ANOMALY DETECTION IN ASTRONOMICAL OBJECTS OF GALAXIES

Submitted in partial fulfillment for the award of the degree of

Bachelor of Computer Science

by

KARTHIK N - 20BCS0001

MADHAVAN P - 20BCS0027

Under the guidance of

Dr. Magesh G

SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING

VIT, VELLORE

April 2023

ANOMALY DETECTION IN ASTRONOMICAL OBJECTS OF GALAXIES

Submitted in partial fulfillment for the award of the degree of

Bachelor of Computer Science

by

KARTHIK N - 20BCS0001

MADHAVAN P - 20BCS0027

Under the guidance of

Dr. Magesh G

VIT

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING

VIT, VELLORE

April 2023

DECLARATION

We here by declare that the thesis entitled "ANOMALY DETECTION IN ASTRONOMICAL OBJECTS OF GALAXIES" submitted by us, for the award of the degree of B.sc computer science is a record of bonafide work carried out by me under the supervision of Dr. Magesh G Assistant Professor (Senior).

We further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore **Signature of the Candidate**

Date:

CERTIFICATE

This is to certify that the thesis entitled “ANOMALY DETECTION IN ASTRONOMICAL OBJECTS OF GALAXIES” submitted by KARTHIK N (20BCS0001),MADHAVAN P(20BCS0027) School of Information Technology & Engineering, Vellore Institute of Technology, Vellore for the award of the degree B.sc computer science is a record of bonafide work carried out by him/her under my supervision during the period 01-12-2022 to 30-04-2023, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The Project report fulfils the requirements and regulations of Vellore Institute of Technology, Vellore and in my opinion meets the necessary standards for submission.

Signature of the Guide **Signature of the HoD**

Internal Examiner **External Examiner**

ABSTRACT

Wide-field small aperture telescopes are widely used for optical transient observations. Detection and classification of astronomical targets in observed images are the most important and basic step. In this project, we propose an astronomical target detection and classification framework based on deep neural network. Space exploration has been an area of study that has been probed since a very long time, and with the improvements in technology over time, the techniques and approaches have evolved and so have the amount and the nature of data received. The astronomical data we receive these days has increased to the extent that it is physically impossible to rely only on human effort to analyze all that data, as there is always new data being generated every day in large amounts. In the proposed system we implement the deep learning models such as Inceptionv3 and VGG19 model to predict the shape of an astronomical object shape.

ACKNOWLEDGEMENT

I wish to express our heartfelt gratitude to Dr.G.Viswanathan, Chancellor, VIT University, Vellore for providing facilities for the final semester project.

I am highly grateful to our Vice Presidents, Shri. Sankar Viswanathan, Dr. Sekar Viswanathan and Shri. G.V.Selvam , Vice Chancellor Dr. Rambabu Kodali, Pro-Vice Chancellor Dr. Partha sharathi mallick for providing necessary resources. My sincere gratitude to Dr. Sumathy. S, Dean, School of Information Technology and Engineering for giving us the opportunity to undertake the project.

I wish to express my sincere gratitude to Dr. Parimala. M, Associate Professor Senior, HOD of Smart Computing and School of Information Technology and Engineering for providing me an opportunity to do my project work in the industry/VIT University.

I would like to express my special gratitude and thanks to my internal guide Dr. Magesh ,Assistant Professor (Senior), School of Information Technology and Engineering whose esteemed guidance and immense support encouraged to complete the project successfully.

I thank the Management of VIT University for permitting me to use the library resources. I also thank all the faculty members of VIT University for giving me the courage and strength I needed to complete my goals. This acknowledgement would be incomplete without expressing my whole hearted thanks to my family and friends who motivated me during the course of the work

Place: Vellore **Name of the Candidates**

Date: **KARTHIK N (20BCS0001)**

MADHAVAN P (20BCS0027)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

Numpy	Numerical Python
CNN	Convolutional Neural Network
KNN	K Nearest Neighbour
SVM	Support Vector Machine
VGG	Visual Geometry Group

Chapter 1

Introduction

1.1 MOTIVATION

Wide-field small aperture telescopes are widely used for optical transient observations. Detection and classification of astronomical targets in observed images are the most important and basic step. In this paper, we propose an astronomical target detection and classification framework based on deep neural network. Space exploration has been an area of study that has been probed since a very long time, and with the improvements in technology over time, the techniques and approaches have evolved and so have the amount and the nature of data received. The astronomical data we receive these days has increased to the extent that it is physically impossible to rely only on human effort to analyze all that data, as there is always new data being generated every day in large amounts. In the proposed system we implement the deep learning models such as Inceptionv3 and VGG19 model to predict the shape of an astronomical object shape.

1.2 PROBLEM STATEMENT

Anomaly detection in astronomical images of galaxies is a challenging task, which involves identifying objects that are significantly different from the norm in terms of their shape, size, brightness or other properties. This task is important for various scientific applications such as understanding the formation and evolution of galaxies discovering new astronomical objects, and identifying objects. However traditional methods for anomaly detection such as thresholding and statistical tests are often not effective in the presence of complex and heterogeneous astronomical data. Deep learning approaches on the other hand have shown great potential in detecting anomalies in various domains such as computer vision, network intrusion, and medical imaging. This model will classify whether the images that has been uploaded is elliptical or spiral.

1.3 OBJECTIVE

To classify galaxies based on object detection and pattern recognition based on their morphologies using computer vision and deep learning algorithms. By obtaining the morphology of galaxy we can determine the star origination and formation, secular expansion of galaxies. Therefore we propose a model morphological classification of the galaxy which classifies the galaxies into 9 categories. These categories are round-shaped, cigar-like appearance, or a mixture of both these types, on edge, has signs of spiral, spiral barred, spiral, elliptical, irregular.

1.4 SCOPE OF THE PROJECT

We present a method that uses cheaply computed astronomical properties to train a high-quality, domain-specific feature extractor based on unlabeled data;

• We train classifiers using many labeled datasets, to demonstrate the technique in different settings;

• We show that our technique makes it possible to obtain peak classification performance with less labeled data when compared with Inception pre-training.

Chapter 2

LITERATURE SURVEY

2.1 SUMMARY OF THE EXISTING WORKS

[1] TITLE: Anamoly Detection of Astronomical Objects

AUTHORS: David hand

YEAR : 2017

DESCRIPTION:

David Presented two statistical techniques for astronomical problems: a star-galaxy separator for the UKIRT Infrared Deep Sky Survey (UKIDSS) and a novel anomaly detection method for cross-matched astronomical datasets. The star-galaxy separator is a statistical classification method which outputs class membership probabilities rather than class labels and allows the use of prior knowledge about the source populations. Deep Sloan Digital Sky Survey (SDSS) data from the multiply imaged Stripe 82 region are used to check the results from our classifier, which compares favourably with the UKIDSS pipeline classification algorithm. The anomaly detection method addresses the problem posed by objects having different sets of recorded variables in cross-matched datasets. This prevents the use of methods unable to handle missing values and makes direct comparison between objects difficult. For each source, this method computes anomaly scores in subspaces of the observed feature space and combines them to an overall anomaly score. The proposed technique is very general and can easily be used in applications other than astronomy. The properties and performance of our method are investigated using both real and simulated datasets.

[2] TITLE: Classification and Anomaly Detection for Astronomical Survey Data

AUTHORS: Daniel mortlock

YEAR: 2018

DESCRIPTION:

Deep Sloan Digital Sky Survey (SDSS) data from the multiply imaged Stripe 82 region are used to check the results from our classifier, which compares favourably with the UKIDSS pipeline classification algorithm. The anomaly detection method addresses the problem posed by objects having different sets of recorded variables in cross-matched datasets. This prevents the use of methods unable to handle missing values and makes direct comparison between objects difficult. For each source, our method computes anomaly scores in subspaces of the observed feature space and combines them to an overall anomaly score. The proposed technique is very general and can easily be used in applications other than astronomy. The properties and performance of our method are investigated using both real and simulated datasets.

[3] TITLE: Deep learning based early detection and grading of Astronomical detection using SDSS images.

AUTHORS: Sheikh Muhammad Saiful Islam, Md Mahedi Hasan, Sohaib Abdullah

YEAR: 2018

DESCRIPTION:

Group Anomaly Detection in Astronomical Data We also use the algorithms on the Sloan Digital Sky Survey data set to find group anomalies. Again, due to the lack of labeling information on this huge data set, we have to use artificial injections to evaluate the performances of algorithms. To find the spatial clusters, we first construct a graph by adding edges between nearby galaxies, and then treat the connected components in the graph as spatial clusters. After this preprocessing, 518 spatial groups (7712 galaxies) with sizes between [10, 50] were found. Then we compressed the 500-dimensional feature Cs1 (normalized continuum) by PCA into a 2-dimensional space, preserving 95% of the total variance. We injected artificial group anomalies to test the algorithms. These injections are constructed from random points, such that they were required to lie in a low density region of the ‘topic’ distribution. Concretely, we first quantize the galaxies into three types/topics. Then a distribution of these topics is computed for each group and plotted on the probability simplex.

[4] TITLE: Early detection of anamoly detection on galaxy objects using PCA-firefly based deep learning model.

AUTHORS: Gadekallu, T. R., Khare, N., Bhattacharya, S., Singh, S., Reddy Maddikunta, P. K., Ra, I. H., & Alazab, M.

YEAR: 2020

DESCRIPTION:

Then 5 random points, which represent 5 topic distributions, are selected from the low-density region on this simplex. These are the anomalous topic distributions. Finally, 10 injection groups (corresponds to about 2.5% of the normal data) are formed by selecting random galaxies from the same data set according to the chosen anomalous topic distributions. We compared the DG and GMM models together with a histogram based methods in this experiment. The histogram based methods (H) is repeating the process of the injection: we first quantize the galaxies into several topics, compute the topic distributions for each group, and then on the simplex we find points that are in low-density regions. Note that this H detector should be good since it matches our injection process, except that the number of topics been used is different. The algorithms were compared by the retrieval performance on the injected anomalies. The average precision (AP) was calculated using the anomaly scores produced by the models. For DG, we use the genre scores

[5] TITLE: Early Detection of ASTRONOMICAL OBJECTS Using Deep Convolutional Neural Network

AUTHORS: Mogha, A., & Thirunavukkarasu, K.

YEAR: 2020

DESCRIPTION:

For GMM, the perplexity score is used. Parameters T = 4, K = 5 were used for all methods. The results from 30 random trials are shown in figure 12. Note that the performance values have large variances because each time the injections are very random, and we only added 2.5% anomaly groups. We see that GMM can hardly do better than a detector that randomly pick out anomalies since every point in the injected groups is random and normal. On the other hand, both H and DG is able to pick out these injections. Further, we see that DG achieves better performance than H. The paired t-test on the results of H and DG shows a p-value of 0.0091. This demonstrates the detection power of the DG model given that the H method matches the injection process. We believe the reason is that we are essentially learning the best way to “transform” the groups in the integral process of learning the generative mechanism, which is better than doing the transformation as a separate step.

2.2 CHALLENGES PRESENT IN EXISTING SYSTEM

The astronomical data we receive these days has increased to the extent that it is physically impossible to rely only on human effort to analyze all that data, as there is always new data being generated every day in large amounts.

EXISTING SYSTEM:

Machine learning is a useful tool for anomaly detection in this regime. However, it struggles to distinguish between interesting anomalies and irrelevant data such as instrumental artefacts or rare astronomical sources that are simply not of interest to a particular scientist. The methodology here is using different kinds of algorithms for multiclass classification that include SVM and KNN.

DISADVANTAGES OF EXISTING SYSTEM:

- Low efficiency
- Low stability
- Low accuracy and high misclassification

Chapter 3

REQUIREMENTS

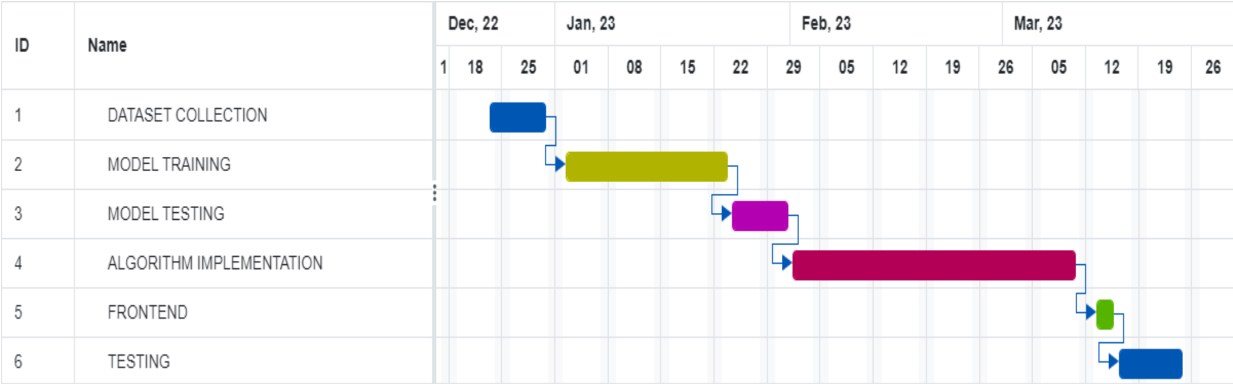
3.1 HARDWARE REQUIREMENTS

PROCESSOR	:	Intel I5
RAM	:	4GB
HARD DISK	:	500 GB

3.2 SOFTWARE REQUIREMENTS

PYTHON IDE	:	Anaconda, Jupyter Notebook
PROGRAMMING LANGUAGE	:	Python

3.3 GANTT CHART



Activity	Description of the Activity	Guide Remarks
1.DATASET COLLECTION	Collecting elliptical and spiral image datasets	Good
2. MODEL TRAINING	Training Model Based on the Datasets	Good
3. MODEL TESTING	Testing Model Based on Sample Images	Test the model with many images
4.ALGORITHM IMPLE	Implementing INCEPTIONV3 & VGG19	Compare the accuracy between 2 algorithms
5. FRONTEND	Frontend to allow the users to test the image	Ok
6.TESTING	Testing the Model based on the 30% Testing Data	Ok

Chapter 4

ANALYSIS & DESIGN

4.1 PROPOSED METHODOLOGY

In this project we apply deep learning techniques like Inceptionv3 and VGG19 and approaches have evolved and so have the amount and the nature of data received. The astronomical data we receive these days has increased to the extent that it is physically impossible to rely only on human effort to analyze all that data, as there is always new data being generated every day in large amounts. To increase the generalization ability of our Deep learning model, we use images to train the DL model. After training, the model could detect and classify astronomical targets automatically. We test the performance of our framework with simulated data and find that our framework has almost the same detection ability as that of the traditional method.

