Gravitational Lensing

When an object is behind something of great mass, the light rays of the object must travel along a warped space-time which distorts how it appears to the observer. The written programme attempts to reproduce this effect. The code will compare the theoretical R value to see how well it is able to give the predicted Einstein Ring. For this purpose, the percentage difference between the inputted R and the experimental R will be shown for each pixel resolution. The hypothesis is that as the resolution of the pixels increase, the percentage difference should reduce. Since the Einstein ring is given in pixel coordinates, but rc is in reduced coordinates [-1,1] which relates to $R = (1-rc^2)^{1/2}$, the radius R in pixel coordinates is taken as a percentage of the resolution and multiplied by 2. This will give its length in reduced coordinates.

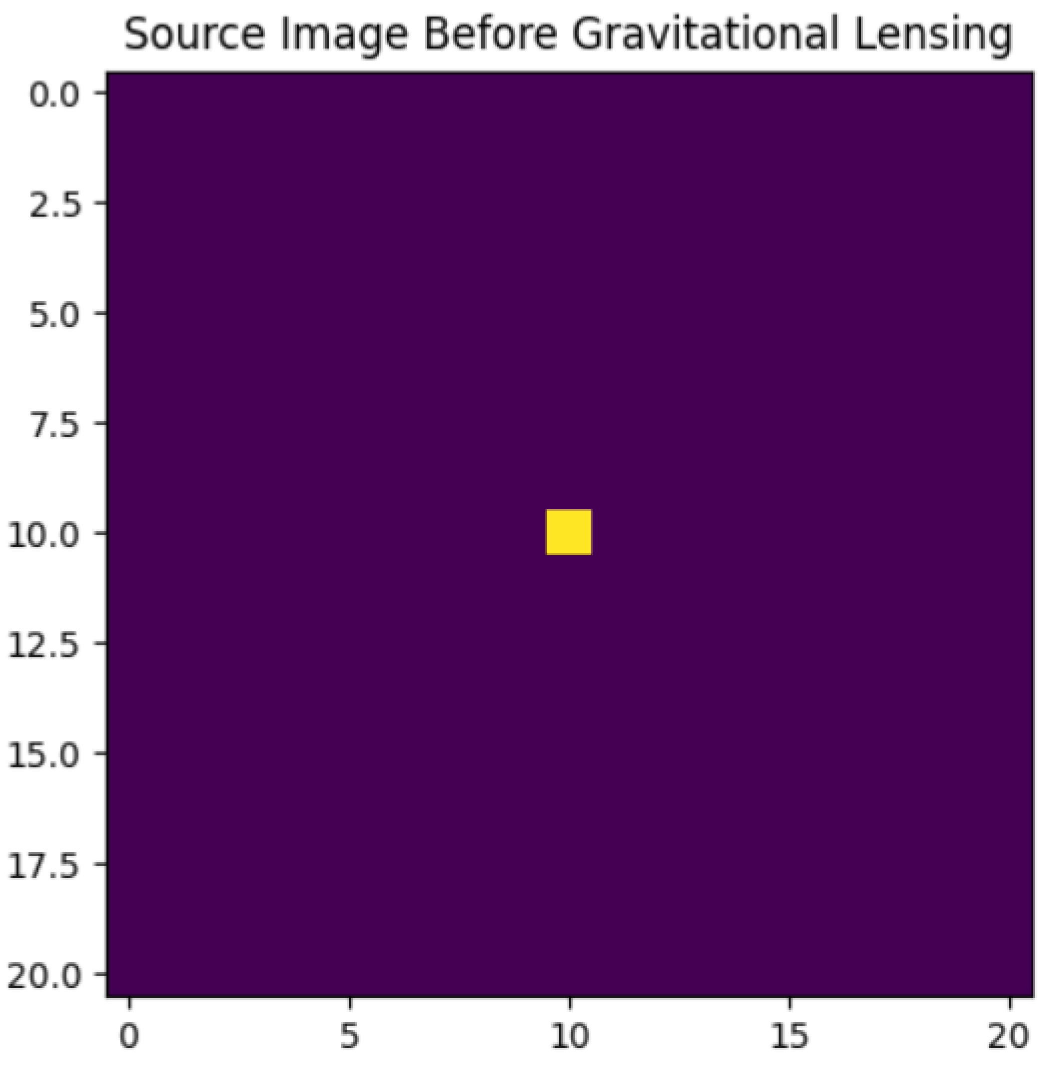

Figure 1. Source Image that is not lensed, centred as a single pixel at 10,10. Total pixel coordinates run from (0,20). This is the test case.

