# Using PyAutoLens for Generation of Adversial Examples for a Convolutional Neural Network Model

Mohammed Asfour

June 17, 2021

**Abstract**

Software for astronomy has had great advances in the previous year to ease the processing and exploration of space phenomena. Such an example is the generation of an image of the black hole in 2019. This report outlines the usage of one of such packages called PyAutoLens. PyAutoLens is a python package for gravitational lens problems, which can be summarized by the bending of the light travelling across the galaxy. The light can bend due to the mass of nearby galaxies or even black holes, which prohibits us from reaching a clear visualization from distant stars past our solar system. The proposed package in this report uses PyAutoLens to generate synthesized images of the gravitational lens phenomenon, and trains a convolutional neural network (CNN) to be able to predict raw images from space telescopes to predict properties of the distant starts to better understand them from their bent light.

## 1 Introduction

Astronomical phenomena has been throughly explored[1][2][3]; yet, it's the trend to use newly released packages for model fitting to explain such phenomena[4][5]. Gravitational Lens is the name given to the phenomenon of bending the lights of stars while travelling past bodies with significant gravitational pull. It's common to call the distant source of the bent light by "source galaxy", while the galaxy with the significant gravitational pull as "lens galaxy". Due to that phenomenon, space telescopes could produce images of distant galaxies or starts that are warped, resulting into straight beams of light to be seen as a circle or a curve. An illustration of this can be seen in Fig. 1

The problem of gravitaional lens is solved using ray tracing. By knowing more about the source galaxy and the lens galaxy, we can simulate how the light is bent while travelling to reach the exact point it had reached in the space telescope image. The ray tracing can then be used to know the real source galaxy.

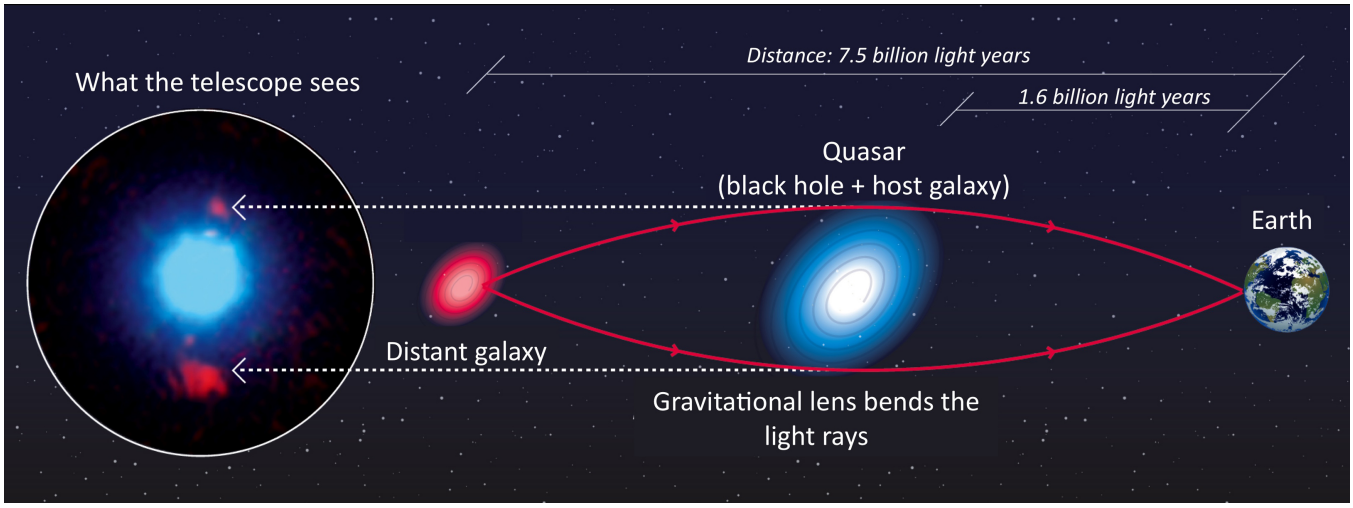


Image credit: F. Courbin, S. G. Djorgovski, G. Meylan, et al., Caltech / EPFL / WMKO

Figure 1: An Example of the Gravitational Lens Phenomenon.