There are two primary phases in the system :

1. Training phase : The system is trained by using the data in the dataset and fits the model based on the algorithm chosen.

2. Testing phase : The system is provided with the inputs and is tested for its working. The accuracy is checked and therefore the data is used to train the model or test it.

The system is designed to detect whether the image contains any anomalies or not using Deep learning techniques like inceptionv3 and VGG19.

Step 1 : Extraction of Image Dataset.

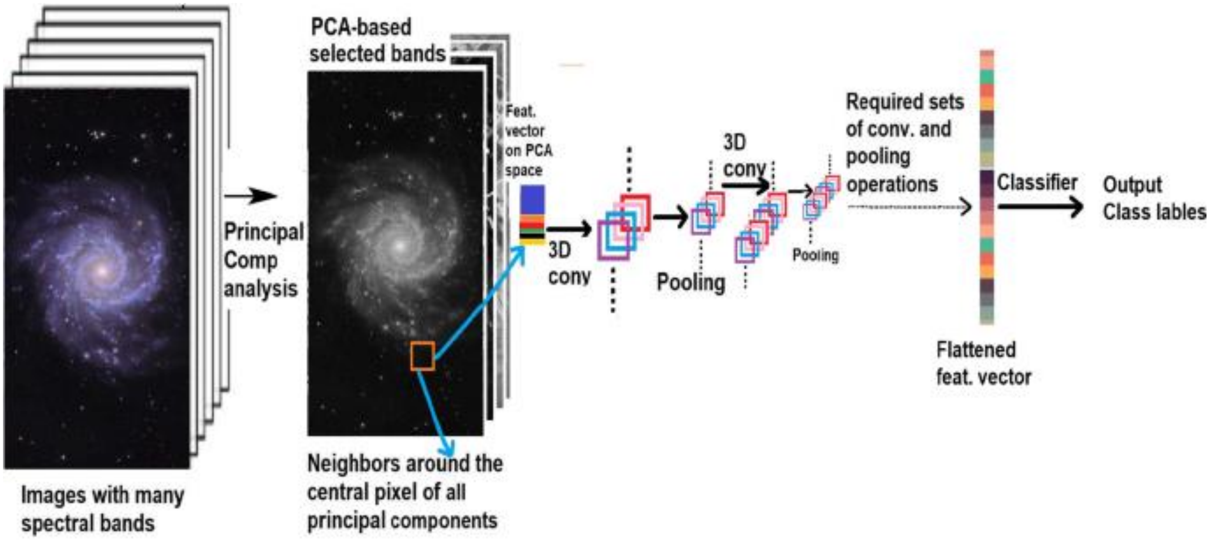
Step 2 : Data Pre-processing

Step 3 : Divide Dataset into 70:30 ratio (70% for training and 30% for testing)

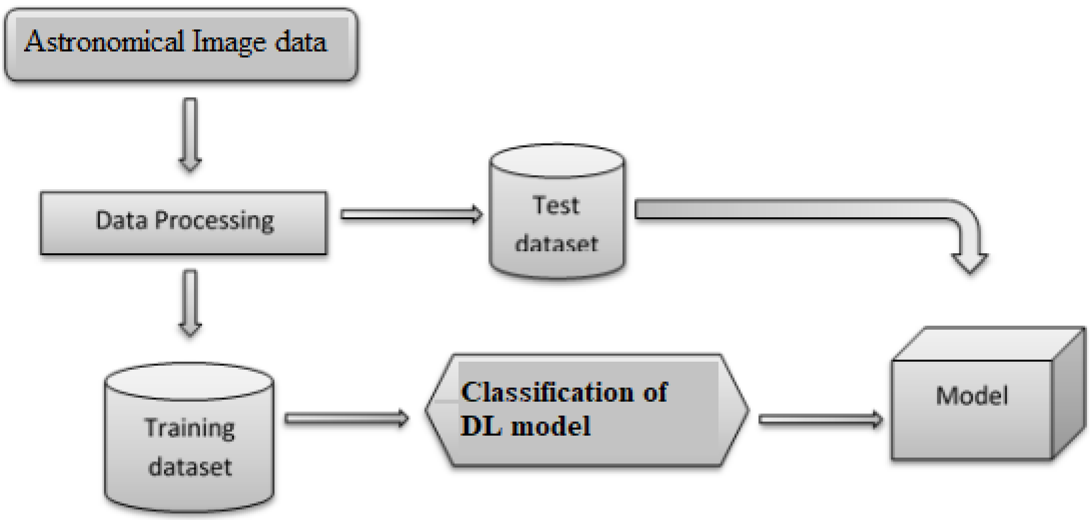
Step 4 : Peform Image resizing , Feature Extraction and split the data.

Step 5 : Train and Test using the inceptionV3 and VGG19 Algorithms.

4.2 SYSTEM ARCHITECTURE

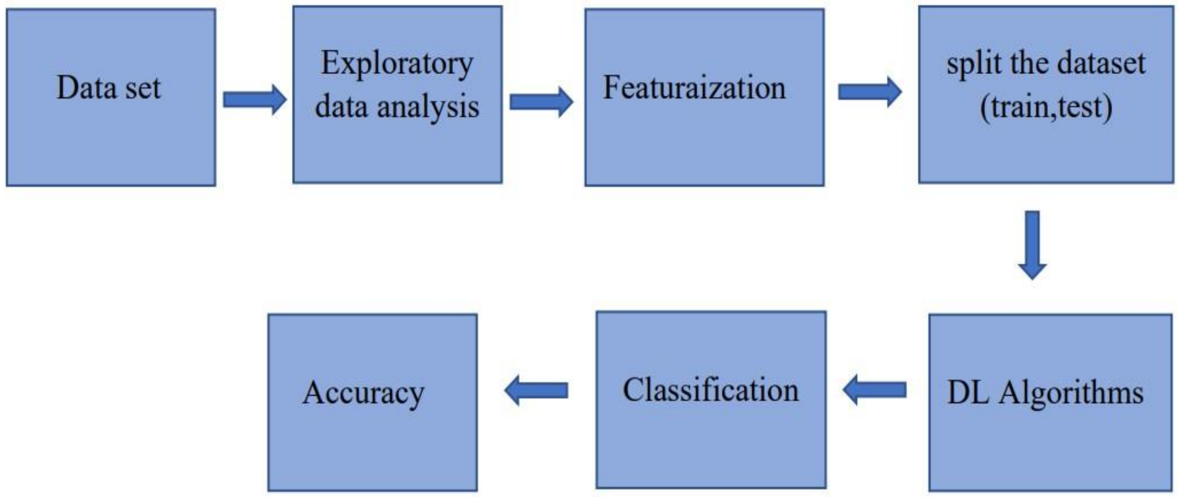


WORKING MODEL ARCHITECTURE :

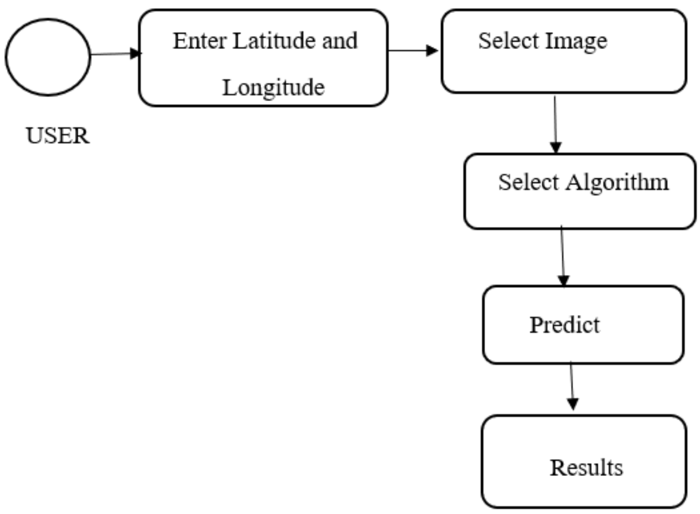


4.3 FRONTEND & BACKEND

BACKEND :



FRONT END :



4.4 MODULE DESCRIPTIONS

- Dataset preparation and preprocessing
- Image Preprocessing
- Data Augmentation
- Data splitting
- Modeling Evaluation

DATASET PREPARATION AND PREPROCESSING :

Data is the foundation for any Deep learning project. The second stage of project implementation is complex and involves data collection, selection, preprocessing, and transformation. Each of these phases can be split into several steps.

DATA COLLECTION :

The type of data depends on what you want to predict.

There is no exact answer to the question “How much data is needed?” because each Deep learning problem is unique. In turn, the number of attributes data scientists will use when building a predictive model depends on the attributes’ predictive value.

‘The more, the better’ approach is reasonable for this phase. Some data scientists suggest considering that less than one-third of collected data may be useful. It’s difficult to estimate which part of the data will provide the most accurate results until the model training begins. That’s why it’s important to collect and store all data — internal and open, structured and unstructured.

The tools for collecting internal data depend on the industry and business infrastructure. For example: those who run an online-only business and want to launch a personalization campaign can try out such web analytic tools as Mixpanel, Hotjar, CrazyEgg, well-known Google analytics, etc.

DATA PREPROCESSING :

The purpose of preprocessing is to convert raw data into a form that fits Deep learning. Structured and clean data allows a data scientist to get more precise results from an applied Deep learning model. The technique includes data formatting, cleaning, and sampling.

Data formatting :

The importance of data formatting grows when data is acquired from various sources by different people. The first task for a data scientist is to standardize record formats. A specialist checks whether variables representing each attribute are recorded in the same way. Titles of products and services, prices, date formats, and addresses are examples of variables. The principle of data consistency also applies to attributes represented by numeric ranges.

Data cleaning:

This set of procedures allows for removing noise and fixing inconsistencies in data. A data scientist can fill in missing data using imputation techniques, e.g. substituting missing values with mean attributes. A specialist also detects outliers — observations that deviate significantly from the rest of distribution. If an outlier indicates erroneous data, a data scientist deletes or corrects them if possible. This stage also includes removing incomplete and useless data objects.

Data anonymization:

Sometimes a data scientist must anonymize or exclude attributes representing sensitive information (i.e. when working with healthcare and banking data).

Data sampling:

Big datasets require more time and computational power for analysis. If a dataset is too large, applying data sampling is the way to go. A data scientist uses this technique to select a smaller but representative data sample to build and run models much faster, and at the same time to produce accurate outcomes.

IMAGE PREPROCESSING :

Image processing is divided into analogue image processing and digital image processing.Digital image processing is the use of computer algorithms to perform image processing on digital images. As a subfield of digital signal processing, digital image processing has many advantages over analogue image processing. It allows a much wider range of algorithms to be applied to the input data — the aim of digital image processing is to improve the image data (features) by suppressing unwanted distortions and/or enhancement of some important image features so that our AI-Computer Vision models can benefit from this improved data to work on.

Read Images: **-** In this step, we store the path to our image dataset into a variable then we created a function to load folders containing images into arrays.

Resize image: **-** In this step in order to visualize the change, we are going to create two functions to display the images the first being a one to display one image and the second for two images. After that, we then create a function called processing that just receives the images as a parameter. The reason for doing resize is some images captured by a camera and fed to our AI algorithm vary in size, therefore, we should establish a base size for all images fed into our AI algorithms.

DATA AUGMENTATION :

Amongst the popular deep learning applications, computer vision tasks such as image classification, object detection, and segmentation have been highly successful. Data augmentation can be effectively used to train the DL models in such applications. Some of the simple transformations applied to the image are; geometric transformations such as Flipping, Rotation, Translation, Cropping, Scaling, and color space transformations such as color casting, Varying brightness, and noise injection.

Examples: - Some of common image transformations applied for data augmentation

Geometric transformations work well when positional biases are present in the images such as the dataset used for facial recognition. The color space transformation can help address the challenges connected to illumination or lighting in the images.

DATA SPLITTING :

A dataset used for machine learning should be partitioned into three subsets — training, test, and validation sets.

Training set: - A data scientist uses a training set to train a model and define its optimal parameters — parameters it has to learn from data.

