>DBSCAN_analysis(encoded_samples, true_labels)</code>	Function that uses as input the encoded image samples and clusters the data using Density-based spatial clustering of applications with noise.
--	--

1.2.5 HIRISE_api.models.Ensemble_Models

Functions

<code>ensemble_model(encoded_data, labels, ...[, ...])</code>	
<code>evaluate_model(model, translation_dataframe, ...)</code>	Function that uses cross-validation and evaluates the stacking model.
<code>get_models([discovery])</code>	Function that defines specified models as an input to the ensemble model.
<code>get_stacking([discovery, all_models])</code>	Function that stacks specified models together as an input to the ensemble model.

Classes

<code>AgglomerativeClusteringWrapper([n_clusters, ...])</code>	
<code>DBSCANWrapper([eps, min_samples, metric, ...])</code>	
<code>HDBSCANWrapper([min_cluster_size, ...])</code>	
<code>OpticsWrapper(*[, min_samples, max_eps, ...])</code>	

1.2.6 HIRISE_api.models.HDBSCAN

Functions

HDBSCAN_analysis(encoded_samples[, ...])	Function that uses as input the encoded image samples and clusters the data using Hierarchical Density-based spatial clustering of applications with noise. The user must specify only the minimum samples, which is the tuning parameters for HDBSCAN.
--	---

1.2.7 HIRISE_api.models.KMeans

Functions

kmeans_analysis(encoded_samples[, clusters, ...])	Function that uses as input the encoded image samples and clusters the data using K Means clustering of applications with noise.
---	--

1.2.8 HIRISE_api.models.Mean_Shift

Functions

mean_shift_analysis(encoded_samples[, plot, ...])	Function that uses as input the encoded image samples and clusters the data using Mean Shift Clustering Method.
---	---

1.2.9 HIRISE_api.models.OPTICS

Functions

OPTICS_analysis(dataframe[, eps, ...])	Function that uses as input the encoded image samples and clusters the data using Ordering Points To Identify Cluster Structure Method.
--	---

1.2.10 HIRISE_api.models.metrics

Functions

<code>calculate_metrics(model, labels[, verbose])</code>	Function that calculates metrics including, rand score, adjusted rand score, mutual information score, normalized mutual information score, adjusted mutual information score, balanced accuracy score, completeness score, homogeneity score and v-score for a given model.
<code>classification_metrics_dataframe(model_list, ...)</code>	Function that creates a metrics dataframe based on the calculated metrics for each model in the model list specified by the user.
<code>generate_precision_dataframe(folder_path, ...)</code>	Function that returns a generated a dataframe of all the precision values evaluated for a true and predicted labels after classification analysis on a dataset.
<code>print_confusion_matrix(folder_path, ...[, ...])</code>	Function that prints the confusion matrix metric for a given set of image clustering results and the associated images.

1.2.11 HIRISE_api.models.utils**Functions**

<code>elbow_curve(encoded_samples[, max_values, ...])</code>
<code>generate_precision_dataframe(folder_path, ...)</code>
<code>translate_labels(translation_list, model_results)</code>

1.3 HIRISE_api.preprocessing**Modules**

<code><i>HIRISE_api.preprocessing.Data_Preparation</i></code>
<code><i>HIRISE_api.preprocessing.Dimension_Reduction</i></code>
<code><i>HIRISE_api.preprocessing.Encoding</i></code>
<code><i>HIRISE_api.preprocessing.Image_Loader</i></code>
<code><i>HIRISE_api.preprocessing.utils</i></code>

1.3.1 HIRISE_api.preprocessing.Data_Preparation

Classes

DataPreparation()	Class that allows for data preparation as part of the preprocessing of the hirise images.
-------------------	---

1.3.2 HIRISE_api.preprocessing.Dimension_Reduction

Functions

PCA_analysis(encoded_samples, labels[, ...])	The PCA analysis allows the user to understand different aspects of the PCA method.
TSNE_analysis(encoded_samples, labels[, ...])	The T-SNE analysis allows the user to understand different aspects of the TSNE method. The user can choose to plot the 2D and 3D visualisations.
UMAP_analysis(encoded_samples[, components, ...])	The UMAP analysis allows the user to understand different aspects of the UMAP method.