# 2 Methodology

## 2.1 Scripts

### 2.1.1 params_and_cli.py

This script is responsible for holding on the hyper-parameters that will control the generation of the dataset, training of the model, and producing the figures.

It contains the hyper-parameters values, as well as the command line interface (CLI) arguments, their description, and their shortcuts. On top of that, it contains the functions that parse the cli passed arguments and modifies the behavior of other scripts. Lastly it includes the functions and information necessary to print out a help table for each of the other two file to the cli using the argument –help while running the other two scripts.

### 2.1.2 generate_data.py

This script is used to randomly generate gravitational lens problem settings using hyperparameters, and saves the generated results as .pickle files in ./dataset with a csv file containing the labels (properties of the source galaxy)

It contains functions that use samples generated from "params_and_cli.py" script to produce the gravitational lens example solve the ray tracing problem and produce the final image that would have been taken by a telescope as an array. The array is then saved as a ".pickle" intermediate file to be used for more processing.

The following code can be run in the cli to use the script

```
1 The hyperparameters controlling the generation process can be
     set using the cli arguments while running the script.
2   For example:
3     python generate_data.py --src-intensity-range=0.3,1.0
4
```

```
To view all the possible arguments and their description use:
    python generate_data.py --help
```

### 2.1.3 model_and_figures.py

This script creates a TensorFlow CNN model using a pre-defined architecture. The script then trains the models, and saves it followed by producing figures locally.

This script contains does of the work of the proposed package as it has to train a convolution model, and produce activation figures of the convolutional layers which could be a bit computationally extensive.

The following code can be run in the cli to use the script

```
The hyperparameters controlling the training and figures can be
   set using the cli arguments while running the script.
  For example:
    python model_and_figures.py --epochs=50 -f=png

To view all the possible arguments and their description use:
    python model_and_figures.py --help
```

## 2.2 Model Architecture

Using a convolutional neural network, the package is able to leverage the deep learning power to learn patterns from images and predict properties of the source galaxy.

The model used is a sequential model, meaning each layer is feeding into the one after it. The whole architecture has a total of 1,214,307 parameters with layers output shapes as follows in Table 1

- Input Layer (None, 64, 64, 1)
- 2D Convolution (None, 64, 64, 32)
- Pooling (None, 32, 32, 32)
- 2D Convolution (None, 32, 32, 64)
- Pooling (None, 16, 16, 64)
- 2D Convolution (None, 16, 16, 128)
- Pooling (None, 8, 8, 128)
- flatten (None, 8192)
- Fully-Connected (None, 128)
- Dropout (None, 128)
- Fully-Connected (None, 3)

The following hyper-parameters were used for training

| Hyperparameter | Default Value |
|---|---|
| valid-split | 0.2 |
| test-split | 0.1 |
| labels | src-redshift, src-center-x, src-center-y |
| epochs | 50 |
| learning-rate | 0.00001 |
| batch-size | 20 |

Table 1: Hyperparameters used in model training

### 2.3 Figures

The figures to be produced will be split into 3 distinguished categories:

- Visualization of the samples generated by the PyAutoLens[6] package
- The losses curves of the training of the neural network
- The visualization of convolutional layer activations.

## 3 Results

### 3.1 The Generated Samples

Firstly we visualize the gravitaional lens adversial examples generated by the PyAutoLens[6] packge. Examples of the samples can be seen in Fig. 2

### 3.2 Model Training

We can then feed the sampled images with the samples properties to train the model, resulting in the following loss curves in Fig. 3

### 3.3 Visualization of Convolutinal Layers Activations

Finally we are able to visualize the activations of the convolutional layers in the model to better understand which characteristics of the images that the model use to predict the properties of the source galaxy. The input image to the model can be seen in Fig. 4, while the activations of the convolutional layers are reported in Fig. 5, Fig. 6, and Fig. 7.