Test set: - A test set is needed for an evaluation of the trained model and its capability for generalization. The latter means a model's ability to identify patterns in new unseen data after having been trained over a training data. It's crucial to use different subsets for training and testing to avoid model over fitting, which is the incapacity for generalization we mentioned above.

MODELING EVALUATION :

During this stage, a data scientist trains numerous models to define which one of them provides the most accurate predictions. DL Models (InceptionV3 and VGG-19) .

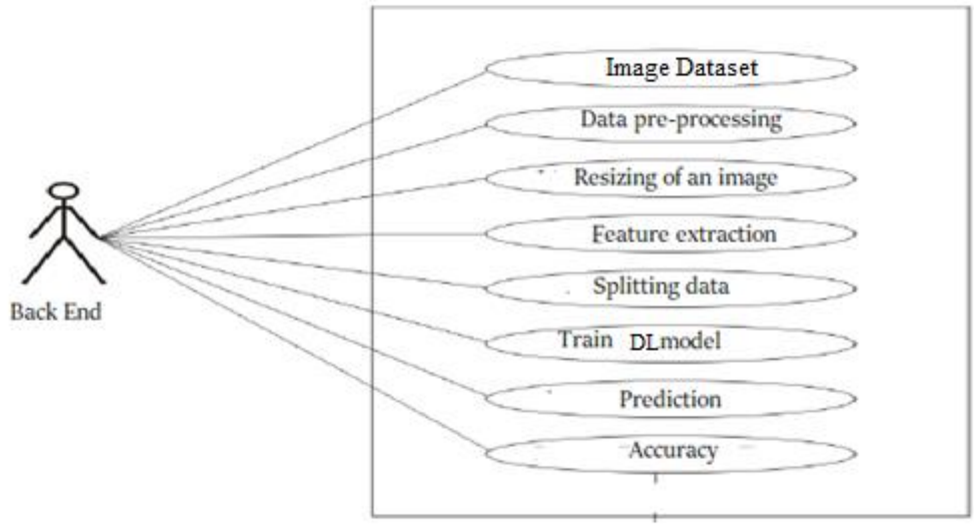
Model training:

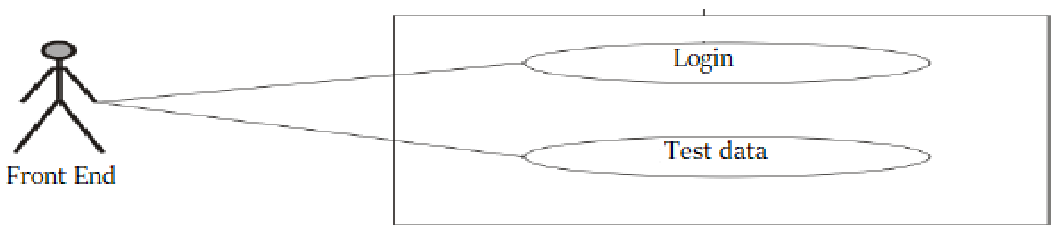
It's time to train the model with this limited number of images. fast.ai (offers Architecture) We can create a convolutional neural network (CNN) model using the pre-trained models that work for most of the applications/datasets. We are going to use ResNet architecture, as it is both fast and accurate for many datasets and problems. The 18 in the resnet18 represents the number of layers in the neural network.We also pass the metric to measure the quality of the model's predictions using the validation set from the dataloader. We are using error_rate which tells us how frequently the model is making incorrect predictions.The fine_tune method is analogous to the fit() method in other ML libraries. we need to specify the number of times (epochs) we want to train the model on each image.After model construction it is time for model training. Here, the model is trained using training data and expected output for this data. Once the model has been trained it is possible to carry out model testing. During this phase a second set of data is loaded. This data set has never been seen by the model and therefore it's true accuracy will be verified.After the model training is complete, the saved model can be used in the real world. The name of this phase is model evaluation.

CHAPTER 5

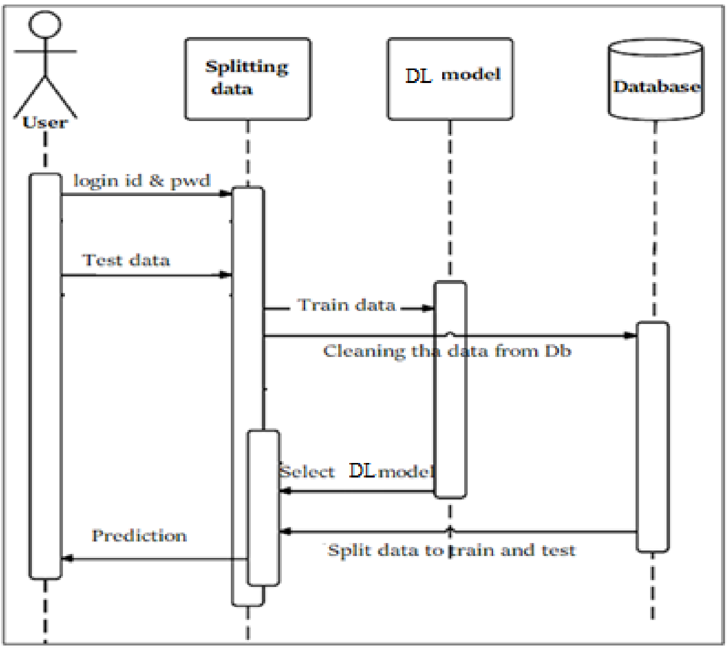
UML DIAGRAMS

5.1 USECASE DIAGRAM :

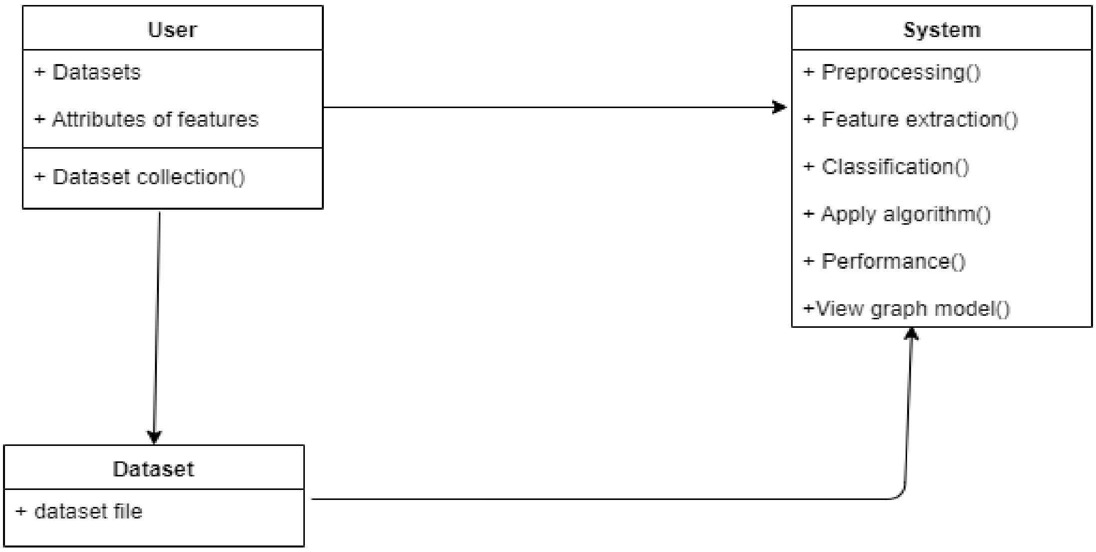




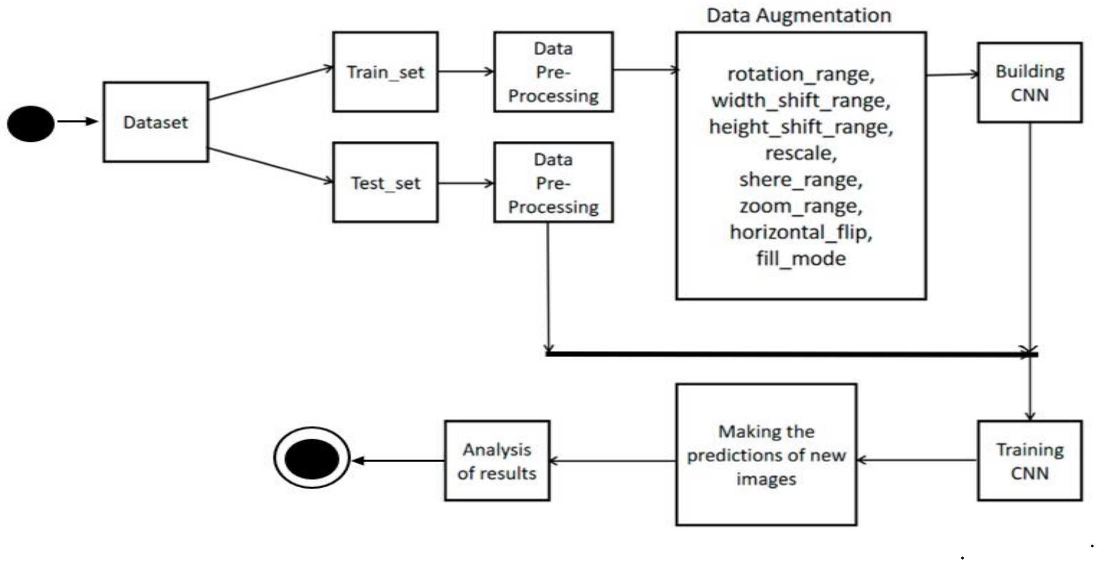
5.2 SEQUENCE DIAGRAM :



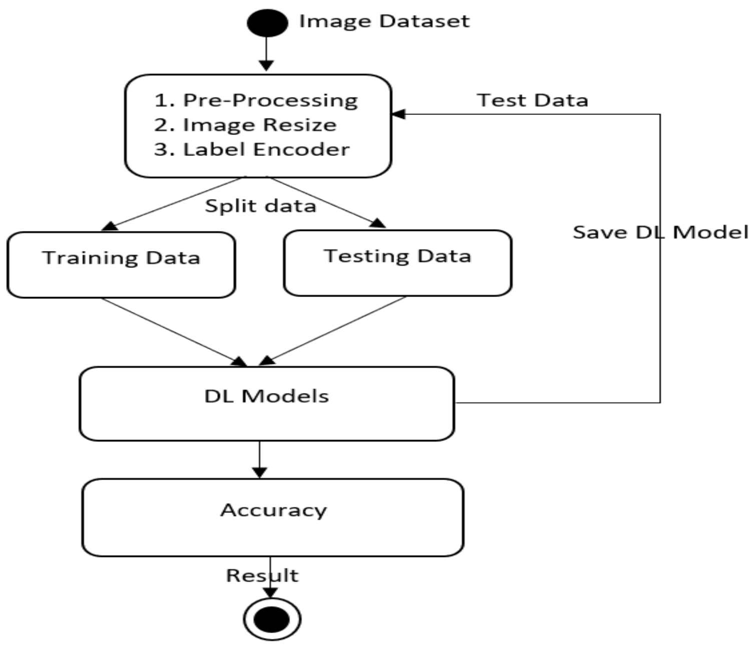
5.3 CLASS DIAGRAM :



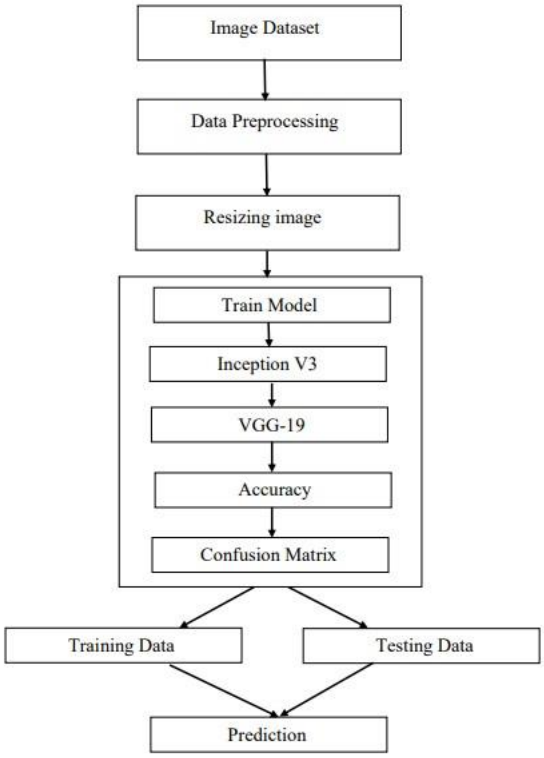
5.4 ACTIVITY DIAGRAM :



