Deep learning final project:
CNNs for galaxy classification

by
Luis Mario Núñez Beltrán
& María Margarita Reyes Sierra

03/11/2023

1 Introduction

Galaxy classification is a complex and evolving field of research whose results contribute to our understanding of galaxy evolution and cosmology. It is a systematic approach used in astrophysics to categorize galaxies based on their apparent morphology and structural features.

The classification of galaxies provides a foundation to investigate their intrinsic properties, which can be used to learn more about their formation and evolution.

As astronomical instruments evolve and amass an unprecedented volume of data, scientists are faced with huge galaxy samples and visual classification of them is becoming impossible.

2 Statement of the problem

Galaxies are most commonly classified according to their shape, brightness, and composition. The most common classification scheme was developed by Edwin Hubble in the 1920s. Hubble divided galaxies into four main types: spirals, ellipticals, lenticulars and irregulars (See Figure 1).

Spiral galaxies have a disk-shaped structure with spiral arms that extend from the center. These galaxies host a range of stellar populations, spanning from young, bluer stars in their spiral arms to older, redder stars in their central bulges.

Elliptical galaxies have a smooth, ellipsoidal shape. While they can vary in size and mass, the largest known galaxies are elliptical. They predominantly contain older stars and generally lack the younger, hot stars characteristic of ongoing star formation seen in spiral galaxies

Lenticular galaxies have a transitional form that shares features with both spiral and elliptical galaxies. They have a disk-like structure, similar to spiral galaxies, but lack the prominent spiral arms. In terms of stellar content, they mostly contain older stars and show little to no ongoing star formation. Their appearance is characterized by a smooth, lens-like shape, from which their name is derived.

Finally, irregular galaxies can be defined as those whose appearance does not fit into any of the other categories.

The task of manually classifying galaxies comes with a set of challenges. Among the main ones are:

The volume of observable galaxies. Since with the advent of advanced telescopes, the number of galaxies that can be observed and recorded has grown exponentially. Manually classifying these large galaxy samples is becoming an untenable endeavor.

The subjective nature of galaxy classification by observers can lead to inconsistencies; two different observers might classify the same galaxy differently based on their interpretations of its features.

The quality of the observational data. Distant galaxies can be challenging to classify accurately, as their full morphology might not be discernible.

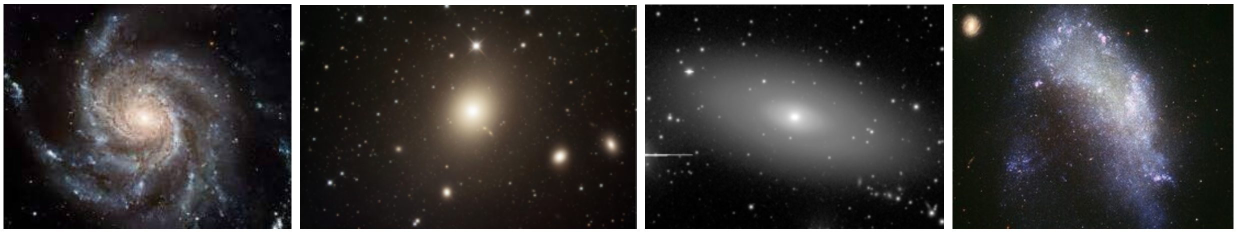
It has then become imperative to develop automated and quantitative approaches for galaxy classification. Leveraging algorithms and machine learning, these systems can efficiently process vast datasets to classify galaxies, often outpacing and outperforming manual efforts

3 Galaxy Zoo: The Project

In July 2007, astronomers from Oxford University had in their possession a data set of ~ 1 million galaxies imaged by the Sloan Digital Sky Survey (SDSS). The galaxies in this data set needed to have their visual morphologies classified. With so many galaxies, it would have taken an individual a thousand lifetimes to classify all of them. Instead, Galaxy Zoo was born.

The Galaxy Zoo project is an ongoing citizen science initiative that enlists the help of the general public to classify galaxies based on their visual morphology. It's one of the pioneering projects in the field of citizen science, where non-professionals collaborate with scientists on real scientific research.

The project was launched in July 2007. Its initial goal was to classify about one million images of galaxies from the SDSS.



(a) Spiral galaxy. (b) Elliptical galaxy. (c) Lenticular galaxy. (d) Irregular galaxy.

Figure 1: Main types of galaxies, as categorized by Edwin Hubble and subsequent observations.

The response was overwhelming:

It was initially assumed that despite outsourcing the work to thousands in the general public, it would still take years for all of the images to be classified. Within the first 24 hours of launch, Galaxy Zoo founders were stunned to be receiving nearly 70,000 classifications an hour. In the end, more than 50 million classifications were received by the project during its first year, contributed by more than 150,000 people.