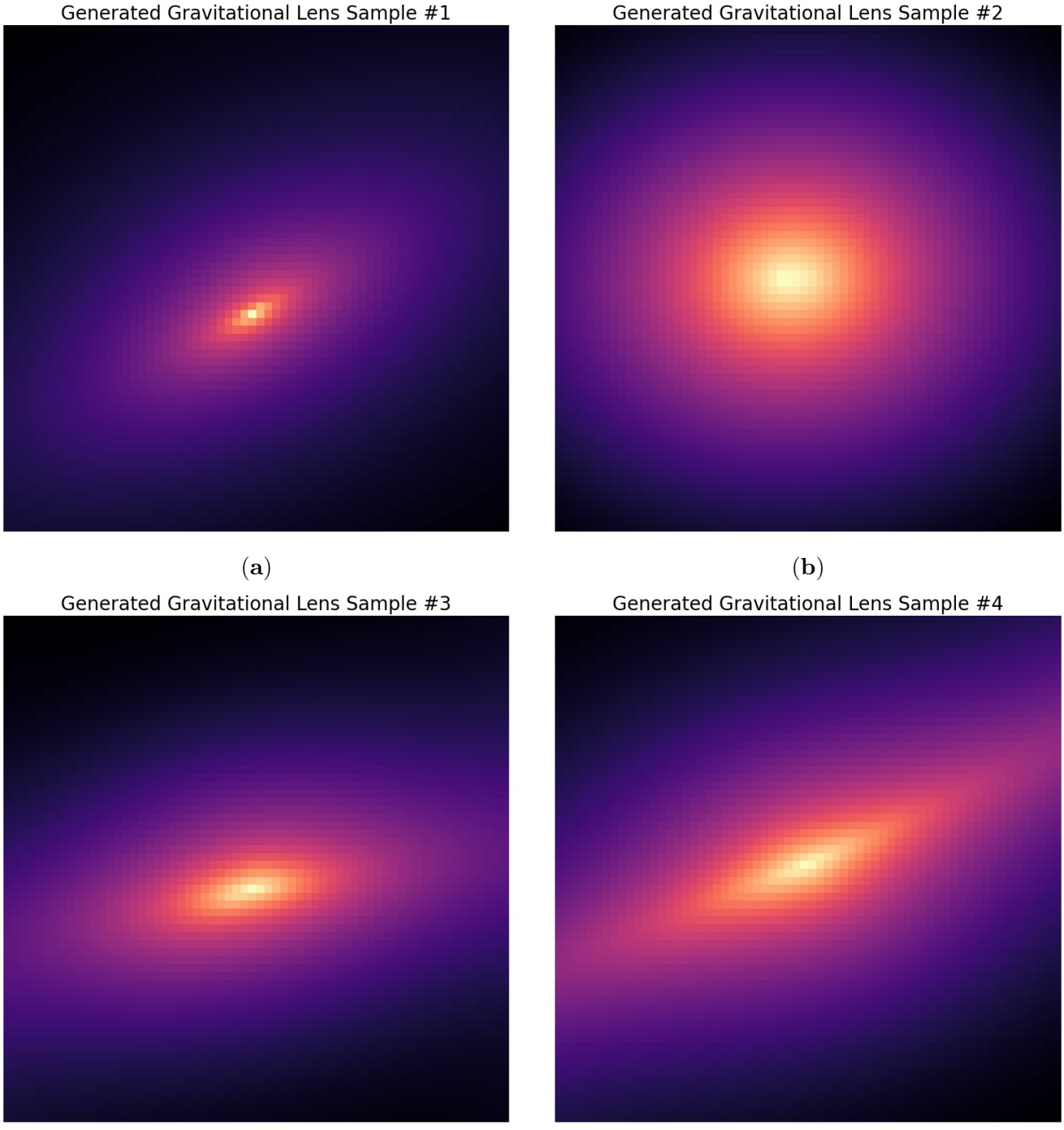


Figure 2: Visualization of samples generated by PyAutoLens[6] of the gravitational lens phenomenon.

