Galaxy Morphology Classification Using Deep Learning

Ezz Eldeen 202202132 Abdel Aziz Elhelaly 202201827

December 2024

1 Introduction

The goal of this project is to classify galaxy images into three distinct morphological categories: Spiral, Lenticular, and Elliptical, using deep learning models. The primary method used is Convolutional Neural Networks (CNN) and Residual Networks (ResNet). This task not only provides an opportunity to explore feature extraction but also allows us to gain insights into the nature of galaxies in astrophysics.

2 Methodology

The methodology of this project involves several stages, ranging from data preparation to model evaluation. The process includes the following key steps: data preprocessing, model design and architecture, training, and evaluation.

2.1 Data Collection and Preprocessing

The dataset consists of 10,000 labeled galaxy images, categorized into three classes: Spiral, Lenticular, and Elliptical. The images are split into three sets:

- **Training Set:** 6,833 images
- **Validation Set:** 3,450 images
- **Test Set:** 450 images

Each image in the dataset is of variable size and resolution, so resizing and normalization are crucial preprocessing steps to ensure the data is suitable for training deep learning models.

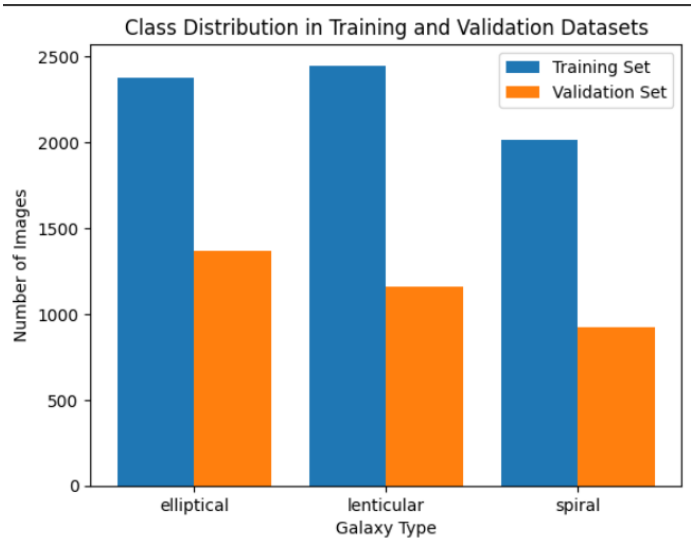

Figure 1: Data Distribution

2.1.1 Data Augmentation

To address overfitting and improve the model's generalization capability, data augmentation techniques are applied to the training data. These transformations include:

- **Rescaling:** The pixel values of the images are scaled to the range [0, 1] by dividing by 255.
- **Shear Range:** Images are randomly sheared to simulate rotation.
- **Zoom Range:** Random zoom transformations are applied to simulate different viewing distances.
- **Horizontal Flip:** The images are randomly flipped horizontally to simulate variations in orientation.

The validation and test datasets are not augmented to ensure an unbiased evaluation of the model's performance.

2.1.2 Data Generators

To handle large datasets and optimize memory usage, images are loaded in batches using Keras' `flow_from_directory` function. This function is used with the following settings:

- **Target Size:** All images are resized to 224x224 pixels to maintain consistency across the dataset.

- **Batch Size:** A batch size of 32 images is used for both training and validation.
- **Class Mode:** The class mode is set to `categorical`, which applies one-hot encoding to the labels for the three classes.
- **Shuffling:** Shuffling is enabled during training to introduce randomness and reduce overfitting, while it is disabled for validation to ensure that the evaluation is done on a consistent and ordered dataset.

These preprocessing steps ensure that the model is trained on varied data while maintaining consistency in the validation set for fair evaluation.

2.2 Model Architecture

The model designed for this project is a Convolutional Neural Network (CNN) with multiple convolutional layers followed by pooling and fully connected layers. The architecture of the model is outlined below:

- **Input Layer:** The input size is set to 224x224x3, where each image is resized to 224x224 pixels, and the RGB color channels are maintained.
- **Convolutional Layers:** The model consists of four convolutional layers with increasing filter sizes: 32, 64, 128, and 256 filters, respectively. These layers help in extracting increasingly abstract and complex features from the input images.
- **Batch Normalization:** After each convolutional layer, batch normalization is applied to improve training speed and model stability by normalizing the output.
- **Max-Pooling Layers:** Max pooling with a 2x2 pool size is applied after each convolutional block to reduce the spatial dimensions and focus on the most relevant features.
- **Global Average Pooling:** This operation reduces the output dimensionality from 3D to 1D, producing a fixed-size output irrespective of the input image size.
- **Dropout Layers:** Dropout is applied after the fully connected layers to prevent overfitting by randomly disabling a fraction of neurons during training.
- **Fully Connected Layer:** A dense layer with 128 units is added, followed by another dropout layer to further regularize the model.
- **Output Layer:** The output layer consists of three units, corresponding to the three galaxy classes (Spiral, Lenticular, Elliptical), and uses the softmax activation function to produce class probabilities.