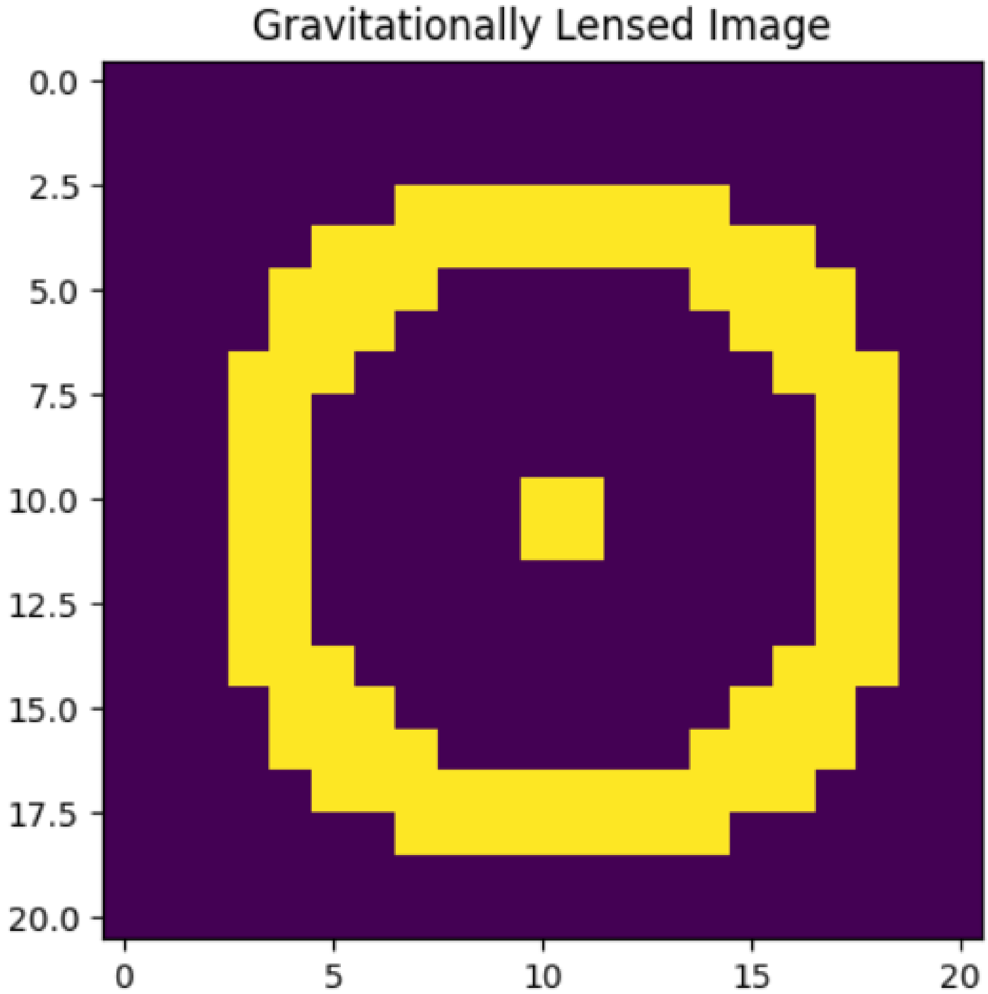

Figure 2. Gravitationally lensed image of source located at 10,10 from figure 1, with epsilon = 0.0, input rc = 0.70, expected R = 0.714, and output R = 0.837. The percentage difference of rc is given as 17.0%. The pixel resolution is quite low at only 21, and therefore the percentage difference is big.

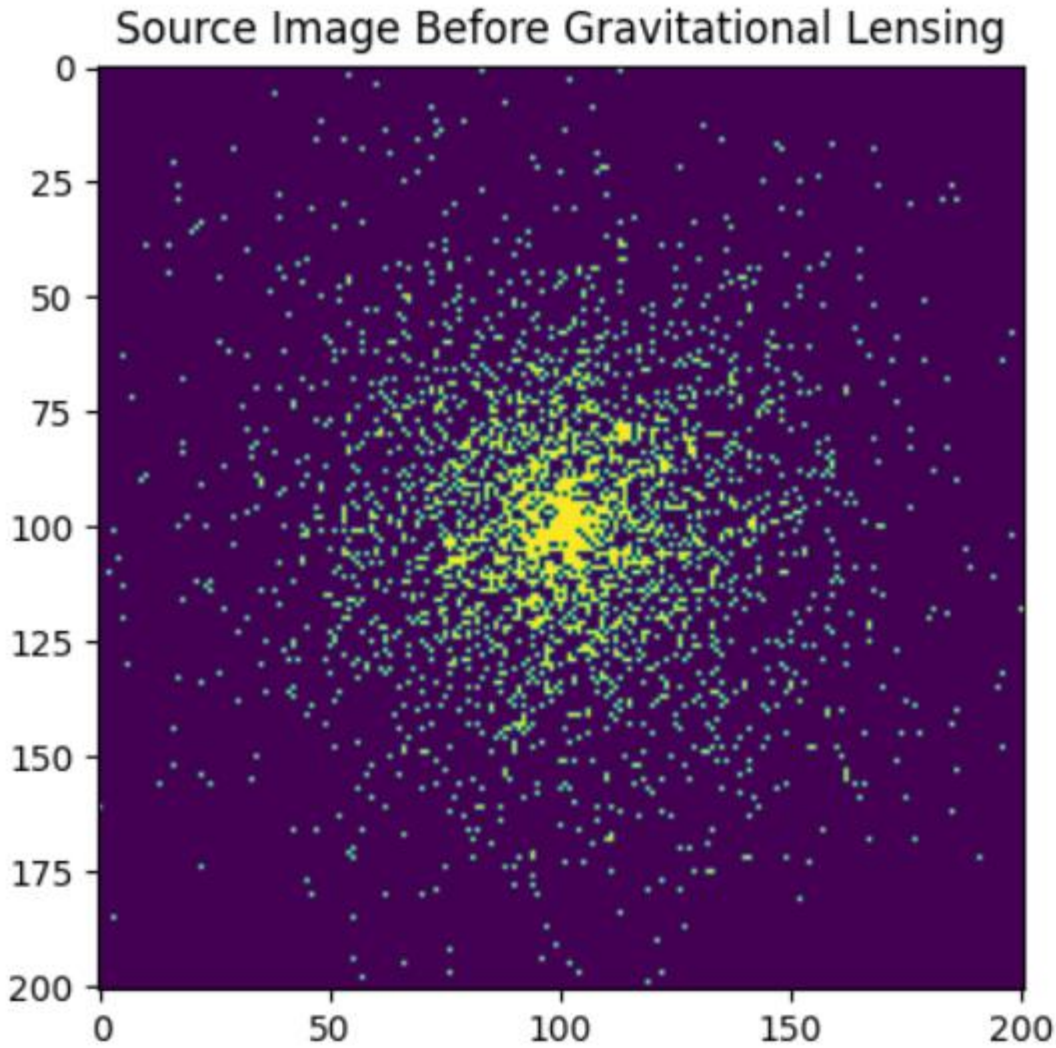

Figure 3. Source image of galaxy cluster with 201x201 pixel resolution, there is a greater concentration towards the centre (100,100) due to the probability decreasing as radius increases. This is given by equation f = f0exp(-r/a), where a =1, f0 =1, and a maximum radius has not been forced on the cluster. The value of rc is calculated rather than set, by substituting given variables of project.

