Astronomy Lab: Report 10

Kiavash Teymoori 99100585
Alireza Talebi 400109821

Introduction

In this experiment our goal is to stack our pictures and calculate PSF, background magnitude, limiting magnitude, star magnitudes and modeling the background in the V filter.
The equipment used for this experiment was an 8-inch telescope and a Nikon D90 camera and the subject was the IC-4665 open cluster. The experiment was done in 3rd and 4th of Ordibehesht 1404. Images were taken from 11:26 pm - 12:44 am with ISO = 800 and exposure time = 2s

Experiment

preprocessing

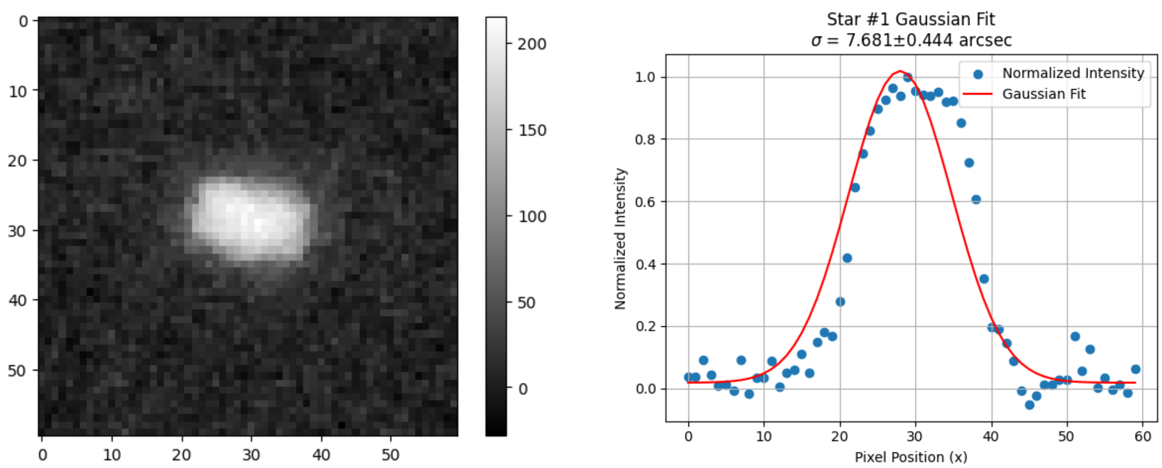
After we made the masterdark and masterflat, we had to align and stack our pictures. In order to do that, I located the same star in every picture and used my shift and align-and-crop-to-average function to align the pictures and crop parts of it that are not present in every picture.
Here is the result:

Figure 1: Stacked picture

PSF

We began by loading the stacked .fit file containing the astronomical image and converted it into a NumPy array. The image dimensions were extracted. Two empty arrays, sigma-hist and delta-sigma-hist, were initialized to store the Gaussian width values and their uncertainties for each selected star.
A custom centroid-finding function was used to compute the center of each star based on weighted averages of pixel intensities.

We then extracted a 60×60 pixel frame around each stars centroid. Since Siril uses an inverted y-axis compared to standard array indexing in Python, we corrected this by adjusting the coordinate transformation during cropping.
We selected a horizontal slice through the star's centroid and normalized the pixel intensity values. This 1D profile was fitted to a Gaussian function using scipy.optimize.curve-fit.
Instead of directly fitting (which could cause division-by-zero errors), we used the parameter "a" and later converted it back to σ. The uncertainty in σ ($\Delta\sigma$) was calculated using error propagation from the covariance matrix returned by the fitting function.



Star 1 Gaussian fit

Figure 2: One of the star images and the respective Gaussian fit

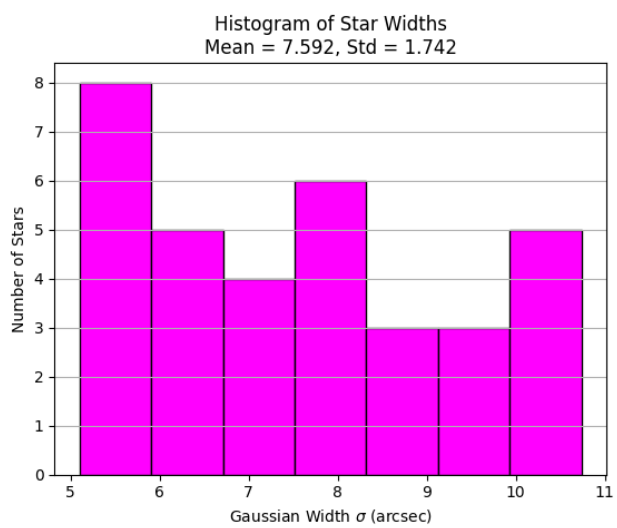


Figure 3: Star Width Histogram

In order to calculate plate scale we can use:

