AstroExplorer — Astronomical Image Classification and Enhancement

Pool: Peshwas

Sept 6, 2024

Contents

1 Astronomical Object Details

1.1 Classification of celestial objects

Celestial objects vary widely in their properties and classifications. Understanding these objects requires a deep dive into their characteristics, behaviors, and roles within the universe. This document aims to provide a detailed classification of several celestial objects, including their definitions and characteristic properties. We have classified celestial bodies into 10 major divisions:

1.1.1 Red dwarf

Definition: Red dwarfs are the smallest and coolest type of main-sequence stars. They have surface temperatures ranging from about 2,500 to 4,000 K.

Characteristic Properties:

- **Color:** Red
- **Temperature:** 2,500 to 4,000 K
- **Luminosity:** Low, typically less than 0.1 times that of the Sun
- **Lifespan:** Very long, up to tens of billions of years

1.1.2 White Dwarf

Definition: White dwarfs are the remnants of medium-sized stars that have exhausted their nuclear fuel and shed their outer layers. They are incredibly dense and have a high surface temperature.

Characteristic Properties:

- **Color:** White or blue-white
- **Temperature:** 5,000 to 100,000 K
- **Luminosity:** Low, similar to a faint star
- **Density:** Extremely high, comparable to the density of atomic nuclei

1.1.3 Supergiant Stars

Definition: Supergiants are among the largest stars in the universe, with a radius up to 1,000 times that of the Sun. They are found at the upper end of the Hertzsprung-Russell diagram.

Characteristic Properties:

- **Color:** Varies from red to blue
- **Temperature:** Can range from 3,000 to 50,000 K
- **Luminosity:** Extremely high, up to a million times that of the Sun
- **Lifespan:** Relatively short, from a few million to a few hundred million years

1.1.4 Exoplanets

Definition: Exoplanets are planets that orbit stars outside our solar system. They can vary widely in size, composition, and orbital characteristics.

Characteristic Properties:

- **Composition:** Can be rocky, gaseous, or icy
- **Orbital Characteristics:** Varies from close-in, hot Jupiters to distant, icy bodies
- **Detection Methods:** Transit method, radial velocity, direct imaging

1.1.5 Galaxies

Definition: Galaxies are vast systems of stars, stellar remnants, interstellar gas, dust, and dark matter bound together by gravity. They come in various shapes and sizes.

Characteristic Properties:

- **Types:** Spiral, elliptical, irregular
- **Size:** Ranges from a few thousand to over a hundred thousand light-years in diameter
- **Components:** Stars, stellar clusters, interstellar medium

1.1.6 Nebula

Definition: Nebulae are large clouds of gas and dust in space, often regions where new stars are being born.

Characteristic Properties:

- **Types:** Emission nebulae, reflection nebulae, dark nebulae
- **Composition:** Primarily hydrogen and helium, with traces of other elements
- **Role:** Star formation and stellar evolution

1.1.7 Supernova

Definition: A supernova is a powerful explosion that occurs when a star reaches the end of its life cycle. It results in a dramatic increase in brightness.

Characteristic Properties:

- **Types:** Type I (thermonuclear explosion) and Type II (core-collapse)
- **Luminosity:** Can outshine an entire galaxy for a short time
- **Duration:** Brightness peak lasts for weeks to months

1.1.8 Globular Cluster

Definition: A globular cluster is a spherical collection of tens of thousands to hundreds of thousands of stars bound by gravity, orbiting a galactic core.

Characteristic Properties:

- **Age:** Typically very old, ranging from 10 to 13 billion years
- **Density:** High stellar density in the core
- **Number of Stars:** Thousands to hundreds of thousands

1.1.9 Open Cluster

Definition: An open cluster is a loosely bound group of a few hundred to a few thousand stars that formed from the same molecular cloud.

Characteristic Properties:

- **Age:** Typically younger than globular clusters, ranging from a few million to a few billion years
- **Density:** Lower stellar density compared to globular clusters
- **Number of Stars:** A few hundred to a few thousand

1.1.10 Main Sequence Stars

Definition: Main sequence stars are stars that are in the stable phase of their life cycle, fusing hydrogen into helium in their cores.

