Independent study: Image and Audio Generation and Editing using Python

Phileas Dazeley Gaist,
College of the Atlantic,
Fall term 2021

- My final will be a short video presentation of a selection of ten digital media recording, editing, and generation processes, as well as a GitHub repository of examples and documentation explaining how to make use of them.

Media files == arrays of data

- Image files = 2d or 3d arrays of shape (height, width, channels)
- Audio files = n-dimensional arrays of shape of shape (channel1, channel2⋯) + a sample rate
- Video files ≈ a list of arrays of images + an audio array

N-dimensional arrays = n-dimensional arrays subdivided N-n times

Where N represents a number of dimensions superior to n.

- A 1d array can be turned into a 2d array by splitting the dimension of the array into a number of smaller arrays (one division).
- A 2d array can be turned into a 3d array by dividing the “deepest” axis of the array into a number of smaller arrays (one division)

To go from a 1d array to a 3d array requires a 2 division of the axis of the 1d array. This principle is generalisable to any number of dimensions.

What this means: the number of dimensions doesn’t really matter

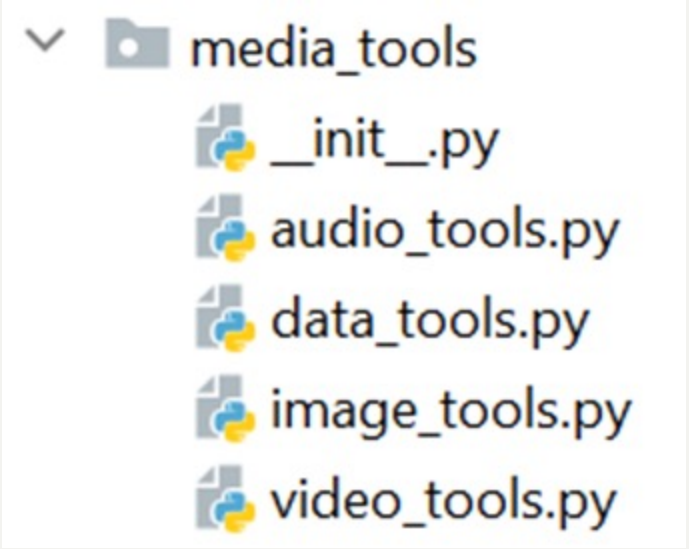
For our purposes, we can edit audio or image arrays using the same tools.

*This does not extent to video because videos are more analogous to lists: they contain arrays of different types and different sizes, which contrary to variances in the number of dimensions, makes our life a little harder.

The `media_tools` package

Data can be

- Mapped
- Transformed
- Generated
- Messed with (transformations, but wacky)



Examples:

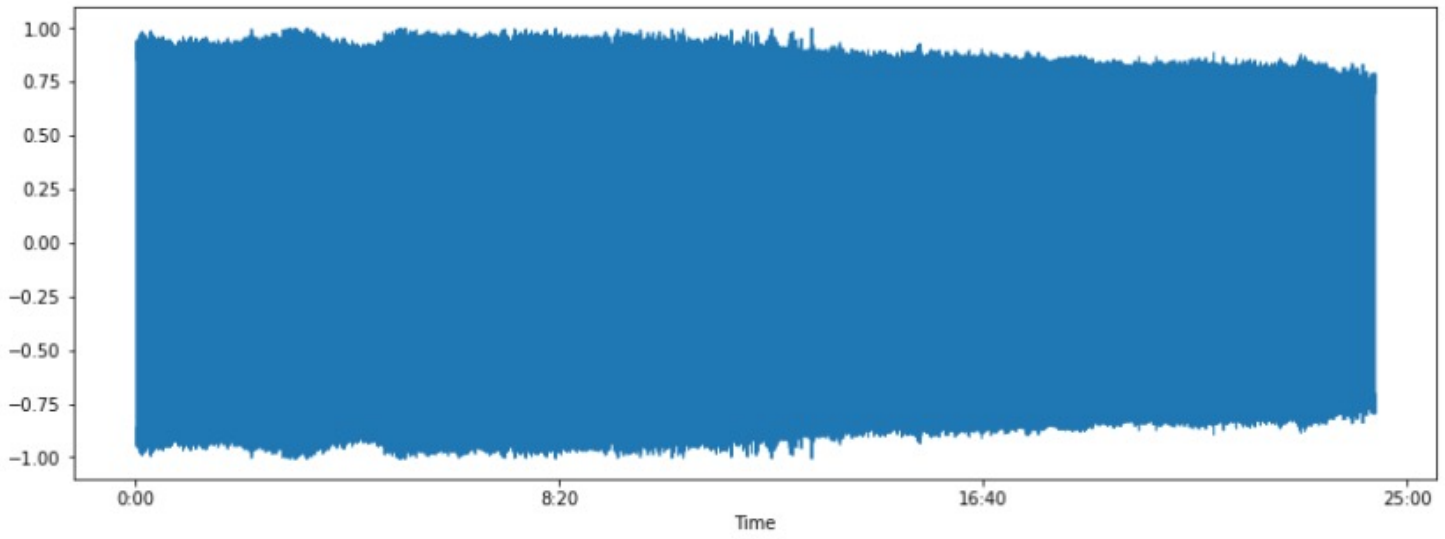
Mapping, Transforming, and Generating data to be saved as media.

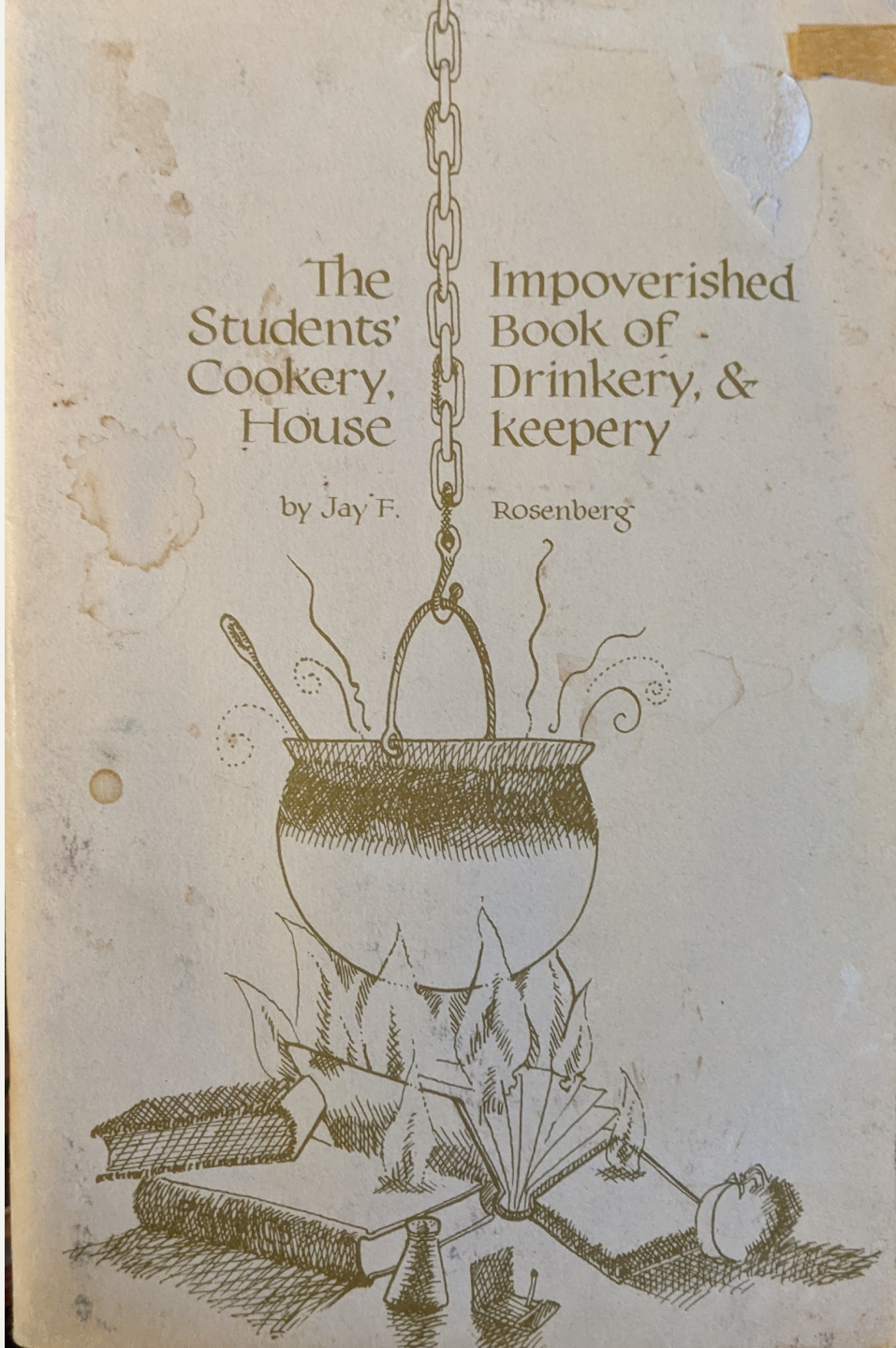
Mapping

Taking data from a media file of a certain type, casting it into the shape of a media file of another type, and reading it as another media file type.

```
# read a picture
picture = image_tools.read_image(path=sample_media+"student_guide.jpg")
image_tools.show_image(picture)

# normalise picture data array
picture_norm = data_tools.normalise_array(picture)

# flatten the image data into a time series
audio_data = data_tools.flatten_array(picture_norm)

# save data to audio file
audio_tools.save_audio(audio_data, path=sample_media+"example_saved_audio.wav"
```

