Scikit-image: Image processing for dummies

Karan Thakker - May 18th 2025

1. Title: Scikit-image: She sci on my kit till I image (CHANGE TITLE)
2. Scikit-image or “skimage”, is an open-source image processing package designed for use in research, education, and industrial applications. Designed for python, the library is written primarily in Python, with some parts written in C, and Cython. It is made to take advantage of open source development, in order to keep it actively updated.
3. I chose this package because I have been working with it for my research, in the Ultra-Fast Optics lab with Dr. Kim. It helped me speed up my research by allowing me to auto-crop onto the region of interest.
4. The skimage package was originally created in August of 2009, created as a combination of various image processing packages, like numpy.ndimage, and matplotlib. Packages like OpenCV, and Pillow, can accomplish a lot of similar tasks to skimage. The current latest version of skimage is version 0.25.2, but the team behind it have been preparing for a 1.0 release.
5. Originally created by Stefan Van der Walt, the package is still regularly maintained and updated by many collaborators, including the original created. The latest version was released on the 18th of February 2025, with new releases coming every few months.
6. The install was very simple and took less than one minute. This is because it makes large use of libraries like matplotlib, and numpy which tend to be pre-installed by most users.
7. The standard methods like pip install, and conda can be used to install the package, as well as allowing users to “build package from source” by forking it off of their github. This last option is primarily for people wanting to contribute as it allows them greater flexibility with the code. Skimage also provides sample datasets, which require further installation of certain dependencies, to fully utilise, if installing from command line.

   On jupyterlab notebook:

   ```
   !pip install scikit.image
   ```

   On the command line:

   ```
   python -m pip install -U scikit-image
   python -m pip install -U scikit-image[data]
   ```

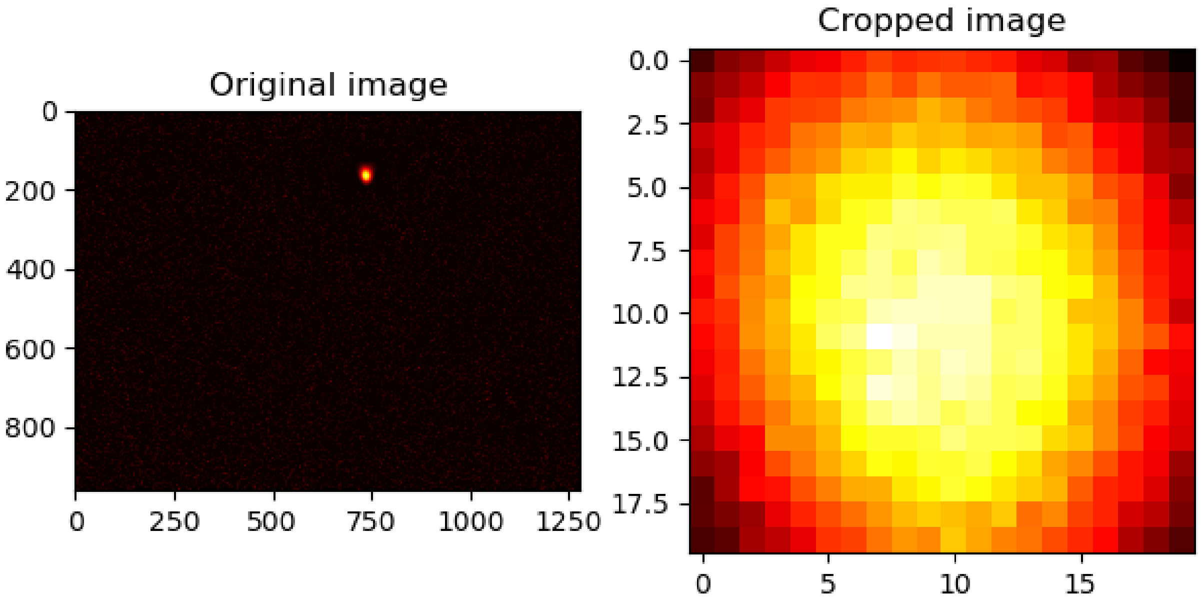
8. The source code is easily available on their github:

   https://github.com/scikit-image/scikit-image

9. Packages like trackpy, which is a particle tracking toolkit for python, directly depend on skimage for object detection, while others like napari, an interactive viewer for

multidimensional images, while not directly dependent, do rely on skimage for plugins for image processing and analysis

10. Skimage cannot be run directly using a command line, but requires a jupyter notebook, or a python script.
11. The accompanying notebook shows how I have used skimage to automatically detect and crop into a region of interest around a laser beam. While very simple, it allowed me to crop very quickly since earlier, it required a lot of manual trial and error to ensure that the region has very little of the background.
12. The package does produce figures, but relies on matplotlib, and other similar packages to do so under the hood.
13. The following is a set of images produced by the notebook which shows the original image and the cropped version, done mainly using automatic detection, and very small manual correction.



14. The package is a mix of Python, and C, or Cython, and thus requires C code
15. The package takes images in various formats as inputs. It even allows the creation of images from scratch using its data.binary_blob() function which generates “synthetic binary image with several rounded blob-like objects.” Furthermore it has several sample datasets to test and experiment with.
16. The package outputs mainly numpy arrays, which can also be displayed as images using imshow().
17. The package includes asv, which is a benchmarking suite used by packages like NumPy and SciPy. Skimage specifically uses asv continuous that runs “relevant performance measurements”
18. To ensure the proper working of the package, the asv suite can be called upon by the user to run benchmarking tests.

19. The package relies heavily on NumPy, and SciPy, and also makes use of matplotlib for image display purposes, and data representation, like histograms.
20. The package has extensive documentation, both on its github repository mentioned above, with release notes and source code, as well as on their website scikit-image.org, which includes documentation on each function and method it has. This documentation was more than enough for me to carry out my work
21. Their preferred citation is :

    Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu and the scikit-image contributors. scikit-image: Image processing in Python. PeerJ 2:e453 (2014) https://doi.org/10.7717/peerj.453

22. References:
    a. skimage
    b. Tutoiralspoint
    c. trackpy
    d. Napari
23. This paper about image generation using neural networks:

    https://arxiv.org/pdf/2312.15289

    As well as this one about recovery of a true image from a distorted one:

    https://proceedings.scipy.org/articles/majora-212e5952-00f.pdf

    Both use scikit-image for various reasons.
    The original paper is mentioned in their preferred citation, is cited 6277 times, according to google scholars, showing that this is a very widely used package
24. This class taught me everything I needed to use this package.
25. Final Disclaimer: As mentioned earlier, I have used this package before as part of my research under Dr. Ki-yong Kim, in his Ultra-Fast Optics Lab. The code I used was written entirely by me, and the image used was produced in collaboration with another undergraduate student.