1.3.3 HIRISE_api.preprocessing.Encoding

Functions

create_encoded_samples_dataframe(folder_path)	Function that uses the autoencoder to encode the samples and return an encoded samples dataframe to the user based on latent dimensions input by the user
plot_autoencoder_results(encoder, decoder, ...)	Function that plots the original and reconstructed images from the autoencoder results
test_batches(encoder, decoder, device, ...)	Function that is used to test the Convolutional Autoencoder and return the mean loss, averaged over all input batches.
train_CAE(encoder, decoder, device, ...)	Function that is used to train using a single batch input into the autoencoder.
train_batches(encoder, decoder, device, ...)	Function that is used to train the Convolutional Autoencoder and return the mean loss, averaged over all input batches.
transfer_learning_encoding(folder_path[, ...])	The Transfer learning function takes in the folder path of the images to

Classes

CAEDecoder(encoded_space_dim, fc2_input_dim)	Class that supports functions needed to define the architecture and forward functions of the decoder in the Convolutional Autoencoder
CAEEncoder(encoded_space_dim, fc2_input_dim)	Class that supports functions needed to define the architecture and forward functions of the encoder in the Convolutional Autoencoder

1.3.4 HIRISE_api.preprocessing.Image_Loader**Functions**

generate_dataloaders(folder_path[, transform])	Function that generates the dataloaders for a HIRISE dataset, given folder path specified by the user
generate_dataset(folder_path[, transform])	Function that generated the HIRISE Dataset given a folderpath of HIRISE Images
initialize_encoder_decoder([latent_dimensions])	Fuction that initialized the encoder and decoder depeining on the latent
show_classes(folder_path[, transform, ...])	Function that shows all classes defined by the user though the Image Folders using the Image Folder dataset
show_encoder_decoder_image_sizes(folder_path)	Function that returns the input and output image sizes of images that have been through the autoencoding process

Classes

HiriseImageDataset(path_to_images[, transform])	Hirise Image Dataset Class that initialize the pytorch ImageLoader Dataset with the folder images to return and image and associated folder name(label)
---	--

1.3.5 HIRISE_api.preprocessing.utils**Functions**

create_image_list(file_path[, transform])	Function that creates a list of all the images in the specified folder in a PIL format
display_all_images(file_path, ...[, ...])	Function to display all the image in the folder in a flat rasterfied format.
display_image_distributions(image_file_path, ...)	Function to display all the image in the folder as images on a distributed map using TSNE,UMAP or PCA as the preprocessing function
normalize_results(encoded_samples)	Function that is used to normalize the values of the encoded samples.
read_encoded_csv(file_path[, autoencoder])	
show_cluster_images(image_file_path, ...[, ...])	Prints the image in a specified cluster, in the form of a grid with rows and columns specified by the user

1.4 HIRISE_api.tests

Modules

HIRISE_api.tests.test_hirise

HIRISE_api.tests.test_models

HIRISE_api.tests.test_preprocessing

1.4.1 HIRISE_api.tests.test_hirise

Functions

test_database_exists()

test_filter_center_latlon()

test_get_all_parameters()

test_get_images()

test_get_individual_parameters()

1.4.2 HIRISE_api.tests.test_models

Functions

test_clustering_results()

test_metrics_function()	Test if metrics function returns all 9 metrics output as expected
-------------------------	---

test_translate_labels()	Test if label translation function for classification metrics translated user defined labels as expected
-------------------------	--

1.4.3 HIRISE_api.tests.test_preprocessing

Functions

test_autoencoder()	Test if output size of the auto-encoded image is as expected
--------------------	--

test_pca_dimension_reduction()	Test if output of PCA Analysis is as expected
--------------------------------	---

test_tsne_dimension_reduction()	Test if output of TSNE Analysis is as expected
---------------------------------	--

test_umap_dimension_reduction()	Test if output of UMAP analysis is as expected
---------------------------------	--

PYTHON MODULE INDEX

h

- `HIRISE_api`, 3
- `HIRISE_api.hirise`, 3
 - `HIRISE_api.hirise.Hirise_Image`, 3
 - `HIRISE_api.hirise.Image_Client`, 4
 - `HIRISE_api.hirise.utils`, 4
- `HIRISE_api.models`, 4
 - `HIRISE_api.models.Affinity_Propagation`, 5
 - `HIRISE_api.models.Agglomerative_Clustering`, 5
 - `HIRISE_api.models.BIRCH`, 6
 - `HIRISE_api.models.DBSCAN`, 6
 - `HIRISE_api.models.Ensemble_Models`, 6
 - `HIRISE_api.models.HDBSCAN`, 7
 - `HIRISE_api.models.KMeans`, 7
 - `HIRISE_api.models.Mean_Shift`, 7
 - `HIRISE_api.models.metrics`, 7
 - `HIRISE_api.models.OPTICS`, 7
 - `HIRISE_api.models.utils`, 8
- `HIRISE_api.preprocessing`, 8
 - `HIRISE_api.preprocessing.Data_Preparation`, 9
 - `HIRISE_api.preprocessing.Dimension_Reduction`, 9
 - `HIRISE_api.preprocessing.Encoding`, 9
 - `HIRISE_api.preprocessing.Image_Loader`, 10
 - `HIRISE_api.preprocessing.utils`, 10
- `HIRISE_api.tests`, 11
 - `HIRISE_api.tests.test_hirise`, 11
 - `HIRISE_api.tests.test_models`, 11
 - `HIRISE_api.tests.test_preprocessing`, 11