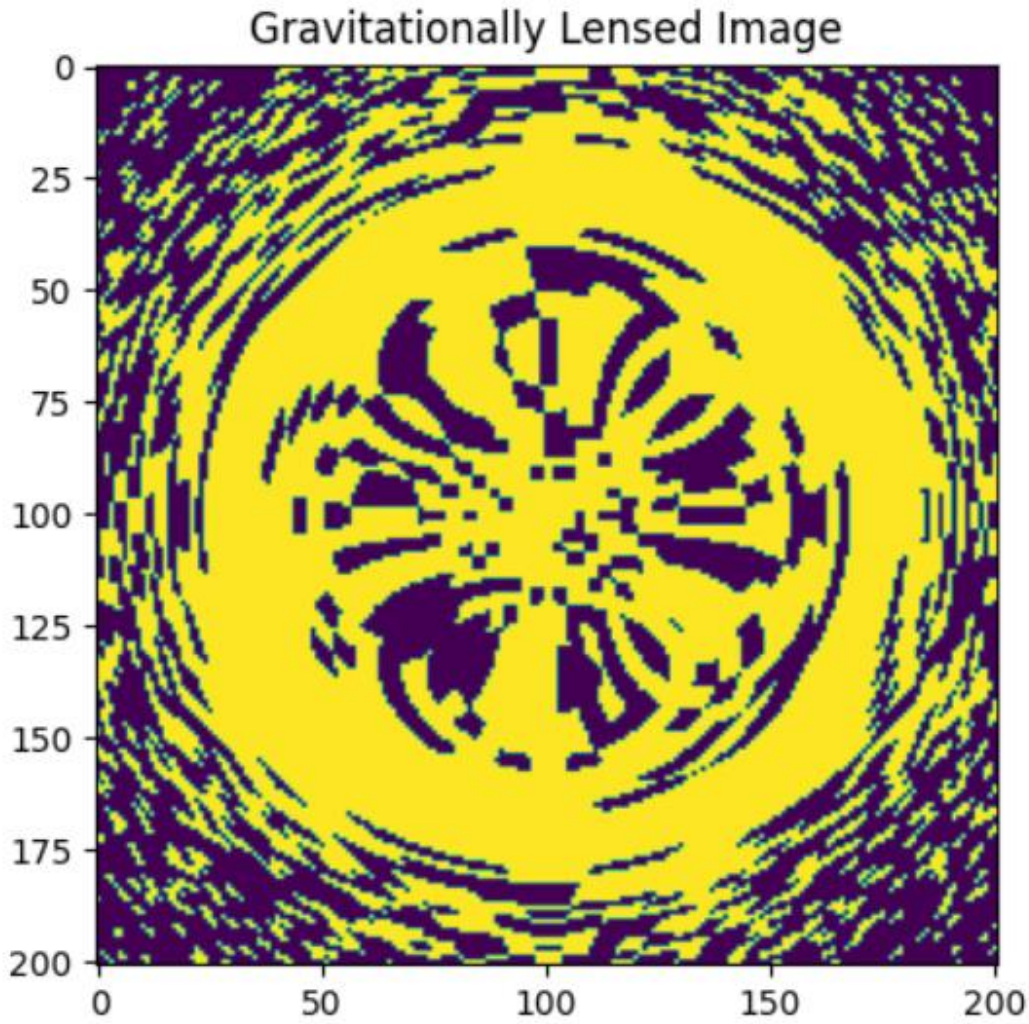

Figure 4. Einstein Ring of figure 3. This image has properties: rc = 0.695, D_s= 878, D_i= 637, D_ls = 441, rc = 0.695, theoretical R= 0.719, output R = 0.948, percentage difference = 31.8%. The percentage difference is lower than that of 21x21 resolution as expected.

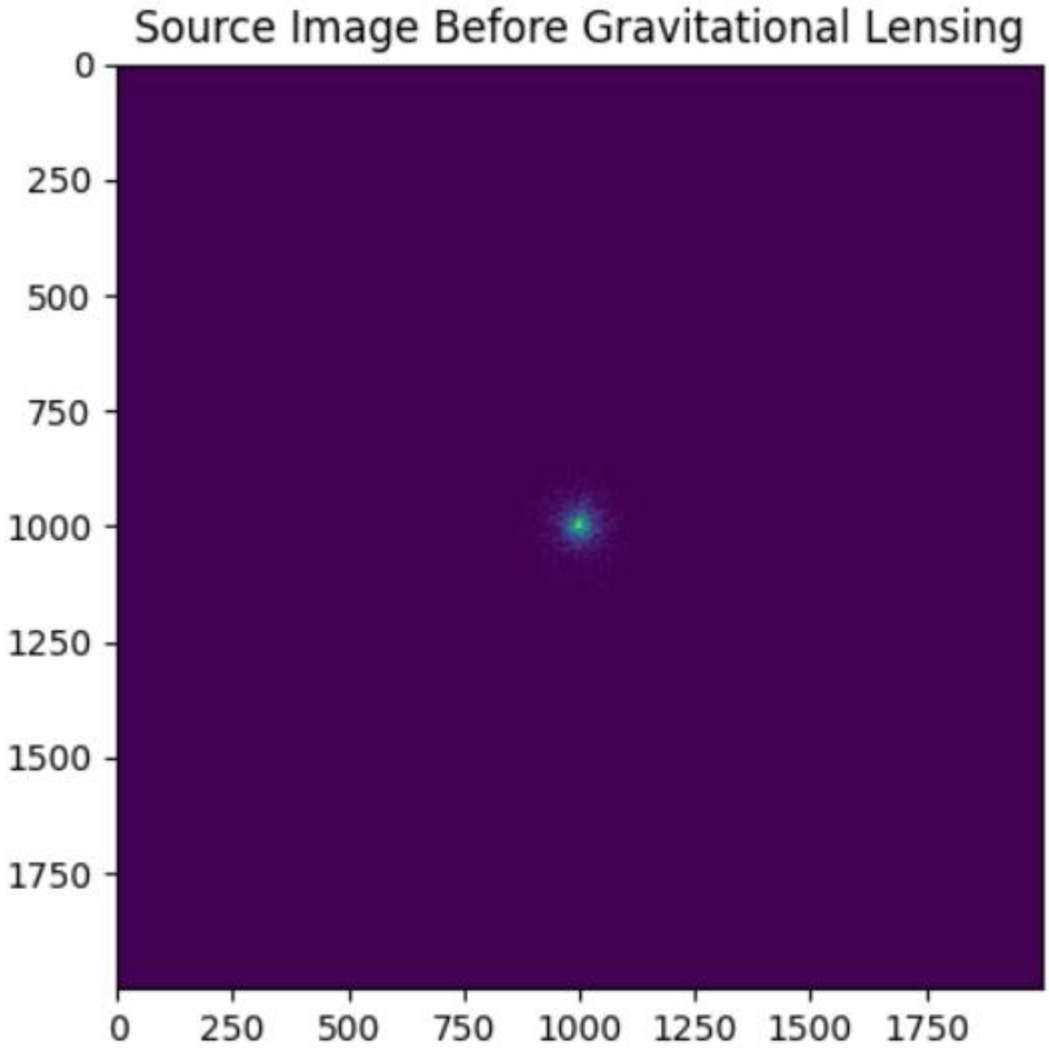

Figure 5. A cluster of galaxies represented by the dispersion of pixels. This is formatted as f = f0exp(-r/a) where the probability of the galaxy decreases as r gets bigger. In this case, r would be defined as the radius from the centre of the cluster. The pixels are 2000x2000, so the resolution of the lensed image produce rc should be better than the prior versions. This cluster is a result of probability so the same cluster will not likely appear again, but it will always be centred around (1000,1000) which is the centre.