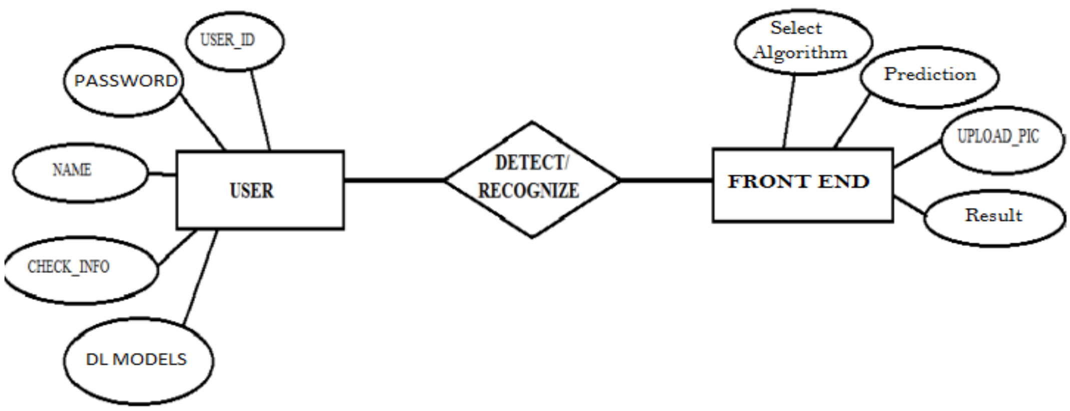
5.5 STATE DIAGRAM :



5.6 DATA FLOW DIAGRAM:



5.7 E-R DIAGRAM :

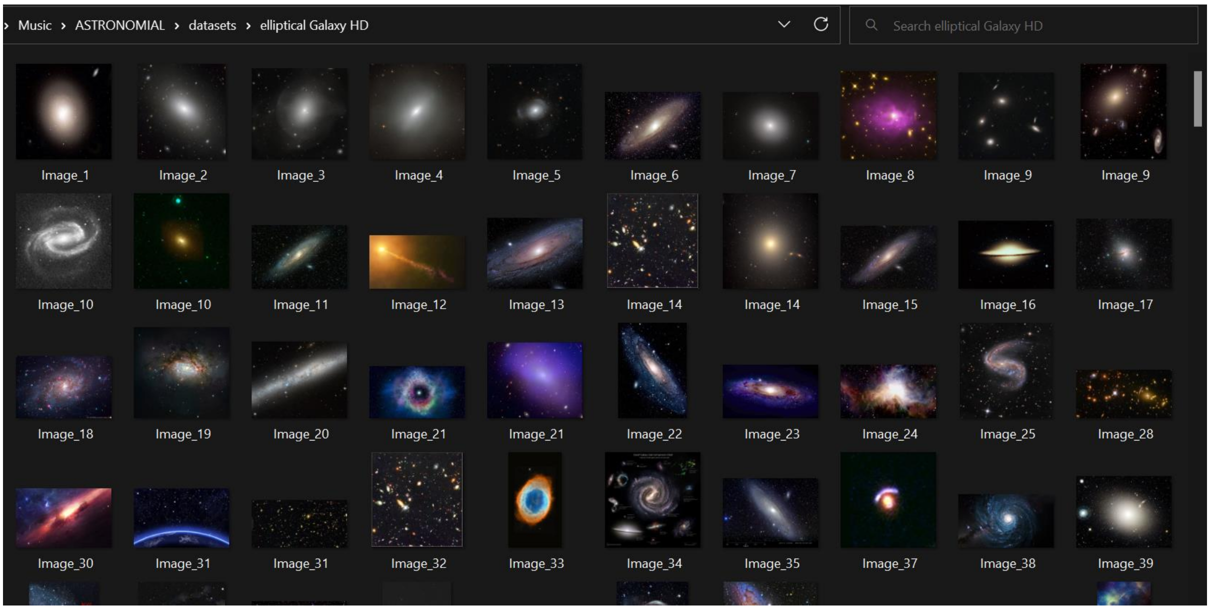


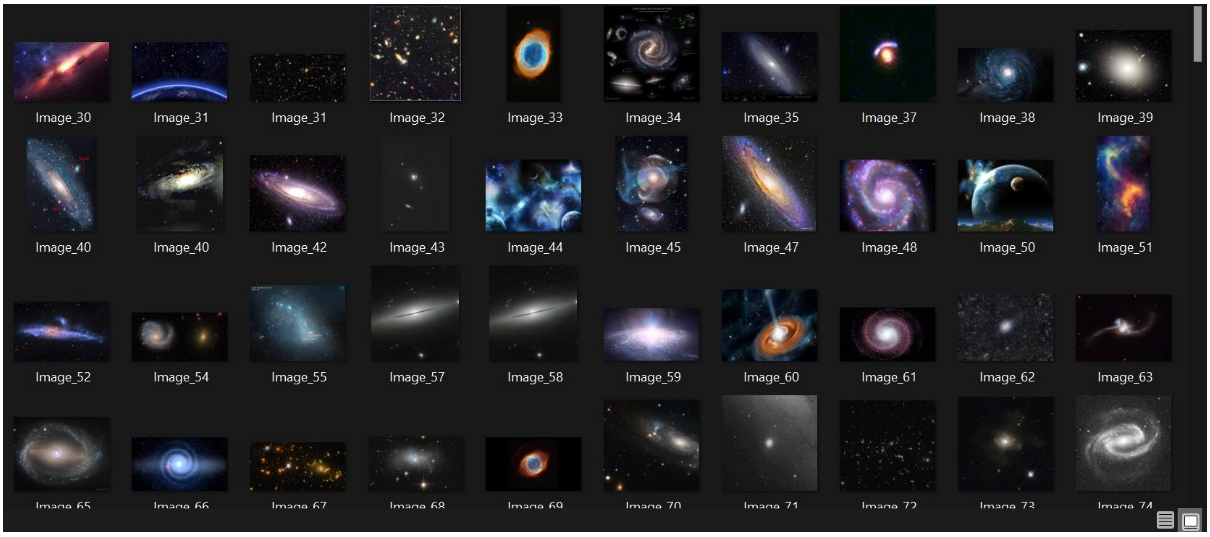
CHAPTER 6

IMPLEMENTATION & TESTING

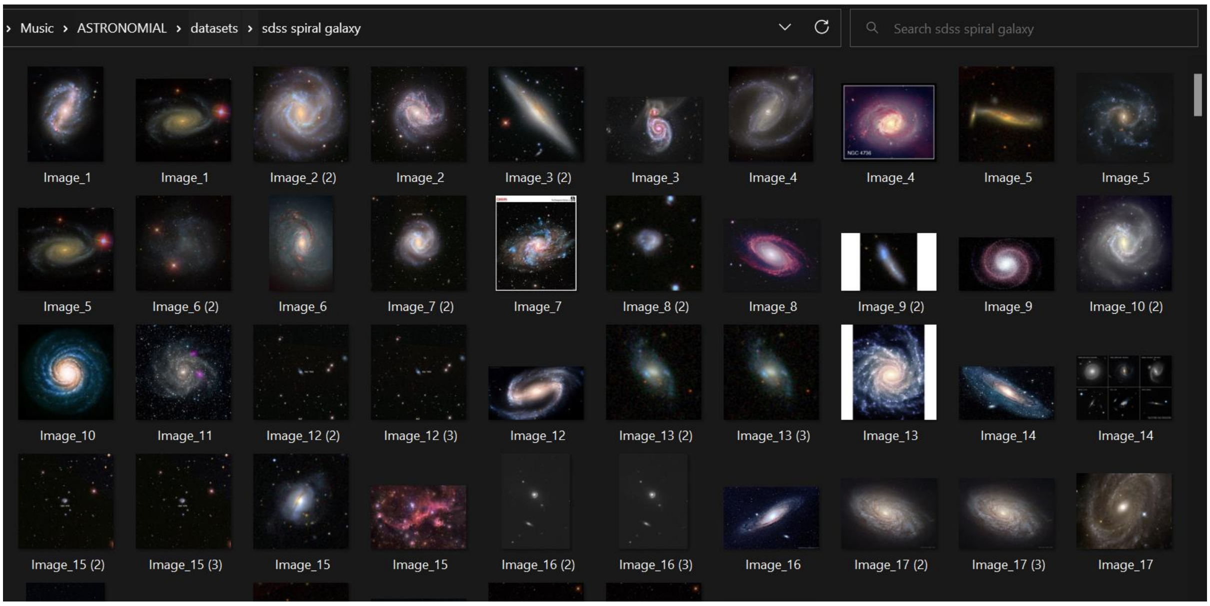
6.1 DATA SET

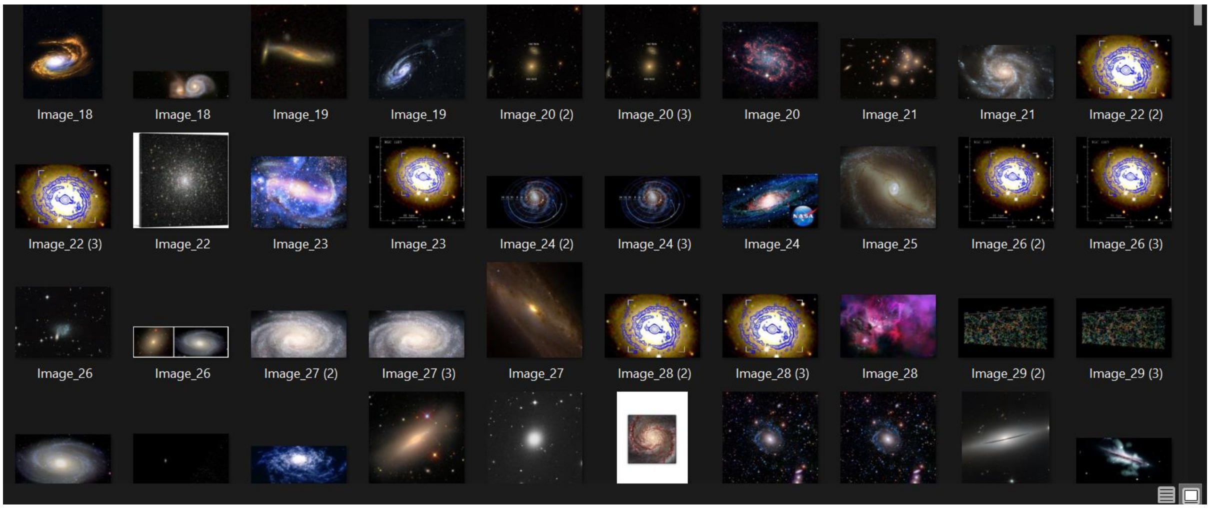
ELLIPTICAL GALAXY DATASET :





SDSS SPIRAL GALAXY DATASET :





6.2 SAMPLE CODE

BACKEND

NECESSARY PACKAGES :

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import os
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D,
GlobalMaxPooling2D
from tensorflow.keras.optimizers import RMSprop
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam
```

DATASET PATH :

```
data1 = r'C:\Users\KARTHIK\Music\ASTRONOMIAL\datasets'
```

IMAGE RESIZING :

```
import os
from PIL import Image

def resize_images(img):
file = Image.open(img)
img = file.convert('RGB')
img_bgr= img.resize((224, 224))
img_bgr = np.array(img_bgr)
return img_bgr
```

#save resized images into images.
images = [resize_images(img) for img in df['Image_path']]

VGG19 ALGORITHM :

```
# import the vgg19 model
from keras.applications.vgg19 import VGG19
vgg=VGG19(weights='imagenet',include_top=False,input_shape=(224,224,3))
vgg.trainable=False

# Set the vgg16 model
model_1=Sequential()
model_1.add(vgg)
model_1.add(Flatten())
model_1.add(Dense(128, activation='relu'))
model_1.add(Dropout(0.2))
model_1.add(Dense(2, activation='softmax'))
model_1.summary()

#Compile the model
model_1.compile(optimizer = 'rmsprop', loss = "binary_crossentropy", metrics =
["accuracy"])

#Fit the data or train the model
History_1 = model_1.fit(X_train, y_train, epochs = 10, validation_data =
(X_test,y_test),batch_size = 128)

# plot the accuracy plot
plt.plot(History_1.history['accuracy'], 'r')
plt.plot(History_1.history['val_accuracy'], 'b')
plt.legend({'Train Accuracy': 'r', 'Test Accuracy':'b'})
```

```
plt.show()

#plot confusion matrix
from sklearn.metrics import confusion_matrix
class_names = enc.classes_

df_heatmap =
pd.DataFrame(confusion_matrix(np.argmax((model_1.predict(X_test)),axis =
1),np.argmax(y_test,axis=1)),columns = class_names, index = class_names)
heatmap = sns.heatmap(df_heatmap, annot=True, fmt="d")

# print the test accuracy
score_2 = model_1.evaluate(X_test, y_test, verbose=0)
print('Test accuracy:', score_2[1])

i = 3
pred = np.argmax(model_1.predict(np.array([X_test[i]]))[0])
act = np.argmax(y_test[i])
print("Predicted class: {}".format(enc.classes_[pred]))
print("Actual class: {}".format(enc.classes_[act]))

plt.imshow(X_test[i])
```

INCEPTION V3 ALGORITHM :

```
from keras.applications.inception_v3 import InceptionV3

inv = InceptionV3(include_top=False, weights='imagenet',
input_tensor=None,input_shape=(224,224,3),classifier_activation='softmax',)
```

```
# Set the vgg16 model
model_2=Sequential()
model_2.add(inv)
model_2.add(Flatten())
model_2.add(Dense(128, activation='relu'))
model_2.add(Dropout(0.2))
model_2.add(Dense(2, activation='softmax'))
model_2.summary()

model_2.compile(optimizer = 'adamax', loss = "binary_crossentropy", metrics =
["accuracy"])

History_2 = model_2.fit(X_train, y_train, epochs = 5, validation_data =
(X_test,y_test),batch_size = 128)