Online participants are presented with galaxy images and are asked to classify them based on their shapes. The basic categories included spiral, elliptical, and irregular.

Given the success of the first project, Galaxy Zoo 2 (GZ2) was launched in February 2009.

GZ2 aimed for more detailed morphological classifications; users were asked to about bars, the shape of the galaxy's core, the presence of spiral arms, their number, and more. The project continued to use the SDSS, but focused on a subset of the most well-resolved images that had already been classified in GZ1 (this often implies that the galaxies in question are at lower redshifts).

4 Deep learning for galaxy classification

In recent years, the field of astrophysics has seen a growing intersection with machine learning, as part of this intersection we have the utilization of convolutional neural networks (CNNs) for various tasks. Prior research on galaxy classification has shown the potential of CNNs in achieving high accuracy rates, outperforming traditional methods, and offering scalability for handling vast datasets.

This project will specifically delve into the application CNNs for galaxy classification. Building on the foundational work in both astrophysics and machine learning, our aim is to harness the power of CNNs to categorize galaxies based on their distinct morphological attributes. Our project employs the data from GZ2, which is the cornerstone for the Kaggle challenge: Galaxy Zoo - The Galaxy Challenge.

5 Data Description

The data we are provided with (and which we have already collected) are:

- **Training images:** Around 62,000 JPG images, which have been extracted from FITS frames showcasing galaxies as observed by the SDSS. Each training image is accompanied by a probability distribution (ground-truth vector of labels), grounded in human visual classifications.

- **Testing images:** Close to 80,000 JPG images, similarly derived from FITS frames of galaxies captured by the SDSS. Our task is to assign probability distributions for each of these images.

5.1 Image acquisition procedure

It is worth repeating that the source of the JPG images in the GZ2 dataset is the SDSS.

The primary mirror of the SDSS telescope at Apache Point Observatory (New Mexico, USA) is 2.5 meters in diameter (Figure 3).

The SDSS imaging camera is designed to capture light in five different optical bands (u, g, r, i, and z). The distinct rows of CCDs in the camera are equipped with filters specific to each of these bands, allowing the camera to simultaneously observe the sky in these bands as the telescope scans across it. The gaps or spaces between these CCD blocks are inherent to the camera's design. When the telescope scans the sky, it moves in a way that ensures the gaps are covered in subsequent observations, so no part of the sky is missed. The imaging camera of the SDSS is composed of 30 CCDs, each of these CCDs contains 2048×2048 pixels.

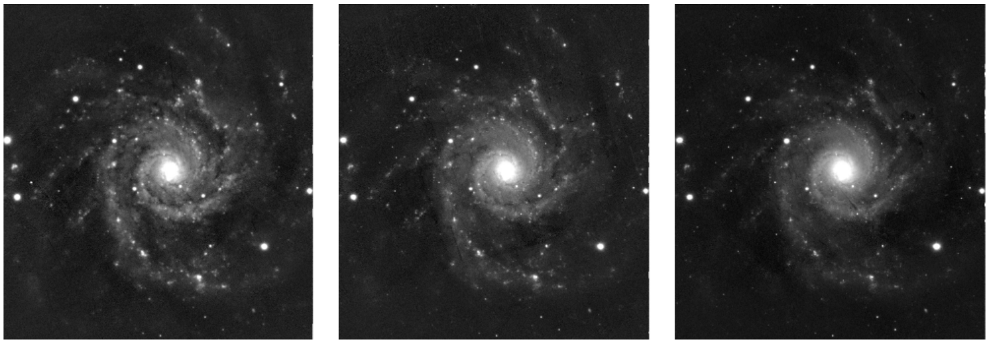


Figure 2: The same spiral galaxy observed with three different wavelength-filters; g (left), r (center), and i (right)

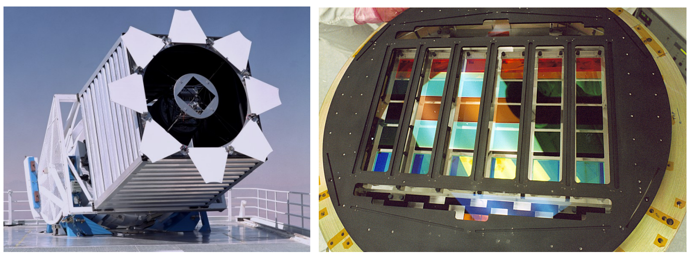


Figure 3: Left: The telescope used to compile the SDSS. Right: The CCD array employed by the SDSS telescope in imaging tasks

When the telescope observes a given galaxy, the result is five images (one per filter) in FITS format, similar to the ones in Figure 2. Those grayscale images indicate the amount of light the CCD captured in each pixel.