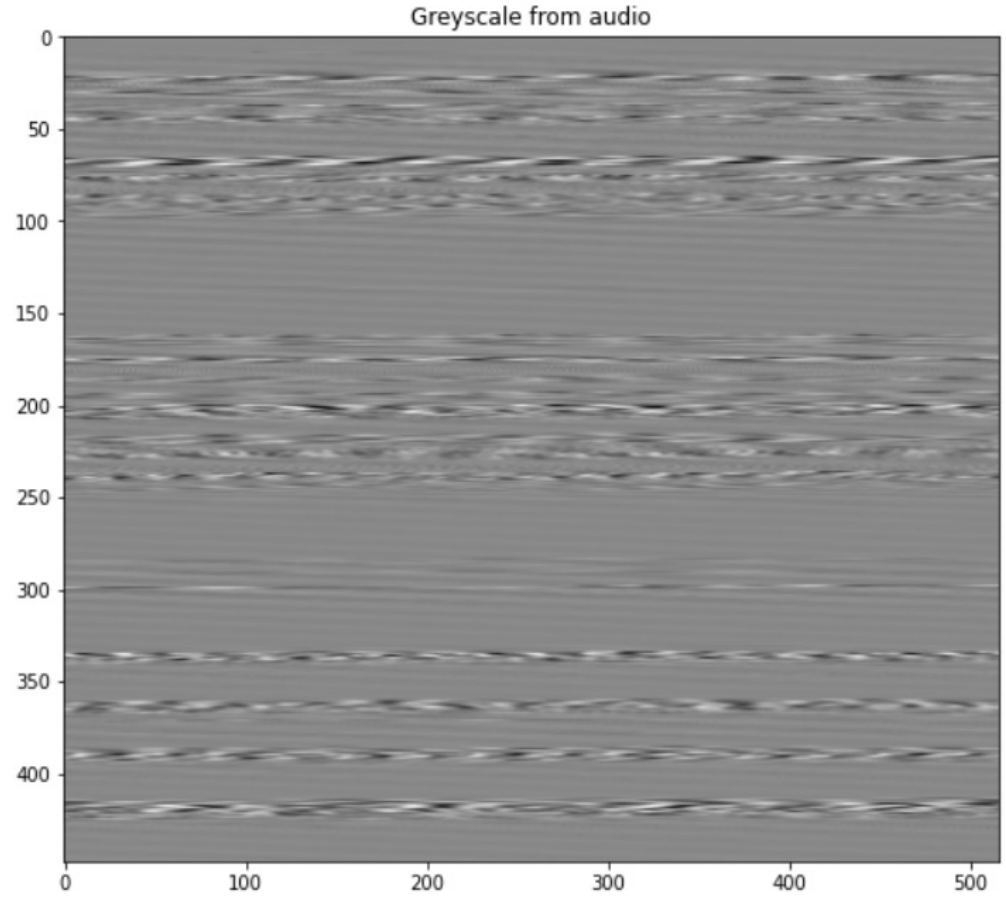
```
audio_data = audio_tools.read_audio(sample_media+"example_saved_audio.wav")
audio_tools.plot_waveform(audio_data)
audio_tools.plot_spectrogram(audio_data)
```





```
8  # load an audio file
   audio_data = audio_tools.read_audio(sample_media+"stereo_sample.wav", mono=True, return_sr=False)

   # converting a mono audio file to a greyscale image
   image_bw = image_tools.array_1d_to_grayscale(audio_data)
   image_tools.show_image(image_bw, axis=True, title="Greyscale from audio")
```

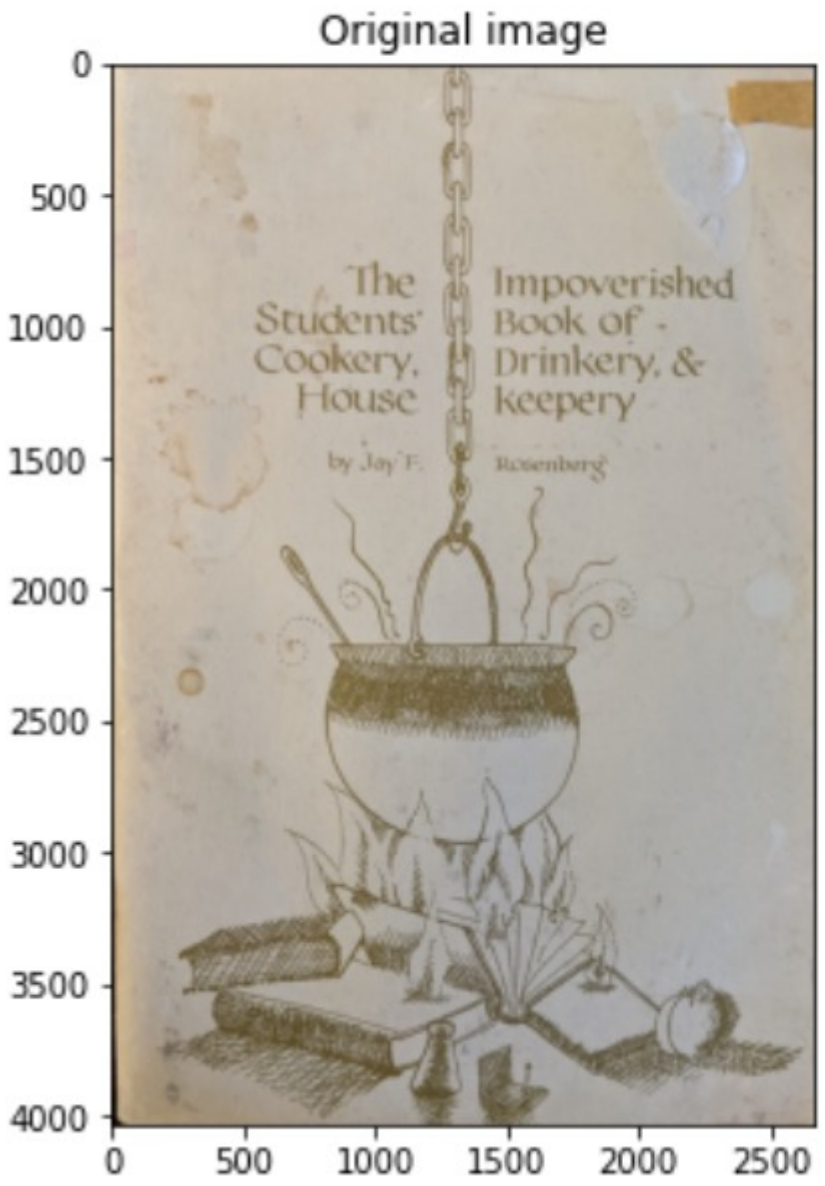
```
Output array shape = [448 516   1]
```



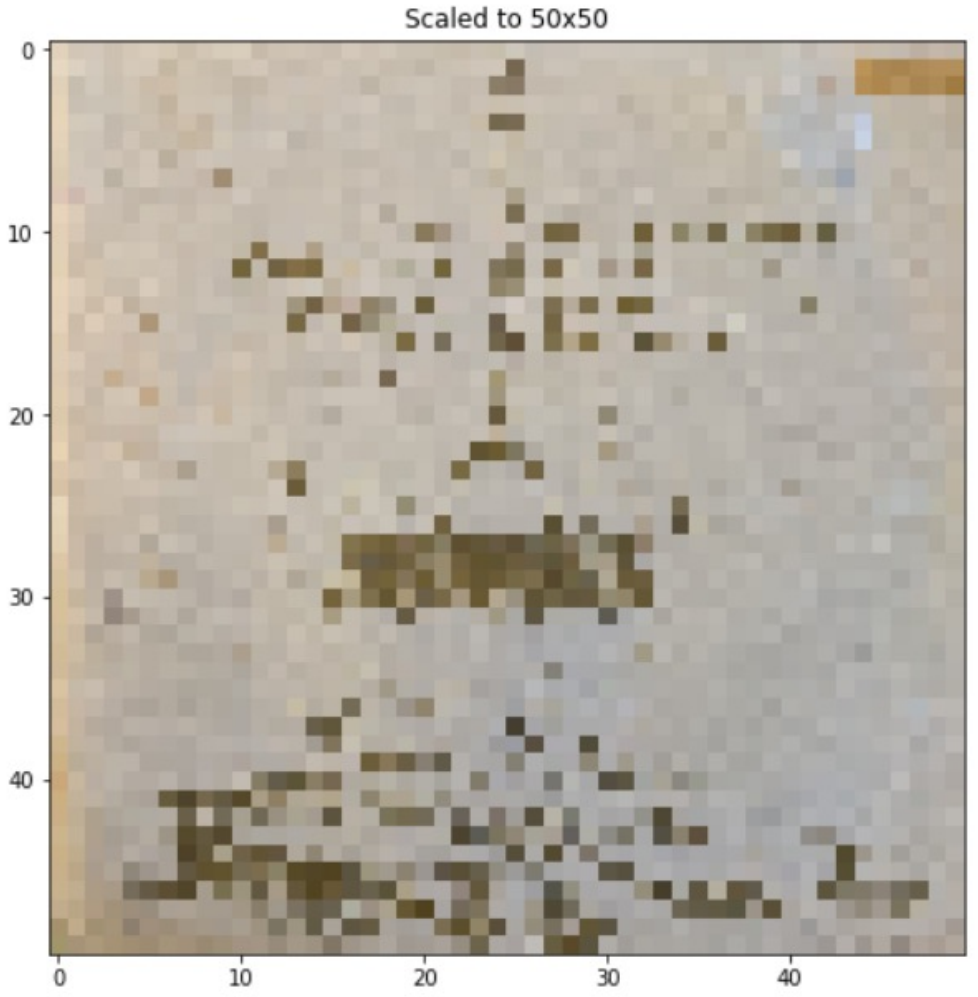
Transformations

- Scaling
- Rotation
- Translation
- Reflection
- Cropping
- Blitting

```
# show the original image
image_tools.show_image(image, axis=True, title="Original image", scale_ratio=1.5)
```




```
# array scaled to a range of 50x50 square
array_scaled_to_50x50 = data_tools.rescale_2d_array(image, (50, 50))
image_tools.show_image(array_scaled_to_50x50, axis=True, title="Scaled to 50x50")
```