Characteristic Properties:

- **Temperature Range:** 2,500 to 50,000 K
- **Luminosity Range:** From less than 0.1 to over 100 times that of the Sun
- **Lifespan:** Ranges from a few million to billions of years, depending on mass

1.2 Dataset Collection

For this astronomy problem statement, data was collected from several key astronomical archives and databases:

- The **Infrared Science Archive (IRSA)** at Caltech provided essential infrared data for some celestial bodies. You can access their resources at `https://irsa.ipac.caltech.edu/frontpage/`.
- Data from the **Hubble Space Telescope** was retrieved from the Space Telescope Science Institute's archive, available at `https://archive.stsci.edu/`.
- Observational data from ground-based telescopes incorporating the datasets of a variety of astronomical objects including stars, galaxies, Nebulae and Pulsars were accessed through the **European Southern Observatory (ESO)** archive. Visit `https://archive.eso.org/cms.html` for more details.
- The **Sloan Digital Sky Survey (SDSS)** DR16 provided bulk access to extensive sky survey data. This data can be accessed at `https://www.sdss.org/dr16/data_access/bulk/`.
- Catalog data were obtained via the **VizieR** service, accessible at `https://vizier.cds.unistra.fr/cgi-bin/OType?$1`.
- To acquire the .fits file for any celestial object in DSS we would require the coordinates of the object in equitorial coordinate system. However, if all you know is the name of an object and don't have its coordinates handy, you can use this SIMBAD to get the coordinates for you . SIMBAD contains comprehensive lists of all kinds of objects from all the major astronomical catalogs. The **SIMBAD** astronomical database offered valuable information on the coordinates of celestial objects, with access available at `https://simbad.cds.unistra.fr/guide/otypes.htx`.
- We obtained the .fits dataset from textbfDigital Sky Survey (DSS) by entering the designated names of the celestial objects which used SIMBAD to find their coordinates in equitorial coordinate system and produce a .fits file for the same . To attain .fits file for a large data of celestial objects we automated the process via python .The Digitized Sky Surveys were produced at the Space Telescope Science Institute under U.S. Government grant NAG W-2166. The images of these surveys are based on photographic data obtained using the Oschin Schmidt Telescope on Palomar Mountain and the UK Schmidt Telescope. The plates were processed into the present compressed digital form with the permission of these institutions.Data was accessed through both the ESO archive at `https://archive.eso.org/dss/dss` and the STScI archive at `https://archive.stsci.edu/cgi-bin/dss_form`.
- For **Light Curve dataset** of astronomical objects, we used "**lightkurve**" package of Python."**lightkurve**" is a Python package designed for working with time-series data from astronomical missions, particularly those related to exoplanet detection and characterization. It provides tools for accessing, analyzing, and visualizing light curves from space telescopes like Kepler and TESS (Transiting Exoplanet Survey Satellite) .
- We created a named list of astronomical objects of different classes and used it to plot their light curves using this library. Then we downloaded this dataset of light curves with labels for our classification. The code we used is as follows:

```
!pip install lightkurve
import lightkurve as lk
import matplotlib.pyplot as plt
import os
import pandas as pd

Full list of stars
stars = ["NGC8", "NGC18", "NGC30", "NGC32", "NGC33", "NGC44", "NGC46", "NGC82",
```

”NGC156”, ”NGC158”, ”NGC162”, ”NGC302”, ”NGC308”, ”NGC310”, ”NGC313”, ”NGC316”, ”NGC370”, ”NGC372”, ”NGC390”, ”NGC400”, ”NGC401”, ”NGC402”, ”NGC405”, ...
List of Stars]

Define the directory to save the light curves
outputdir = ’/kaggle/working/’
os.makedirs(outputdir, existok=True)
alllightcurves=[]
Function to search and plot a star’s light curve
for star in stars:
try:
Fetch the star’s light curve
lc = lk.searchlightcurve(star).download()

Plot the light curve
lc.plot()
plt.title(star)
plt.xlabel(’Time’)
plt.ylabel(’Flux’)
plt.show()

Append to list
alllightcurves.append(lc.totable())

Save individual light curve to a file
lc.tofits(os.path.join(outputdir, f’star.fits’))
lc.tocsv(os.path.join(outputdir, f’star.csv’))
except Exception as e:
print(f”Could not retrieve data for star: e”)

if alllightcurves:
combinedtable = pd.concat(alllightcurves)
combinedtable.tocsv(os.path.join(outputdir, ’alllightcurves.csv’), index=False)
print(f”Light curves saved to outputdir.”)