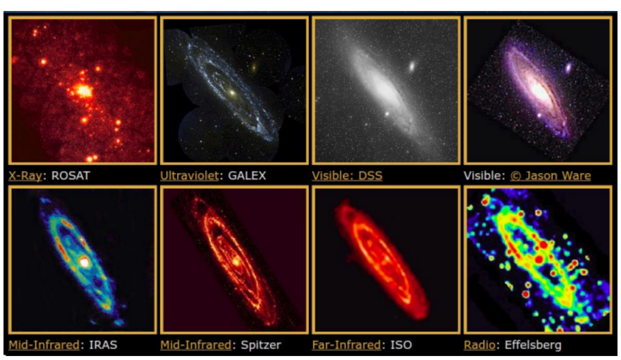


Figure 4: The Andromeda galaxy (M31) observed in different wavelenghts.

One of the benefits of observing a galaxy in different wavelengths is that each one delivers different information. That can already be seen from the different appearances of the galaxy in Figure 2 across different filters. A particularly striking example is shown in Figure 4 wherein the Andromeda galaxy (M31) is mapped from X-rays to radio waves.

For each galaxy in the GZ2 sample, three of those FITS images; the g, r, and i bands, were combined by the SDSS to generate 424 × 424 RGB-color JPG images. Figure 5 exemplifies this.

For this project, we will be utilizing the JPG images provided by the SDSS as our data set, as opposed to the FITS files from which they were composed. The rationale for this decision is multifold, but it is primarily driven by the time-consuming nature of the data reduction process required to work with FITS files, in comparison to a more accessible format like JPG.

The creation of the JPG images from raw FITS files from an astronomical survey is not a straightforward task; it involves several intricate steps, each of which is time-intensive and computationally demanding. Some examples of the data reduction processes include:

Calibration: This involves correcting the raw FITS images for any distortions or biases introduced by the imaging equipment. It typically requires the subtraction of dark frames to remove dark current noise, and flat-fielding to correct for variations in the sensitivity of the detector pixels.

Background estimation and subtraction: FITS images often contain background noise from the sky or instrumental sources, which must be carefully subtracted to enhance the visibility of the galaxy itself.

Intensity Scaling and Color Mapping: The dynamic range of FITS images is typically much greater than can be displayed in a standard JPEG image. Converting the high dynamic range data into a visual format involves scaling the intensity levels and mapping them to color channels, which is a complex process that often requires manual tuning to produce aesthetically pleasing and scientifically informative images.

Given the extensive nature of these tasks, and considering the constraints of time and resources inherent to a class project, we have decided to use the pre-processed JPG images. That decision allows us to focus on the deep learning aspect of our project rather than the preprocessing of the images.

The galaxy features that are pertinent to our project's objectives — such as spiral arm structure, bar classification, and the presence of galactic bulges — remain discernible and analyzable within the JPG format

Nevertheless, to foster a better understanding of the procedures involved in data reduction, we have included an appendix detailing some of the most common data reduction steps for raw FITS files.

Figure 5: Three galaxies in the GZ2 sample

5.2 Ground-truth labels

Figure 6 contains the classification procedure that resulted in the ground-truth labels for the training data. That is, for each image, the volunteer was asked to go through these 11 "tasks", each task consists of assigning a response to the corresponding questions.

The weighted voting system used in the Galaxy Zoo project to classify galaxies:

Task	Question	Responses	Next
01	*Is the galaxy simply smooth and rounded, with no sign of a disk?*	smooth	07
		features or disk	02
		star or artifact	**end**
02	*Could this be a disk viewed edge-on?*	yes	09
		no	03
03	*Is there a sign of a bar feature through the centre of the galaxy?*	yes	04
		no	04
04	*Is there any sign of a spiral arm pattern?*	yes	10
		no	05
05	*How prominent is the central bulge, compared with the rest of the galaxy?*	no bulge	06
		just noticeable	06
		obvious	06
		dominant	06
06	*Is there anything odd?*	yes	08
		no	**end**
07	*How rounded is it?*	completely round	06
		in between	06
		cigar-shaped	06
08	*Is the odd feature a ring, or is the galaxy disturbed or irregular?*	ring	**end**
		lens or arc	**end**
		disturbed	**end**
		irregular	**end**
		other	**end**
		merger	**end**
		dust lane	**end**
09	*Does the galaxy have a bulge at its centre? If so, what shape?*	rounded	06
		boxy	06
		no bulge	06
10	*How tightly wound do the spiral arms appear?*	tight	11
		medium	11
		loose	11
11	*How many spiral arms are there?*	1	05
		2	05
		3	05
		4	05
		more than four	05
		can't tell	05

Figure 6: The GZ2 decision tree, comprising 11 tasks an 37 responses. The "task" number is an abbreviation only and does not necessarily represent the order of the task within the decision tree. The text "Question" and "Responses" are displayed to volunteers during classification. "Next" gives the subsequent task for the chosen response.