$$Plate\ scale\ (arcsec/pixel) = \frac{206.265 \times pixel\ size\ (\mu m)}{focal\ length\ (mm)}$$

but I did not remember what our focal length was so I selected two stars HD 161621 and HIP 86944, used stellarium to look up their RA (right ascension) and DEC (declination) to calculate plate scale as follows:

$$\theta = arccos(sin(\delta_1)sin(\delta_2) + cos(\delta_1)cos(\delta_2)cos(\alpha_1 - \alpha_2)$$
$$arcsec\ per\ pixel\ =\ stars\ arcsec\ distance\ /\ stars\ pixel\ distance$$

in which δ is DEC(in radians) and α is RA(in radians) and θ is angular seperation (in radians at first).

$$Arc\ seconds\ per\ pixel = 1.132 \pm 0.002$$

Relative Magnitude Calculation

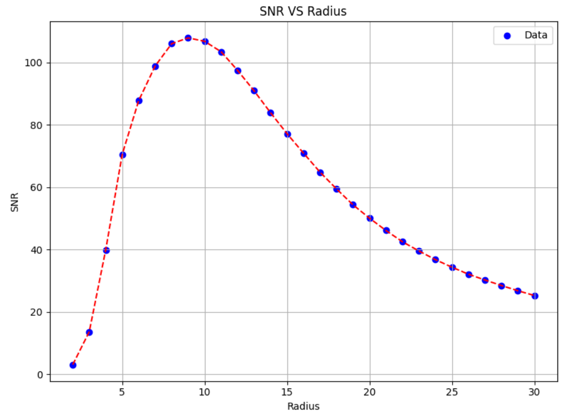
Method

In aperture photometry, we set a circular area which we call the object aperture, on the star to calculate magnitude. then in order to remove noise we select an area outside the previous area (which we set between two and three times the the star radius)
These are the equations we used to calculate magnitude and noise and errors:

$$m = -2.5log(\frac{star\ brightness - sky\ brightness}{exposure\ time} + C$$

$$noise = N\sqrt{noise_s^2ky + noise_r^2ead}$$
$$\frac{\Delta m}{m} = \frac{1.08}{SNR}$$
$$SNR = \frac{Brightness}{noise}$$

In order to find the best radius, we gradually raise radius and calculate SNR (singal to noise ratio) and pick the radius in which SNR is at it's peak.



Reference star

Figure 4: Max SNR method to find best radius

We calculated center of mass to find the center of each star. Here is the resulting picture for reference star with magnitude m = 7.36(assuming from stellarium):

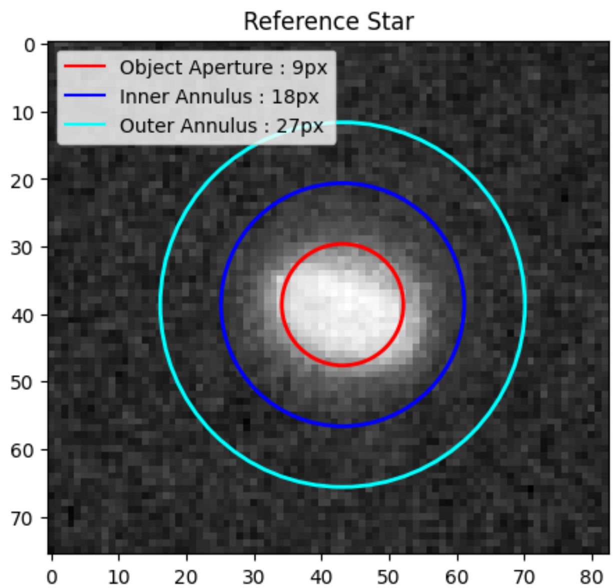


Figure 5: SNR = 107.87

Background Magnitude

The process is similar to how we find stars magnitude relative to a reference star. We selected these four patches and calculated their magnitude, then took the average to calculate the background magnitude.