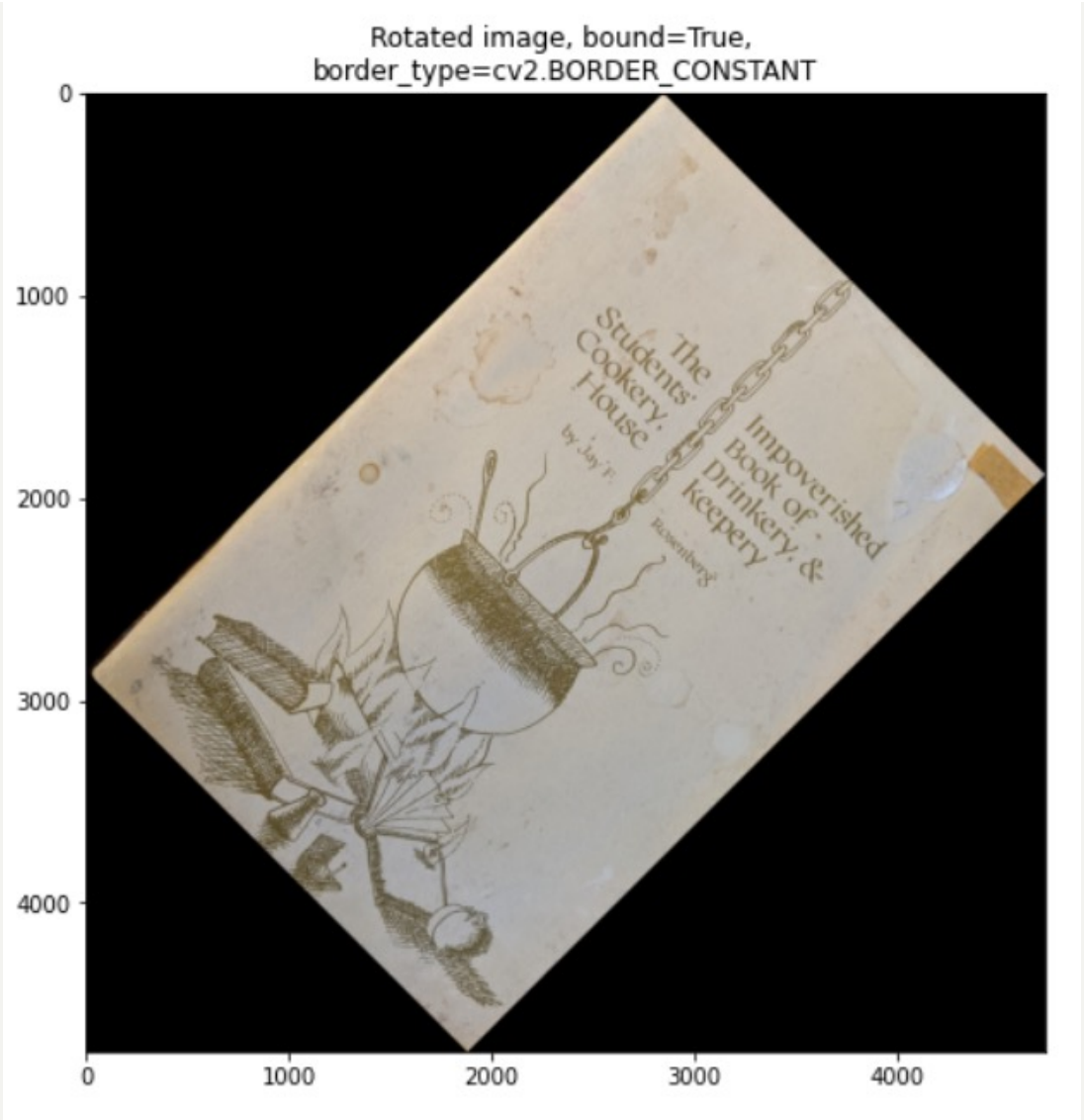


1. Rotation with preserved array size:

```
# rotate the image by 45 degrees, but crop the image to the original size:
img_rotated = image_tools.rotate_image(image, angle=45, bound=False, border_type=cv2.BORDER_CONSTANT, border_value=(0, 0, 0))
image_tools.show_image(img_rotated, axis=True, title="Rotated image, bound=False, \nborder_type=cv2.BORDER_CONSTANT")
```

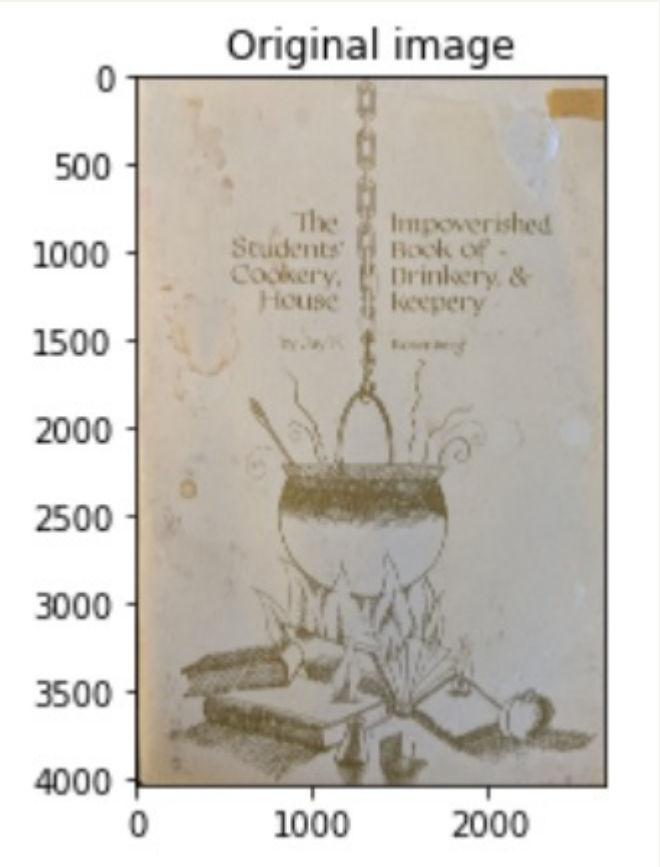
2. Rotation with preserved array data:

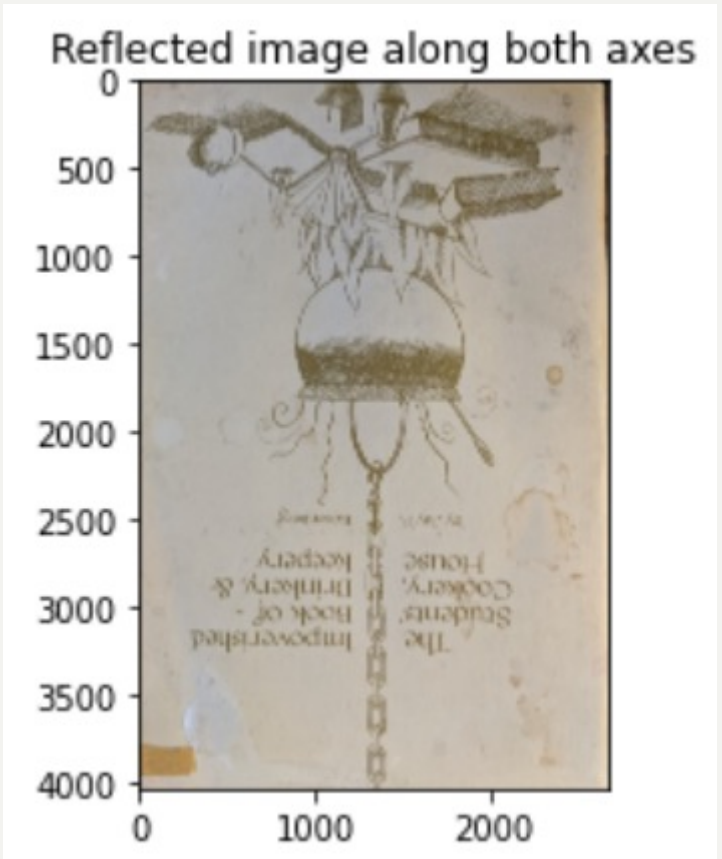
```
# rotate the image by 45 degrees without losing any part of the image:
img_rotated = image_tools.rotate_image(image, angle=45, bound=True, border_type=cv2.BORDER_CONSTANT, border_value=(0, 0, 0))
image_tools.show_image(img_rotated, axis=True, title="Rotated image, bound=True, \nborder_type=cv2.BORDER_CONSTANT")
```

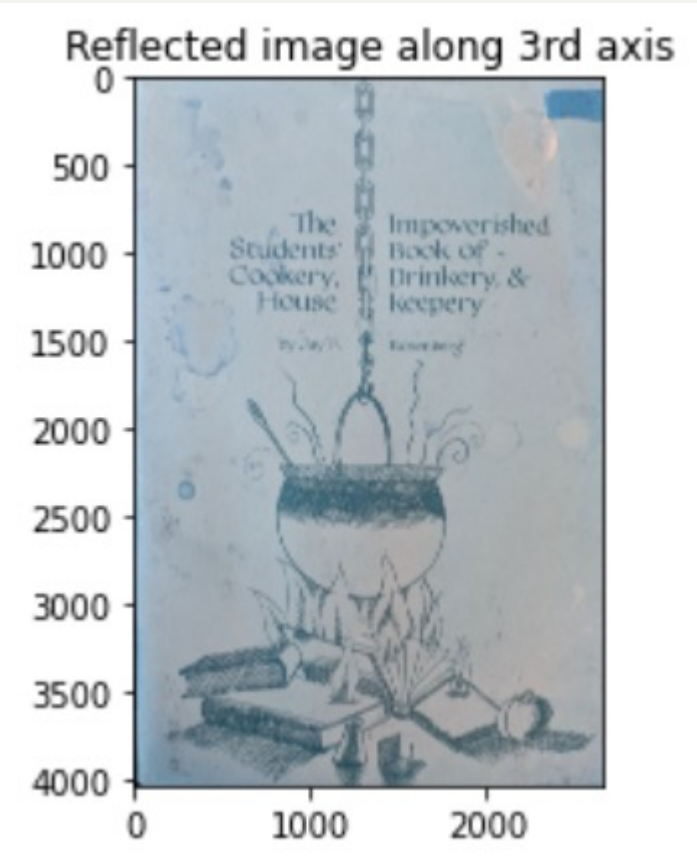


```
# show the start image
image_tools.show_image(image, axis=True, title="Original image", scale_ratio=1)
```

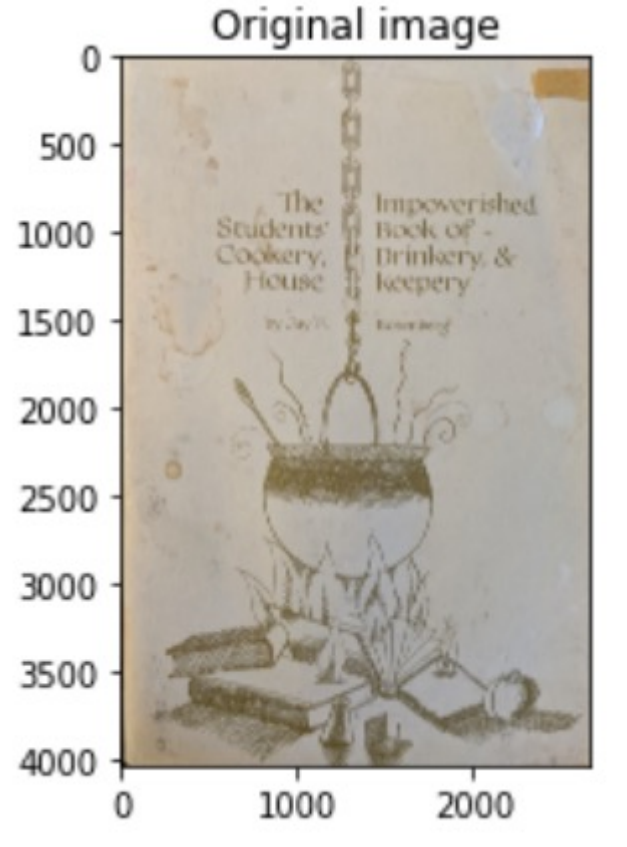
```
# get image reflection along both axes
image_reflected = data_tools.flip_2d_array(image, axes=(0, 1))
image_tools.show_image(image_reflected, axis=True, title="Reflected image along both axes", scale_ratio=1)
```

