Reading .png image into a numerical vector

To be able to work with images reasonably (learn from them and classify them), we shall transform each image into a numerical vector. Each number in the vector will correspond to the intensity of certain pixel in the image.

TLDR

With the use of Pillow [https://pillow.readthedocs.io/en/stable/] and Numpy [https://numpy.org/] librari
the whole thing is easy:

```
from PIL import Image
import numpy as np

impath = 'train_data/img_1112.png'
image_vector = np.array(Image.open(impath)).flatten()
```

Explanation

If you are interested in what the above code actually does, let's do it step by step:

```
from PIL import Image
import numpy as np

impath = 'train_data/img_1112.png'
im = Image.open(impath)
print(type(im))
im2d = np.array(im)
print(type(im2d), im2d.shape)
im1d = im2d.flatten()
print(type(im1d), im1d.shape)
print(im1d)
```

After running the above code, you should get output similar to this:

```
<class 'PIL.PngImagePlugin.PngImageFile'>
<class 'numpy.ndarray'> (10, 10)
<class 'numpy.ndarray'> (100,)
[230 202 168 139 124 129 147 180 206 221 227 181 126  84  51  49  80 145
 206 241 227 169 102  50  18   7  27  96 183 233 212 156  92  40  21  15
```

```
 25   76 164 224 196 136   78   47   33   32   39   73 141 203 175 118   64   41
 42   45   48   66 122 184 151   87   39   24   35   34   28   35   84 156 139   76
 36   26   35   37   30   38   69 130 152 106   87   99 116 114   95   77   82 122
198 180 186 217 237 226 199 168 152 172]
```

Explanation:

- Function `Image.open()` reads in the image and returns an instance of class `PngImageFile`.
- Function `np.array()` converts the image into an instance of class `numpy.ndarray`, i.e., it creates a 2D array of size 10×10.
- Method `ndarray.flatten()` turns 10×10 matrix into a vector of length 100; its contents are at the end of the output.

courses/be5b33kui/semtasks/05_ml1/image.txt · Last modified: 2024/02/18 20:07 by xposik