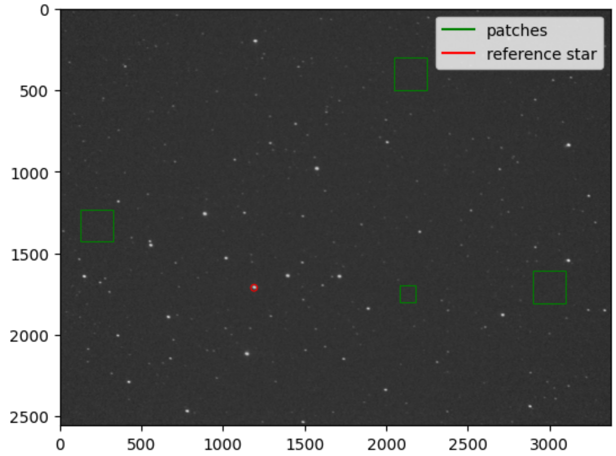


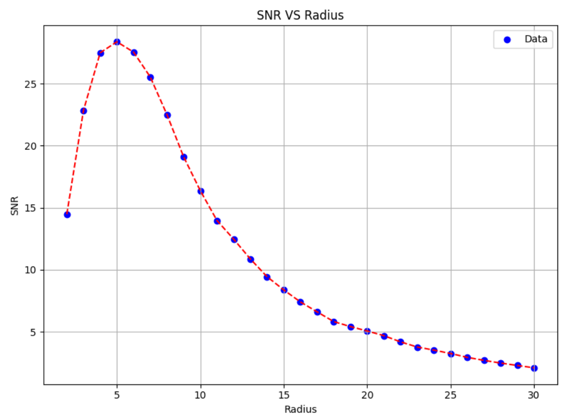
Figure 6: Background areas without stars

Title	*magnitude*
Area 1	15.61
Area 2	15.28
Area 3	15.62
Area 4	16.04
Avg ± Err	15.64 ± 0.27

Table 1: resulting magnitude

Limiting Magnitude

Limiting magnitude is the least amount of magnitude we can measure. I did not want to manually calculate every faint stars magnitude and see which one is lowest because it is not efficient and I may miss some stars. So I wrote a code that detects candidates for stars using thresholding. It loops over all detected bright pixels and slices out a 31 × 31 patch and runs a full SNR and photometry analysis with the functions we already used for relative photometry.
Now we have everything that could be a star in the picture. The problem is for low snrs, there might be noise instead of a star. I manually checked the resulting stars based on snr and it looks like from snr = 15 we can find some super faint stars but i'm not confident on calling them the limit magnitude but I think at around snr = 25, we can confidently differentiate them from noise. So the limiting magnitude is 10.09 ± 0.04



Reference star

Figure 7: Max SNR method to find best radius

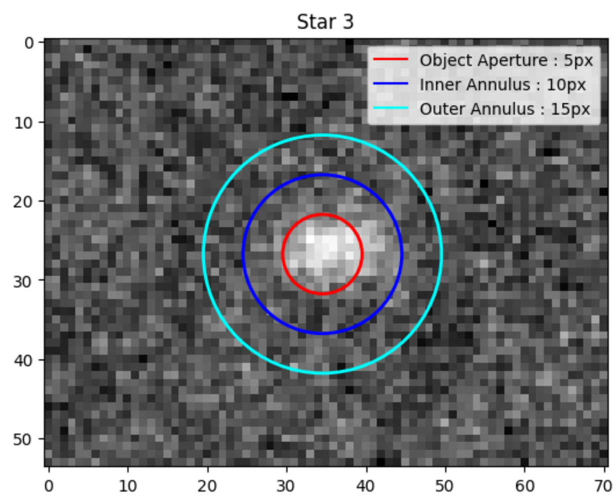
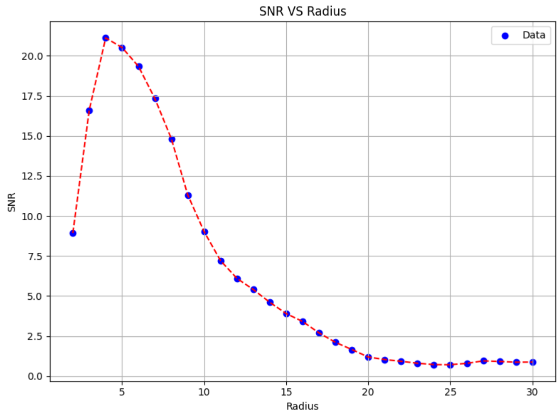


Figure 8: SNR = 28.37, m = 10.09 ± 0.04

here is the picture for a super faint star that i'm not confident on calling it a star (maybe they are the limit magnitude):