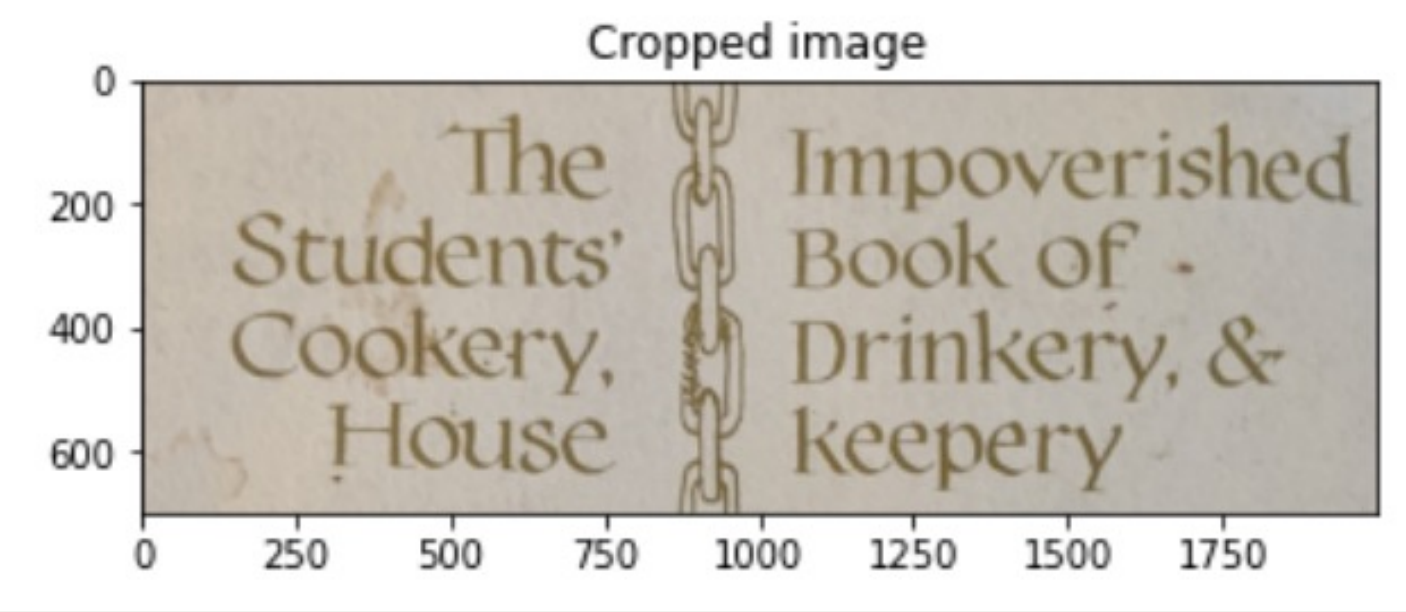




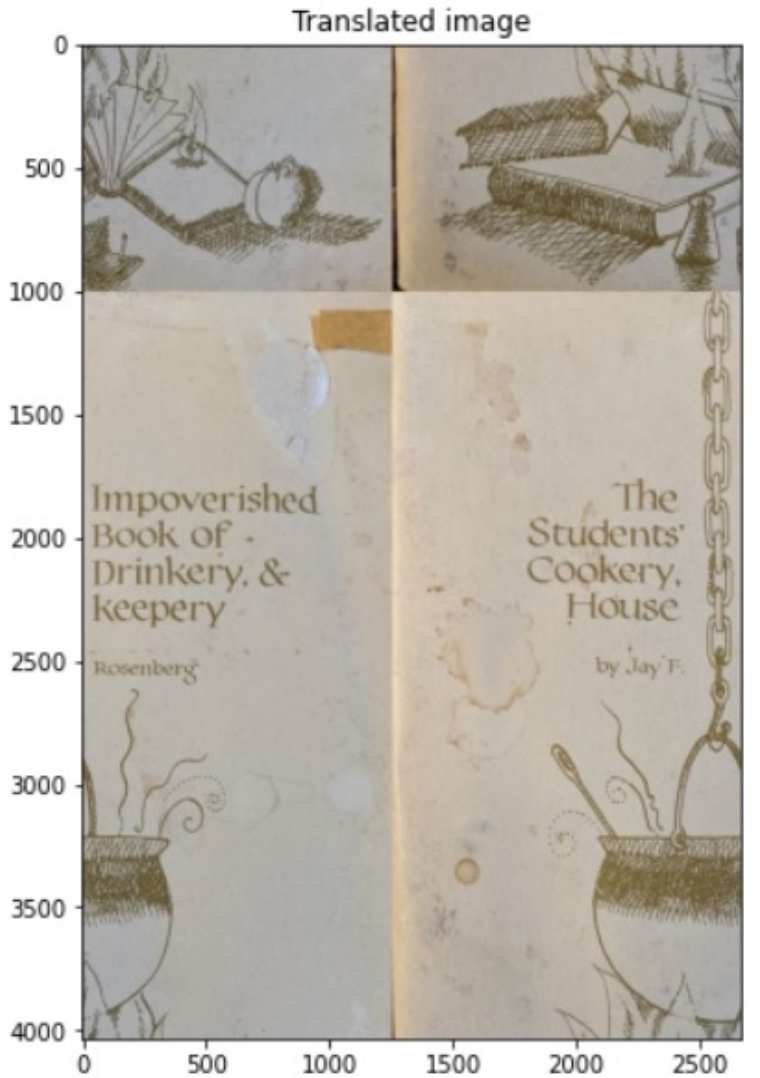


```
# show the start image
image_tools.show_image(image, axis=True, title="Original image", scale_ratio=1)

# crop an image to a range described by the top left and bottom right corners
image_cropped = image_tools.crop_image(image, top_left=(700, 400), bottom_right=(1400, 2400))
image_tools.show_image(image_cropped, axis=True, title="Cropped image", scale_ratio=1)
```

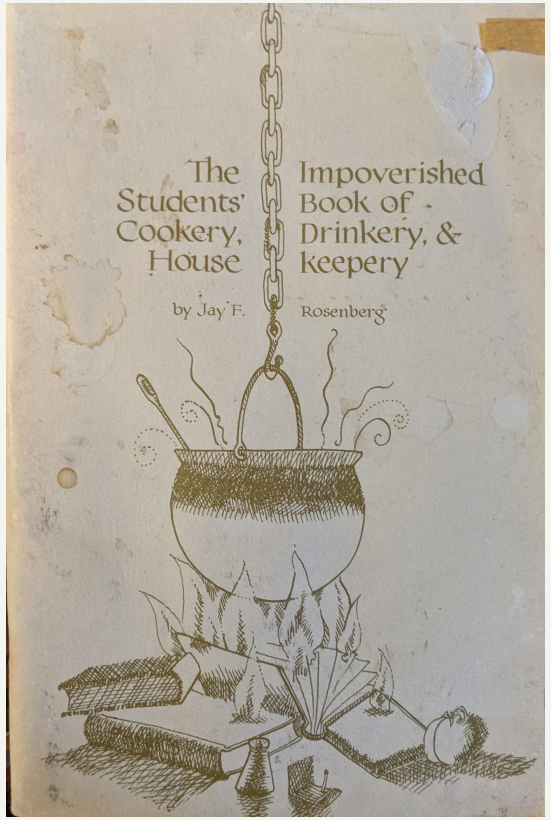


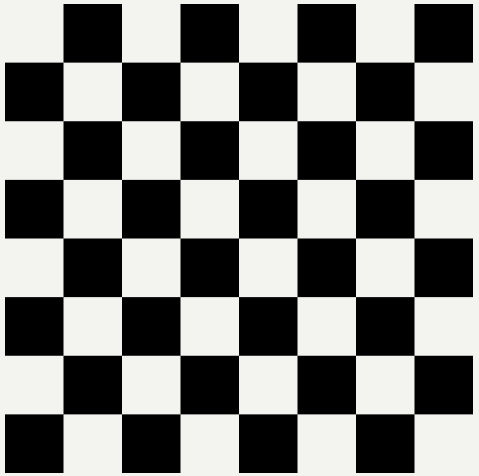


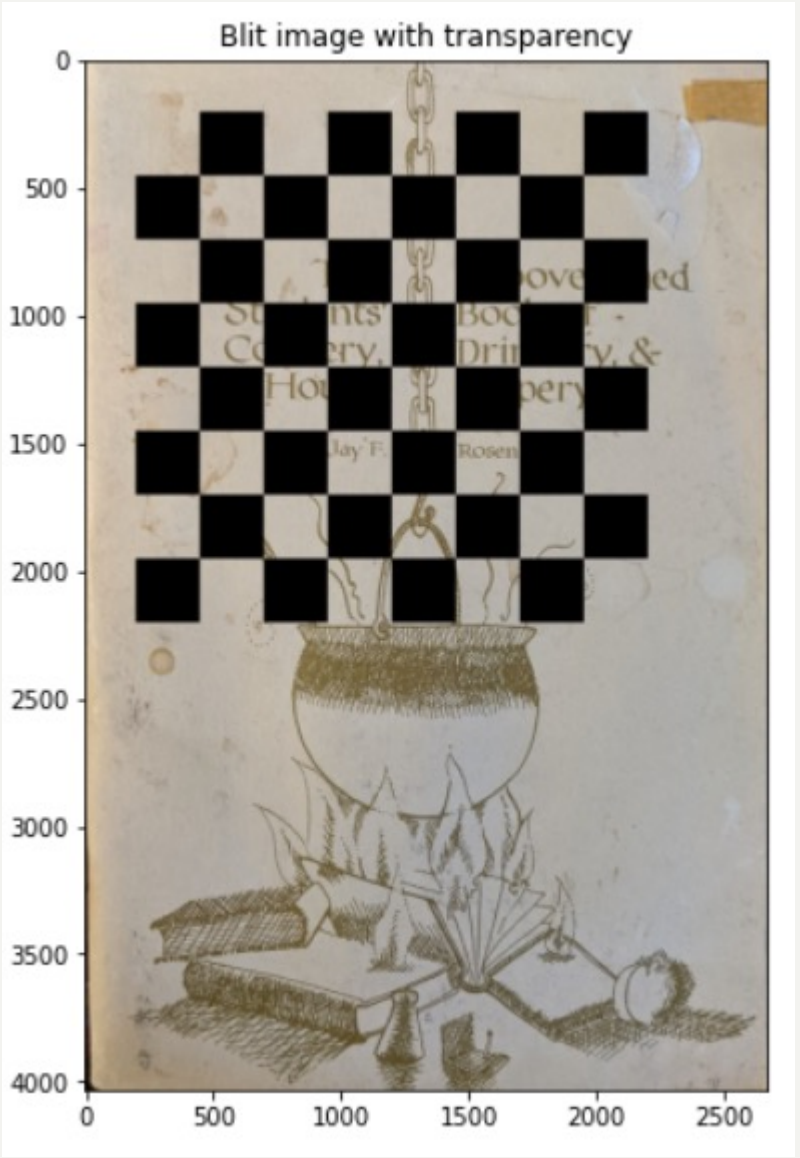
```
# every translation can be described as a combination of translations along the axes of an array.
# example 2d array translation:
image_translated = data_tools.translate_2d_array(image, 1255, 1000, border_type=cv2.BORDER_WRAP, border_value=(0, 0, 0))
image_tools.show_image(image_translated, axis=True, title="Translated image")
```



```
# blit chessboard onto the destination
img_blit = data_tools.blit(img_source, img_dest, *(200, 200))
image_tools.show_image(img_blit, axis=True, title="Blit image with transparency")
```





