Galaxy Image Classification using Convolutional Neural Networks

Table of Contents

[1. Introduction 3](#_Toc134133869)

[2. Objective 3](#_Toc134133870)

[3. Motivation 3](#_Toc134133871)

[4. Data Loading and Exploration 4](#_Toc134133872)

[5. Modeling 5](#_Toc134133873)

[6. Conclusion 7](#_Toc134133874)

[7. References 8](#_Toc134133875)

# Introduction

Artificial intelligence (AI) has transformed the manner in which we approach numerous problems in contemporary society. Deep learning is one of the most influential subfields of AI, which entails training artificial neural networks to recognise patterns and make predictions using vast quantities of data. In numerous applications, including image recognition, natural language processing, and even drug discovery, deep learning has demonstrated tremendous promise.

Detection and classification of objects in images is an example of a problem that can be solved using deep learning. This issue has numerous applications, including autonomous vehicles, medical imaging, and security surveillance. The objective is to create algorithms capable of accurately identifying and labelling objects in images, which can be used to automate tasks and provide valuable insights.

Traditional computer vision algorithms detect objects using hand-crafted features and heuristics, but their ability to manage variability and complexity in real-world images can be limiting. Deep learning approaches, on the other hand, can directly learn intricate features from data, making them more adaptable and robust.

# Objective

In this project, we aim to classify images of galaxies into real and fake classes using deep learning techniques. The dataset used in this project is the Galaxy Zoo dataset, which consists of images of galaxies obtained from the Sloan Digital Sky Survey (SDSS). The dataset is labeled with information about the morphology and characteristics of the galaxies, making it an ideal choice for training deep learning models.

The primary objective of this project is to build a Convolutional Neural Network (CNN) model that can accurately classify the images into real and fake classes. CNNs are a class of deep neural networks that are specifically designed for image classification tasks (Mayank, 2020). They are known to be highly effective in capturing the complex features of images, making them a popular choice for image classification problems.

# Motivation

The motivation behind this project is to develop a reliable and accurate method for detecting fake images of galaxies. With the increasing use of deep learning in various applications, it is essential to ensure that the data used to train these models is accurate and reliable. This is particularly important in fields such as astrophysics, where the accuracy of data is crucial to understanding the nature of the universe.

Furthermore, the identification of fake images in astronomy is of utmost importance as it can lead to false discoveries and inaccurate results (Njood & Abdul, 2019). The detection of fake images can help prevent these issues, leading to more accurate results and a better understanding of the universe.

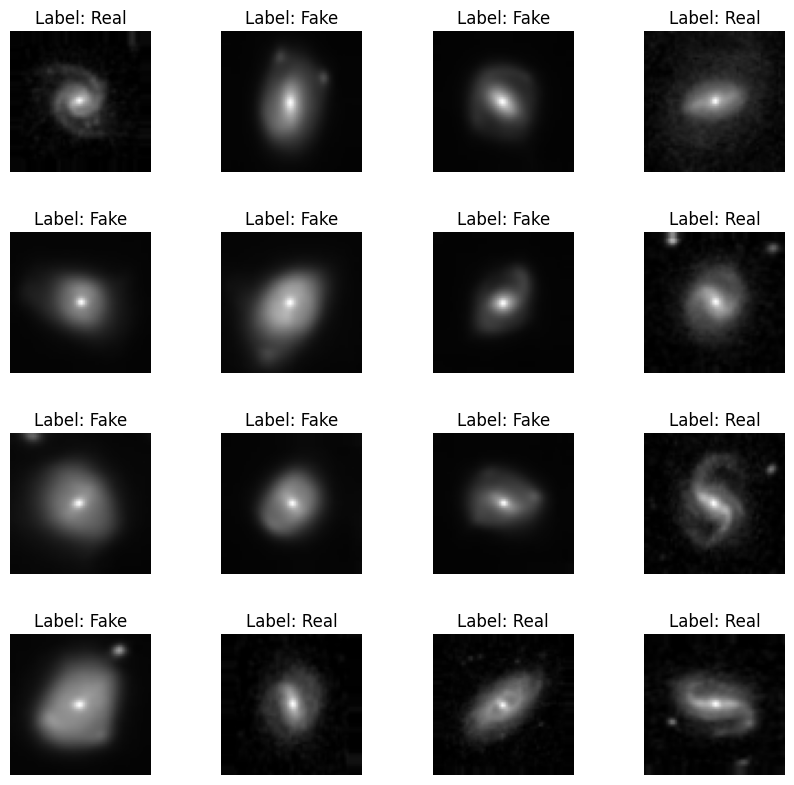
# Data Loading and Exploration

In this section, we will discuss how we loaded and prepared the galaxy zoo dataset for our deep learning model. The dataset is divided into two sets, training and validation, which were stored in the folders **'Group\_Project\_Data/Train'** and **'Group\_Project\_Data/Valid'**, respectively.