The CNN model is summarized in the following architecture:

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 222, 222, 32)      896
batch_normalization          (None, 222, 222, 32)      128
max_pooling2d (MaxPooling2D) (None, 111, 111, 32)      0
conv2d_1 (Conv2D)            (None, 109, 109, 64)      18496
batch_normalization_1        (None, 109, 109, 64)      256
max_pooling2d_1 (MaxPooling2D) (None, 54, 54, 64)      0
conv2d_2 (Conv2D)            (None, 52, 52, 128)       73856
batch_normalization_2        (None, 52, 52, 128)       512
max_pooling2d_2 (MaxPooling2D) (None, 26, 26, 128)      0
conv2d_3 (Conv2D)            (None, 24, 24, 256)       295168
batch_normalization_3        (None, 24, 24, 256)       1024
max_pooling2d_3 (MaxPooling2D) (None, 12, 12, 256)      0
global_average_pooling2d     (None, 256)               0
dropout (Dropout)            (None, 256)               0
dense (Dense)                (None, 128)               32896
dropout_1 (Dropout)          (None, 128)               0
dense_1 (Dense)              (None, 3)                 387
=================================================================
Total params: 423,619 (1.62 MB)
Trainable params: 422,659 (1.61 MB)
Non-trainable params: 960 (3.75 KB)
```

The ResNet 50 model is summarized in the following architecture:

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Functional)        (None, 7, 7, 2048)        23,587,712
global_average_pooling2d_5   (None, 2048)              0
dense_7 (Dense)              (None, 256)               524,544
dropout_3 (Dropout)          (None, 256)               0
dense_8 (Dense)              (None, 3)                 771
=================================================================
Total params: 24,113,027 (91.98 MB)
Trainable params: 525,315 (2.00 MB)
Non-trainable params: 23,587,712 (89.98 MB)
```

2.3 Model Training

The model is trained using the Adam optimizer with a categorical cross-entropy loss function, as the problem is a multiclass classification task. The following training strategies are employed:

- **Early Stopping:** To prevent overfitting, early stopping is used to halt training if the validation loss does not improve for 5 consecutive epochs. The model with the best validation loss is restored after training.
- **Learning Rate Reduction:** The learning rate is reduced by a factor of 0.5 if the validation loss does not improve for 3 epochs. The minimum learning rate is set to 0.0001 to avoid excessively small updates.

2.4 Model Evaluation

The performance of the model is evaluated using the test set, which consists of 450 images. The model's accuracy is calculated to assess its classification performance across all three galaxy categories.

3 Comparison of ResNet50 and Custom CNN Models

In this study, we compared the performance of two deep learning architectures for galaxy morphology classification: a pre-trained ResNet50 model and a custom Convolutional Neural Network (CNN). The evaluation was conducted using calculation of accuracy

3.1 Analysis of Results

The ResNet50 model achieved superior performance across all metrics compared to the custom CNN model. This improvement can be attributed to the following factors:

- **Transfer Learning:** ResNet50 leverages pre-trained weights on large-scale datasets, enabling better feature extraction for galaxy morphology.
- **Deeper Architecture:** ResNet50's deep structure facilitates learning complex patterns compared to the relatively shallow custom CNN.
- **Regularization:** Techniques like batch normalization in ResNet50 reduce overfitting.

In contrast, the custom CNN model, while simpler and computationally less expensive, exhibited slightly lower performance due to limited feature extraction capacity. Future improvements could involve augmenting the custom CNN with additional layers or incorporating transfer learning principles.

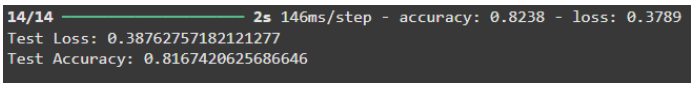

Figure 2: CNN Accuracy

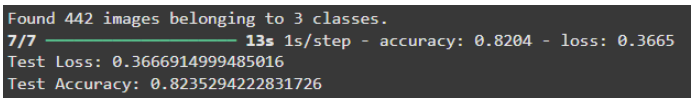

Figure 3: ResNet 50

4 Challenges

Several challenges were encountered during the project:

- **Overfitting**: Despite applying data augmentation and dropout, overfitting remained an issue. This was mitigated by using early stopping and further tuning the dropout rate.
- **Imbalanced Dataset**: The distribution of galaxy classes in the dataset was not perfectly balanced, which might have affected the model's performance on underrepresented classes.
- **Computational Complexity**: The training process was computationally expensive, especially with deep models. Optimizing hyperparameters helped reduce training time.

5 Conclusion

The project successfully developed a deep learning model for classifying galaxy images into Spiral, Lenticular, and Elliptical categories. The model's performance was satisfactory, but further improvements can be made by using more advanced architectures such as ResNet, tuning hyperparameters, and increasing the dataset size. Future work could explore transfer learning and fine-tuning on pre-trained models to improve accuracy.