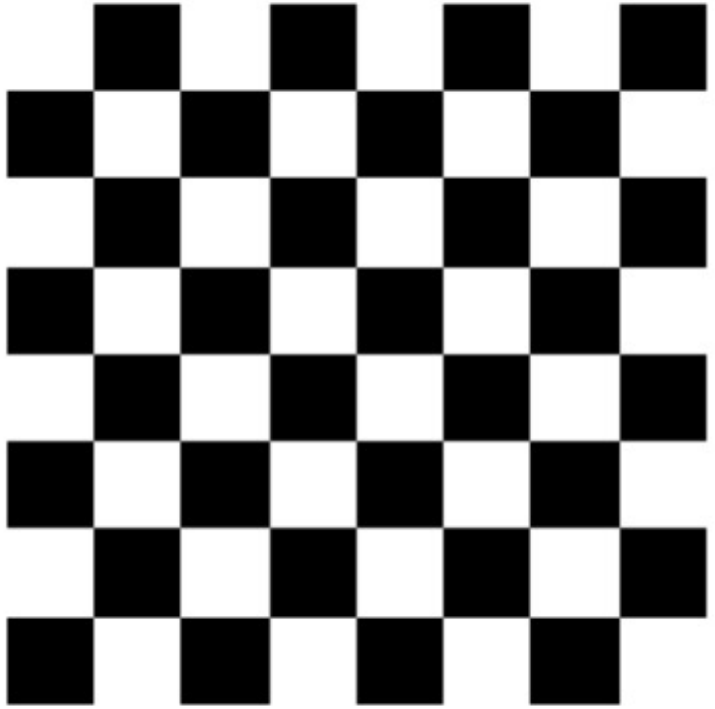


Generation

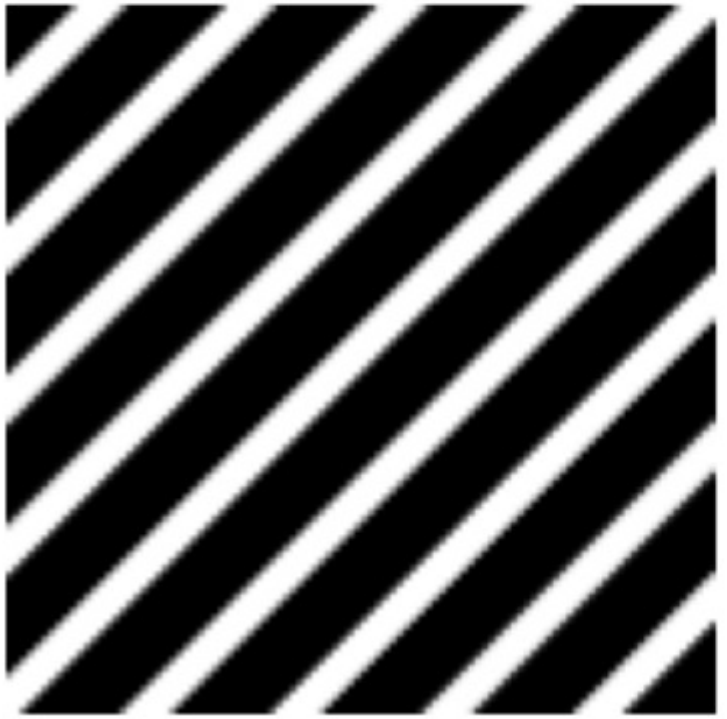
A couple of my favourite data generation tools (available in Data_generation.ipynb in the project documentation)

- Chessboards
- Stripes
- Gradients

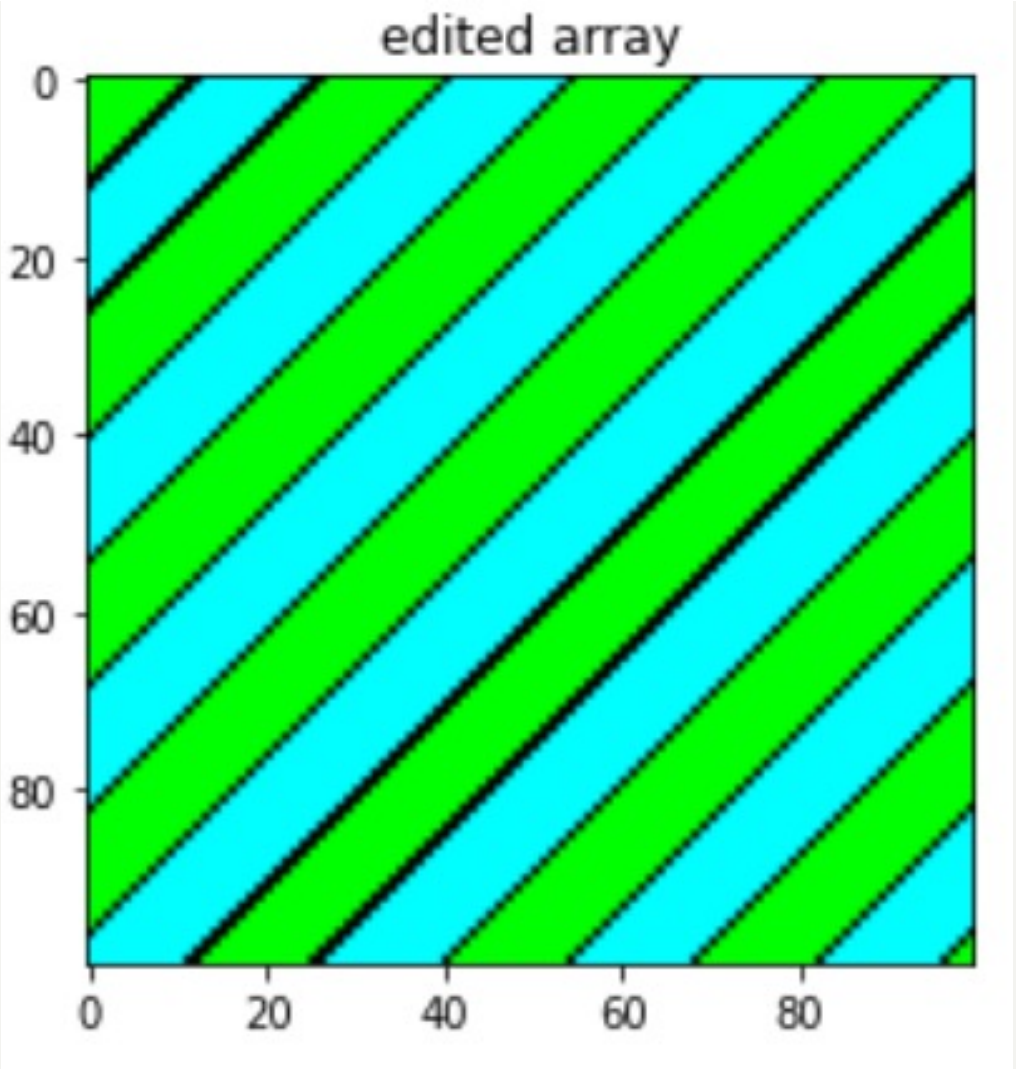
```
chessboard = create_chessboard()
image_tools.show_image(chessboard, scale_ratio=1)
```

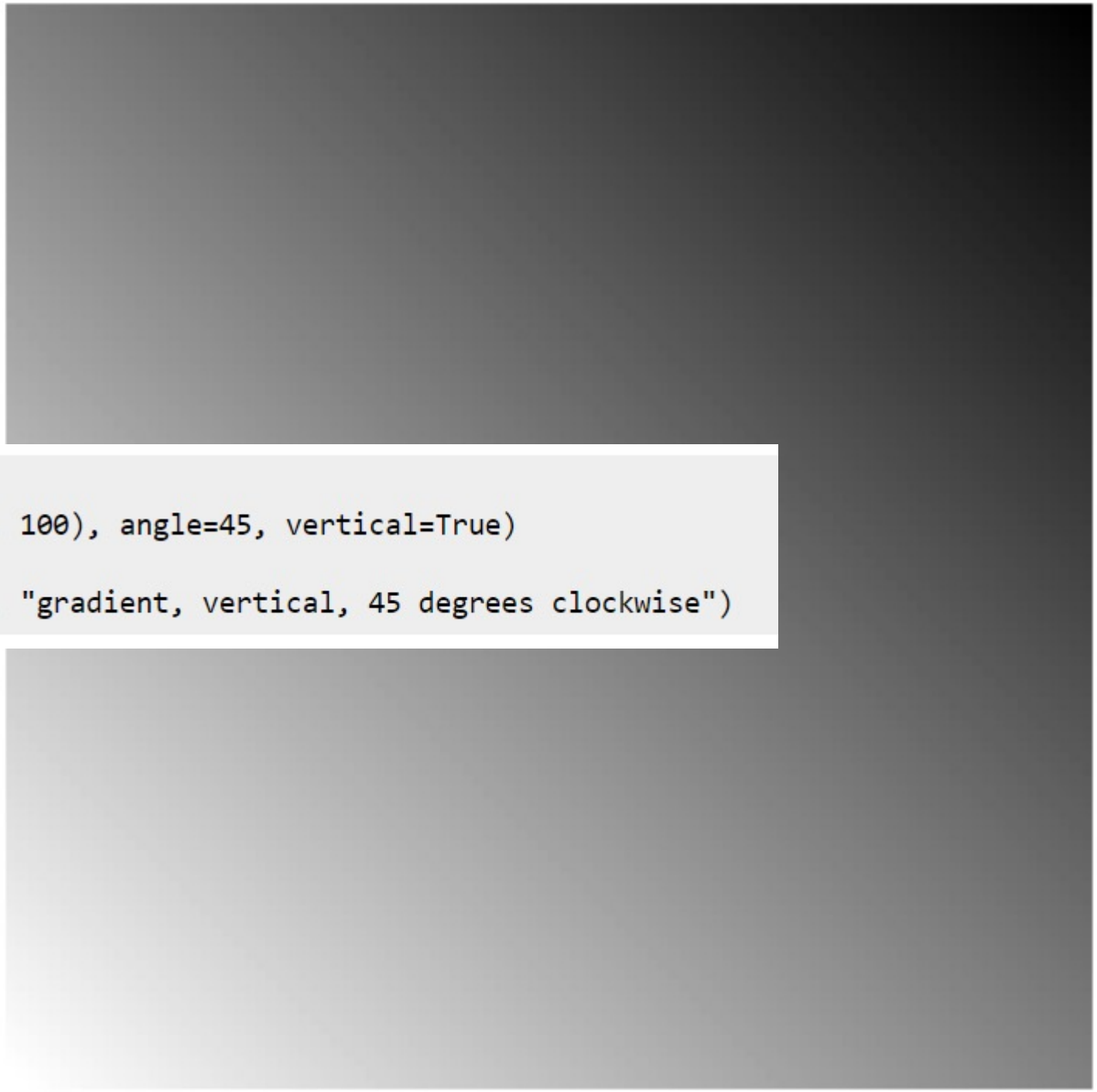


```
stripes = create_alternating_stripes((100, 100, 4), 10, 5, 45)
image_tools.show_image(stripes, scale_ratio=1)
```