- **Initial Classification:**

  Users classify galaxies into primary categories (smooth, features/disk, star/artifact). The results give initial probabilities for each category. These probabilities sum to 1.0 for each galaxy since each galaxy is classified into one of these primary categories by every user.

  For example, if for a given galaxy:

  - 80% of users think the galaxy is smooth, then Class1.1 = 0.80

  - 15% think it has features/disk, then Class1.2 = 0.15

  - 5% think it's a star or artifact, then Class1.3 = 0.05

  therefore, the first three entries of the ground-truth vector for that galaxy will be (0.80, 0.15, 0.05,....)

- **Subsequent Classification:**

For each primary classification (like "smooth"), additional questions pertain to more specific morphological features. The probabilities derived from these questions are multiplied by the probability of the initial classification to determine the probability for that sub-classification.

Continuing with the example above:

- Of the 80% who thought it is smooth: If 50% think it's completely round, then Class 7.1 = 0.80 * 0.50 = 0.40

If 25% think it's in-between, then Class 7.2 = 0.80 * 0.25 = 0.20

If 25% think it's cigar-shaped, then Class 7.3 = 0.80 * 0.25 = 0.20

Thus, the ground-truth vector for this galaxy will also contain those entries in their corresponding indices: (..., 0.40, 0.20, 0.20,...)

For each galaxy, the result is a vector of ground-truth labels with 37 entries.

The goal of the Kaggle challenge is to build a deep-learning classifier capable of predicting the ground-truth vector for the galaxies in the test set.

5.3 Performance evaluation

The Kaggle challenge asks competitors to evaluate the performance of the models over the training data with the RMSE:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(p_i - a_i)^2}$$

Where:

- N is the number of galaxies times the total number of responses. This represents the total number of data points (or predictions) being evaluated.

- p_i is the predicted value for the i-th data point.

- a_i is the actual value for the ith data point.

ALL OF THE FOLLOWING IS ADDITIONAL STUFF. WE MAY USE SOME OF IT IF SOMEONE HAS RELATED QUESTIONS. IT IS NOT NECESSARY FOR THE PROJECT, WE DO NOT USE IT, I PLAN TO PUT THIS IN AN APPENDIX GIVEN THAT THIS IS RELATED TO FITS FILES

RGB from FITS

Combining FITS images from three different wavebands to create a composed RGB image is a common practice in astronomy. Here's how it's typically done:

Select Three Wavebands:

Choose three wavebands or filters corresponding to Red (R), Green (G), and Blue (B). In many astronomical datasets, these might not directly map to the visual RGB, but you can use the images from three filters and assign them to RGB channels as you see fit. For instance, in SDSS, you might use the 'i' band for Red, the 'r' band for Green, and the 'g' band for Blue.

This consists of using image processing software or programming tools (like Python's Astropy and Matplotlib libraries) to assign each processed (data reduced already, more on this later) FITS image to an RGB channel:

- R channel: Image from the reddest waveband

- G channel: Image from the intermediate waveband

- B channel: Image from the bluest wamveband

After the initial combination, you might need to adjust the balance between channels to get a natural or scientifically informative look. This can involve tweaking the color balance, and brightness/contrast of the composed image, among others.

But before we can combine the images, we need to "process them". Surveys like the SDSS do it for us, so, we do not have to do it ourselves, which would take a very long time. I next outline some of the things that may be good to know about this processing.

Image Processing:

Adjusting the contrast

Adjusting contrast in an image means modifying the difference between the darkest and the lightest areas to make them either stand out or blend in.

At its core, changing the contrast of an image is no more than applying a transformation to the pixel values of that image. The goal of this transformation is to modify the range and distribution of pixel intensities, thus altering the perceived contrast.

Here are a few methods and the transformations commonly applied:

Linear Stretch

Mathematical representation:

$$\text{new\_pixel\_value} = \frac{\text{pixel\_value} - \text{min\_value}}{\text{max\_value} - \text{min\_value}} \times 255$$

Non-linear Stretch:

For example, a logarithmic transformation might be expressed as:

$$\text{new\_pixel\_value} = c \times \log(1 + \text{pixel\_value})$$

Where c is a constant.

Gamma Correction:

$$\text{new\_pixel\_value} = \text{pixel\_value}^{\gamma}$$

Where γ is a value that dictates the level of correction.

The exact transformation used depends on the desired outcome.

Notice that When you adjust the contrast of a gray-scale image, you might not utilize the full range of the 8 bits per pixel (0-255).

High Contrast:

- The differences between the darkest and the lightest areas of the image are pronounced.

- Can sometimes result in loss of detail in very dark or very light areas (e.g the stretching can push the darkest tones towards pure black and the lightest tones towards pure white. When this happens, any subtle variations or details in those regions can be lost because multiple tones might get clamped to either the maximum or minimum values.).