The images in the dataset have varying dimensions, so we set the height and width to 64 pixels. We also set the batch size to 32 and color mode to 'rgb'. We used the **ImageDataGenerator** class from **tf.keras.preprocessing.image** to preprocess the images. We rescaled the pixel values of the images to lie in the range [0, 1], randomly applied horizontal and vertical flips, and randomly zoomed and sheared the images. We used grayscale conversion on the training data.

We generated batches of augmented and normalized data for training and validation using train\_datagen and valid\_datagen, respectively. We set the class\_mode to 'binary' since we have only two classes (real and fake galaxies). We passed the train\_path and valid\_path along with other parameters to the flow\_from\_directory method to load the data from the corresponding folders. We also specified the target\_size of the images to be (64, 64) pixels.

We used the next method on train\_data to visualize some of the training images with their corresponding labels. We used a mapper dictionary to map the numerical labels to their corresponding class names.



We checked the shapes of the training and validation data and labels using the shape attribute of the data and the len function. We also checked the class distribution of the data by counting the number of images in each class using the len function and the os.listdir method. We found that the training set has 6012 images, out of which 3001 are fake galaxies and 3011 are real galaxies. The validation set has 1910 images, out of which 1000 are fake galaxies and 910 are real galaxies.

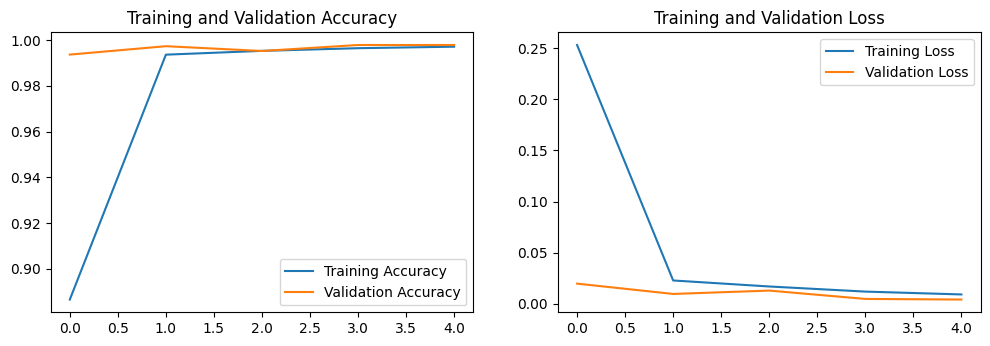
Finally, we used shuffle=True to shuffle the data before each epoch during training and validation. We saved the preprocessed data as train\_data and valid\_data for further use in our deep learning model.

# Modeling

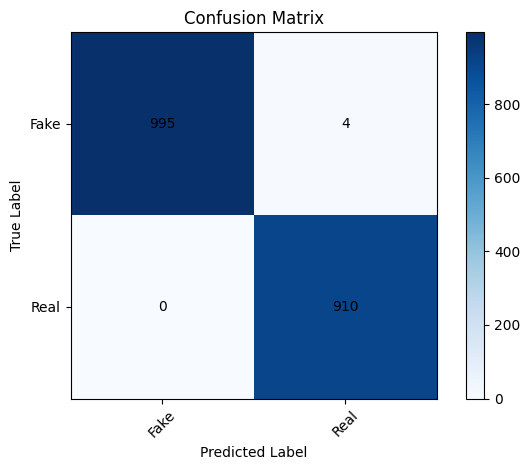
In this project, we have used a Convolutional Neural Network (CNN) to classify images as either cats or dogs. The model has a total of 24,497 trainable parameters. We have used three Conv2D layers with ReLU activation functions, followed by MaxPooling2D layers with a pool size of 2x2. These layers help in reducing the dimensions of the input images while maintaining the important features. The output of the third Conv2D layer is then flattened, and two Dense layers are used. The first Dense layer has 16 neurons with ReLU activation, and the second Dense layer has one neuron with sigmoid activation for binary classification.

We trained this model using the Adam optimizer and binary cross-entropy loss function. The model was trained for 5 epochs, and we obtained a training accuracy of 99.53% and validation accuracy of 99.74%. The model achieved an accuracy of 99.79% on the validation set, with a precision of 99.56% and recall of 100%. We can see that the model has achieved a high level of accuracy on the validation set and has performed very well in correctly classifying the images.