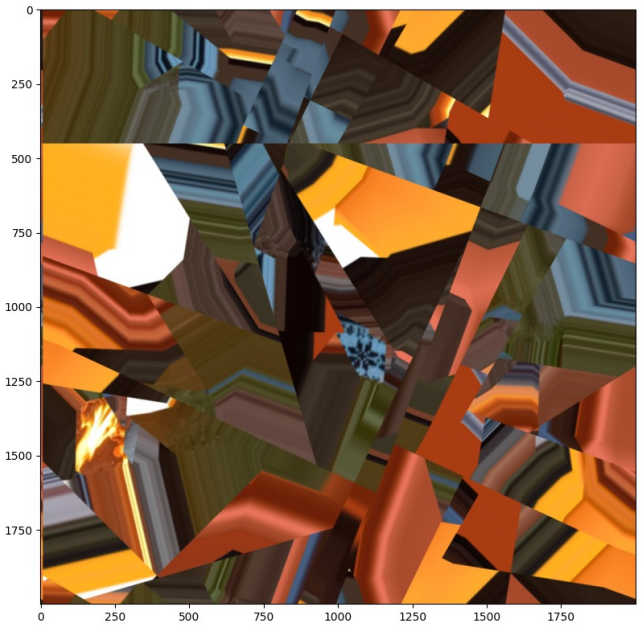
```
array2edit = np.zeros((100, 100, 3))
edit_mask = create_alternating_stripes(array2edit.shape, neg_stripe_width=10, pos_stripe_width=10, angle=-45, vertical=False)

array2edit[edit_mask == 1] = (0, 255, 0)
array2edit[edit_mask == 0] = (0, 255, 255)
# show the edited array
image_tools.show_image(array2edit, axis=True, title="edited array", scale_ratio=1)
```