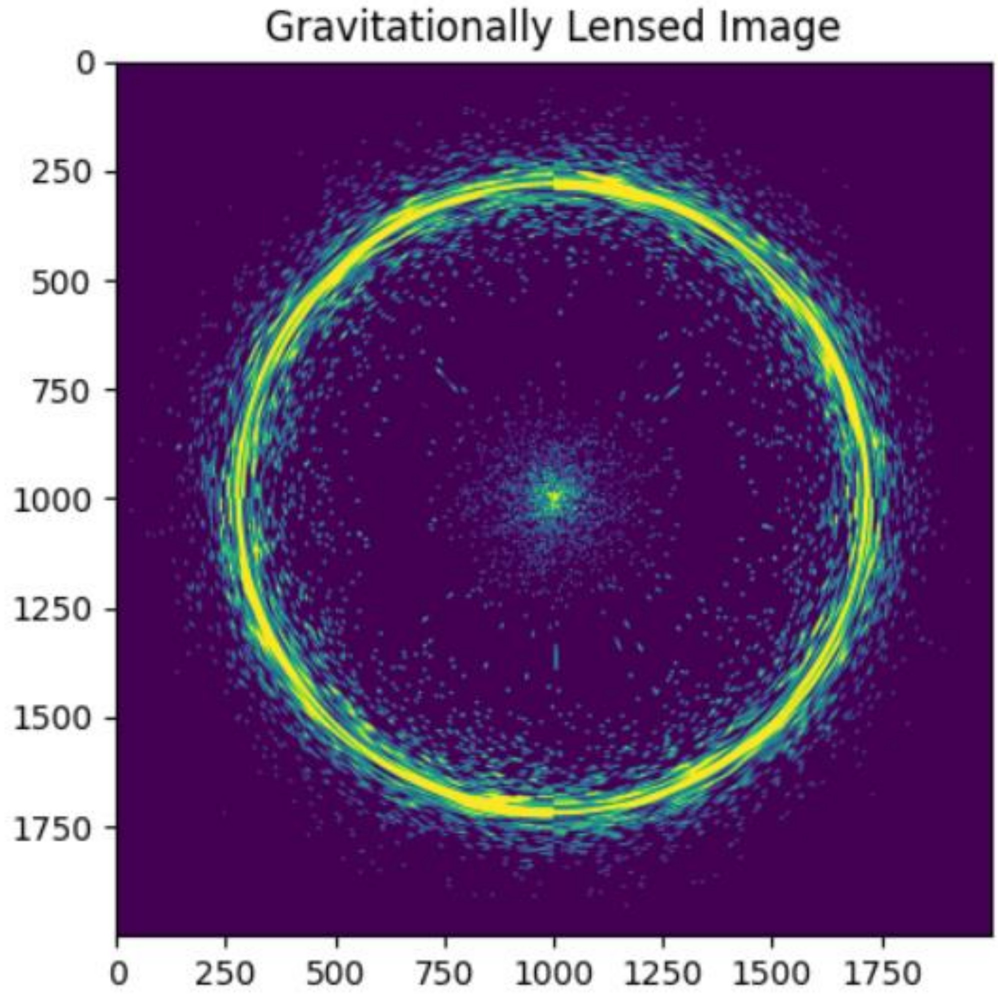

Figure 6. Gravitationally lensed image of figure 5, where Ds = 878, Di=637, Dls= 441, R = 718, rc = 0.695, theoretical R= 0.719, output R = 0.720, percentage diff = 0.139%. The percentage difference for this would differ by iteration, as the galaxy cluster is also randomised by probability. Method of calculating rc to place in can be seen in the code, and the values that were used are as suggested.

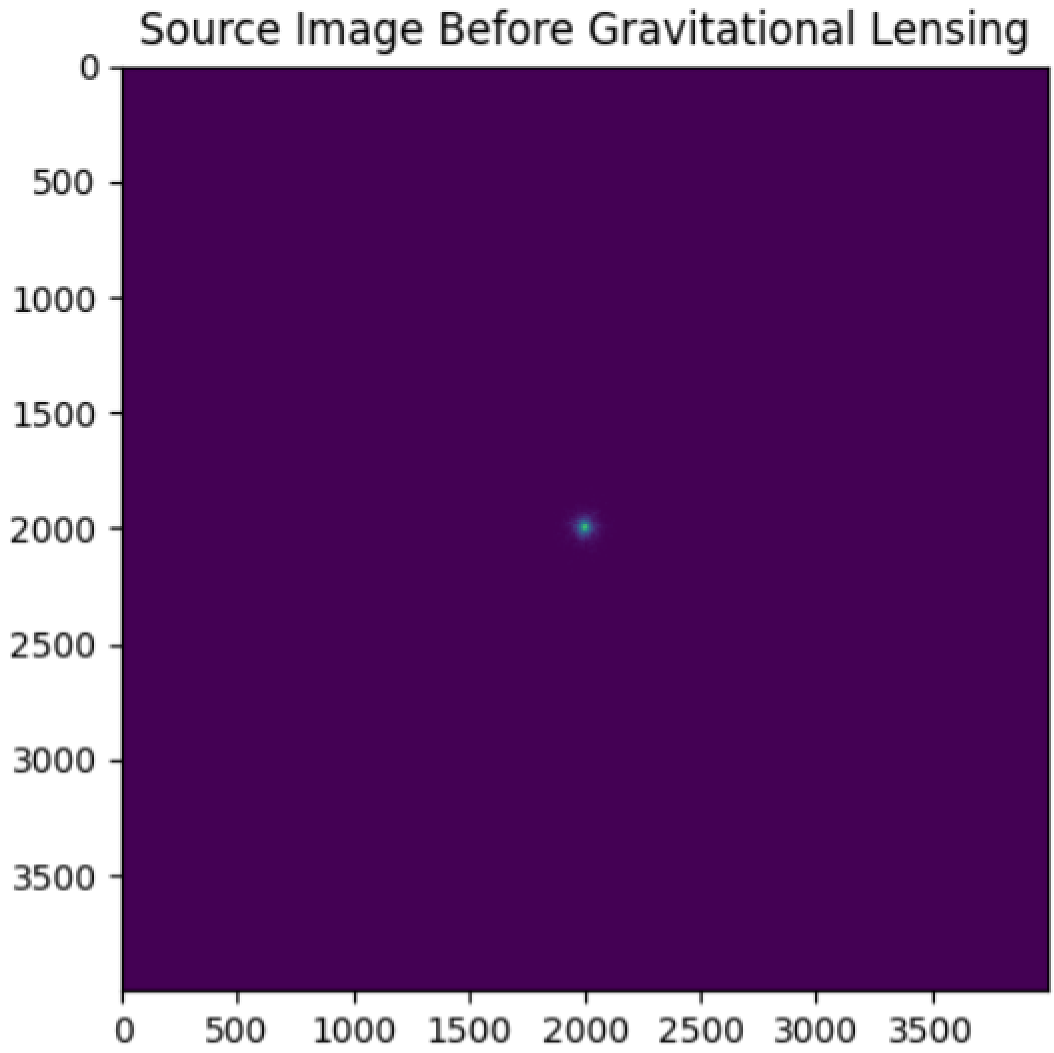

Figure 7. A source image of galaxy cluster with function f = f0exp(-r/a). The pixel resolution is 4000x4000. It is centred around (2000,2000). The probability cluster is the same as those prior source images to figure 7, but not as figure 1.

