
imagediffer Documentation

Release 0.1

Jan Slifka

January 31, 2017

1	Submodules	3
1.1	imagediffer.app package	3
1.2	imagediffer.core package	3
2	Indices and tables	7
	Python Module Index	9

Image Differ is a tool for image comparison. It loads 2 images and generates the diff image that shows where the images are different. It also calculates:

- Percentage of pixels where the images differ (with adjustable threshold for difference)
- Mean Square Error for the images
- Structural Similarity Index

It can also calculate comparison stats for individual color channels or ignore colors.

All the functions for image comparison and manipulations are separated in `imagediffer.core` module and can be reused without the application.

Submodules

1.1 imagediffer.app package

This package is for the application itself. It uses `imagediffer.core` under the hood and provides clear user interface to use the library.

1.1.1 imagediffer.app.app module

class `imagediffer.app.app.App`
Bases: `object`

PyQt5 application for using core library.

To run the application, simply create an instance and call `run` method.

```
from imagediffer.app.app import App

App().run()
```

run()
Run the application.

1.2 imagediffer.core package

This package contains tools for loading and saving images, various color manipulations and comparing images.

1.2.1 imagediffer.core.differ module

Differ module contains functions for comparing images and creating diff images

`imagediffer.core.differ.calculate_mse(image1, image2)`

Calculate **mean squared error** for given images. The higher the value of MSE is, the more different the images are.

Parameters

- **image1** – Numpy image array
- **image2** – Numpy image array

Returns Float number representing mean squared error

`imagediffer.core.differ.calculate_ssim(image1, image2)`

Calculate [structural similarity index](#) for given images. If the value is 1.0 the images are same. The lower the value is, the more different the images are.

Parameters

- **image1** – Numpy image array
- **image2** – Numpy image array

Returns Float number from -1.0 to 1.0 representing SSIM

`imagediffer.core.differ.chebyshev_distance(image1, image2)`

Calculate [chebyshev distance](#) for each pixel in image1 and image2. Images must have same dimension.

Parameters

- **image1** – Numpy image array
- **image2** – Numpy image array

Returns Array of chebyshev distances

`imagediffer.core.differ.diff(image1, image2, compare_colors_method, tolerance=0, diff_color=(1.0, 0, 1.0, 1.0))`

Create a diff image of image1 and image2. Pixels are considered different if the distance of colors is greater than tolerance using given compare_colors_method. The result image is created by blending image1 and image2 together and replacing different pixels with the diff_color.

Parameters

- **image1** – Numpy image array
- **image2** – Numpy image array
- **compare_colors_method** – Method that takes two numpy image arrays and return array of color distances in range from 0.0 to 1.0. You can use `euclidean_distance`, `chebyshev_distance` or implement your own method.
- **tolerance** – Defines the color distance that is acceptable and colors are considered the same
- **diff_color** – RGBA color that should be used for different pixels

Returns

Tuple containing:

- **diff_image** Numpy image array
- **diff_pctg** Percentage of pixels where the color distance exceeded the acceptable tolerance

`imagediffer.core.differ.euclidean_distance(image1, image2)`

Calculate [euclidean distance](#) for each pixel in image1 and image2. Images must have same dimension.

Parameters

- **image1** – Numpy image array
- **image2** – Numpy image array

Returns Array of euclidean distances

1.2.2 imagediffer.core.loader module

Loader module contains functions for loading and saving images.

`imagediffer.core.loader.load_image_from_file(file)`

Load image as a numpy array from given file.

Parameters `file` – Path to the image file

Returns Numpy image array with color and alpha channels values from 0.0 to 1.0

`imagediffer.core.loader.load_image_from_url(url)`

Load image as a numpy array from given URL.

Parameters `url` – Image URL

Returns Numpy image array with color and alpha channels values from 0.0 to 1.0

`imagediffer.core.loader.save_image(image, path)`

Save given image to the PNG file. If the file extension is not part of the `path` or is missing, it will be autocompleted or replaced.

Parameters

- **image** – Numpy image array
- **path** – Path to the image file.

1.2.3 imagediffer.core.utils module

Utils module contains various functions for manipulating images.

`imagediffer.core.utils.extract_colors(image)`

Extract individual color channels and alpha channel from numpy image array.

Parameters `image` – Numpy image array

Returns Array of width × height numpy arrays for each color and alpha channels

`imagediffer.core.utils.norm_color(image)`

Normalize colors to be in range from 0.0 to 1.0 instead of from 0 to 255.

Parameters `image` – Numpy image array with colors from 0 to 255

Returns Numpy image array with colors normalized to floats

`imagediffer.core.utils.to_blue(source)`

Convert source image to image using blue channel for all color channels.

Parameters `source` – Numpy image array

Returns Numpy image array

`imagediffer.core.utils.to_grayscale(source)`

Convert image to grayscale using [PAL/NTSC conversion](#).

Parameters `source` – Numpy image array

Returns Numpy image array

`imagediffer.core.utils.to_green(source)`

Convert source image to image using green channel for all color channels.

Parameters `source` – Numpy image array

Returns Numpy image array

`imagediffer.core.utils.to_red(source)`

Convert source image to image using red channel for all color channels.

Parameters **source** – Numpy image array

Returns Numpy image array

Indices and tables

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

i

`imagediffer.app.app`, 3
`imagediffer.core.differ`, 3
`imagediffer.core.loader`, 5
`imagediffer.core.utils`, 5

A

App (class in imagediffer.app.app), 3

C

calculate_mse() (in module imagediffer.core.differ), 3
calculate_ssim() (in module imagediffer.core.differ), 4
chebyshev_distance() (in module imagediffer.core.differ),
4

D

diff() (in module imagediffer.core.differ), 4

E

euclidean_distance() (in module imagediffer.core.differ),
4
extract_colors() (in module imagediffer.core.utils), 5

I

imagediffer.app.app (module), 3
imagediffer.core.differ (module), 3
imagediffer.core.loader (module), 5
imagediffer.core.utils (module), 5

L

load_image_from_file() (in module imagediffer.core.loader), 5
load_image_from_url() (in module imagediffer.core.loader), 5

N

norm_color() (in module imagediffer.core.utils), 5

R

run() (imagediffer.app.app.App method), 3

S

save_image() (in module imagediffer.core.loader), 5

T

to_blue() (in module imagediffer.core.utils), 5

to_grayscale() (in module imagediffer.core.utils), 5
to_green() (in module imagediffer.core.utils), 5
to_red() (in module imagediffer.core.utils), 6