TAKNEEK QUEST PS

ASTRO-EXPLORER

Documentation

TEAM MEMBERS:

Abstract

we've developed an image classification model that is pretrained over FITS images of various astronomical objects like asteroids, black holes, comets, galaxies, stars, nebulas and the cosmic space. We've also developed an image enhancement model that's capable of improving the dataset for better training. Finally, we created a light curve classification model that identifies an object based on the wavelength of light wave emitting from it.

Contents

1. Introduction **3**

2. Astronomical Object Details **3**

2.A. Stars . . . 3
2.B. Planets . . . 3
2.C. Galaxies . . . 4
2.D. Nebulae . . . 4
2.E. Black Holes . . . 5
2.F. Constellations . . . 5
2.G. Asteroids . . . 6

3. Model Design and Approach **7**

4. Enhancement Techniques **9**

1. Introduction

In this document, we will explore two main topics. First, we will delve into the classification and key characteristics of various astronomical objects, including stars, planets, galaxies, nebulae, black holes, constellations, and asteroids. This section highlights how these objects are differentiated based on properties such as size, luminosity, composition, and structure. Following this, we will discuss the process of analyzing astronomical images using FITS files, including image resizing and label extraction, which is crucial for tasks like image classification and machine learning in astronomy.

2. Astronomical Object Details

2.A. Stars

Properties: Stars are massive celestial bodies composed primarily of hydrogen and helium that emit light and heat. Key characteristics include:

- Luminosity: The total amount of energy a star emits per unit of time can be inferred from its absolute magnitude and distance.

- Size: Can range from a few times the radius of the Sun (for dwarf stars) to hundreds of times the Sun's radius (for giant stars).

Differentiation : Stars are differentiated by their color (related to temperature), brightness (luminosity) and size. For instance, red giants are cooler and more luminous than blue supergiant.

2.B. Planets

Properties: Planets are celestial bodies that orbit a star and have cleared their orbital path of other debris. Key characteristics include:

- Orbital Parameters: Semi-major axis, eccentricity, and orbital period.

- Surface Conditions: Presence of atmospheres, surface temperatures, and potential for liquid water.

- Size and Composition: Rocky planets (terrestrial) versus gas giants (Jovian).

Differentiation: Planets are classified based on their size (terrestrial vs. gas giant), orbital characteristics, and surface conditions. For example, Earth-like planets have a solid surface and atmosphere, while gas giants do not.

2.C. Galaxies

Properties: Galaxies are vast systems of stars, stellar remnants, interstellar gas, dust, and dark matter. Key characteristics include:

- Morphological Type: Elliptical, spiral, or irregular.
- Star Formation Rate: Indicates the activity and evolutionary stage of the galaxy.

Differentiation: Galaxies are differentiated by their shape and structure. Spiral galaxies have a well-defined disk with spiral arms, while elliptical galaxies are more rounded and lack a distinct structure.

2.D. Nebulae

Properties: Nebulae are vast clouds of gas and dust. Key characteristics include:

- Type: Emission nebulae (glowing due to ionized gas), reflection nebulae (reflecting light from nearby stars), and dark nebulae (blocking light from background stars).
- Composition: Mainly hydrogen and helium with traces of heavier elements.

Differentiation: Nebulae are classified based on their emission properties and the role they play in star formation. For instance, emission nebulae are often found in regions of active star formation.

2.E. Black Holes

Properties:

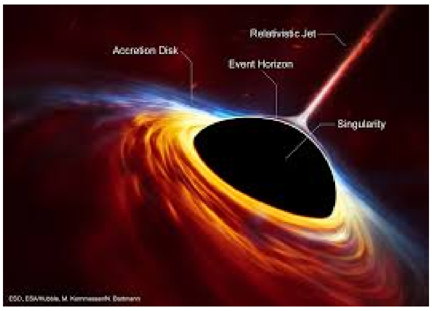
- Event Horizon: The boundary surrounding a black hole beyond which no light or other radiation can escape.

- Singularity: The core of a black hole where matter is thought to be infinitely dense and the gravitational pull is infinitely strong. It is a point where the known laws of physics break down.

- Mass: Black holes can vary greatly in mass. Stellar black holes typically range from about 5 to 20 solar masses, while supermassive black holes can range from hundreds of thousands to billions of solar masses.

- Spin: Black holes can rotate, and this rotation affects their structure.

Differentiation: Formed from the remnants of massive stars that have ended their life cycles. Typically have masses between 5 and 20 solar masses. Around the time of alignment, extreme gravitational lensing of the galaxy is observed.



2.F. Constellations

Properties:

Patterns: Constellations are patterns formed by groups of stars that appear to be grouped from Earth's perspective. These patterns often have historical, cultural, or mythological significance.

- Constellation Boundaries: The sky is divided into 88 official constellations, each with well-defined boundaries established by the International Astronomical Union (IAU).

Differentiation: Constellations are useful in assisting astronomers and navigators in locating certain stars, in the northern sky.

2.G. Asteroids

Properties:

- Composition: Asteroids are rocky bodies left over from the formation of the solar system. They are composed of various materials including metals, silicates, and carbonaceous compounds.