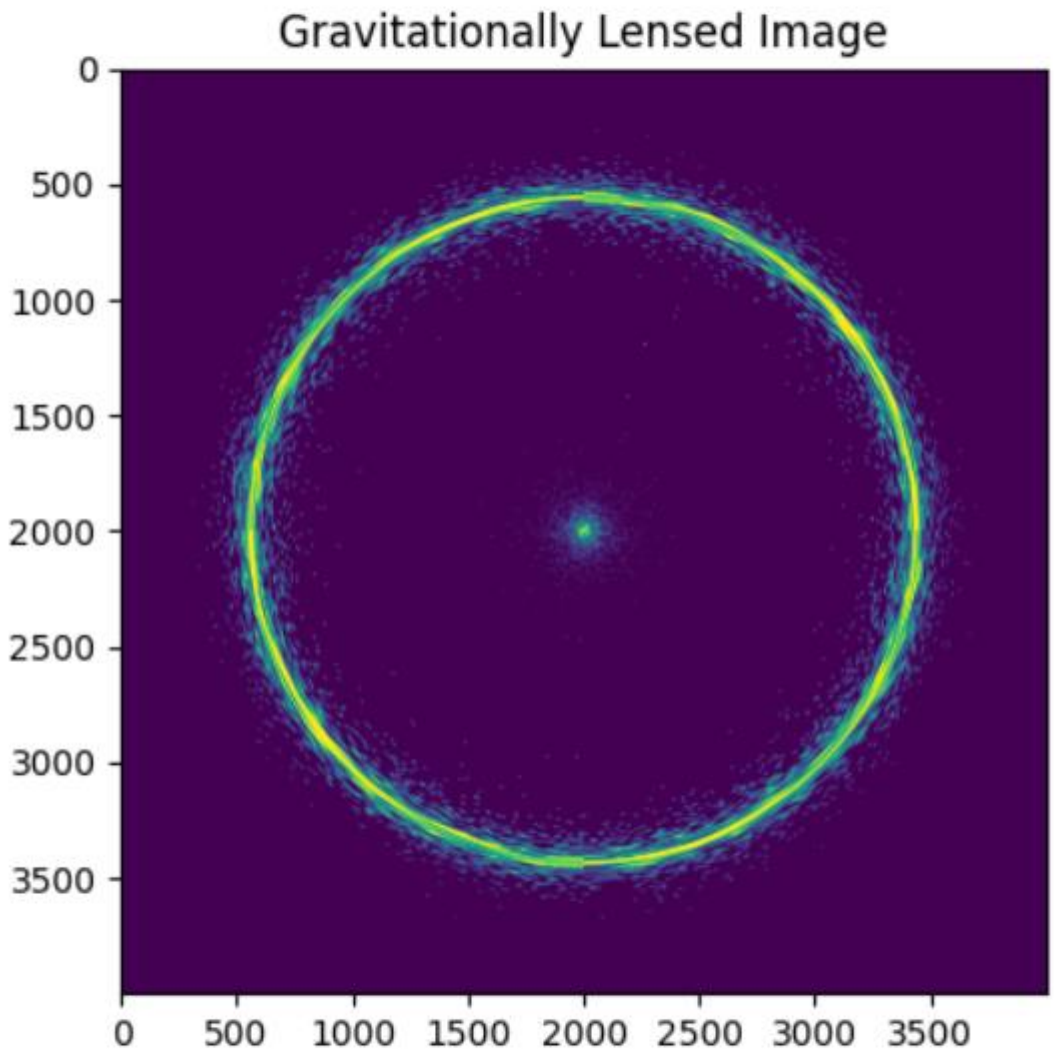

Figure 8. Gravitationally lensed image of figure 7, where Ds = 878, Di=637, Dls= 441, R = 1390, rc = 0.695, theoretical R = 0.719, output R = 0.717, and percentage difference = 0.278%. As was predicted by the hypothesis, the percentage difference of the rc once again decreases as the resolution increased. This is higher than 2000x2000 and is not as expected, perhaps this can be explained by considering that the random distribution may hold higher weight than the pixel resolution at such a high number.