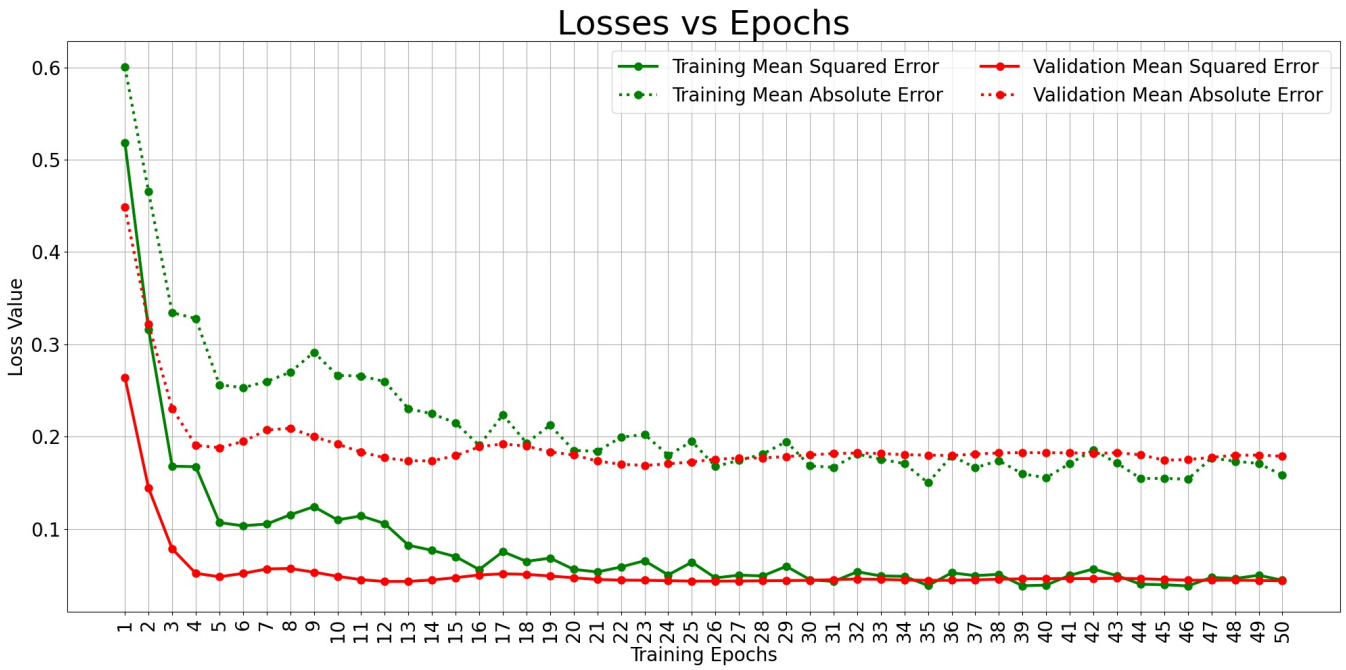


Figure 3: The training and validation losses of the model during training

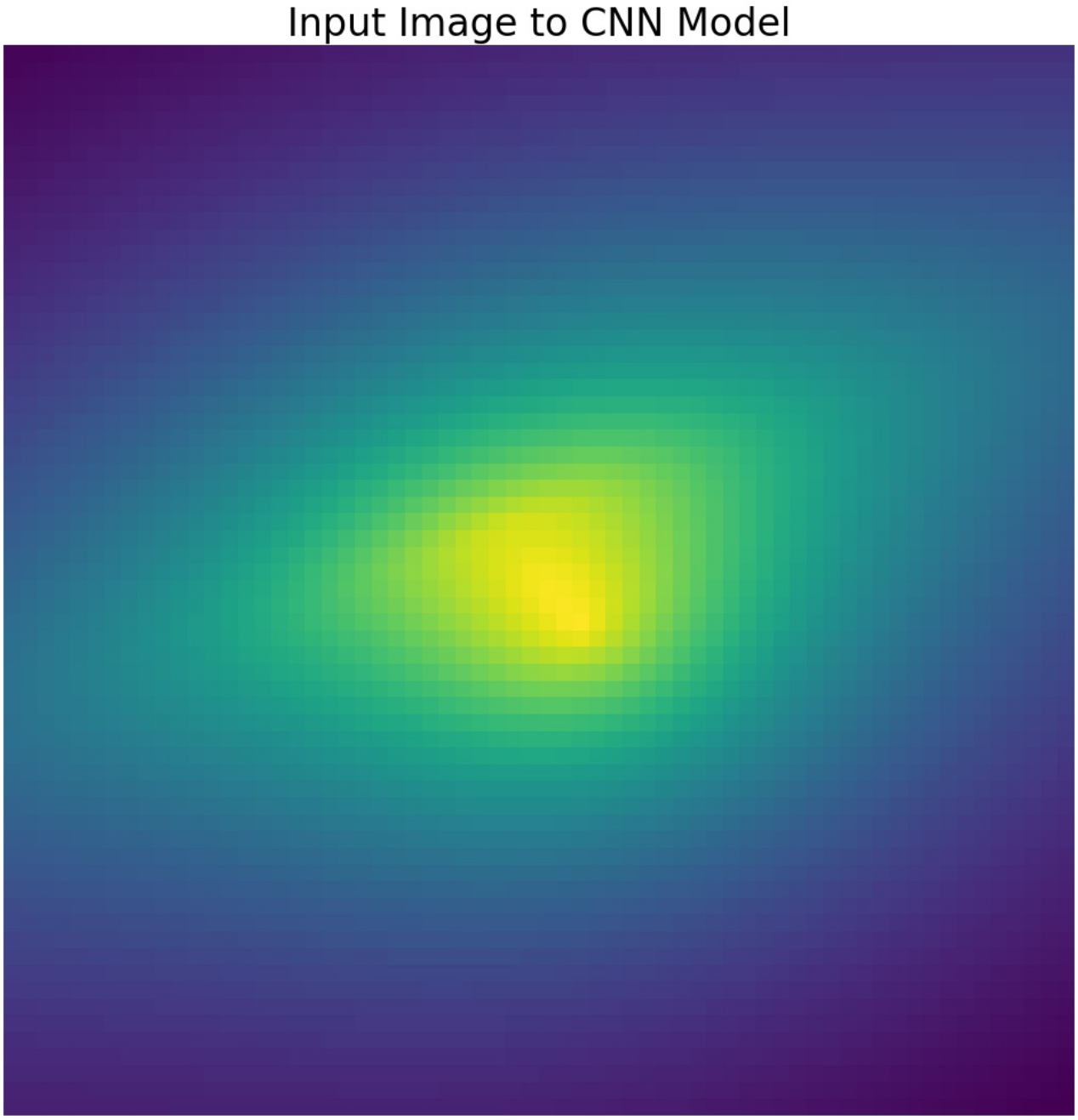