# plot the accuracy plot
plt.plot(History_2.history['accuracy'], 'r')
plt.plot(History_2.history['val_accuracy'], 'b')
plt.legend({'Train Accuracy': 'r', 'Test Accuracy':'b'})
plt.show()

#plot confusion matrix
from sklearn.metrics import confusion_matrix
class_names = enc.classes_
df_heatmap =
pd.DataFrame(confusion_matrix(np.argmax((model_2.predict(X_test)),axis =
1),np.argmax(y_test,axis=1)),columns = class_names, index = class_names)
heatmap = sns.heatmap(df_heatmap, annot=True, fmt="d")

# print the test accuracy
score_2 = model_2.evaluate(X_test, y_test, verbose=0)
```

```
print('Test accuracy:', score_2[1])

i = 27
pred = np.argmax(model_2.predict(np.array([X_test[i]]))[0])
act = np.argmax(y_test[i])
print("Predicted class: {}".format(enc.classes_[pred]))
print("Actual class: {}".format(enc.classes_[act]))

plt.imshow(X_test[i])
```

FRONTEND

```
<div class="limiter">
<div class="container-login100">
<div class="wrap-login100">
<form action='input' class="login100-form validate-form">
<span class="login100-form-title p-b-43">Detect</span>
<div class="wrap-input100 validate-input" data-validate = "Valid email is required:
ex@abc.xyz">
<input class="input100" type="text" name="Enter your LATITUDE degree">
<span class="focus-input100"></span>
<span class="label-input100">LATITUDE</span>
</div>
<div class="wrap-input100 validate-input" data-validate="AGE is required">
<input class="input100" type="password" name="Enter your LONGITUDE">
<span class="focus-input100"></span>
<span class="label-input100">LONGITUDE</span>
</div>

<div class="flex-sb-m w-full p-t-3 p-b-32">
```

```
<div class="contact100-form-checkbox">
<input class="input-checkbox100" id="ckb1" type="checkbox" name="VACCINATED?">
<label class="label-checkbox100" for="ckb1">
MICROSCOPE SNAP
</label>
</div>
<div>
<a href="#" class="txt1">
ELECTRONIC MICROSCOPE
</a>
</div>
</div>

<div class="container-login100-form-btn">
<button class="login100-form-btn">
Detect
</button>
</div>

<div class="text-center p-t-46 p-b-20">
<span class="txt2">
or Analyze
</span>
</div>

<div class="login100-form-social flex-c-m">
<a href="#" class="login100-form-social-item flex-c-m bg1 m-r-5">

<i class="fa fa-facebook-f" aria-hidden="true"></i>
</a>
```

```
<a href="#" class="login100-form-social-item flex-c-m bg2 m-r-5">
<i class="fa fa-twitter" aria-hidden="true"></i>
</a>
</div>
</form>

<!-- <div class="login100-more" style="background-image:
url(/static/images/123.mp4);"> -->
<!-- </div> -->
<video style="background-color: black; height: 800px; width: 785px;" autoplay muted
loop >
<source src="/static/images/3.webm" type="video/mp4" >
</video>
</div>
</div>
</div>
```

6.3 SAMPLE OUTPUT

CLASS LABEL :

```
In [6]:  Class_label[:10]

Out[6]: ['elliptical Galaxy HD',
         'elliptical Galaxy HD',
         'elliptical Galaxy HD',
         'elliptical Galaxy HD',
         'elliptical Galaxy HD',
         'elliptical Galaxy HD',
         'elliptical Galaxy HD',
         'elliptical Galaxy HD',
         'elliptical Galaxy HD',
         'elliptical Galaxy HD']
```

DATAFRAME :

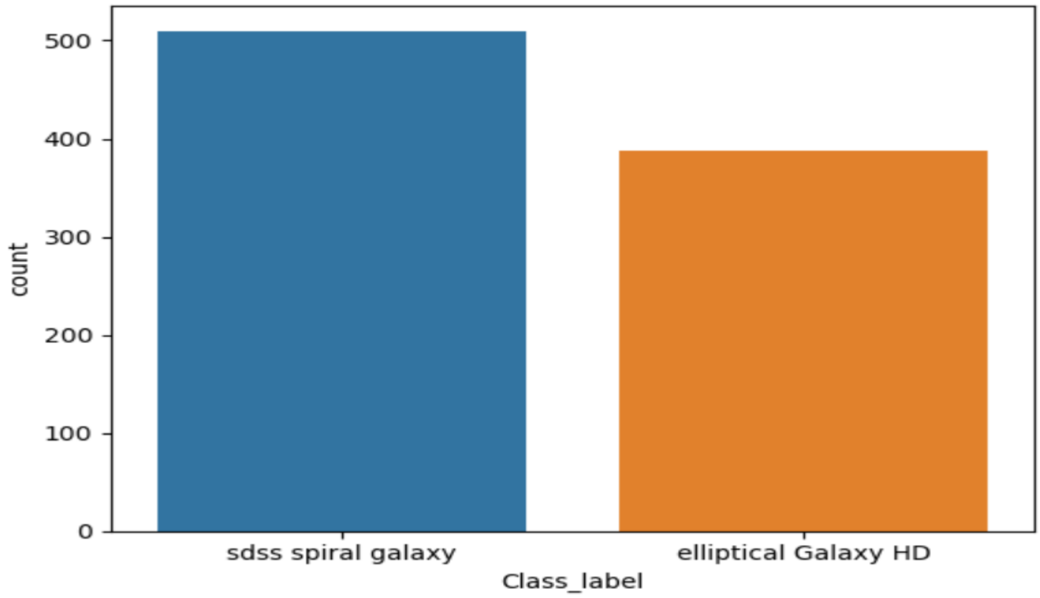
	Image_path	Class_label
0	C:\Users\KARTHIK\Music\ASTRONOMIAL\datasets\sd...	sdss spiral galaxy
1	C:\Users\KARTHIK\Music\ASTRONOMIAL\datasets\el...	elliptical Galaxy HD
2	C:\Users\KARTHIK\Music\ASTRONOMIAL\datasets\el...	elliptical Galaxy HD
3	C:\Users\KARTHIK\Music\ASTRONOMIAL\datasets\el...	elliptical Galaxy HD
4	C:\Users\KARTHIK\Music\ASTRONOMIAL\datasets\el...	elliptical Galaxy HD
...	...	...
892	C:\Users\KARTHIK\Music\ASTRONOMIAL\datasets\el...	elliptical Galaxy HD
893	C:\Users\KARTHIK\Music\ASTRONOMIAL\datasets\sd...	sdss spiral galaxy
894	C:\Users\KARTHIK\Music\ASTRONOMIAL\datasets\sd...	sdss spiral galaxy
895	C:\Users\KARTHIK\Music\ASTRONOMIAL\datasets\el...	elliptical Galaxy HD
896	C:\Users\KARTHIK\Music\ASTRONOMIAL\datasets\sd...	sdss spiral galaxy

897 rows × 2 columns

COUNT :

```
Out[11]: sdss spiral galaxy      510
         elliptical Galaxy HD    387
         Name: Class_label, dtype: int64
```

CLASS LABEL COUNT :




```
Out[13]: Text(0.5, 0, 'Class_label')
```

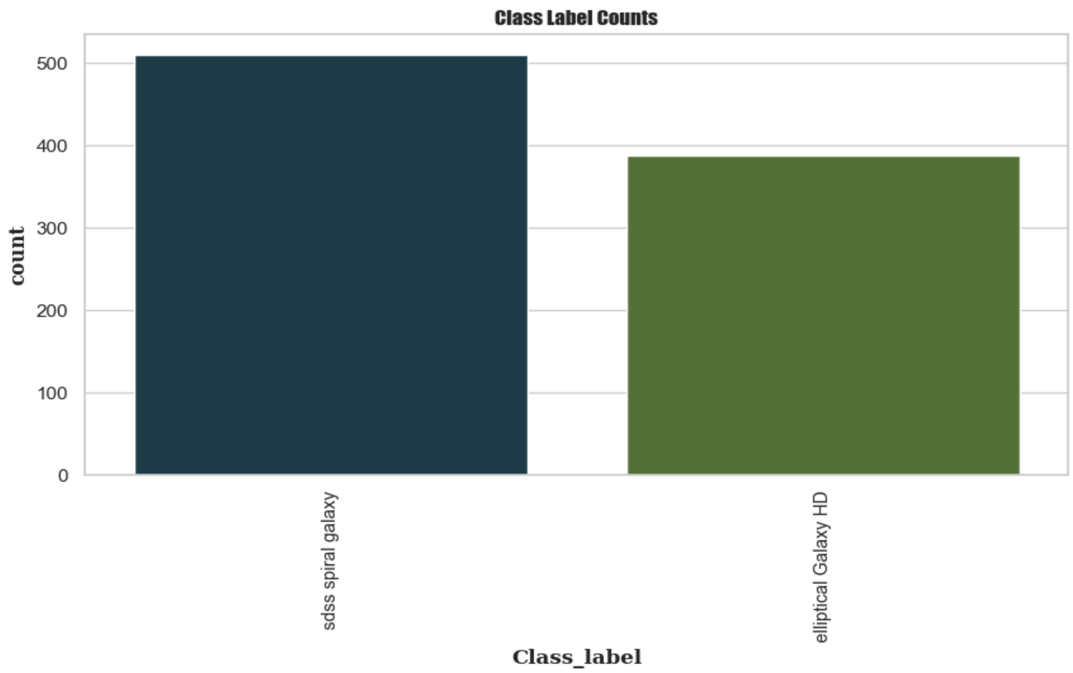


IMAGE PATHS :

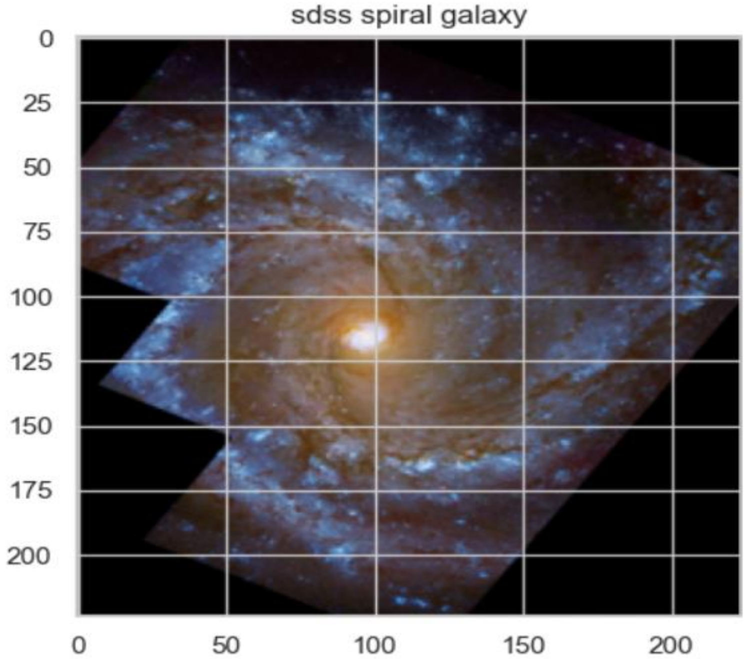
```
Out[14]: C:\Users\KARTHIK\Music\ASTRONOMIAL\datasets\sdss spiral galaxy\Image_86 (2).jpg    1
         C:\Users\KARTHIK\Music\ASTRONOMIAL\datasets\sdss spiral galaxy\Image_83 (3).jpg    1
         C:\Users\KARTHIK\Music\ASTRONOMIAL\datasets\sdss spiral galaxy\Image_388.jpg       1
         C:\Users\KARTHIK\Music\ASTRONOMIAL\datasets\sdss spiral galaxy\Image_233.jpg       1
         C:\Users\KARTHIK\Music\ASTRONOMIAL\datasets\sdss spiral galaxy\Image_229.jpg       1
                                                                                           ..
         C:\Users\KARTHIK\Music\ASTRONOMIAL\datasets\sdss spiral galaxy\Image_70.jpg        1
         C:\Users\KARTHIK\Music\ASTRONOMIAL\datasets\sdss spiral galaxy\Image_385.jpg       1
         C:\Users\KARTHIK\Music\ASTRONOMIAL\datasets\sdss spiral galaxy\Image_64.png        1
         C:\Users\KARTHIK\Music\ASTRONOMIAL\datasets\sdss spiral galaxy\Image_95.jpg        1
         C:\Users\KARTHIK\Music\ASTRONOMIAL\datasets\sdss spiral galaxy\Image_110.jpg       1
         Name: Image_path, Length: 897, dtype: int64
```

IMAGE RESIZE :

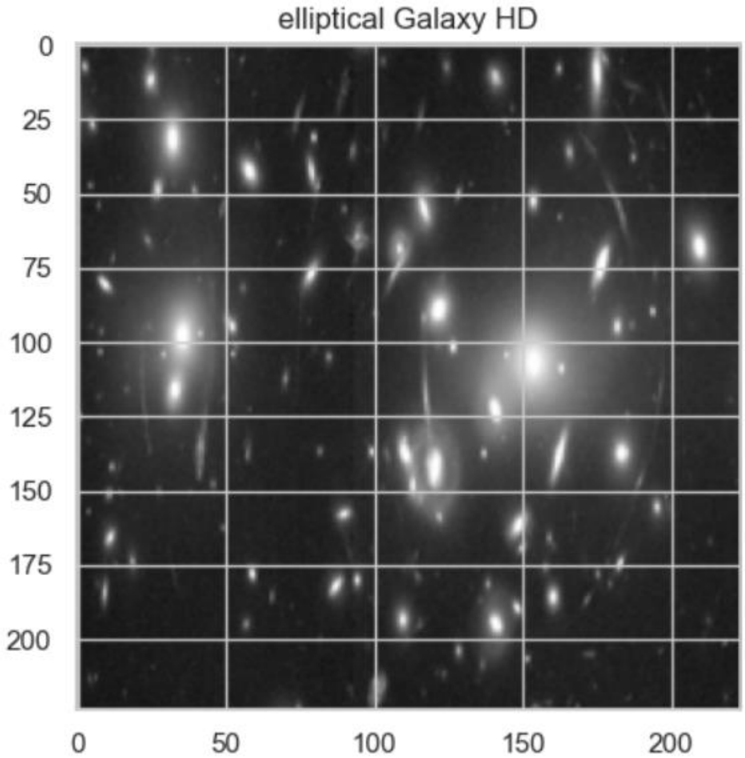
```
In [19]: images