Reference star

Figure 9: Max SNR method to find best radius

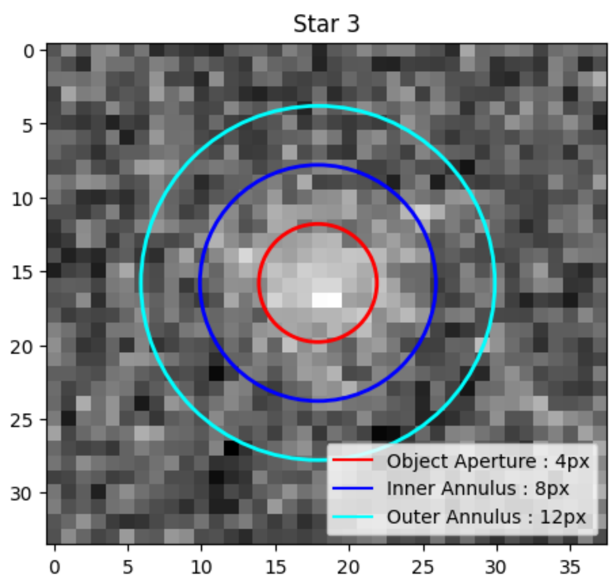


Figure 10: SNR = 21.12, m = 10.89 ± 0.05

Theoretical Limiting Magnitude

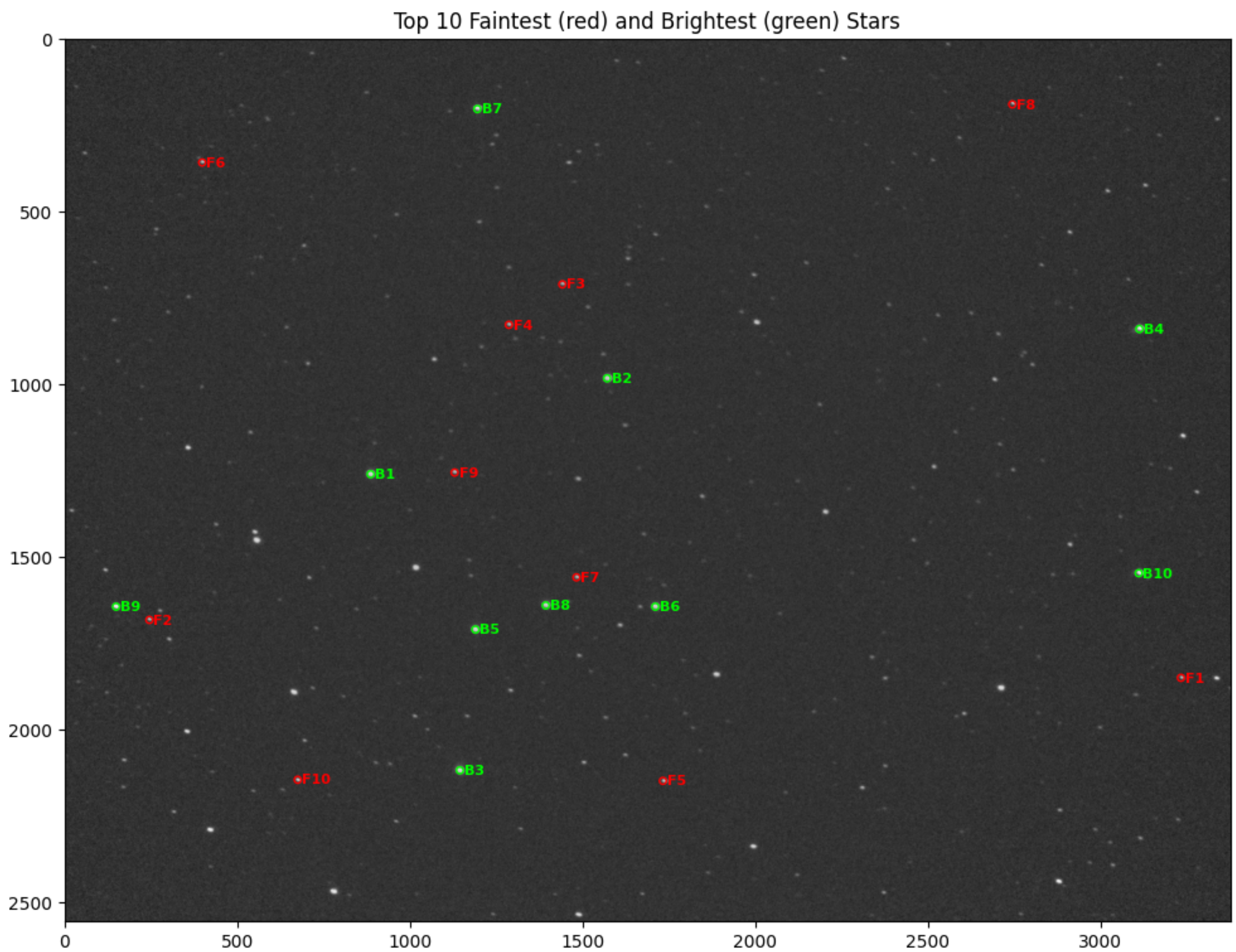
instead of picking the lowest magnitude star to find limiting magnitude, we can theoretically calculate it as follows:

$$F = SNR_min.\sqrt{F/gain + N_{pix}(BPP/gain + ReadNoise^2)}$$
$$m_lim = m_ref - 2.5log_10(\frac{F_lim}{F_ref})$$

since I have the PSF of the picture I can calculate what the magnitude would be for the said PSF, which is 9.747 ± 0.0002 and is pretty close to what we have found in the previous section.

