Galaxy Classifier – Project 3

Mitchell, Trent, Rob, Tony

Executive Summary

We developed a convolutional neural network (CNN) to classify a dataset of galaxy images. Our goal was to automate the manual classification of galaxy images using our image classification model.

The model effectively distinguishes between different types of galaxies, demonstrating relatively high multi-class accuracy of 75% and potential for further application in astronomical image analysis.

Data Collection

We used Galaxy Zoo 2 data, a subsample of the original Galaxy Zoo data, which measures more detailed morphological features.

The morphological features include galactic bars, spiral arm and pitch angle, bulges, edge-on galaxies, relative ellipticities, and many others.

Data source: https://data.galaxyzoo.org

Package source: https://pypi.org/project/galaxy-datasets/

Exploratory Data Analysis

Example Raw Image

We successfully downloaded and processed the Hart et al. (2016) dataset from Galaxy Zoo 2 containing over 209,000 images, validating the integrity by checking for missing values, and verifying data types.

Using the Python Imaging Library (PIL), we further examined individual images and their properties, ensuring consistency in image dimensions and formats across the dataset.

Data Clean Up

Using Hart et al. (2016), we identified the y-value column, which consisted of the highest p-value. This data column was manually labeled ‘summary’ with 7 possible classifications.

We developed a target_formatter.py file to parse the summary column with OneHotEncoder to create our multi-class y variable.

We developed a datasets_util.py file to handle the download of the Galaxy Library and create numpy arrays from the raw image data.

Detailed Approach

We developed an overarching classification notebook that called our target_formatter function and datasets_util function. This loaded, preprocessed, and split the data.

This notebook then called the cnn_model.py file to create the CNN model, which contained our parameter set.

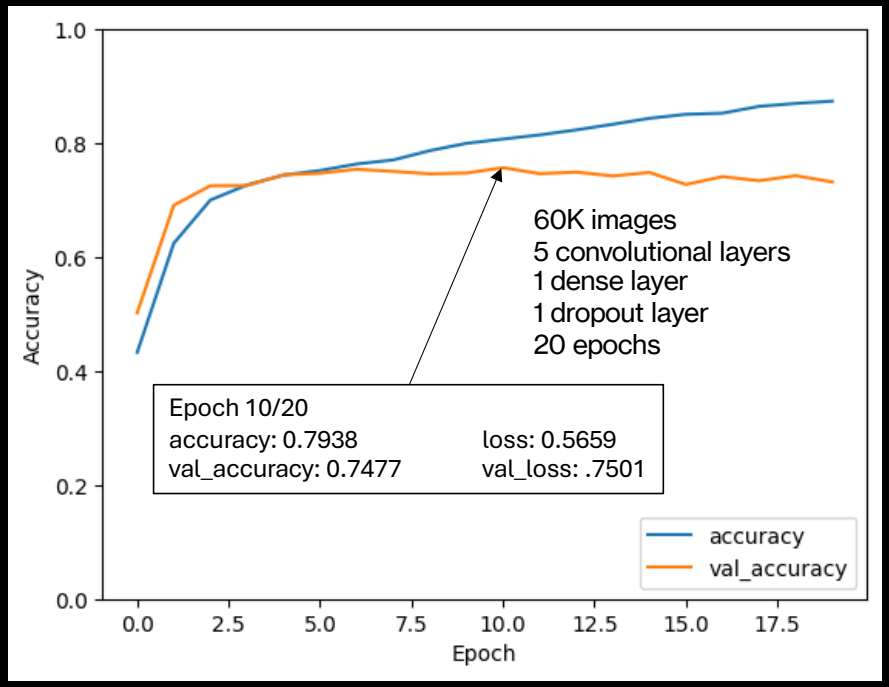
The CNN model was then trained and evaluated. Adjustments were made as necessary to dataset size and CNN parameters to improve the model.