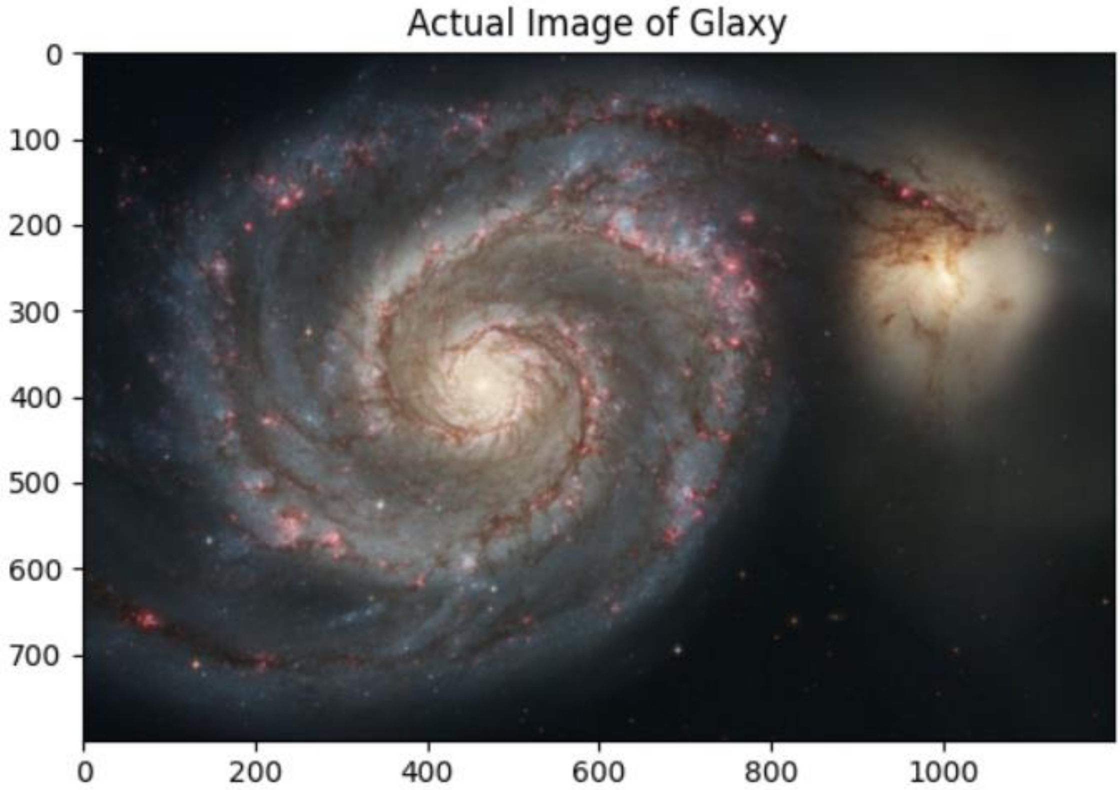

Figure 9. A single galaxy found online [1]. Its dimensions are 1200x800, and the difference between the lengths are 33.3%. A percentage of the difference in length is worked out as the code is optimised for square dimensions, which may explain anomalies.

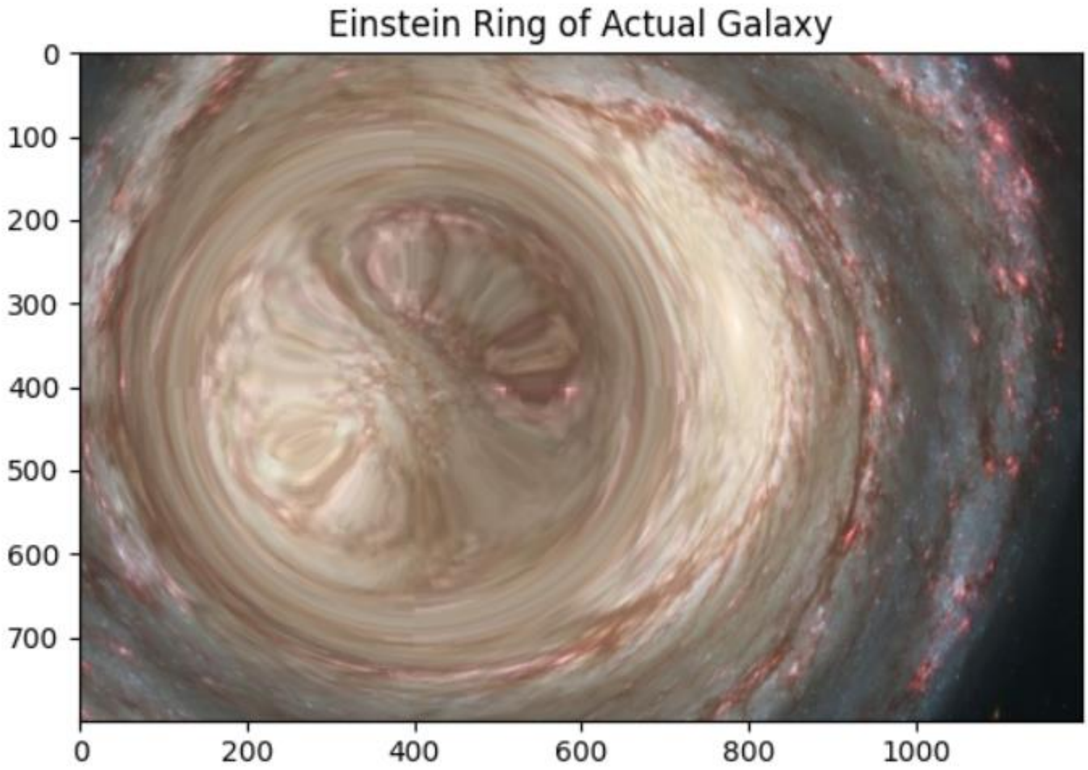

Figure 10. Gravitationally lensed image of a single galaxy which was found from the image of figure 9. The programme’s circle properties cannot be worked out automatically as this is 3D, and the function only works for 2D. Consequently, it is worked out by hand and found to be output R = 0.712, and theoretical R = 0.719 from rc = 0.695. The percentage difference the percentage difference is therefore a small 0.974%.