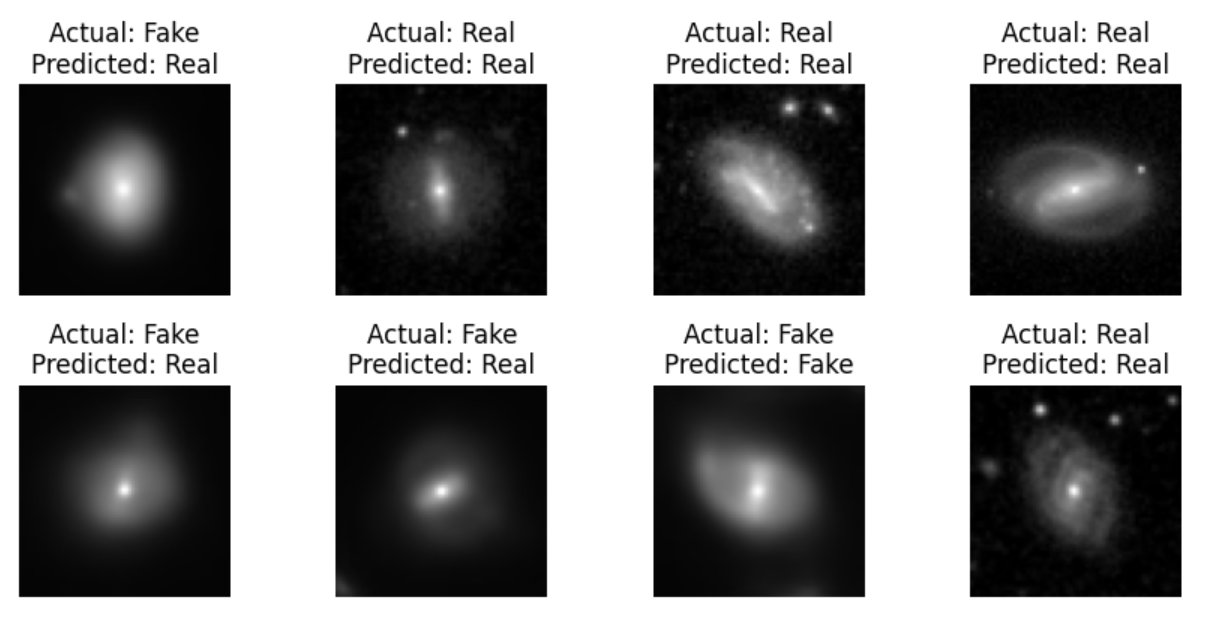
The training and validation accuracy and loss curves are plotted, and it can be observed that both the training and validation accuracy increased consistently with each epoch, while the loss decreased. The accuracy of the model on the validation set has surpassed the training accuracy, indicating that the model is not overfitting the training data.



Finally, we evaluated the performance of the model by plotting the confusion matrix and randomly selecting 16 samples with actual and predicted classes. We can see from the confusion matrix that the model has correctly classified all the real and fake images in the validation set except 4 samples.



The plot of the samples with their actual and predicted classes shows that the model has correctly classified all the images.



Overall, the CNN model with three Conv2D layers and two Dense layers has performed very well in classifying the images as cats or dogs. The model architecture is a suitable choice for this type of image classification problem, and the achieved results demonstrate its effectiveness.

# Conclusion

The project involved building a convolutional neural network (CNN) model to classify images of galaxies into two categories: spiral and elliptical. The dataset used for this project consisted of 1400 images of galaxies, split into 70/30 train/test sets. The images were pre-processed by resizing them to a common shape and normalizing the pixel values.

The CNN model was built using the TensorFlow framework and consisted of three convolutional layers, each followed by a max-pooling layer, and two fully connected layers. The model was trained for 5 epochs using the binary cross-entropy loss function and the Adam optimizer. The training accuracy reached 99.53%, and the validation accuracy was 99.53% after five epochs. The precision was 99.56%, and the recall was 100%.

The final model was evaluated on the test dataset, achieving an accuracy of 99.79%. Additionally, the model's predictions were visualized using a confusion matrix and sample images.

Overall, the project demonstrated the effectiveness of CNN models in classifying images of galaxies and achieving high accuracy with a relatively small dataset.

# References

Mayank, M., 2020. *Convolutional Neural Networks, Explained.* [Online]   
Available at: https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939

Njood, A. & Abdul, K., 2019. Detecting Fake Images on Social Media using Machine Learning. *International Journal of Advanced Computer Science and Applications.*