The Journal of Open Source Software

The SHINIER the Better: An Adaptation of the SHINE Toolbox on Python

Mathias Salvas-Hébert [1*¶], **Nicolas Dupuis-Roy** [2*], **Catherine Landry** [1], **Ian Charest** [1], and **Frédéric Gosselin** [1]

[1] Département de Psychologie, Université de Montréal, CP 6128, succ. Centre-ville, Montréal, QC, H3C 3J7, CANADA [2] Elephant Scientific Consulting, Canada ¶ Corresponding author * These authors contributed equally.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

DRAFT

Summary

The SHINIER (Spectrum, Histogram, and Intensity Normalization, Equalization, and Refinement) toolbox, written in Python, is an open-source package that replicates and extends the functionality of the original SHINE toolbox (Willenbockel et al., 2010), written in MATLAB. Like SHINE, it includes functions for normalizing and scaling mean luminance and contrast, for specifying either the full Fourier amplitude spectrum or its rotational average, and for exact histogram specification. In addition, SHINIER supports color images, better memory management, implements image dithering algorithms for improving pixel depth, and offers improved exact histogram equalization methods, among other enhancements. The original SHINE toolbox (Willenbockel et al., 2010), written in MATLAB, has been cited more than 1,350 times according to Google Scholar—an average of about 100 citations per year—clearly indicating its popularity and usefulness in vision science research. SHINIER aims to provide the same benefits to the research community, while expanding accessibility and functionality.

Statement of need

When conducting experiments with humans, animals, or machines, the choice of stimuli is critical. We usually intend participants to rely on invariant features that genuinely support recognition in real life. However, experimental image sets—necessarily small subsets of the virtually infinite possible exemplars—often contain accidental features that can be exploited instead. For example, in a dog–cat categorization task, participants might succeed not because they attend to diagnostic shape or texture cues, but because the dog images (often taken outdoors in bright sunlight) have luminance histograms with higher means and greater variance than cat images (typically photographed indoors under dim lighting). These luminance differences are artifacts of illumination, not reliable distinguishing properties of dogs versus cats in the real world. One way to avoid such confounds is to use artificially generated stimuli with fully controlled low-level properties. Another is to rely on very large naturalistic image databases, such as the Natural Scenes Dataset (Allen et al., 2022), where idiosyncratic correlations tend to average out. When working with natural images and such large-scale resources are unavailable—or impractical due to time constraints with human or animal participants—normalizing and adjusting low-level image properties becomes essential.

The SHINE (Spectrum, Histogram, and Intensity Normalization and Equalization) toolbox has become the standard for this type of image normalization (Willenbockel et al., 2010). Since its release, it has been cited over 1,350 times according to Google Scholar, highlighting its popularity and utility in vision research. Written in MATLAB, SHINE addressed the dominant programming environment of its time for image processing, experimental control, and data analysis in vision science. With Python now the most widely used programming language in the scientific community (Srinath, 2017), a Python version of SHINE was needed. This project evolved into a more versatile toolbox: SHINIER. By sharing it with the broader research community, we hope it will be as useful as SHINE has been over the past 15 years.

SHINIER replicates the core functionalities of SHINE while introducing new features. It supports all equalization techniques from SHINE, applied individually or in combination. Users can directly specify full Fourier amplitude spectra and rotational averages, control luminance histograms, normalize and scale luminance distributions without altering their shape, and separately equalize foreground and background luminance. Functions for plotting Fourier spectra and average energy across spatial frequencies are included, as in the original toolbox. In addition, it implements the exact histogram specification algorithm (Coltuc et al., 2006), supports color images, dynamically manages image storage, and implements the noisy-bit image dithering technique to improve pixel depth (Allard & Faubert, 2008).

Availability and usage

SHINIER is available as a pip-install package using `pip install shinier` with its source code hosted on Charestlab's GitHub. Exhaustive documentation is available in the README.

Acknowledgements

The author would like to thank the original contributors of the SHINE toolbox.

References

Allard, R., & Faubert, J. (2008). The noisy-bit method for digital displays: Converting a 256 luminance resolution into a continuous resolution. *Behavior Research Methods*, *40*(3), 735–743.

Allen, E. J., St-Yves, G., Wu, Y., Breedlove, J. L., Prince, J. S., Dowdle, L. T., Nau, M., Caron, B., Pestilli, F., Charest, I., & others. (2022). A massive 7T fMRI dataset to bridge cognitive neuroscience and artificial intelligence. *Nature Neuroscience*, *25*(1), 116–126.

Coltuc, D., Bolon, P., & Chassery, J.-M. (2006). Exact histogram specification. *IEEE Transactions on Image Processing*, *15*(5), 1143–1152.

Srinath, K. (2017). Python–the fastest growing programming language. *International Research Journal of Engineering and Technology*, *4*(12), 354–357.

Willenbockel, V., Sadr, J., Fiset, D., Horne, G. O., Gosselin, F., & Tanaka, J. W. (2010). Controlling low-level image properties: The SHINE toolbox. *Behavior Research Methods*, *42*(3), 671–684.

DRAFT