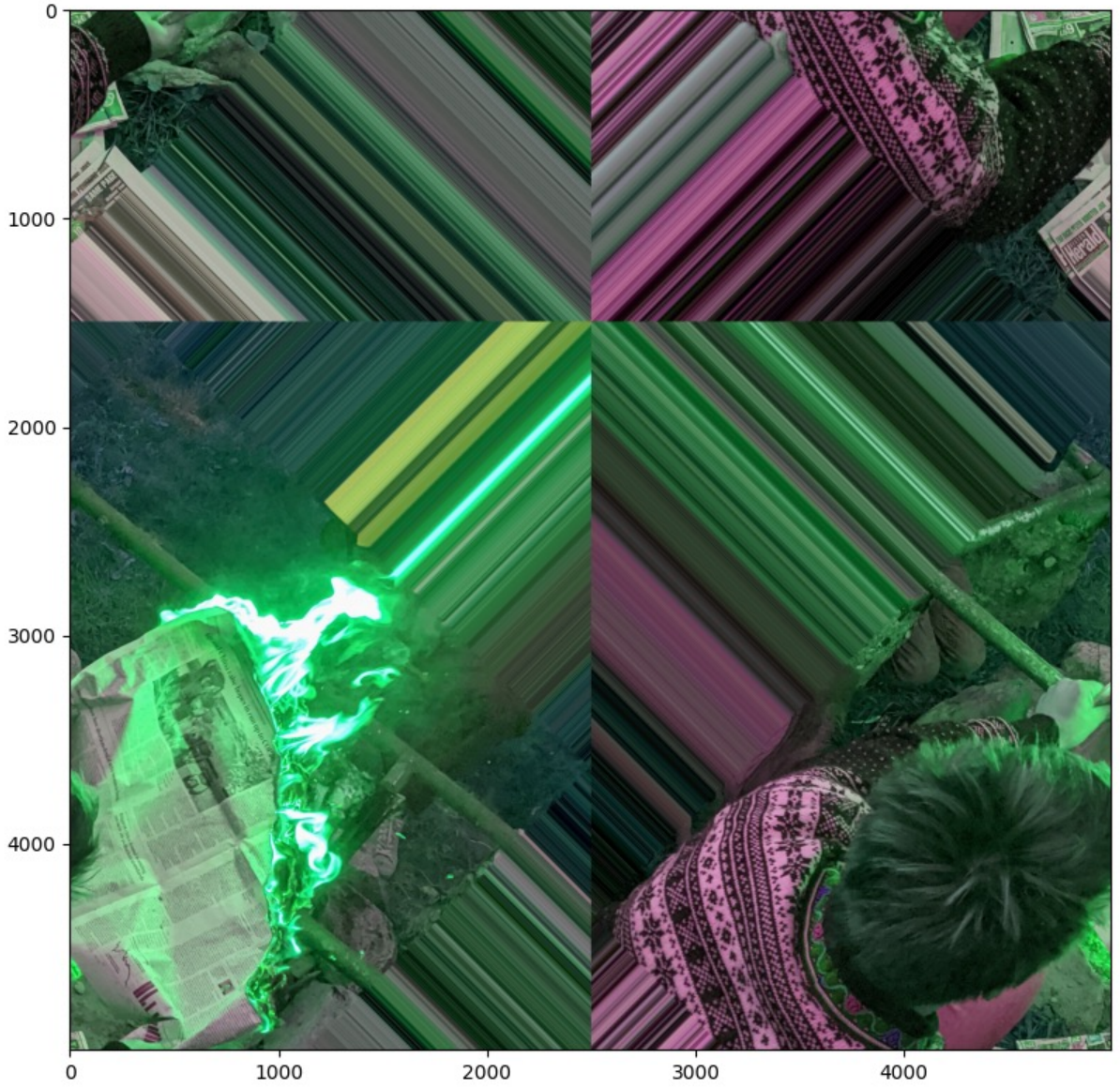


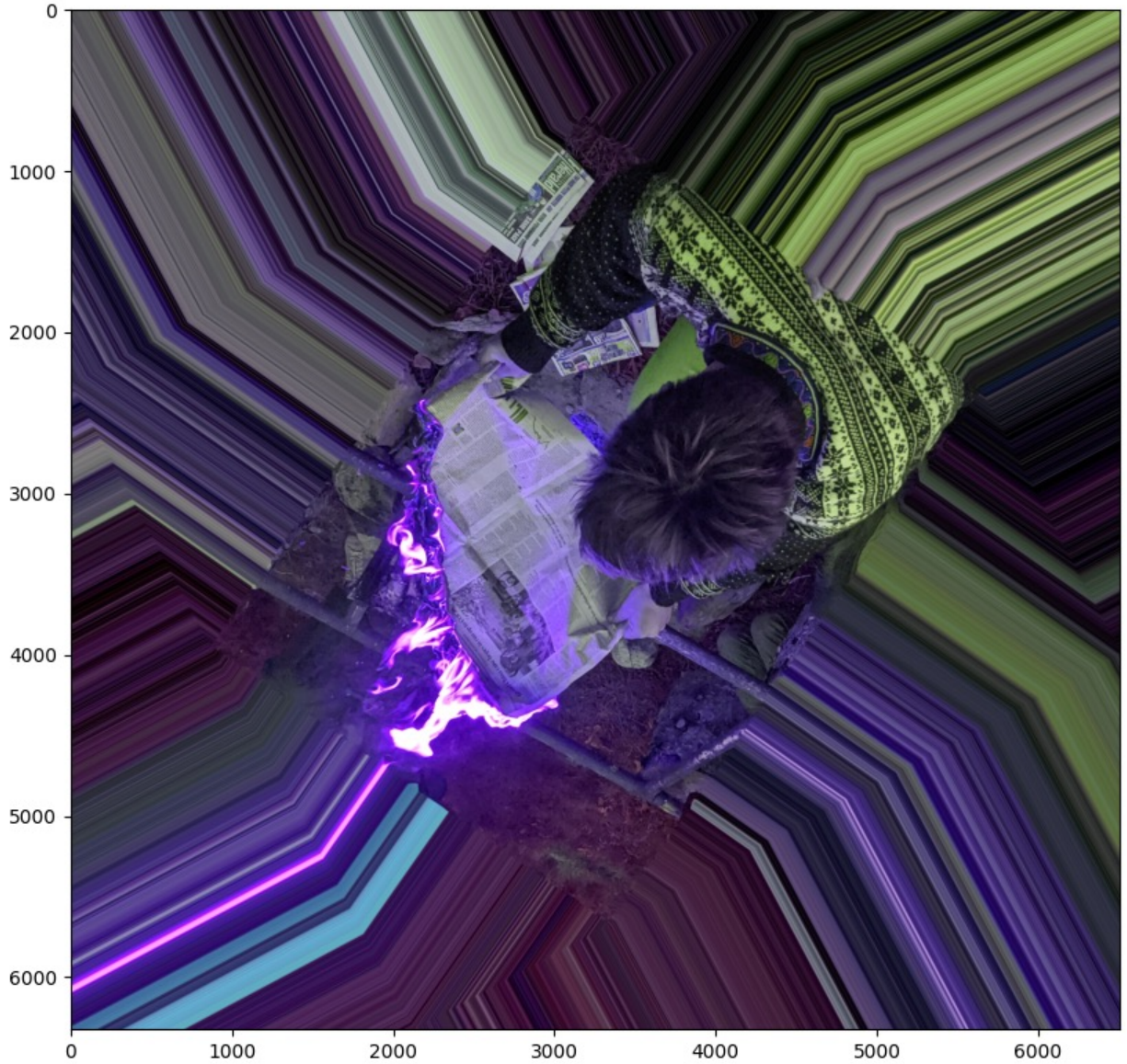


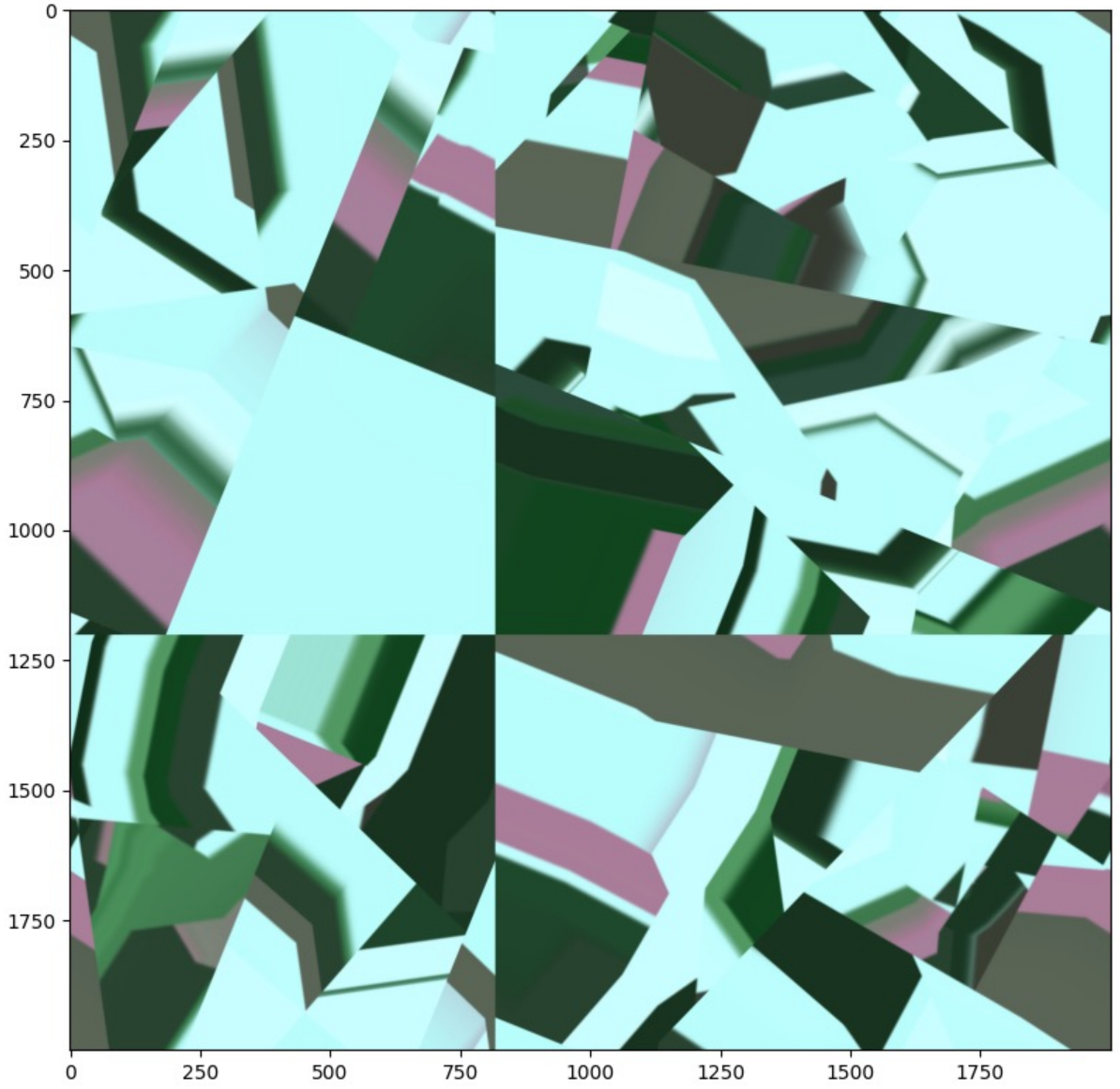
```
# create a gradient
gradient = create_gradient((100, 100), angle=45, vertical=True)
# show the gradient
image_tools.show_image(gradient, "gradient, vertical, 45 degrees clockwise")
```

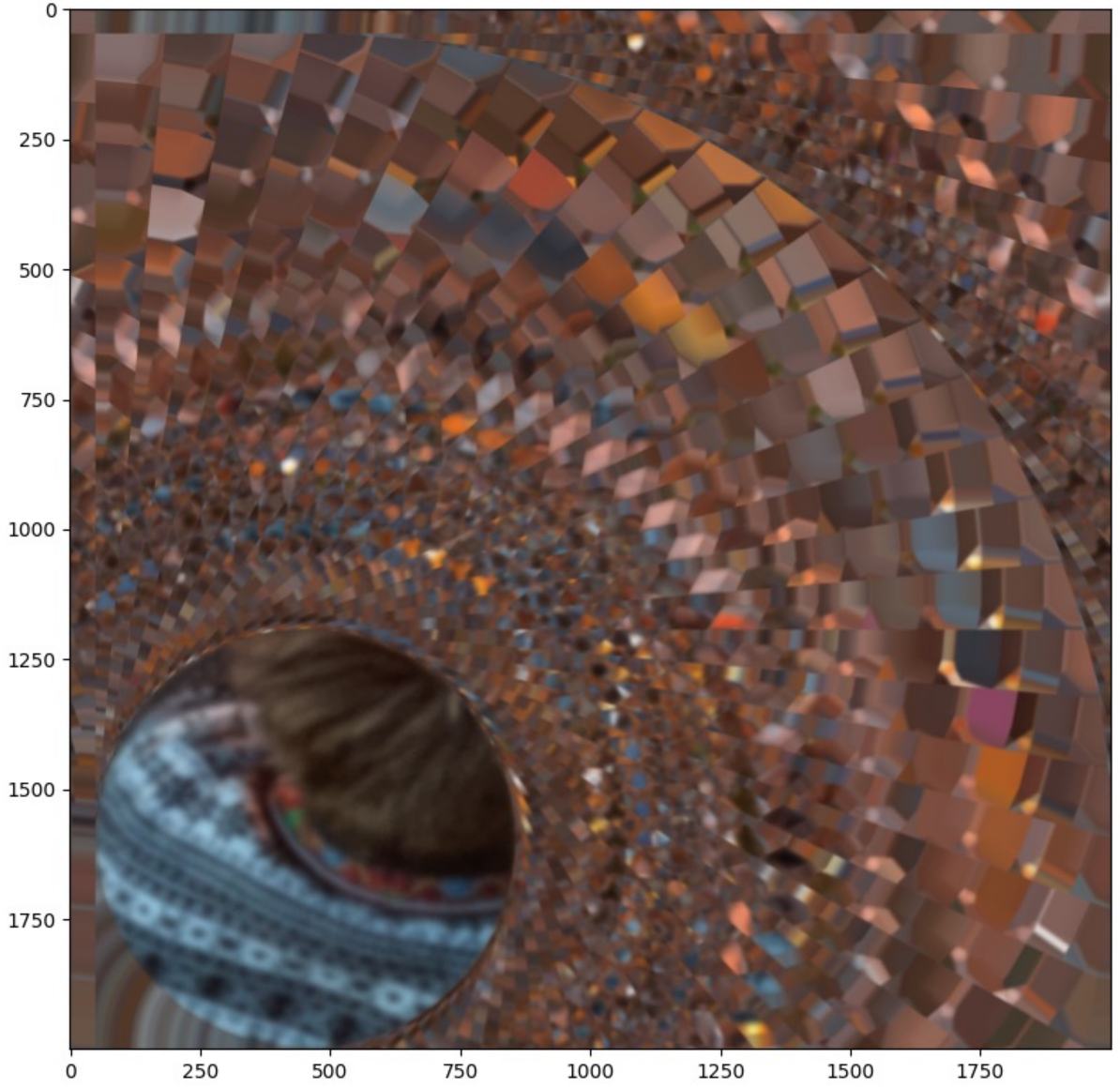
```
n=30
for i in range(n):
    # rotate the copy
    img_copy = image_tools.rotate_image(img_copy, randint(0, 359), False, border_type=cv2.BORDER_REPLICATE)
    # roll the color channels
    img_copy = data_tools.roll_array(img_copy, 1, 2)
    # translate the copy
    img_copy = data_tools.translate_2d_array(img_copy, randint(-img_copy.shape[0], img_copy.shape[0]),
                                             randint(-img_copy.shape[1], img_copy.shape[1]))
    # rescale the copy
    img_copy = data_tools.rescale_2d_array(img_copy, (2000, 2000))
```