DS9 and contrast

When you use DS9 and set the scale to "linear" with a clipping at "99.5%", you're instructing DS9 to use a linear stretch for displaying the image but to clip or saturate pixel values beyond the 99.5th percentile.

With the "99.5%" setting, DS9 calculates the pixel value that corresponds to the 99.5th percentile of all pixel values in the image. Any pixel values above this threshold are set to this 99.5th percentile value. This method helps in avoiding the influence of extreme pixel values (like outliers or cosmic ray hits) that might otherwise skew the contrast of the image. For the "99.5%" setting, we clip or saturate 0.5% of the pixel values on both the lower and upper ends.

AND WHAT IF INSTEAD OF "99.5%" I CHOOSE "MIN MAX"?

DS9 will scan the entire image to find the minimum pixel value ('Min') and the maximum pixel value ('Max').

With the "Min Max" option:

- The 'Min' value is displayed as black.

- The 'Max' value is displayed as white.

- All the other pixel values in the image are linearly scaled between these two extremes.

POTENTIAL DISADVANTAGES OF MIN MAX

The "Min Max" method might sometimes result in a less optimal display if there are extreme outliers. For instance, if there's a single super-bright pixel (due to a cosmic ray hit, an error, etc.), it might cause the rest of the image to appear too dark because most of the dynamic range of the display gets used up by that outlier. On the other hand, using a percentile option like "99.5%" would clip such outliers, providing a more balanced view of the majority of the image's content.

IF I KEEP THE RIGHT BOTTOM OF MY MOUSE PRESSED IN DS9, I SEE THAT THE IMAGE GETS BRIGHTER AND DARKER, WHAT AM I DOING WITH THIS?

In DS9, when you click and drag with the right mouse button, you're adjusting the display contrast and brightness interactively.

Vertical Movement:

- Dragging the mouse up and down adjusts the contrast of the image.

- Moving the mouse upward increases the contrast, making the differences between pixel values more pronounced.

- Moving the mouse downward decreases the contrast, making the differences between pixel values less pronounced or more subtle.

Horizontal Movement:

- Dragging the mouse left and right adjusts the brightness (or bias) of the image.

- Moving the mouse to the right makes the image brighter.

- Moving the mouse to the left makes the image darker.

When you choose "linear, 99.5%" in DS9:

You establish a baseline linear mapping of the pixel values based on the data's range, excluding the lowest and highest 0.5% to set the black and white points respectively. This creates a linear transformation based on those cutoffs. Then, by pressing and dragging with the right mouse button: you can fine-tune the slope (contrast) and position (brightness) of that transformation.

So, the initial "linear, 99.5%" setting determines an initial transformation, and the interactive adjustments with the right mouse button allow you to modify this transformation to achieve the desired visual outcome. You're essentially interactively tweaking the display representation while keeping the underlying data unchanged.

Figure 7: Low contrast vs High contrast.

Tweaking color balance

This is usually done after you already tweaked both the contrast and bias of each independent channel.

After aligning and combining the channels into an RGB image, you might then tweak the color balance to ensure the combined image has a desired or accurate color representation. For example, if an RGB image appears too "blue," you might enhance the red and green channels relative to the blue to achieve a more neutral color balance.

Here is an example of how this could be done:

The following technique is sometimes referred to as "color matrix transformations" or "color grading using matrices." (the use of a matrix for color transformations is not the only way to adjust colors).

To adjust the color balance of an image, you can use a 3x3 matrix to transform the RGB values of every pixel in the image. The idea is that each new color channel (R', G', B') is a weighted combination of the original channels (R, G, B).

Let's represent the original RGB values of a pixel as a column vector:

$$\mathbf{v} = \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

A 3x3 transformation matrix $\mathbf{M}$ for color balancing could look like:

$$\mathbf{M} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

The new RGB values $\mathbf{v}'$ are then calculated by:

$$\mathbf{v}' = \mathbf{M} \times \mathbf{v}$$

In other words:

$$R' = a \times R + b \times G + c \times B$$

$$G' = d \times R + e \times G + f \times B$$

$$B' = g \times R + h \times G + i \times B$$

To give a simple example, suppose we want to enhance the red channel and slightly decrease the influence of the green channel:

$$\mathbf{M} = \begin{bmatrix} 1.2 & -0.1 & 0 \\ 0 & 0.9 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

In this example:

- The red value R' is 120% of the original R minus 10% of the original G.

- The green value G' is 90% of the original G.

- The blue value remains unchanged.

Figure 8: Notice that the background appears to have a reddish hue rather than a deep black typical of space images. This image needs some color tweaking.

