

The distance of 2 images

If we represent the images as vectors of numbers [\[wiki/courses/be5b33kui/semtasks/05_ml1/image\]](/wiki/courses/be5b33kui/semtasks/05_ml1/image), we need to compute the distance of two vectors. (Which does not sound that weird as "computing a distance of two images", right?)

Using functions from `numpy`, we can compute the (Euclidean) distance in the following two ways (see `d1` and `d2`):

```
from PIL import Image
import numpy as np

im1 = np.array(Image.open('train_data/img_1112.png')).astype(int).flatten()
im2 = np.array(Image.open('train_data/img_1113.png')).astype(int).flatten()

diff = im1 - im2
d1 = np.sqrt(np.sum(np.square(diff)))
d2 = np.linalg.norm(diff)
```

Variables `d1` and `d2` should contain the same result.

Our images are grayscale with shades of gray ranging from 0 to 255 (represented by `numpy.uint8` datatype). If we used them with their native datatype `uint8`, underflow errors may happen when computing the difference `im1 - im2` (`uint` cannot represent negative numbers), and overflow errors may happen when computing the squares of differences. That's why it is better to transform all numbers to `int` which can represent both large and negative numbers (it is important to use `.astype(int)` after reading in the image.)

[courses/be5b33kui/semtasks/05_ml1/distance.txt](#) · Last modified: 2024/02/18 20:07 by xposik

Copyright © 2024 CTU in Prague | Operated by [IT Center of Faculty of Electrical Engineering](#) |
Bug reports and suggestions [Helpdesk CTU](#)