- Size: Asteroids range in size from a few meters to hundreds of kilometers in diameter. The largest asteroid, Ceres, has a diameter of about 940 kilometers.

- Orbit: Most asteroids are found in the asteroid belt between Mars and Jupiter, though some have orbits that bring them close to Earth (Near-Earth Asteroids) or cross the orbits of other planets (Apollo asteroids).

Differentiation: Most asteroids can be found orbiting the Sun between Mars and Jupiter within the main asteroid belt.

3. Model Design and Approach

1. **Importing Libraries**
   - numpy is imported as np for array manipulation.
   - fits from astropy.io is used to read the FITS file.
   - Image from PIL (Python Imaging Library) is used to convert images into a format that allows easy resizing.
   - LabelEncoder from sklearn.preprocessing can be used to encode image labels into integer format (currently commented out).

2. **Paths and Initialization**
   - fits_file_path specifies the location of the FITS file.
   - image_data_list and label_list are lists for storing image data and corresponding labels, respectively.
   - target_size is set to (256, 256) to define the desired image size.

3. **Opening and Iterating over FITS File**
   - fits.open(fits_file_path) opens the FITS file.
   - The loop for hdu in hdulist[1:-1] iterates over the HDUs (Header Data Units) starting from index 1 and ending before the last HDU, which contains labels.
     - *Reasoning:* Index 0 (primary HDU) is skipped, as it usually contains metadata, not image data. The last HDU might contain a table of labels.
   - Each HDU is checked with isinstance(hdu, fits.ImageHDU) to ensure it contains image data.

4. **Image Data Extraction and Resizing**
   - For each valid image HDU, image_data = hdu.data extracts the image data as a NumPy array.
   - The image data is converted to a PIL Image using Image.fromarray(image_data) for resizing.
   - The image is resized to the target_size of 256x256 pixels using image.resize(target_size).
   - The resized image is converted back to a NumPy array via np.array(resized_image) and appended to image_data_list.
   - The label for the image is fetched from the HDU's header using hdu.header.get('OBJECT'), and appended to label_list.

5. **Converting to NumPy Arrays**
   - After processing all the images, image _data_list and label_list are converted to NumPy arrays: image_data_array and label_array.

6. **Label Encoding**
   - The commented-out section shows the option to encode the image labels as integers using LabelEncoder. This is useful for classification tasks, though it's not activated in this code.

7. **Printing Results**
    - The shape of image_data_array is printed to verify that the images have been processed and resized correctly.
    - The label array is printed as well.

The file is a Jupyter Notebook containing Python code related to displaying FITS images (a format used for astronomical data). It uses the following tools and techniques:

Imports and libraries:

matplotlib.pyplot for plotting. astropy.io.fits for reading and handling FITS files. Functionality:

A function display_fits_image(fits_file, title="FITS Image") is defined to load and display images from a FITS file. It opens the FITS file, extracts the image data, and then displays it using matplotlib in grayscale. Model and Output:

The output is the display of the FITS image in a grayscale format using the imshow function from matplotlib. The FITS file in use is specified as '/content/output_images.fits'. The model itself is a basic image rendering system for FITS files, focusing on visualization rather than enhancement. The output is a visual representation of the FITS image in a 10x10 plot.

4. Enhancement Techniques

1. **Processing FITS Data into RGB Bands**
   - The function create_separate_band_fits() takes an input FITS file and creates separate FITS files for the RGB channels.
   - Three methods for processing the bands are used:
     - Histogram Equalization (method='hist' for red band): Improves contrast by spreading out the intensity values.
     - Adaptive Histogram Equalization (method='adapthist' for green band): Locally enhances contrast while preserving details.
     - Gamma Correction (method='gamma' for blue band): Adjusts the brightness of the image using a gamma value.
   - Each processed band is written to separate output FITS files for the red, green, and blue bands.
2. **Displaying FITS Image**
   - The display_fits_image() function reads and displays any FITS image using Matplotlib. It visualizes grayscale images, which represent the intensity of the FITS data.
3. **Enhancing and Combining Color Bands**
   - The enhance_color_bands() function reads the three FITS files corresponding to the RGB channels.
   - Depending on the object_type (like 'nebula', 'galaxy', etc.), it calls specific enhancement functions, such as enhance_nebula_color() or enhance_galaxy_color().
   - The preprocess_band() function is applied to each band, which rescales intensity and applies histogram equalization.
4. **Displaying the Enhanced RGB Image**
   - Once the bands are processed and combined into an RGB image, the display_image() function is used to show the enhanced, color-combined image.

Customization Based on Object Type:

- Black Hole: Standard enhancement combining the bands without extra sharpening or smoothing.
- Constellation: Sharpened using unsharp masking to highlight stars.
- Galaxy/Nebula: Sharpening applied to highlight structures.
- Cosmos: Smoothed using Gaussian kernel to soften wide-field images.
- Planet: Mild sharpening applied to emphasize planetary details.
- Star: Sharpening to enhance star clarity.

References