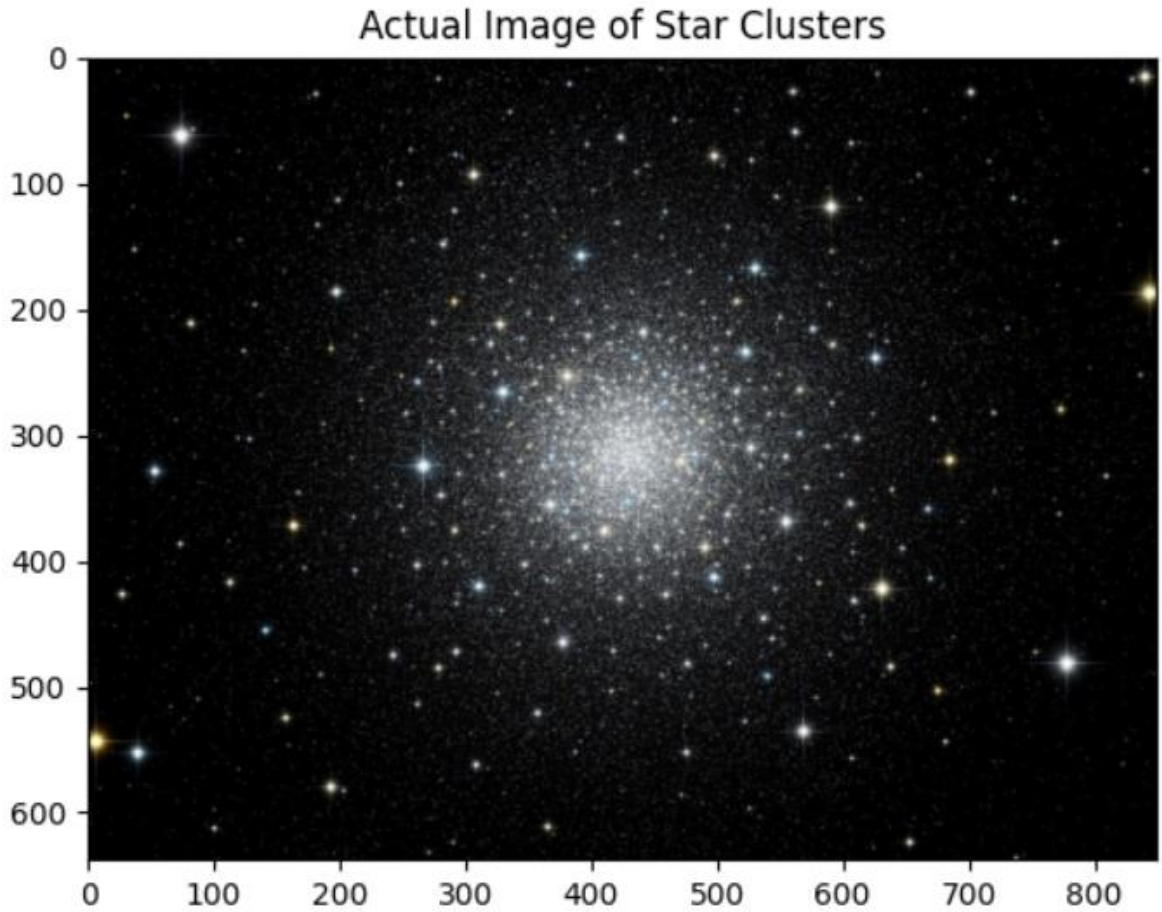

Figure 11. Image of star clusters found from online [2]. This may give insight into how multiple light sources would be distorted by gravitational lensing. The dimension of this picture is given by 850x638, the difference between the lengths are 24.9%.

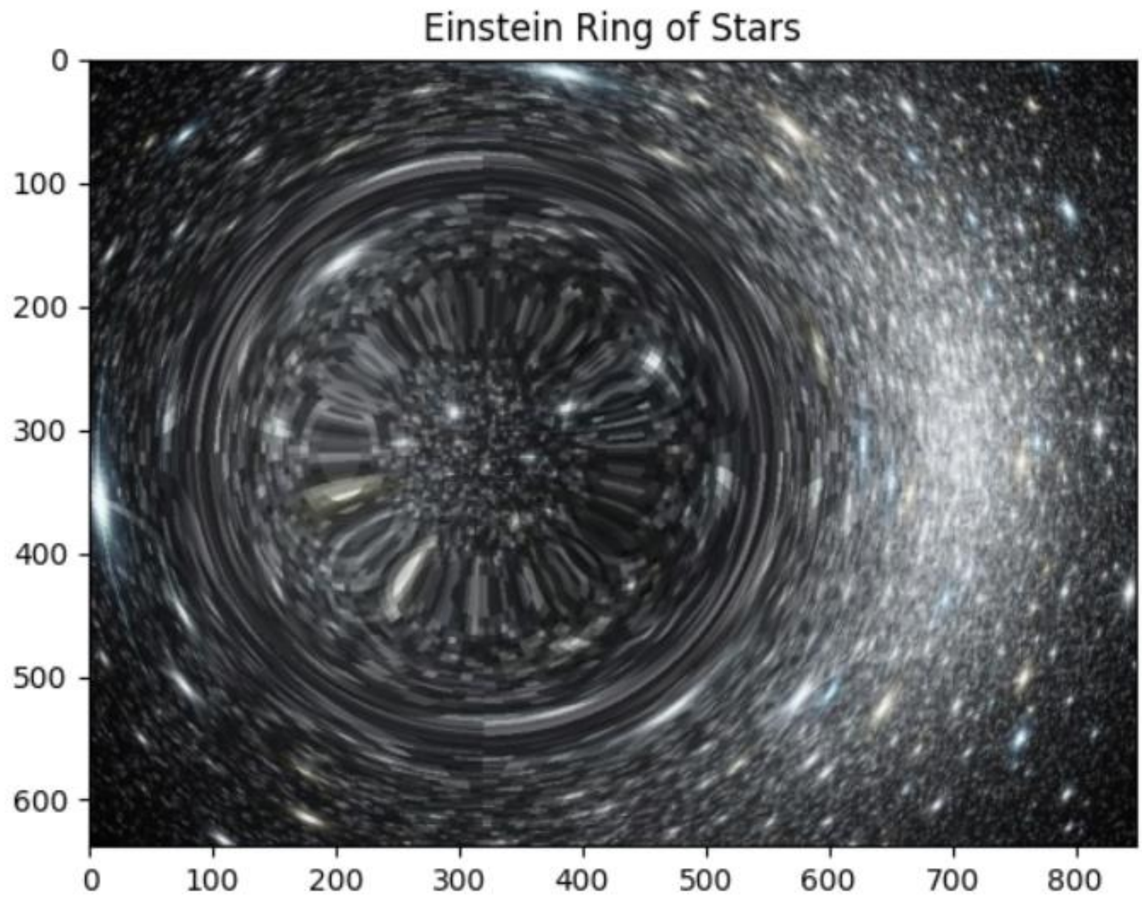

Figure 12. Einstein Ring Found from gravitational lensing of figure 11. Experimental R = 0.572, theoretical R = 0.719, ,rc = 0.695, percentage difference = 20.4%. This again had to be worked out by hand as it is a 3D-coloured picture (RGB), and the error likely arises from its dimensional pixels not being square.