Data reduction

When using your CCD to observe the sky, the "data cleaning" process you'll follow is often termed "data reduction" in astronomy. The primary goal of this process is to remove instrumental and observational artifacts from your images, making them suitable for scientific analysis.

For large surveys like SDSS, these calibration procedures are systematically applied to all data, and the resulting calibrated images are what's made available to the public and to researchers. Thus, when you access SDSS data, you're getting images that have already undergone these crucial calibration steps, making them ready for scientific analysis.

Here are the standard data reduction steps you would typically follow:

Bias Subtraction, dark current subtraction, flat fielding, cosmic ray removal, atmospheric correction,...

Bias

Every CCD has an inherent electronic signal even without any light exposure, with the shutter closed and zero exposure time (so the CCD doesn't have time to accumulate any dark current). This is the bias level. You can think of this bias as the intrinsic electronic signature of the CCD that's present in every image.

This bias is a consequence of the following: To ensure that the digital value (Analog to Digital Conversion) is always positive (even for pixels that captured no light), the electronics of the system (charge to voltage convertor, amplifier,...) introduce a constant offset voltage before the conversion to a digital value. This offset ensures that the digital values will always be positive and that the inherent electronic noise doesn't result in negative values. This constant offset is the "bias".

Ensuring that the digital values are always positive is important for several reasons: In the digital storage system of the CCD camera, pixel values are typically represented as unsigned integers. Unsigned integers can only take on non-negative values. If the signal was allowed to go negative due to inherent electronic noise or other factors, it could cause errors in the storage system. And, for example, electronic noise is inherent in any electronic system, including CCDs. Without a bias offset, fluctuations from this noise could cause pixel values to oscillate around zero, leading some to go negative. By introducing a positive bias, even pixels that experience a negative fluctuation due to noise will remain in the positive value domain.

It is important to say that while the raw output from CCD detectors typically has non-negative pixel values due to the bias offset, several factors can lead to negative pixel values in processed FITS images.

Bias subtraction is critical not just for the science images but also for the calibration frames themselves. For example, before a dark frame can be used to correct for dark current, it first needs to have the bias subtracted out.

Correcting for bias:

Capture Bias Frames:

- A bias frame is an image taken with the CCD camera with zero exposure time and the shutter closed. It captures the electronic signal of the CCD without any light exposure.

- Multiple bias frames are usually taken to, for example, average out any random noise.

Create a Master Bias Frame:

- Combine the individual bias frames to create a master bias. This can be done by averaging the pixel values across all the bias frames.

- Median combining is often preferred over a simple average, as it is less sensitive to outliers like cosmic ray hits.

Subtract the Master Bias Frame:

- For every image subtract the master bias frame from it, pixel by pixel.

- This subtraction removes the electronic bias signal, leaving behind the actual light signal from the sky (plus other sources of noise like dark current that you may have not considered up to this point).

I would like to say that readout noise is not a form of bias, but both are inherent electronic characteristics of CCDs that need to be accounted for in data processing, but they are not the same.

Readout Noise:

- Readout noise arises from the inherent uncertainty in the process of reading the charge from each pixel and converting it into a digital value. This includes the shifting of charges, amplification, and analog-to-digital conversion.

- It's a random noise, meaning it varies from readout to readout and is not consistent like the bias.

- It's characterized by a Gaussian distribution around a mean value (which includes the bias).

In data reduction, you would subtract the bias (using a master bias frame) to remove the constant offset. However, the readout noise remains as a fundamental, random uncertainty in the data. While you can't eliminate readout noise, averaging multiple frames can help reduce its effect, as the noise will average down with the square root of the number of frames combined.

Dark current

Over time, a CCD accumulates charge even without exposure to light, known as the dark current.

How do we correct for it?

Take several dark frames, which are exposures of the same duration as your science images but with the shutter closed, so no light enters. After subtracting the master bias from these darks, average them to create a master dark frame. Subtract this master dark from your science frames.

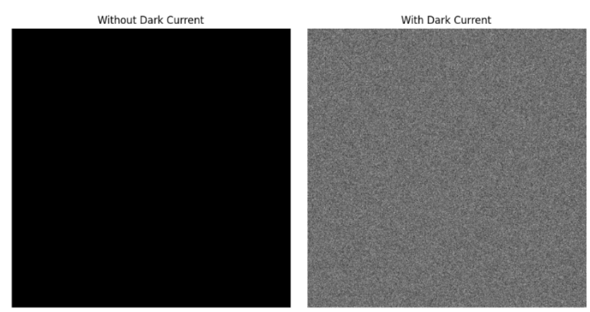


Figure 9: Here I assume that the bias has already been subtracted, therefore, the left image appears full black instead of grayish

The Poisson distribution is commonly used to model the number of events happening in a fixed interval of time, given a fixed average rate of occurrence. The number of photons arriving at a detector in a given period of time, or the number of electrons generated in a pixel due to thermal activity (dark current), are inherently random processes that often follow a Poisson distribution.