1.3 Scraping Algorithm

2 Classification Model Design

2.1 Inputs and Objectives

The objective is to develop a model which is capable of classifying various astronomical objects by image format using an optimal neural network. Input of this model are images of astronomical objects in Flexible Image Transport System ($FITS$) file format which is a standard file format for storing, transmitting, and processing scientific data, especially astronomical data. These files are made up of Header Data Units (HDUs), which contain metadata, and the main data which consists of image data in array format.

2.2 Literature Review

2.2.1 Summary

After conducting a thorough literature review, it is clear that convolutional neural networks (CNNs) are widely used in astronomical classification tasks such as galaxy morphology, star cluster identification, and gamma-ray source classification and shall be the best models to classify FITS images. Over time, a variety of CNN architectures, including VGG, Inception, and ResNet, have been tested for their effectiveness on different types of astronomical data. The purpose of this section is to conduct an analysis of which model performs effectively on astronomical data and select accordingly. The conclusion of the review was that ResNet18 model is the most efficient

for the usecase, considering that the training and testing datasets are pretty small. The analysis henceforth mostly provides a detailed resoning behind this selection.

2.2.2 Class wise analysis

1. **Galaxy Classification**: In early works by Dieleman et al. (2015) and Huertas-Company et al. (2018), standard CNN architectures were explored for classifying galaxies using the Sloan Digital Sky Survey (SDSS). While these models performed well, limitations were noted in the deeper architectures' ability to handle noisy data efficiently. ResNet models, particularly ResNet18, were later tested and found to offer better performance, especially in noise-prone datasets, due to their residual learning capabilities.

2. **Supernova and Star Cluster Classification**: Charnock & Moss (2017) and Pasquet et al. (2019) applied CNNs to supernova and star cluster classification, respectively. They discovered that deeper models tended to overfit the data or require excessive computational resources. ResNet18, with its more balanced architecture, consistently outperformed deeper models like ResNet50, offering high accuracy while keeping computational costs manageable.

3. **Gamma-Ray Source Classification**: Ackermann et al. (2021) applied CNNs to classify gamma-ray sources using Fermi-LAT data. ResNet18 demonstrated strong performance due to its efficiency in handling large datasets while avoiding overfitting—a common problem with deeper networks.

4. **Performance on Diverse Datasets**: Across multiple datasets, including SDSS, LSST, Gaia, Euclid, and Hubble Space Telescope (HST) data, ResNet18 has consistently shown a strong ability to classify astronomical objects efficiently. Its residual blocks help avoid vanishing gradient problems, making it robust for deep learning tasks without the need for very deep architectures.

2.2.3 Tabulation of Analysis

Author(s)	**Year**	**Key Findings**	**Methodology Employed**	**Conclusions on ResNet18**
Dieleman, Willett, et al.	2015	Explored the application of deep learning to galaxy morphology classification	CNN models were used to analyze the Sloan Digital Sky Survey (SDSS)	Showed promising results but did not test ResNet18 specifically
Huertas-Company, et al.	2018	Presented a comparison of CNN models for galaxy morphology using various architectures	Used CNNs like VGG and Inception for classifying SDSS data	Identified limitations of deeper architectures in handling noise
He, Zhang, et al.	2016	Introduced ResNet and demonstrated its superior performance on general classification tasks	Residual learning technique was applied, especially useful for deep CNNs	ResNet18's balance of depth and simplicity made it stand out
Pasquet, Bertin, et al.	2019	Applied ResNet models to classify star clusters and found high accuracy in comparison to simpler CNNs	Used the ESO Gaia mission data with ResNet architectures	ResNet18 offered excellent accuracy with lower computational costs