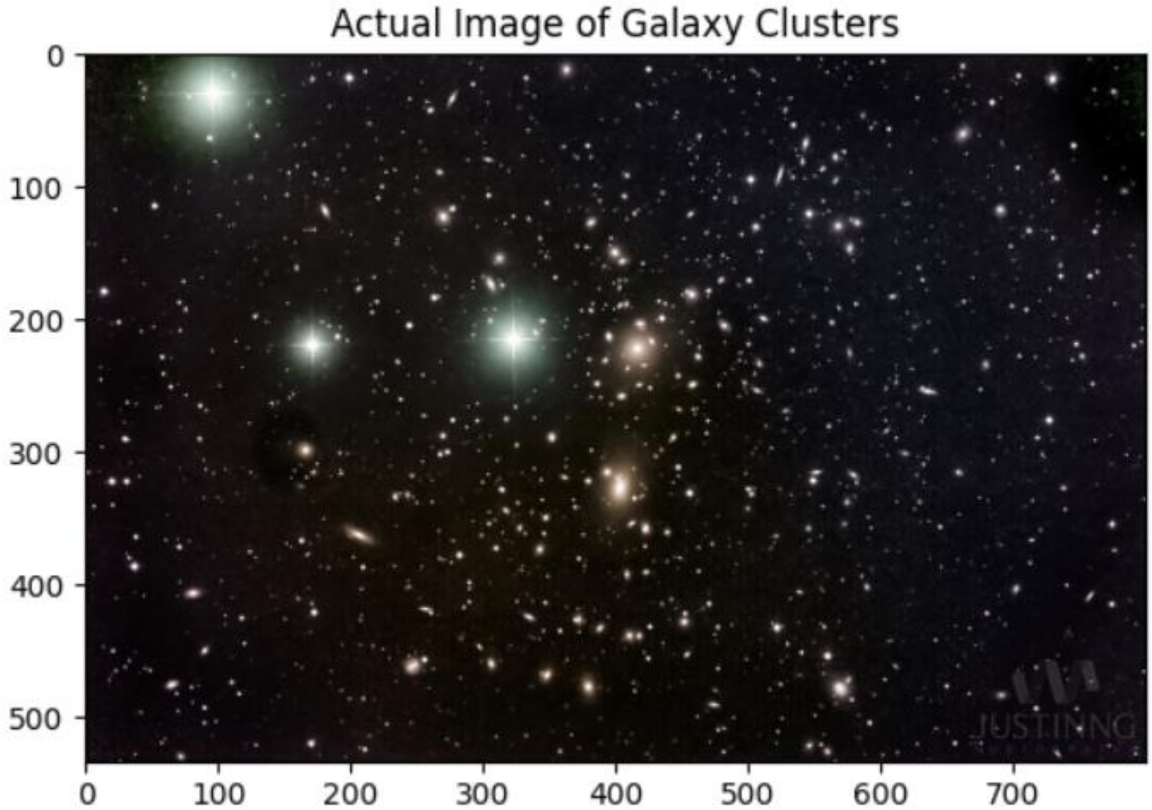

Figure 13. Image found online of a cluster of Galaxies; these will give more insight to how a gravitational lensing may work on multiple light sources. The dimension of the picture is 800x533. The difference between the lengths is 33.3%.

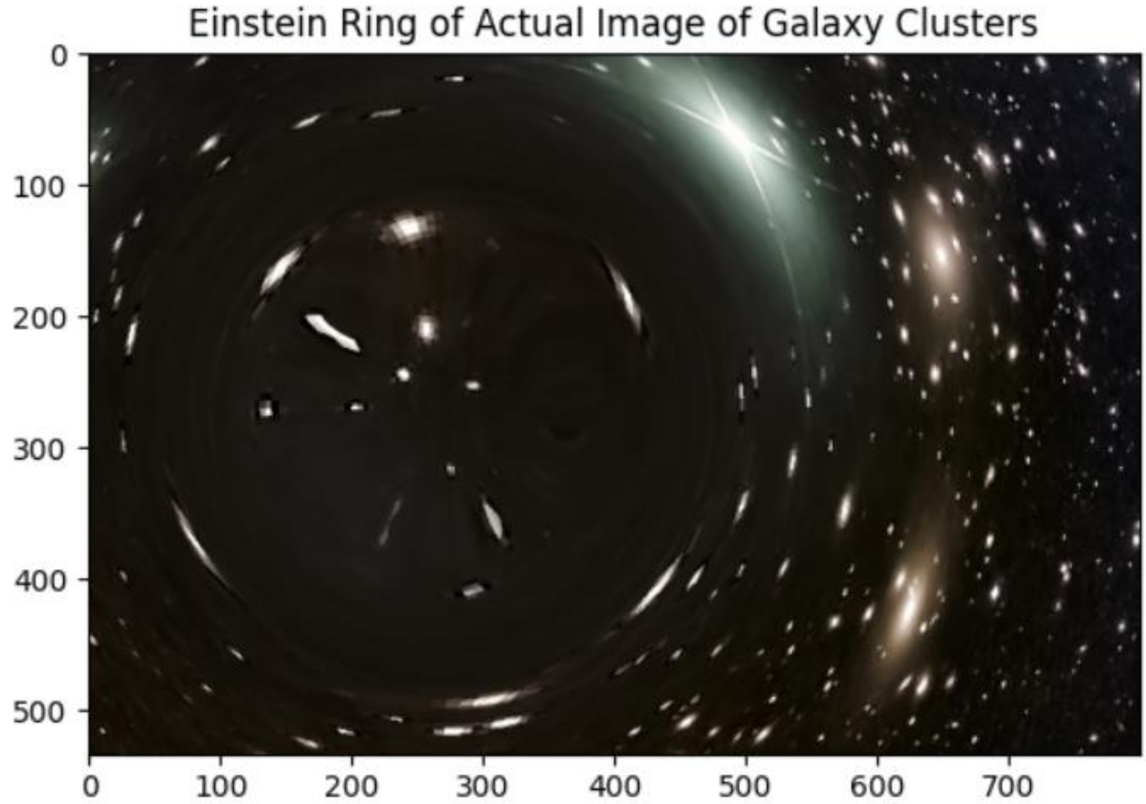

Figure 14. Gravitationally lensed image of figure 13. The experimental R = 0.978, theoretical rc = 0.791, the percentage difference = 36.0%.

Conclusion

From the real-life images, we may conclude that most of the error came from the dimensions of the pictures being unequal. The code was optimised for square dimensions, such as y x y, where y is the dimension of the pixels. However, it would not be clear to conclude that as the difference in the pixel lengths relatively increased the percentage difference of R also increased. It would appear to be more plausible to conclude that as the number of pixels increased it gave a smaller percentage difference of R. This was very clear when looking at the square dimension pictures which are randomly self-generated by code. As we move from (201x201), (2000x2000), and (4000x4000), the percentage difference decreases from 31.8%, 0.139%,0.278% respectively. The slight rise at the end may be explained by random nature of the generation of galaxy clusters in the function.

This may be due to the percentage error being inversely proportional to the sensitivity of our measuring equipment, in this case the number of pixels. Another source of uncertainty may come from the fact that for the coloured images that were from real life situations, R was calculated by eye as the radius could not be measured out precisely. This was not the case for the self-generated source images as the code was able to work with 2D images (black and white or purple and yellow in this case).

References:

[1] https://abrancoalmeida.com/tag/galaxia-whirlpool/, accessed 16/12/2021

[2]https://toppng.com/uploads/preview/stars-galaxy-star-cluster-shining-sparkling-bright-11570075216ggutqsgaf7.jpg, accessed 16/12/2021

[3]https://lmanuela-osorio.medium.com/materia-oscura-no-tan-oscura-1d6438103990, accessed 16/12/2021

[4]Get_Radius function inspired by Stefan van der walt:

https://stackoverflow.com/questions/28281742/fitting-a-circle-to-a-binary-image/28289147, accessed 14/12/2021