

# Galaxy Image Classification

---

Shardul Junagade

23110297

GitHub Link - [ShardulJunagade/Cosmic-Curator](https://github.com/ShardulJunagade/Cosmic-Curator)

## 1. Introduction

In this project, the goal was to classify galaxies into three categories—Spiral, Elliptical, and Uncertain—using deep learning. Given the strong performance of transformer-based models in computer vision, I chose to use a **Swin Transformer (Tiny variant)** pretrained on ImageNet as the core model.

## 2. Data Preparation and Preprocessing

The dataset contained astronomical images with corresponding labels in a CSV file. I cleaned the data by removing entries for images that were missing from the directory. To make sure each class was well represented, I used stratified sampling to split the data into 90% training (5400 images) and 10% validation (600 images).

For the training set, I applied random flips and rotations to augment the data and help the model generalize better. For the validation and test sets, I only applied resizing and normalization to maintain consistency during evaluation.

## 3. Model Architecture

At the heart of the model is the Swin-Tiny transformer, which is **pretrained on ImageNet**. On top of that, I added a custom classification head made up of:

- A linear layer with 256 output features
- ReLU activation
- Dropout for regularization
- A final output layer with 3 neurons (for the 3 classes)

The model was trained using **CrossEntropyLoss**, and I used the **AdamW optimizer** with a learning rate and weight decay of 0.01.

## 4. Training and Evaluation

I trained the model for **30 epochs** and tracked both the training/validation losses and the **macro F1-score**, which is especially important here because the classes were imbalanced.

To avoid overfitting and select the most reliable models, I saved the **top 2-3 checkpoints** based on their F1-scores on the validation set. I also plotted loss and F1 trends over epochs to get a better idea of how training progressed. All of this was done using the **P100 GPU**

available on Kaggle, which sped things up significantly.

Plots of loss and F1-score vs epochs were saved for performance visualization. The training was conducted on P100 GPU available on Kaggle.

## 5. Experimentation

I conducted multiple experiments to understand what works best for this classification task:

- **Model Variants:**

- **ResNet-50:** Served as a baseline CNN model. It showed reasonable performance but was outperformed by transformer-based models.
- **Swin-Tiny (Swin-T)**
- **Swin-Small (Swin-S)**
- **Swin-Base (Swin-B)**

- **Preprocessing with Denoising:**

Astronomical images often contain noise. Denoising was applied using OpenCV's `fastNlMeansDenoisingColored` function to enhance the image quality before feeding into the model.

## 6. Results and Conclusion

The model demonstrated good convergence and generalization. Macro F1-score was used as the primary metric due to class imbalance, ensuring fair evaluation across all galaxy types. The use of Swin Transformer, combined with appropriate augmentations and evaluation metrics, led to a robust solution for the galaxy classification task.