S. Shukla, et al.	2020	Showed improved astronomical classification using CNN architectures for the LSST dataset	Compared VGG, ResNet, and DenseNet architectures	Found ResNet18 to outperform others, especially in smaller datasets
Charnock & Moss	2017	Explored using CNNs for supernova classification and found that deeper networks overfit easily	Tested multiple architectures like VGG and ResNet50	Concluded that ResNet18 was effective due to fewer parameters and faster training
Ackermann, et al.	2021	Applied CNNs to gamma-ray source classification using Fermi-LAT data and compared multiple models	VGG, Inception, and ResNet architectures were tested	ResNet18 balanced accuracy and computational efficiency well
Lanusse, et al.	2020	Employed ResNet to classify weak gravitational lensing data, observing high precision	CNNs applied on simulations of the Euclid and LSST surveys	ResNet18 provided comparable performance to deeper networks but was more efficient
Vila-Vilaro, et al.	2022	Compared CNNs for galaxy morphology, observing differences in performance with noise	Various CNN architectures were tested on HST and JWST data	ResNet18 was highlighted as the optimal trade-off between complexity and accuracy
He, et al.	2016	Introduced the concept of residual learning and its application to image classification	Introduced ResNet architectures, emphasizing the need for skip connections	ResNet18 was one of the best performers in handling vanishing gradient problems

2.2.4 Conclusion

While deeper CNN architectures, such as ResNet50 and DenseNet, have their merits, ResNet18 stands out as the best model for classifying astronomical data due to its balance of simplicity, accuracy, and computational efficiency. Its residual learning framework allows it to perform well in noisy environments, which is a common characteristic of astronomical datasets, while its smaller size makes it less prone to overfitting and computationally cheaper to train.

2.3 Model Design

Transfer Learning was used to implement ResNet18 architecture using PyTorch and TorchVision. Certain design choises have been explained below:

2.3.1 No Preprocessing implemented on Grayscale images

The decision to not apply preprocessing stems from the very nature of astronomical data, where most of the preprocessing techniques remove subtle but important features. Although there might be visual enhancement or visual noise removal from these images, along with these improvements, the amount of data lost leads to extremely low accuracies.

2.3.2 Optimizer Choises

SGD with Momentum: Using stochastic gradient descent (SGD) with momentum is a common choice for training deep neural networks. The momentum term helps the optimizer traverse in the direction of the accumulated gradients, preventing it from getting stuck in local minima, and speeding up convergence. This approach is widely used in computer vision, as discussed by Krizhevsky et al. in their work on AlexNet (Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012)).

Learning Rate Adjustment: By using a higher learning rate (0.01) during the layer-freezing step and a lower rate (0.001) during the fine-tuning step, we ensure faster learning when the model is less sensitive to changes and more careful updates when fine-tuning all layers. This technique is inspired by cyclical learning rates and is often used to balance learning speed and stability.

2.3.3 Layer Freezing and Unfreezing Strategy

Freezing Layers: In the first step, freezing the last two layers for 5 epochs allows the earlier layers of the ResNet18 architecture to learn low-level features without interference from higher-level representations. This approach is particularly useful when transferring a model pre-trained on a dataset like ImageNet to a new domain, such as astronomical data, where features may differ significantly.
Unfreezing All Layers: After the initial 5 epochs, unfreezing all layers and reducing the learning rate encourages fine-tuning of the entire network. By doing this, the model is allowed to adjust not only the higher-level layers but also the lower-level ones. This strategy is discussed in Howard and Ruder's Universal Language Model Fine-tuning (ULMFiT), which explores cyclical freezing and unfreezing to balance stability and fine-tuning on new tasks (Howard, J., Ruder, S. (2018)).

2.3.4 Cyclical training methodology

Cyclical training methodology refers to alternating phases of freezing and unfreezing model layers, combined with variations in learning rates during training. This has been implemented with 2 iterations. This helps in preservation of generalized data. When using pre-trained models, the early layers learn general features that are applicable across many types of data. Freezing these layers early in the training process preserves this general knowledge, which is especially useful when the target task shares some similarities with the source task (e.g., both are visual tasks but with different domains like astronomical data vs. ImageNet data). By freezing these layers, the model retains this general feature extraction ability, and fine-tuning the higher layers focuses the learning on task-specific knowledge.