redpil Documentation

Release 0.0.5+0.gd616002.dirty

Mark Harfouche

Jul 14, 2019

Contents

CHAPTER 1

redpil

Join the wonderland of python, and decode all your images in a numpy compatible way.

Pillow is a great library for image manipulation. However, many operations fall outside what Pillow can do. As such, many scientific applications require the image to be available as a numpy array. However, Pillow's memory system is largely incompatible with numpy's. imageio has created an efficient bridge between numpy and Pillow (see benchmarks below). Unfortunately, Pillow's multitude of options remain confusing it is challenging to understand how they all operate together. Furthermore, the code base is rather old, written in C, meaning that it is difficult to extend the functionality of existing decoders.

For large images, having to understand the details of both Pillow and numpy is a serious bottleneck. The goal of the library it to read and write images in a manner natural to numpy users. Images are presented as the values they hold (not indices in a color table) allowing for direct data analysis.

As much as possible, the library is written in python allowing for new decoding algorithms to be played around with.

1.1 Bitmap images

Generally, this library will not load memory in a C-contiguous array. Rather the memory order will mostly match what was saved on disk.

Bitmap images will be stored in an order similar to how they arranged in RAM.

1.2 Supported file formats

Reading BMP is almost fully supported. Writing is still limited.

- BMP: 1, 4, or 8bit per pixel. Wikipedia

1.3 Future file formats

- BMP: more coverage
- JPEG, JPEG2000
- GIF
- PNG
- SVG
- TIFF

1.4 Benchmarks

I don't have a fancy benchmarking service like scikit-image or dask has, but here are the benchmarks results compared to a PIL backend. This is running on my SSD, a Samsung 960 Pro which claims it can write at 1.8GB/s. This is pretty close to what `redpil` achieves.

1.4.1 8 bit BMP grayscale images

Saving images:

	mode		
shape	redpil	pillow	imageio
(128, 128)	93.4±1µs	254±30µs	369±20µs
(1024, 1024)	720±30µs	936±50µs	1.60±0.3ms
(2048, 4096)	5.25±0.7ms	5.20±0.1ms	10.4±2ms
(32768, 32768)	480±10ms	489±5ms	1.34±0.09s

Reading image

	mode		
shape	redpil	pillow	imageio
(128, 128)	131±5µs	293±10µs	130±2µs
(1024, 1024)	194±10µs	1.03±0.1ms	192±5µs
(2048, 4096)	1.69±0.05ms	8.55±1ms	1.67±0.03ms
(32768, 32768)	350±3ms	230±5µs	354±10ms

Note, Pillow refuses to read the 1GB image because it thinks it is a fork bomb.

Patched up imageio

As it can be seen, the team at imageio/scikit-image are much better at reading the pillow documentation and understanding how to use it effectively. Their reading speeds actually match the reading speeds of redpil, even though they use pillow as a backend. They even handle what pillow thinks is a forkbomb.

Through writing this module, two bugs were found in imageio that affect the speed of saving images imageio PR #398, and how images were being read imageio PR #399

With PR 398, the saving speed of imageio+pillow now matches that of redpil. Note I'm always using the computer when running benchmarks, so take the exact numbers with a grain of salt.

Saving

	mode		
shape	redpil	pillow	imageio
(128, 128)	98.3±4μs	245±7μs	350±4μs
(1024, 1024)	714±20μs	921±30μs	997±20μs
(2048, 4096)	4.83±0.3ms	5.30±0.4ms	5.26±0.2ms
(32768, 32768)	520±40ms	516±30ms	489±9ms

Reading

	mode		
shape	redpil	pillow	imageio
(128, 128)	129±0.7μs	284±2μs	129±0.7μs
(1024, 1024)	191±2μs	1.12±0.1ms	190±0.9μs
(2048, 4096)	1.62±0.03ms	8.88±1ms	1.63±0.02ms
(32768, 32768)	357±9ms	223±4μs	361±8ms

CHAPTER 2

Contents

2.1 Installation

2.1.1 Stable release

To install redpil, run this command in your terminal:

```
$ pip install redpil
```

This is the preferred method to install redpil, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

2.1.2 From sources

The sources for redpil can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/hmaarrfk/redpil
```

Or download the tarball:

```
$ curl  -OL https://github.com/hmaarrfk/redpil/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

2.2 Usage

redpil is still heavily experimental. To use it in your project, make sure you understand the limitations of each plugin. Then import the plugin you wish to use with:

```
from redpil.bmp import imread, imwrite
```

2.3 API

In the future, when the API is more established, we will support a few functions, that provided the filename and the extension, will attempt to infer which plugin to call. For now, these functions do nothing.

Top-level package for redpil.

redpil.**imread**()

redpil.**imwrite**()

2.3.1 redpil.bmp module

2.4 History

2.4.1 0.0.1 (2018-09-22)

- First release on PyPI.

Python Module Index

r

Index

I

R