The average rate of dark current (i.e., electrons generated per pixel per unit time) is usually provided by the CCD manufacturer or can be measured. This rate serves as the lambda (λ) parameter for the Poisson distribution.

I used a Poisson distribution to generate the image with dark current.

Cosmic Ray Removal

Cosmic rays are high-energy particles (mostly protons) originating from space that, when they hit the Earth's atmosphere or come directly into contact with a detector in space, can produce secondary particles (e.g. neutrinos, neutrons, muons,...). When these secondary particles or the primary cosmic rays strike the silicon of the CCD, they can ionize many atoms. This ionization process means that these high-energy particles can bump a large number of electrons out of their normal positions, promoting them into the conduction band of the silicon.

As a result, a significant number of electrons get collected in the pixel well where the cosmic ray hit occurred. This collection of electrons is what gets read out as a high-intensity signal, making the pixel (or pixels) appear unusually bright, sometimes creating a streak if the cosmic ray particle traveled through multiple pixels;if a cosmic ray hits the CCD at an oblique angle, it can travel through the CCD and affect a path of pixels rather than just one. CCDs are not just 2D grids on a flat plane; they have depth. Each pixel is essentially a small well dug into the silicon where electrons can be collected. A cosmic ray that penetrates the CCD can travel through this depth and, depending on its energy and angle, might pass through the boundaries between adjacent pixel wells. As it travels, it can ionize silicon atoms along its path, producing free electrons in multiple pixels.

When imaging with a CCD, especially in space where there's no atmosphere to shield against cosmic rays, you can sometimes capture these cosmic ray hits. These show up as bright spots or streaks that aren't present in the actual scene you're trying to image.

Correction

One effective way to remove cosmic ray hits is by taking multiple exposures of the same scene and then combining these images. Here's why this works:

Randomness of Cosmic Ray Hits:

The occurrence of cosmic ray hits on a detector is a random process. This means that the likelihood of a cosmic ray hitting the exact same pixel in two different exposures is extremely low.

Combining Multiple Exposures:

If you've taken multiple exposures (or "frames") of the same scene, each frame might have its own unique cosmic ray hits. But because of their randomness, these hits will appear in different places in each frame.

Stacking and Rejection:

When you "stack" (or combine) these frames, you can employ algorithms that identify the pixels that are unusually bright compared to the same pixels in other frames. Because a cosmic ray hit is bright and isolated to a particular frame, these algorithms can easily identify and reject it.

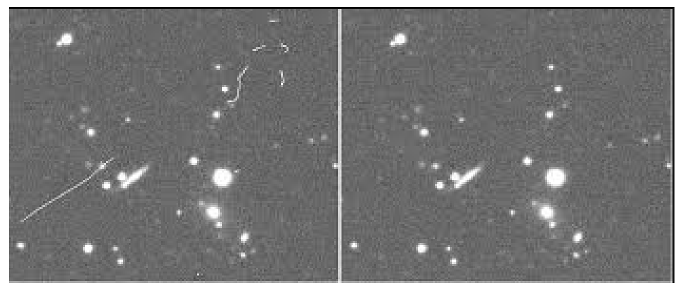


Figure 10: The left image has some "cosmic-ray" detections

Flat Fielding

Not all pixels in a CCD are created equal. Some pixels might be more sensitive to incoming light than others, resulting in a brighter response even when exposed to the same amount of light.

Procedure:

Take Flat Field Frames:

A flat field frame is an image of a uniformly illuminated field. This could be the twilight sky (just after sunset or just before sunrise; because the sky is still somewhat bright during twilight, most stars are not visible, ensuring that they don't interfere with the flat image. This is essential as stars would introduce non-uniformities into the flat field frame), a white screen with a light behind it, or the interior of an observatory dome with lights on.

The goal is to have a smooth, evenly illuminated image where variations are only due to the CCD and the optical system, not the light source.

Subtract Bias (and Dark) from Flats:

Before using the flat frames, you must correct them for inherent system noise. The master bias (and if needed, master dark) is subtracted from each flat frame to ensure any pixel-to-pixel variation is truly from the CCD/optical system and not from electronic noise.

Create a Master Flat:

Once the individual flat frames are cleaned of noise, they are combined (averaged) to create a "master" flat. This averaging helps in reducing random noise and gives a smooth representation of the system's non-uniformities.

Normalize the Master Flat:

Before you use the master flat to correct your science images, it's important to normalize it. This means ensuring that its average value is 1 (or some standard value). This way, when you divide your science image by the master flat, you're not changing the overall brightness of the image, just correcting for the non-uniformities.