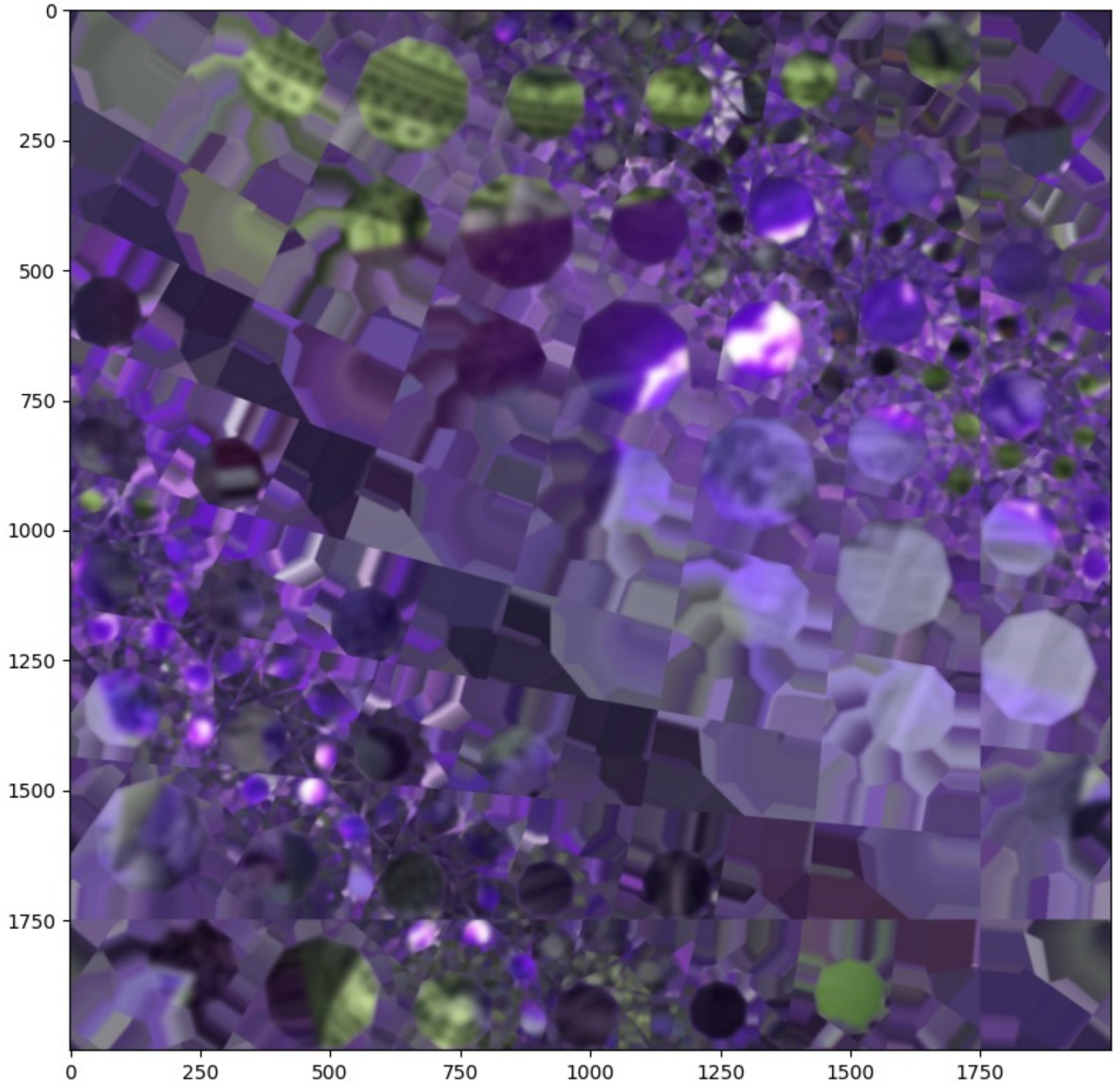


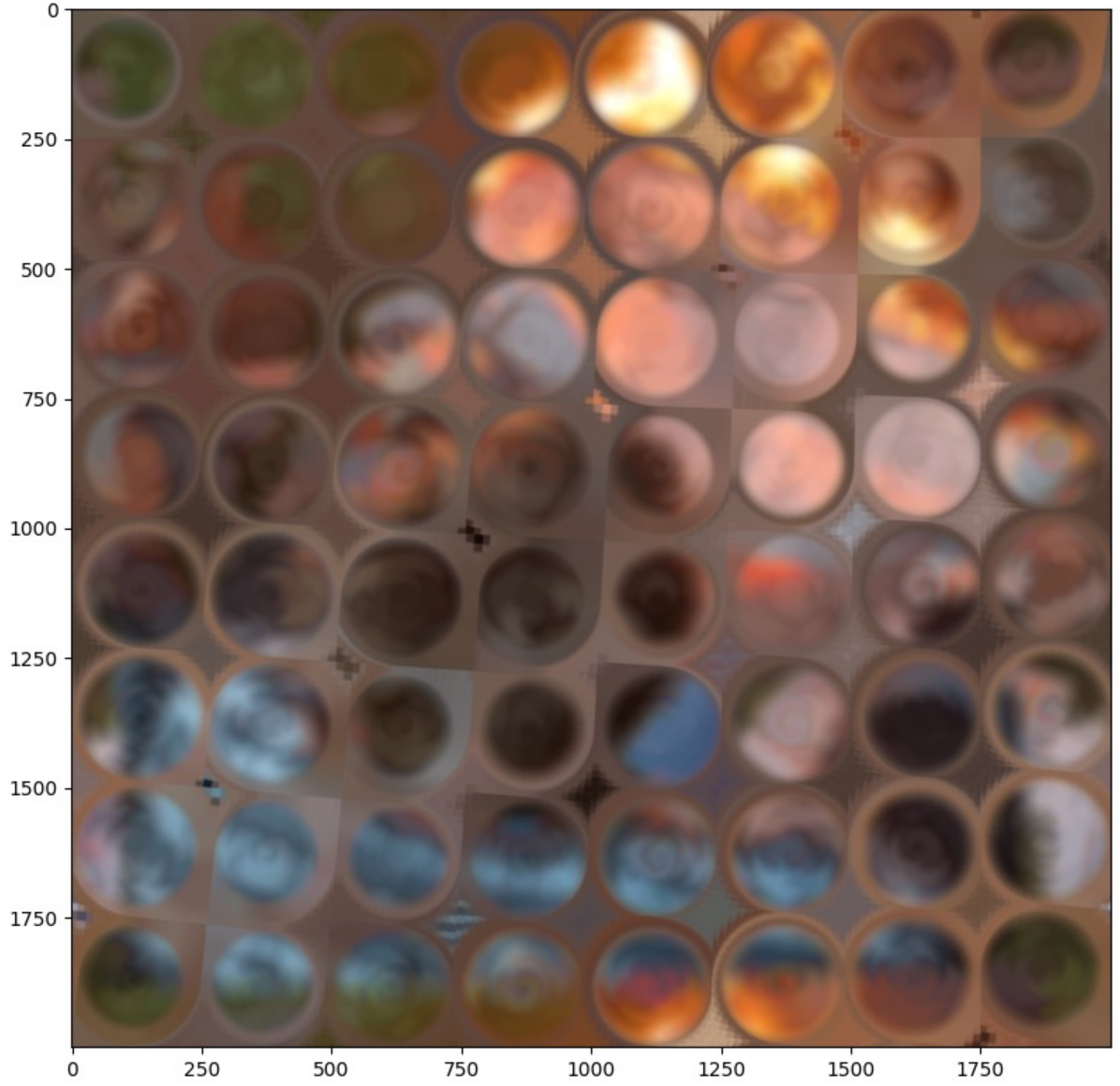

0
1000
2000
3000
4000
0
1000
2000
3000
4000


0
1000
2000
3000
4000
5000
6000
0
1000
2000
3000
4000
5000
6000


0
250
500
750
1000
1250
1500
1750
0
250
500
750
1000
1250
1500
1750


0
250
500
750
1000
1250
1500
1750
0
250
500
750
1000
1250
1500
1750


0
250
500
750
1000
1250
1500
1750
0
250
500
750
1000
1250
1500
1750


0
250
500
750
1000
1250
1500
1750
0
250
500
750
1000
1250
1500
1750

How to use this project yourself

---

1. Download Python
2. Head to https://github.com/phileasdg/Image_and_Audio_Generation_and_Editing_using_Python and download my code
3. Install the dependencies using the command:

   ```
   pip install -r requirements.txt
   ```

4. Mess around and have fun!

Bibliography of project sources

“Animated Image Using a Precomputed List of Images — Matplotlib 3.4.3 Documentation.” Accessed September 30, 2021. https://matplotlib.org/stable/gallery/animation/dynamic_image.html.

“Animation Example Code: Dynamic_image.Py — Matplotlib 2.0.2 Documentation.” Accessed September 30, 2021. https://matplotlib.org/2.0.2/examples/animation/dynamic_image.html.

Cross Validated. “Data Transformation - What Does ‘Normalization’ Mean and How to Verify That a Sample or a Distribution Is Normalized?” Accessed September 27, 2021. https://stats.stackexchange.com/questions/70553/what-does-normalization-mean-and-how-to-verify-that-a-sample-or-a-distribution.

GeeksforGeeks. “Different Ways to Create Pandas Dataframe,” November 14, 2018. https://www.geeksforgeeks.org/different-ways-to-create-pandas-dataframe/.

“How to Use Loc and Iloc for Selecting Data in Pandas | by B. Chen | Towards Data Science.” Accessed September 26, 2021. https://towardsdatascience.com/how-to-use-loc-and-iloc-for-selecting-data-in-pandas-bd09cb4c3d79.

Stack Overflow. “Image - Matplotlib Imshow(): How to Animate?” Accessed September 30, 2021. https://stackoverflow.com/questions/17212722/matplotlib-imshow-how-to-animate.

“Librosa — Librosa 0.8.1 Documentation.” Accessed September 26, 2021. https://librosa.org/doc/main/index.html.

Bibliography of project sources (cont'd)

Cross Validated. "Normalization - How to Normalize Data to 0-1 Range?" Accessed September 27, 2021. https://stats.stackexchange.com/questions/70801/how-to-normalize-data-to-0-1-range.

"NumPy Array Slicing." Accessed September 26, 2021. https://www.w3schools.com/python/numpy/numpy_array_slicing.asp.

"Numpy.Clip — NumPy v1.21 Manual." Accessed September 27, 2021. https://numpy.org/doc/stable/reference/generated/numpy.clip.html.

"OpenCV: Color Space Conversions." Accessed September 26, 2021. https://docs.opencv.org/3.4.15/d8/d01/group__imgproc__color__conversions.html.

"OpenCV: OpenCV-Python Tutorials." Accessed November 18, 2021. https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html.

"Pyplot vs Object Oriented Interface · Matplotblog." Accessed September 26, 2021. https://matplotlib.org/matplotblog/posts/pyplot-vs-object-oriented-interface/.

Stack Overflow. "Python - How to Load All Modules in a Folder?" Accessed September 26, 2021. https://stackoverflow.com/questions/1057431/how-to-load-all-modules-in-a-folder.

Stack Overflow. "Python - How to Normalize a NumPy Array to within a Certain Range?" Accessed September 27, 2021. https://stackoverflow.com/questions/1735025/how-to-normalize-a-numpy-array-to-within-a-certain-range.

Bibliography of project sources (cont'd)

Stack Overflow. "Python - Specifying and Saving a Figure with Exact Size in Pixels." Accessed September 27, 2021. https://stackoverflow.com/questions/13714454/specifying-and-saving-a-figure-with-exact-size-in-pixels.

Stack Overflow. "Python - Understanding Slice Notation." Accessed September 26, 2021. https://stackoverflow.com/questions/509211/understanding-slice-notation.

Stack Overflow. "Python - Using OpenCV to Overlay Transparent Image onto Another Image." Accessed November 18, 2021. https://stackoverflow.com/questions/40895785/using-opencv-to-overlay-transparent-image-onto-another-image.

Stack Overflow. "Python: Want to Display Red Channel Only in Opencv." Accessed September 27, 2021. https://stackoverflow.com/questions/44554125/python-want-to-display-red-channel-only-in-opencv.

Roberts, Leland. "Understanding the Mel Spectrogram." *Analytics Vidhya* (blog), March 14, 2020. https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53.

The Coding Train. *Coding Challenge #47: Pixel Sorting in Processing*, 2016. https://www.youtube.com/watch?v=JUDYkxU6J0o.

"The N-Dimensional Array (Ndarray) — NumPy v1.21 Manual." Accessed September 27, 2021. https://numpy.org/doc/stable/reference/arrays.ndarray.html.

Bibliography of project sources (cont'd)

"Using Display.Specshow — Librosa 0.8.1 Documentation." Accessed September 26, 2021. https://librosa.org/doc/main/auto_examples/plot_display.html#sphx-glr-auto-examples-plot-display-py.

Science ABC. "Why Is The Loudness Of Sound Expressed In Negative Decibels Sometimes?," December 1, 2018. https://www.scienceabc.com/pure-sciences/why-negative-decibels-are-a-thing.html.

"Matplotlib.Animation.ArtistAnimation — Matplotlib 3.4.3 Documentation." Accessed September 30, 2021. https://matplotlib.org/stable/api/_as_gen/matplotlib.animation.ArtistAnimation.html.

Stack Overflow. "Minimum Value on a 2d Array Python." Accessed September 27, 2021. https://stackoverflow.com/questions/44431412/minimum-value-on-a-2d-array-python.