Stars Magnitude

For this part, I used the list of stars that we detected in the previous chapter and for high SNRs (>100), got the top ten brightest and faintest stars (note that these are not as faint as the previous ones since we are using the high SNR stars):



Reference star

Figure 11: image with detected stars

ID	magnitude ± error	SNR	X	Y	Type
F1	8.84± 0.02	70.95	3233.6	1850.7	Faint
F2	8.60± 0.02	70.43	245.2	1682.9	Faint
F3	8.58± 0.02	72.05	1441.3	710.9	Faint
F4	8.58± 0.02	72.35	1286.6	828.0	Faint
F5	8.56± 0.02	72.27	1732.8	2149.2	Faint
F6	8.56± 0.02	72.31	397.9	357.8	Faint
F7	8.55± 0.02	71.92	1482.4	1559.7	Faint
F8	8.52± 0.01	75.09	2744.9	189.4	Faint
F9	8.50± 0.01	76.32	1129.4	1256.0	Faint
F10	8.49± 0.01	76.18	674.8	2146.0	Faint
B1	7.13± 0.01	112.53	886.0	1260.8	Bright
B2	7.16± 0.01	110.65	1571.1	983.1	Bright
B3	7.16± 0.01	108.38	1144.6	2118.7	Bright
B4	7.22± 0.01	119.56	3113.2	840.3	Bright
B5	7.35± 0.01	108.40	1189.7	1710.8	Bright
B6	7.36± 0.01	108.05	1710.8	1644.6	Bright
B7	7.36± 0.01	109.34	1195.2	203.0	Bright
B8	7.36± 0.01	107.95	1393.3	1641.1	Bright
B9	7.46± 0.01	115.83	148.6	1644.6	Bright
B10	7.48± 0.01	114.39	3111.1	1547.5	Bright

Table 2: resulting magnitude

Background Modeling

for this part I selected a lot of patches without stars and fit a 2D rectangular plane(to not make things complicated) to it. The result is as follows:

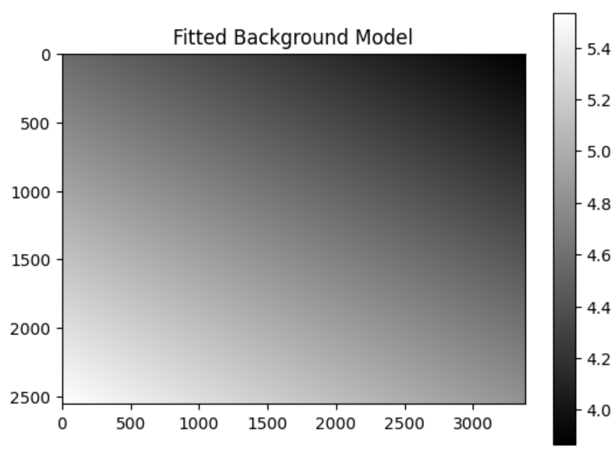


Figure 12: Background model

It is interesting that it made a gradiant shade like that.

Although we will not see a significant impact but if we subtract our original image it would become:

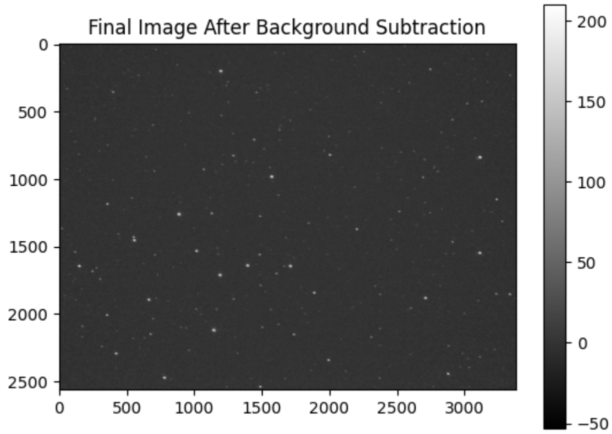


Figure 13: Original image after subtracting background model