Figure 4: Input image to the CNN model generated by PyAutoLens[6] package

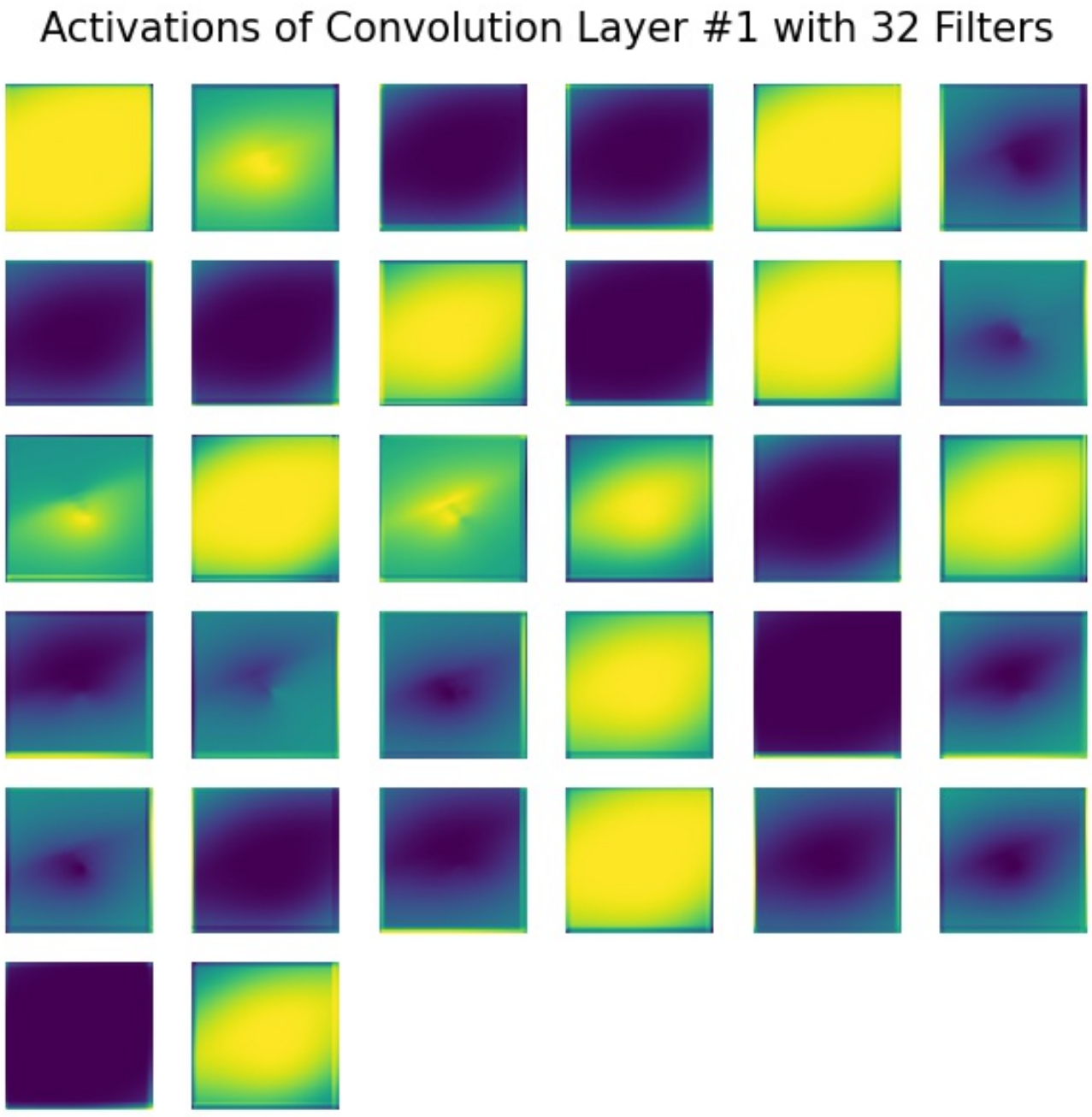


Figure 5: Activation of the filters of the first convolutional layer corresponding to the sample image generated by PyAutoLens[6] package