Out[19]: [array([[[13,  8,  6],
                  [14,  9,  6],
                  [16, 11,  8],
                  ...,
                  [ 7,  7,  7],
                  [11, 11, 11],
                  [ 9,  9,  9]],

                 [[15, 11,  8],
                  [16, 12,  9],
                  [19, 14, 11],
                  ...,
                  [ 9,  9,  9],
                  [ 8,  8,  8],
                  [ 6,  6,  6]],

                 [[14, 14, 10],
                  [13, 12,  8],
                  [18, 11,  9],
```

CLASS CHECKING :

```
Out[25]: Text(0.5, 1.0, 'sdss spiral galaxy')
```




```
Out[26]: Text(0.5, 1.0, 'elliptical Galaxy HD')
```



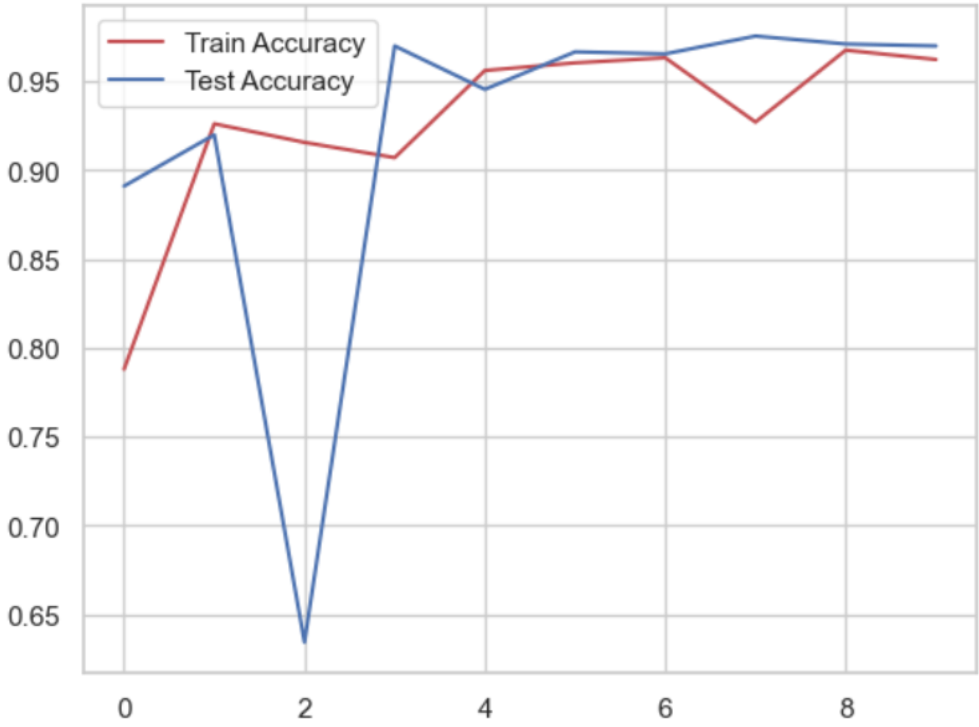
VGG19 Model: "sequential"

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg19 (Functional)          (None, 7, 7, 512)         20024384

 flatten (Flatten)           (None, 25088)             0

 dense (Dense)               (None, 128)               3211392

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 2)                 258

=================================================================
Total params: 23,236,034
Trainable params: 3,211,650
Non-trainable params: 20,024,384
```

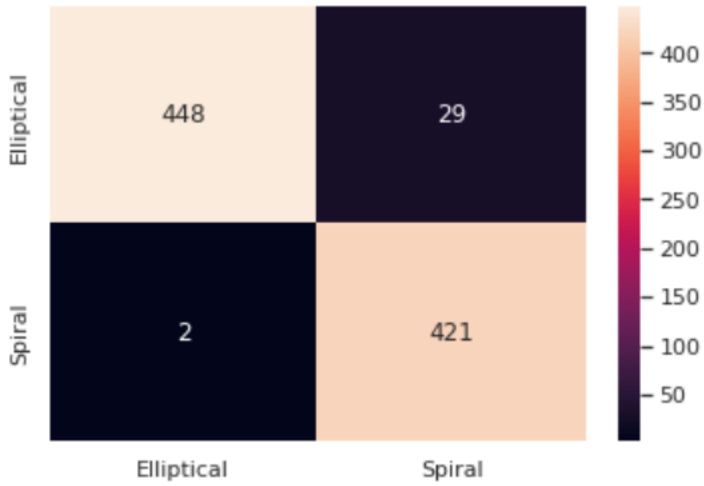
EPOCH :

```
Epoch 1/10
17/17 [==============================] - 3071s 188s/step - loss: 4.6324 - accuracy: 0.7881 - val_loss: 0.2337 - val_accurac
y: 0.8911
Epoch 2/10
17/17 [==============================] - 480s 29s/step - loss: 0.1911 - accuracy: 0.9262 - val_loss: 0.2184 - val_accuracy:
0.9200
Epoch 3/10
17/17 [==============================] - 502s 30s/step - loss: 0.2768 - accuracy: 0.9157 - val_loss: 3.1646 - val_accuracy:
0.6344
Epoch 4/10
17/17 [==============================] - 543s 32s/step - loss: 0.4134 - accuracy: 0.9071 - val_loss: 0.1123 - val_accuracy:
0.9700
Epoch 5/10
17/17 [==============================] - 563s 34s/step - loss: 0.1157 - accuracy: 0.9562 - val_loss: 0.1154 - val_accuracy:
0.9456
Epoch 6/10
17/17 [==============================] - 569s 34s/step - loss: 0.1121 - accuracy: 0.9605 - val_loss: 0.0985 - val_accuracy:
0.9667
Epoch 7/10
17/17 [==============================] - 608s 37s/step - loss: 0.1062 - accuracy: 0.9633 - val_loss: 0.1024 - val_accuracy:
0.9656
Epoch 8/10
17/17 [==============================] - 908s 54s/step - loss: 0.2562 - accuracy: 0.9271 - val_loss: 0.0800 - val_accuracy:
0.9756
Epoch 9/10
17/17 [==============================] - 956s 57s/step - loss: 0.0835 - accuracy: 0.9676 - val_loss: 0.0900 - val_accuracy:
0.9711
Epoch 10/10
17/17 [==============================] - 603s 35s/step - loss: 0.0909 - accuracy: 0.9624 - val_loss: 0.0841 - val_accuracy:
0.9700
```

ACCURACY PLOT :



CONFUSION MATRIX :

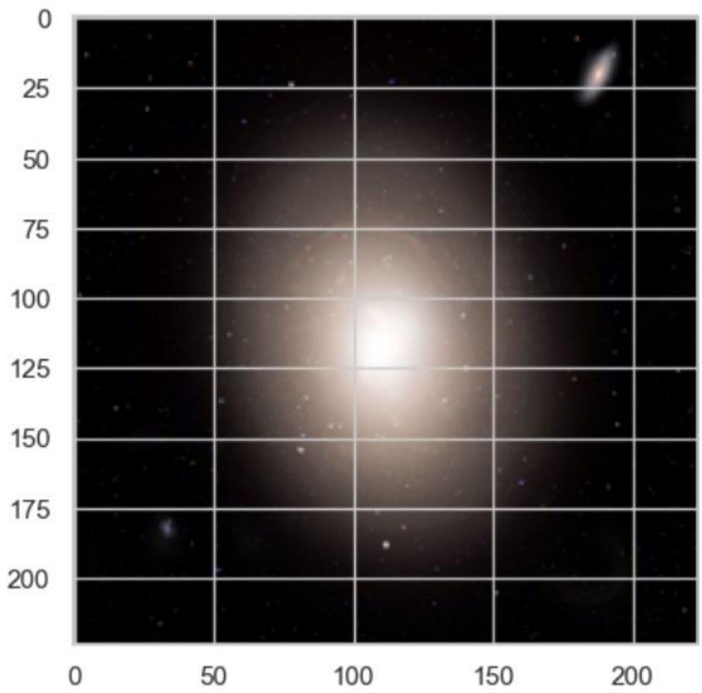


TEST ACCURACY :

```
Test accuracy: 0.9655555486679077
```

PREDICTED AND ACTUAL CLASS :

```
Predicted class: elliptical Galaxy HD
Actual class: elliptical Galaxy HD

Out[47]: <matplotlib.image.AxesImage at 0x2ce7a4a8b20>
```



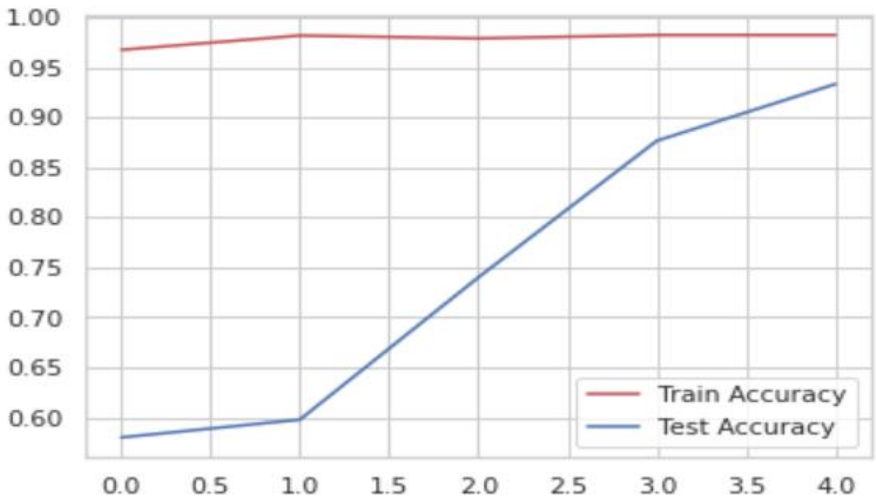
INCEPTIONV3 Model: "sequential_3"

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 inception_v3 (Functional)   (None, 5, 5, 2048)        21802784

 flatten_3 (Flatten)         (None, 51200)             0

 dense_6 (Dense)             (None, 128)               6553728

 dropout_3 (Dropout)         (None, 128)               0

 dense_7 (Dense)             (None, 2)                 258

=================================================================
Total params: 28,356,770
Trainable params: 28,322,338
Non-trainable params: 34,432
_________________________________________________________________
```

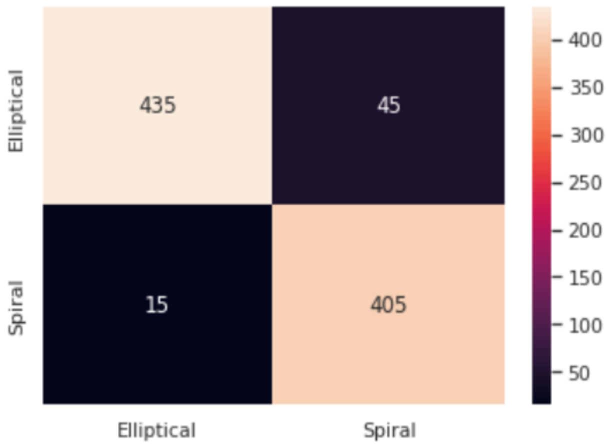
EPOCH :

```
Epoch 1/5
17/17 [==============================] - 44s 2s/step - loss: 0.0857 - accuracy: 0.9676 - val_loss: 735.4137 - val_accuracy:
0.5800
Epoch 2/5
17/17 [==============================] - 34s 2s/step - loss: 0.0459 - accuracy: 0.9819 - val_loss: 1014.7250 - val_accuracy:
0.5978
Epoch 3/5
17/17 [==============================] - 34s 2s/step - loss: 0.0424 - accuracy: 0.9790 - val_loss: 193.9220 - val_accuracy:
0.7400
Epoch 4/5
17/17 [==============================] - 34s 2s/step - loss: 0.0383 - accuracy: 0.9824 - val_loss: 25.3984 - val_accuracy:
0.8767
Epoch 5/5
17/17 [==============================] - 34s 2s/step - loss: 0.0366 - accuracy: 0.9824 - val_loss: 4.9556 - val_accuracy: 0.
9333
```

ACCURACY PLOT :



CONFUSION MATRIX :

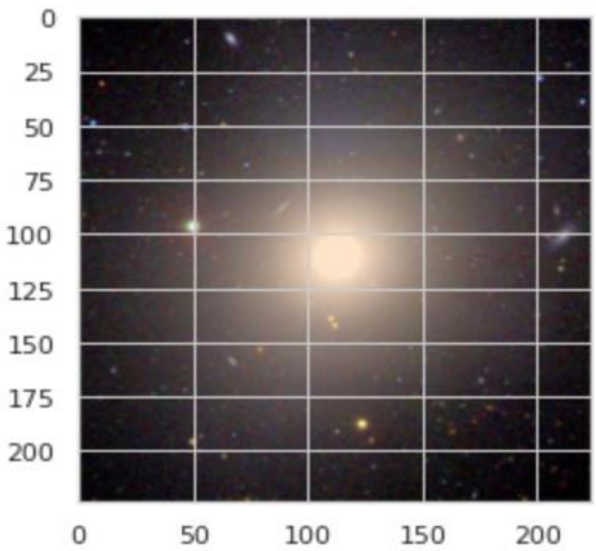


TEST ACCURACY :