Divide Science Frames by Master Flat:

After normalization, you then divide each of your science frames (the actual images of your astronomical target) by the master flat. This process will correct the science frame by adjusting pixel values based on the variations captured in the master flat, thus ensuring uniform sensitivity and illumination across the entire image.

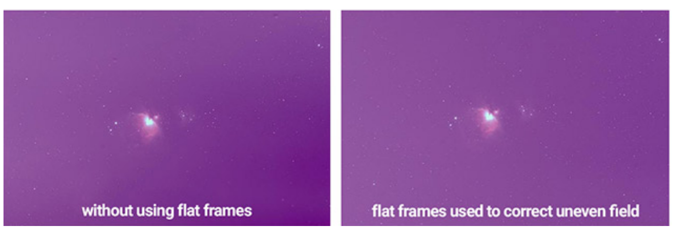


Figure 11: Correction with flat fields. Notice that the corrected image appears to have a more uniform distribution of light intensity

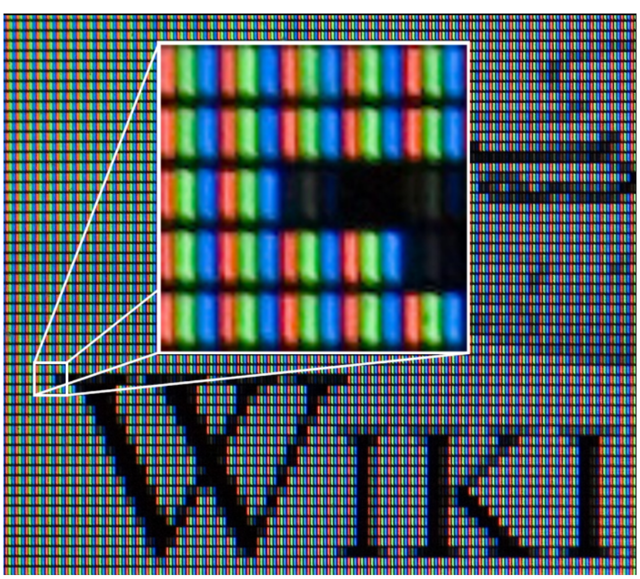


Figure 12: Pixels and subpixels in a screen

If you magnify a portion of your screen significantly, you will see that each pixel is made up of three smaller subpixels: one red, one green, and one blue. When a pixel appears white to our eyes, it means all three sub-pixels are illuminated at full intensity. Conversely, when a pixel appears black, all three sub-pixels are turned off.

The blending of these sub-pixels at varying intensities produces the wide range of colors we see on our screens.

For example:

A fully illuminated red sub-pixel with the green and blue sub-pixels turned off will make the pixel appear red. A fully illuminated green sub-pixel with the red and blue sub-pixels turned off will make the pixel appear green. And so on for other combinations.

If an image is in grayscale, each pixel on your screen will still use the red, green, and blue sub-pixels, but they'll be illuminated to the same intensity:

For a pure white pixel in a grayscale image, the red, green, and blue sub-pixels will all be fully illuminated.

For a pure black pixel, none of the sub-pixels will be illuminated.

For shades of gray in between, all three sub-pixels will be illuminated to the same level. For example, a medium gray might have each of the red, green, and blue sub-pixels illuminated to 50% intensity.

The fact that all three sub-pixels are illuminated to the same level ensures that the color remains "gray", but the intensity of that gray can vary based on the illumination level. So, in grayscale images, the combination of the three colors (red, green, and blue) at the same intensity gives the appearance of various shades of gray.

An interesting question is the following; when we employ plotting tools like "imshow" to display grayscale images, we do not provide "imshow" with more than one float per pixel. How does it manage to plot a grayscale image in this case?

What happens under the hood when you display a grayscale image using a tool like "imshow" is a bit of implicit color mapping:

1. The tool will interpret each float value in the matrix as a grayscale intensity. A value of 0 might represent black, and a maximum value (e.g., 1.0 for normalized float images) might represent white. Intermediate values represent shades of gray.

2. When the image is actually rendered on your screen, the tool will convert each of these grayscale intensities into RGB values where R, G, and B are all equal to that intensity. For example, an intensity of 0.5 might get converted to an RGB value of (127, 127, 127) if using 8-bit color depth.

3. Your screen will then use its red, green, and blue subpixels to display this RGB value.

ONCE YOU GET THE THREE FITS IMAGES THAT YOU WANT TO COMBINE TO CREATE AN RGB FILE, IT IS ENOUGH TO SUM UP THE PIXELS OF EACH IMAGE (PIXEL BY PIXEL)?

No, simply summing up the pixel values of the three FITS images is not the correct approach to create an RGB composite. Instead, each FITS image should be mapped to one of the RGB channels, and then the channels are combined to create a single color image. Do not simply sum the pixel values across the images. Instead, you're essentially stacking the images in different color layers.