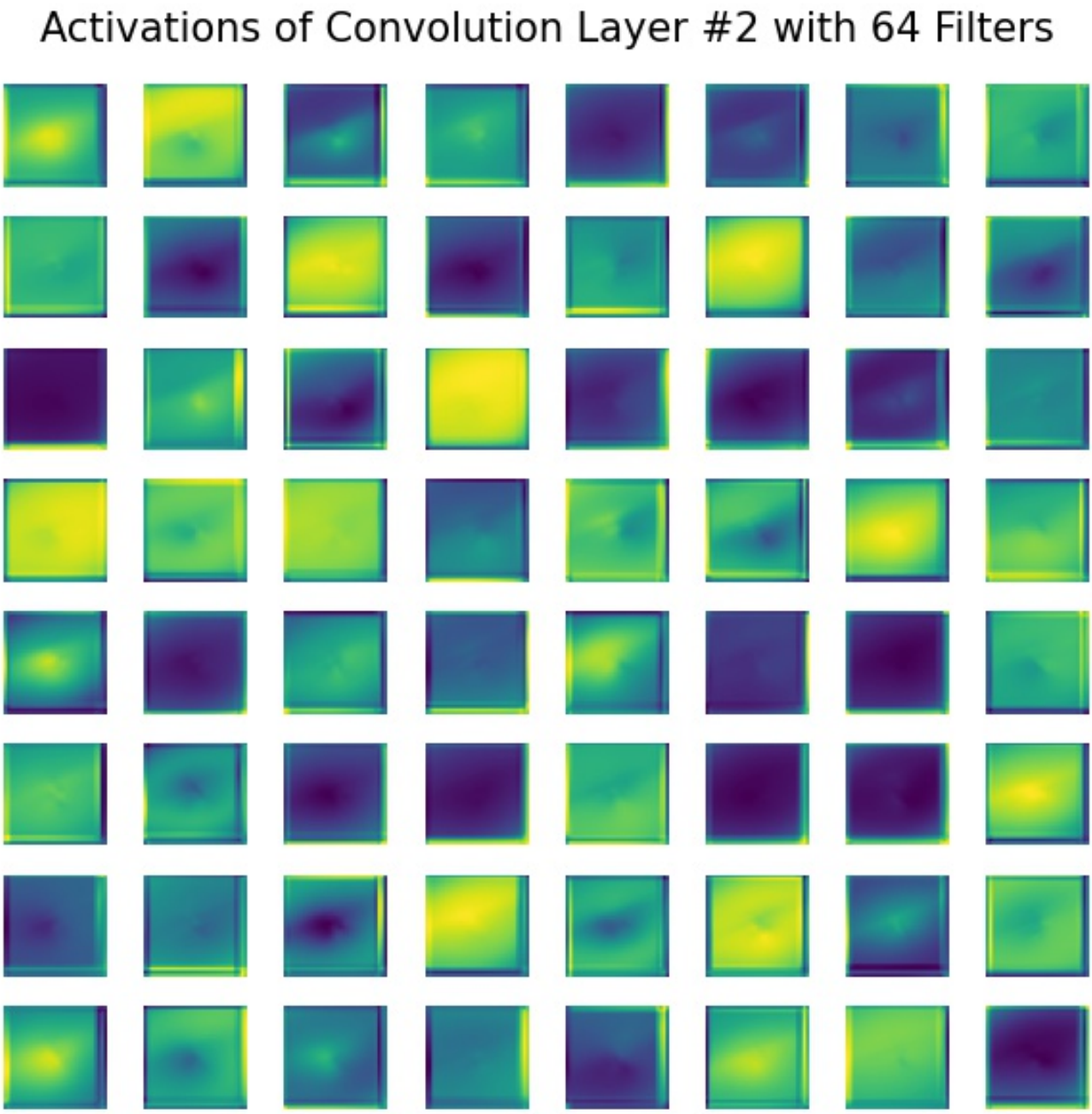


Figure 6: Activation of the filters of the second convolutional layer corresponding to the sample image generated by PyAutoLens[6] package