```
Test accuracy: 0.9333333373069763
```

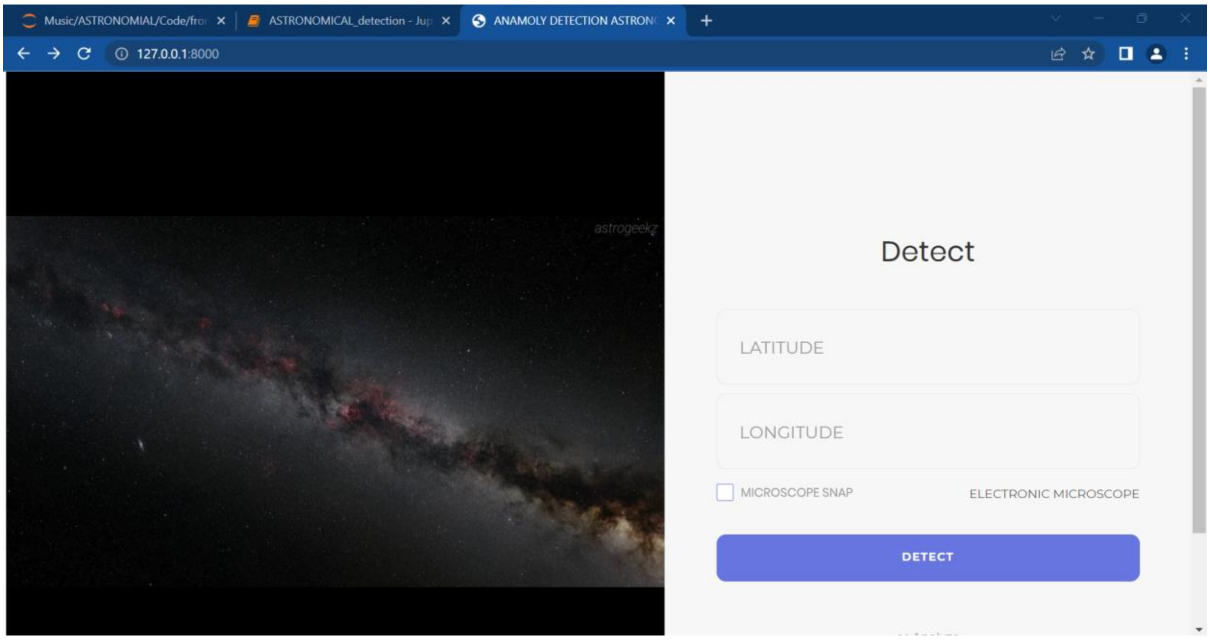
PREDICTED AND ACTUAL CLASS :