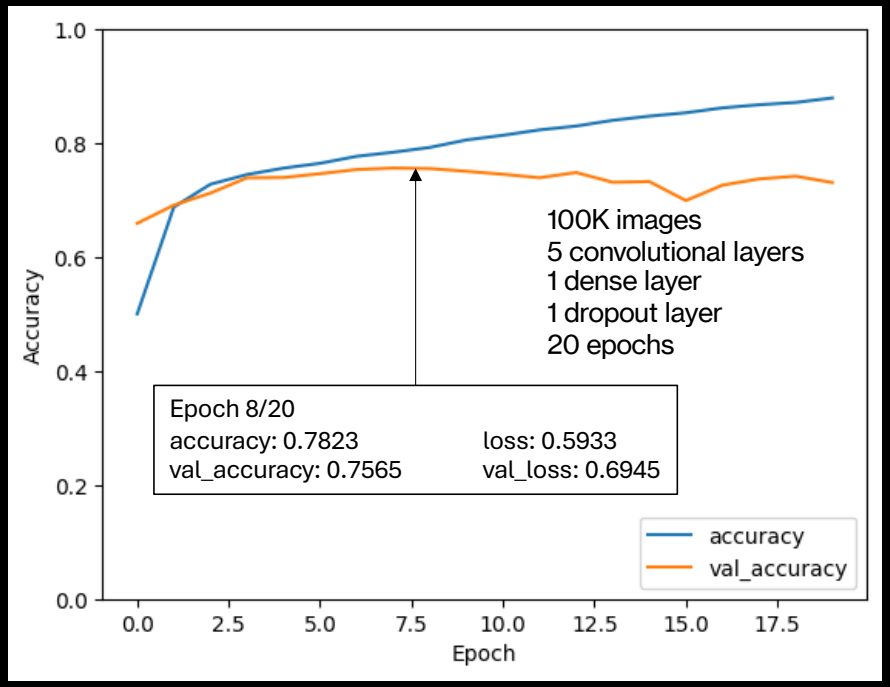
Future Research & Hindsight Changes

Lessons Learned:

- Missed nulls in the summary column.
- Local machines typically lacked sufficient RAM to process datasets larger than 50K images.
- Colab Pro Account with ample paid compute time is necessary to process large volumes of images.
- More sufficient hardware would allow greater experimentation to increase accuracy with such techniques as batch normalization, data augmentation, edge detection, and CNN parameters.

Results & Conclusions



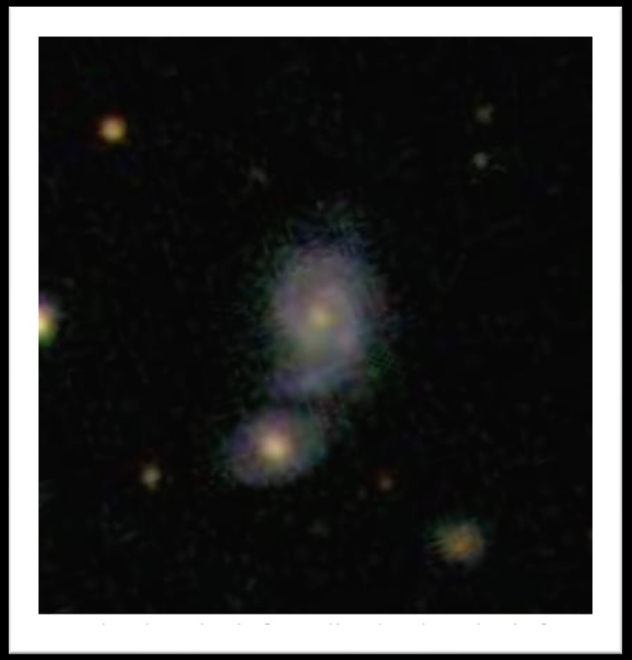


Learning occurred rapidly and then tapered off due to overfitting.

Model improvement occurred moving from 10K to 100K images and from three to five convolutional layers.

Classification Examples – Correct

Smooth Cigar



Unbarred Spiral

Smooth In-Between

Classification Examples – Incorrect

Label: Barred Spiral
Prediction: Unbarred Spiral

Label: Edge on Disk
Prediction: Unbarred Spiral

Questions