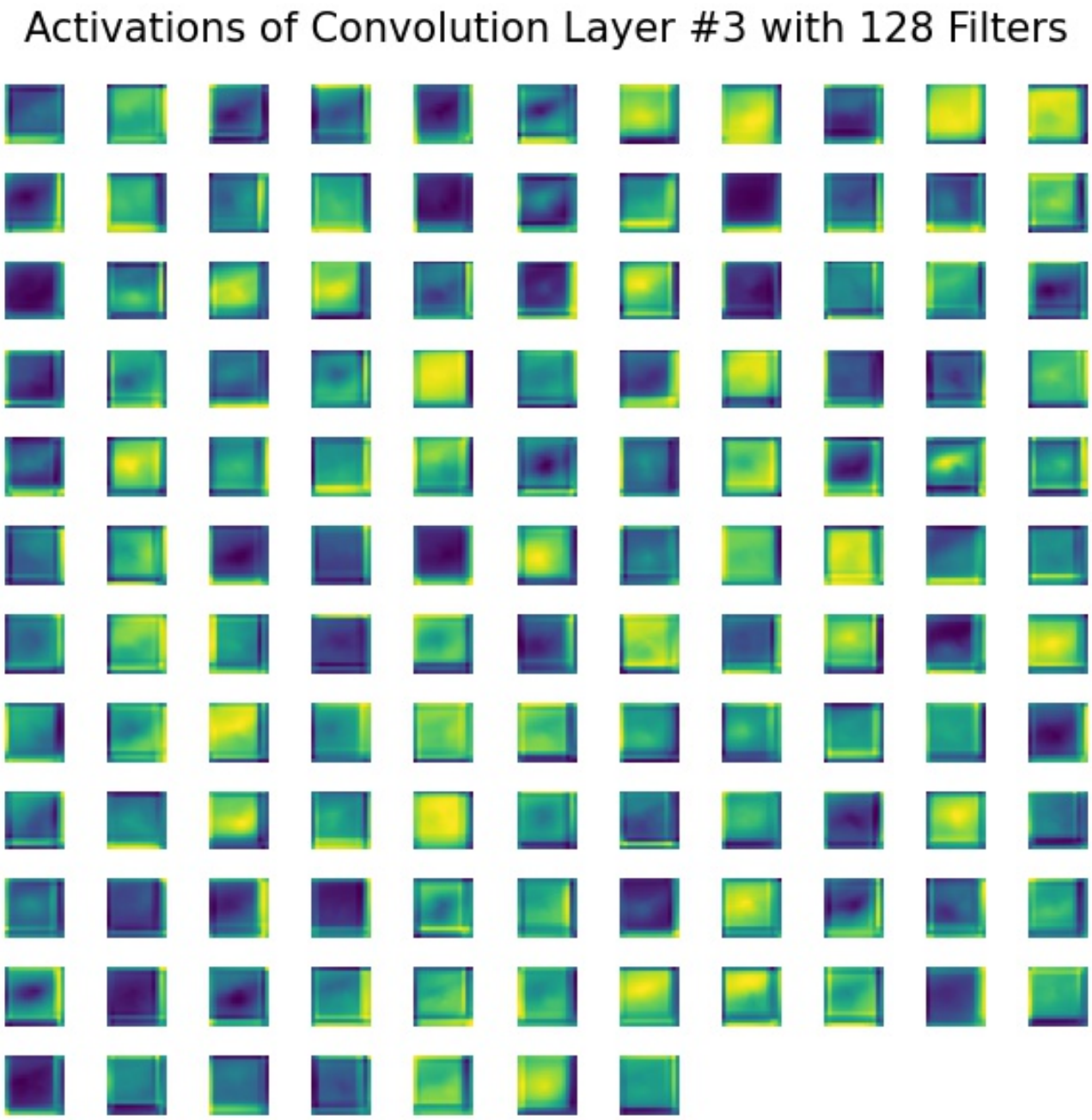


Figure 7: Activation of the filters of the third convolutional layer corresponding to the sample image generated by PyAutoLens[6] package

# References


[1] J. Maresca, S. Dye, and N. Li, "Auto-identification of unphysical source reconstructions in strong gravitational lens modelling," *Monthly Notices of the Royal Astronomical Society*, vol. 503, 02 2021.

[2] S. Birrer, A. Shajib, D. Gilman, A. Galan, J. Aalbers, M. Millon, R. Morgan, G. Pagano, J. Park, L. Teodori, N. Tessore, M. Ueland, L. Vyvere, E. Wempe, L. Yang, X. Ding, T. Schmidt, D. Sluse, and A. Amara, "lenstronomy ii: A gravitational lensing software ecosystem," *"*, 06 2021.

[3] J. Nightingale, R. Hayes, A. Kelly, A. Amvrosiadis, A. Etherington, Q. He, N. Li, X. Cao, J. Frawley, S. Cole, A. Enia, C. Frenk, D. Harvey, R. Li, R. Massey, M. Negrello, and A. Robertson, "Pyautolens: Open-source strong gravitational lensing," *"*, 06 2021.

[4] J. Bramburger, S. Alexander, and E. McDonough, "Dark disk substructure and superfluid dark matter," *Physics Letters B*, 08 2019.

[5] D. Powell, S. Vegetti, J. McKean, C. Spingola, F. Rizzo, and H. Stacey, "A novel approach to visibility-space modelling of interferometric gravitational lens observations at high angular resolution," *Monthly Notices of the Royal Astronomical Society*, vol. 501, 09 2020.

[6] J. Nightingale, R. Hayes, A. Kelly, A. Amvrosiadis, A. Etherington, Q. He, N. Li, X. Cao, J. Frawley, S. Cole, A. Enia, C. Frenk, D. Harvey, R. Li, R. Massey, M. Negrello, and A. Robertson, "Pyautolens: Open-source strong gravitational lensing," *Journal of Open Source Software*, vol. 6, p. 2825, 02 2021.