```
Predicted class: Spiral
Actual class: Spiral

Out[98]: <matplotlib.image.AxesImage at 0x7f015e9f35d0>
```

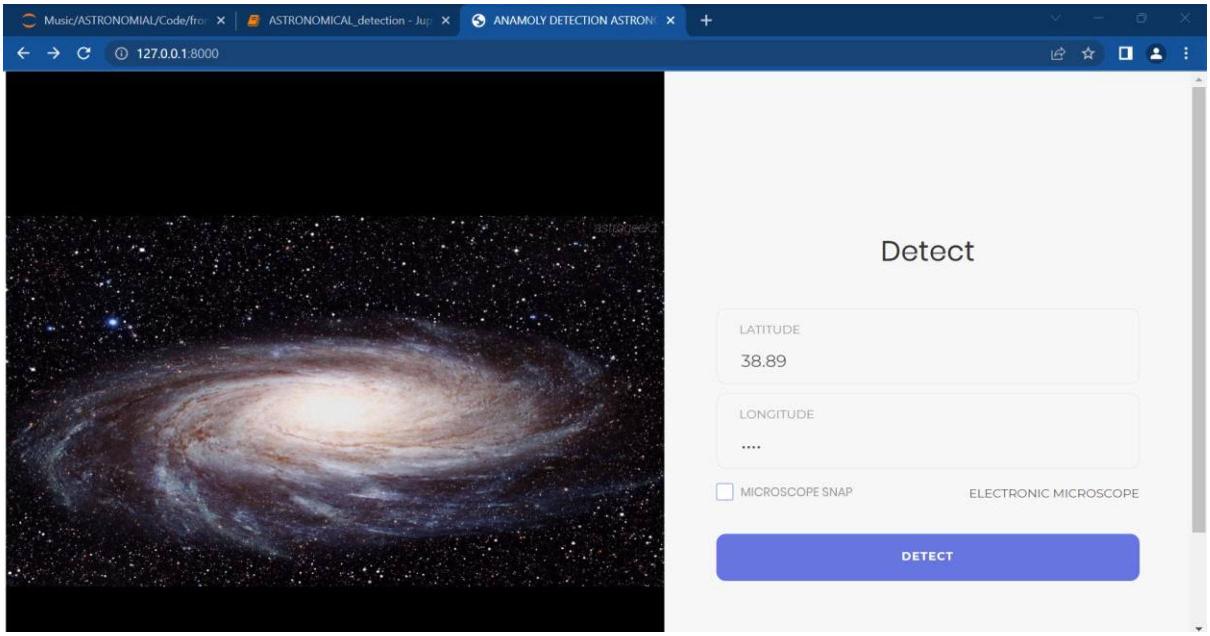


FRONTEND :

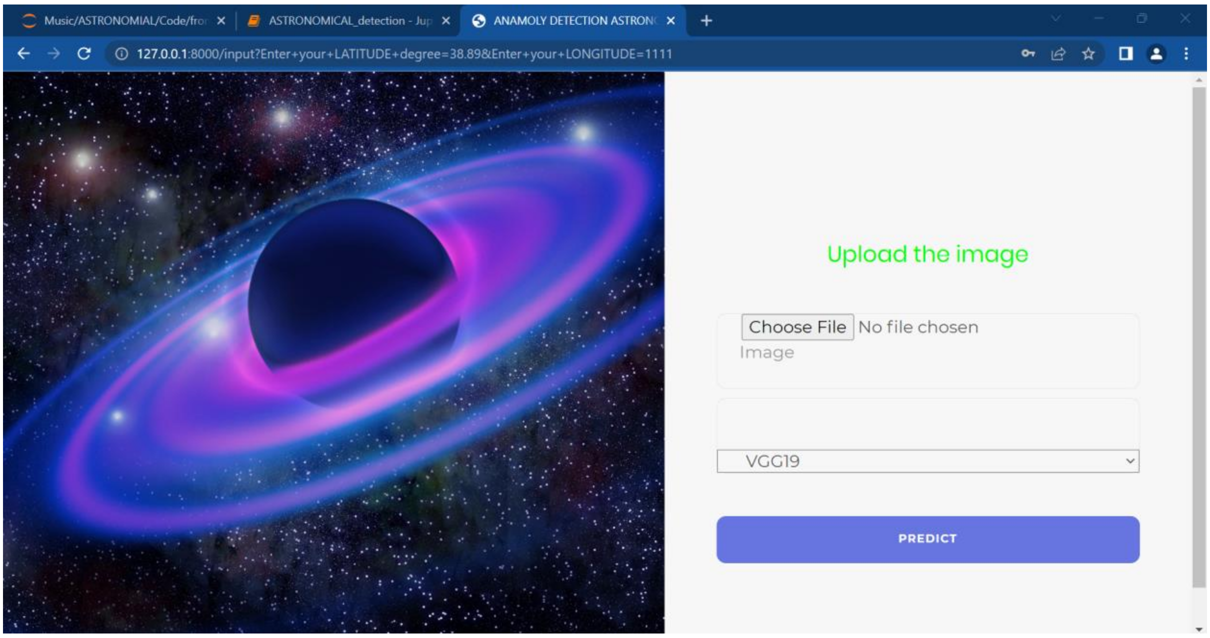
Initial Page



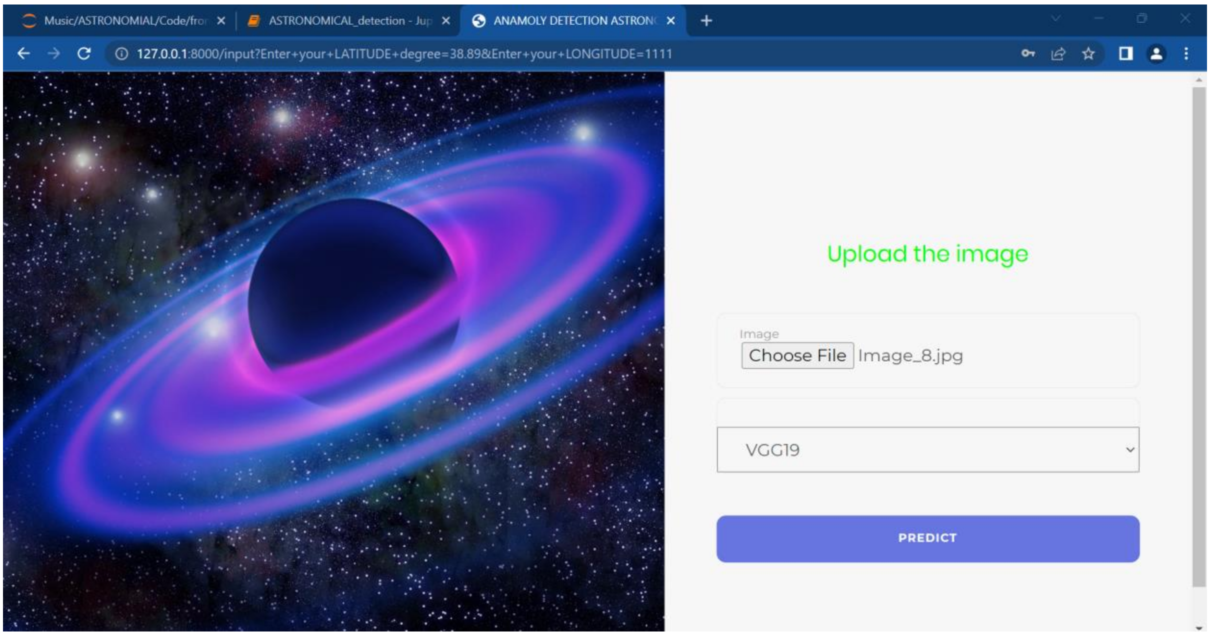
The User need to Enter the Latitude and Longitude , press detect to proceed.



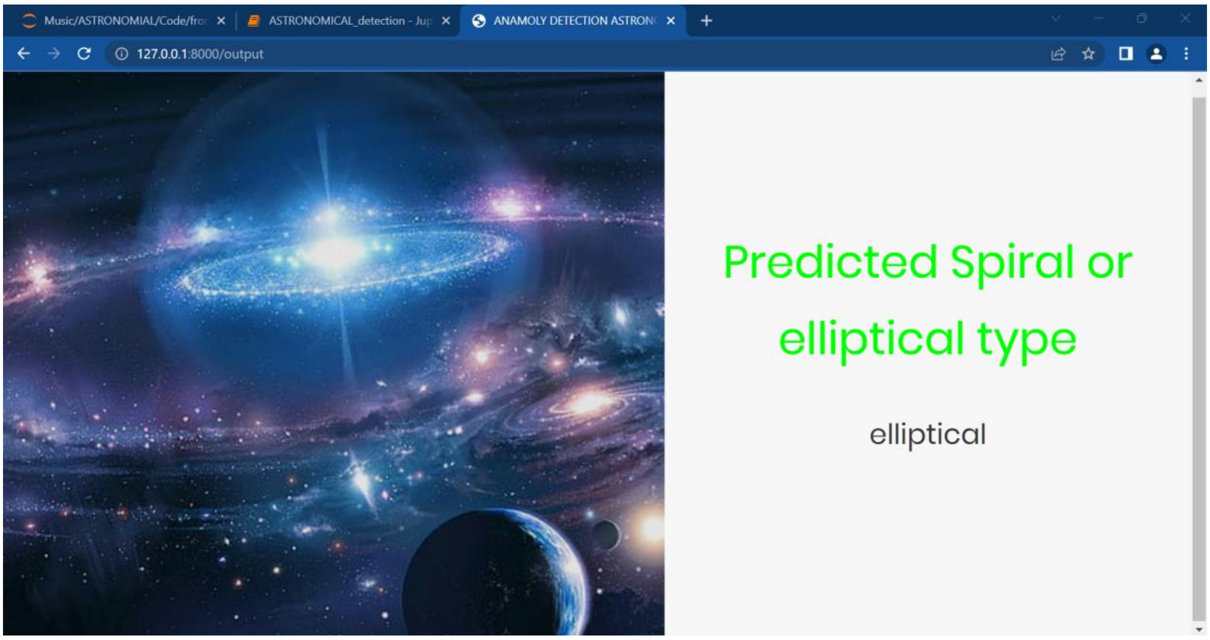
Now The user need to select the image from the Database and need to select the Algorithm (VGG19 or INCEPTIONV3)



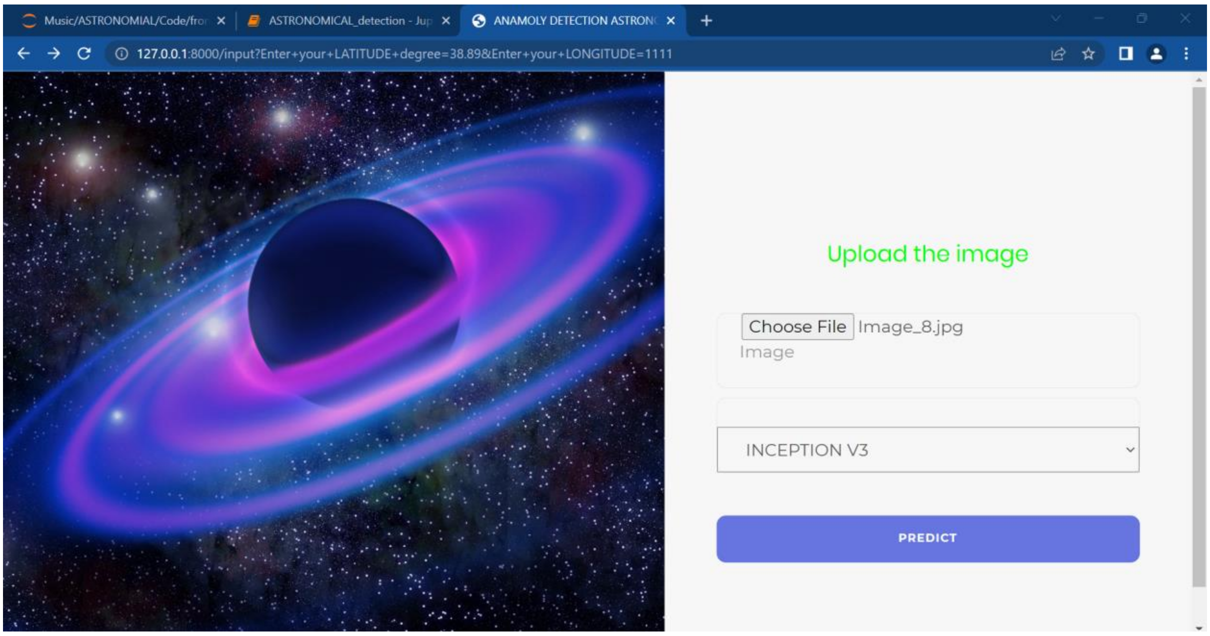
The Image is choosed and VGG19 Algorithm is selected.



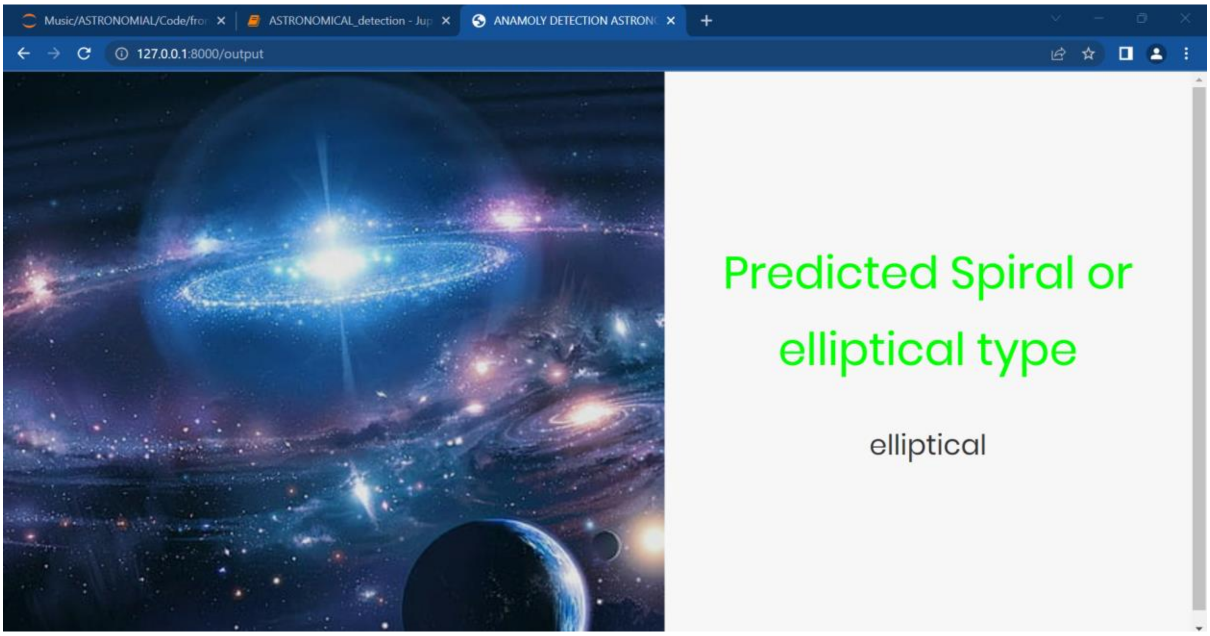
Now the model predicted that the choosed image is elliptical (based on VGG19)



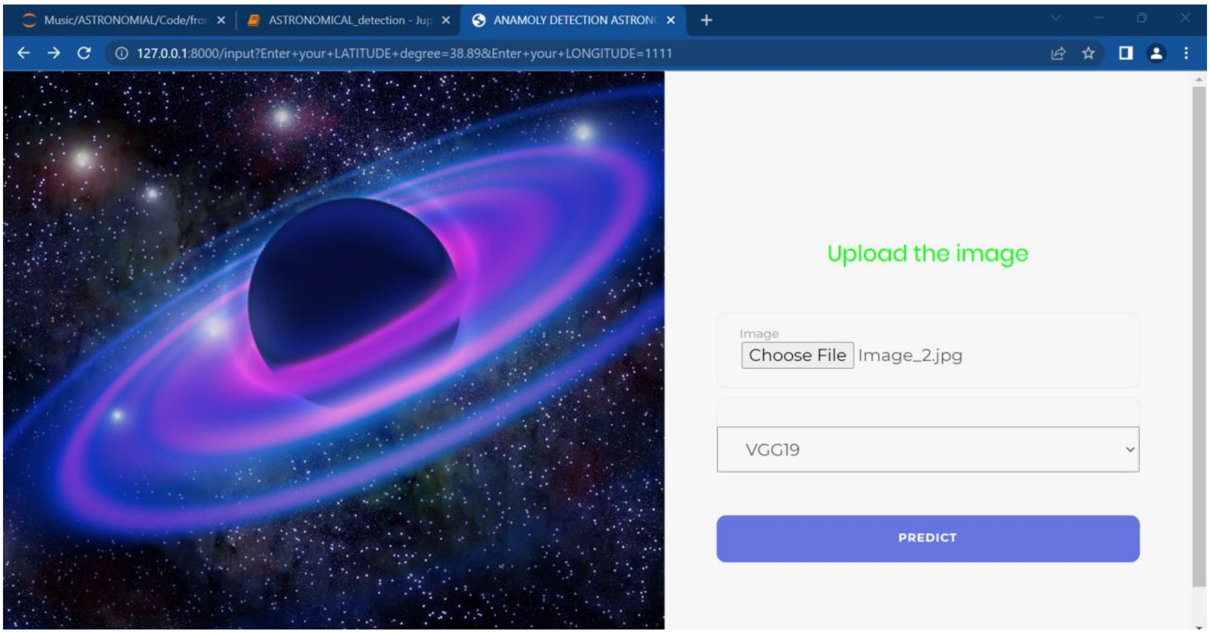
Now the same image is choosed to Test with the INCEPTIONV3 Algorithm



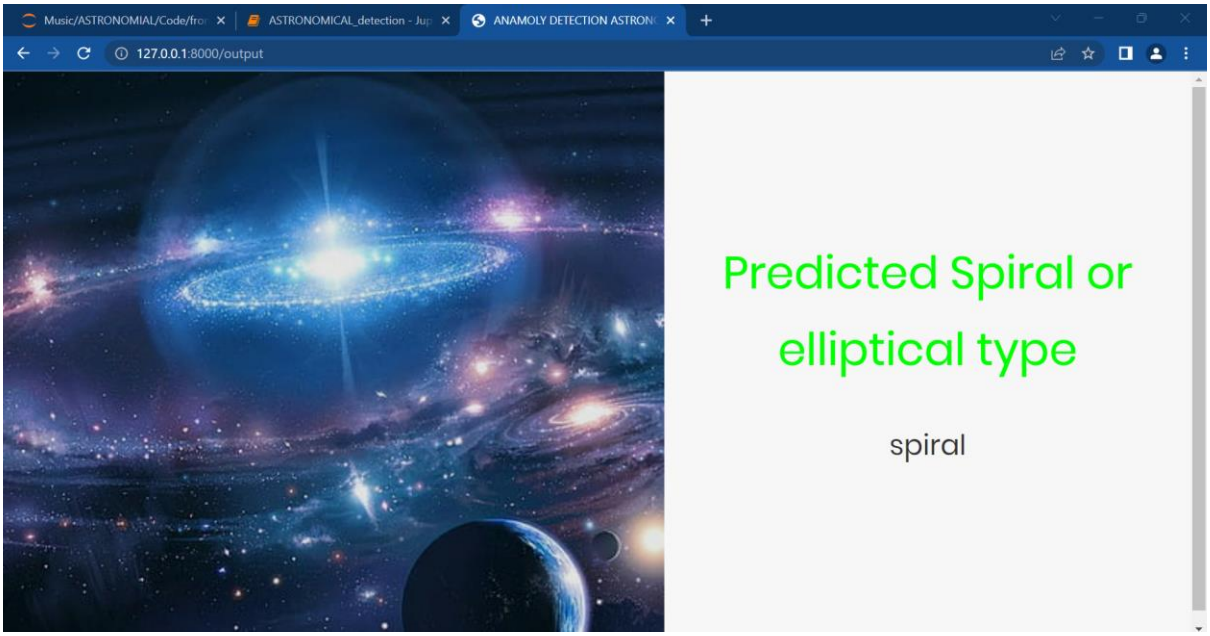
The model predicted that the choosed image is elliptical (based on INCEPTIONV3)



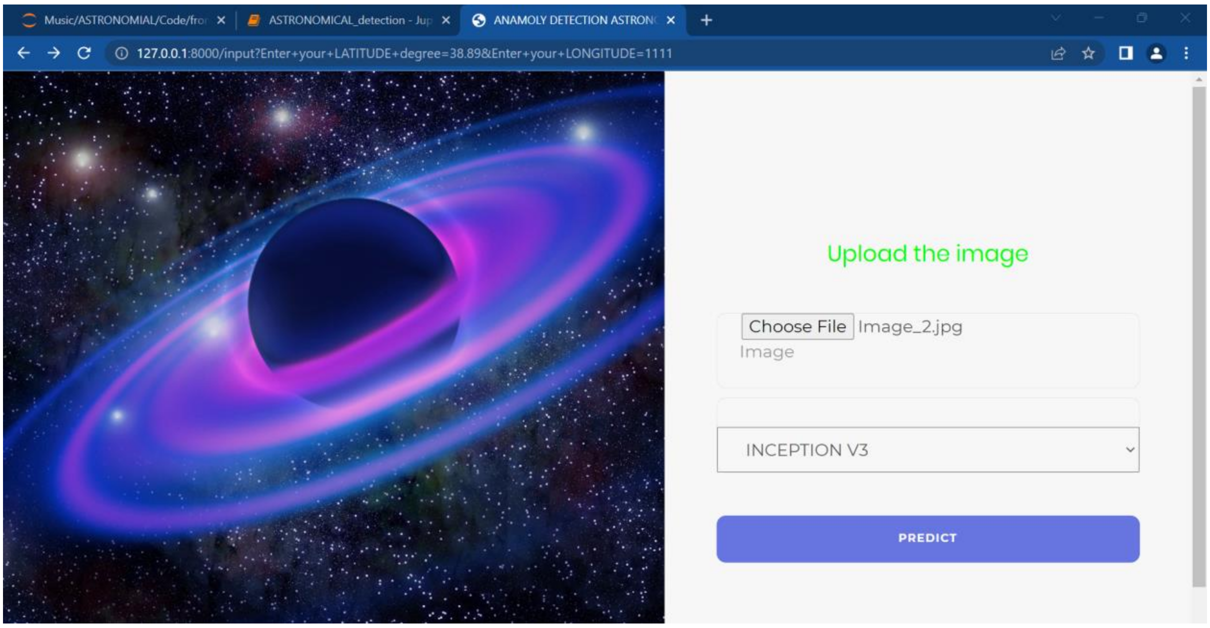
Now the spiral image is choosed and VGG19 Algorithm is selected.



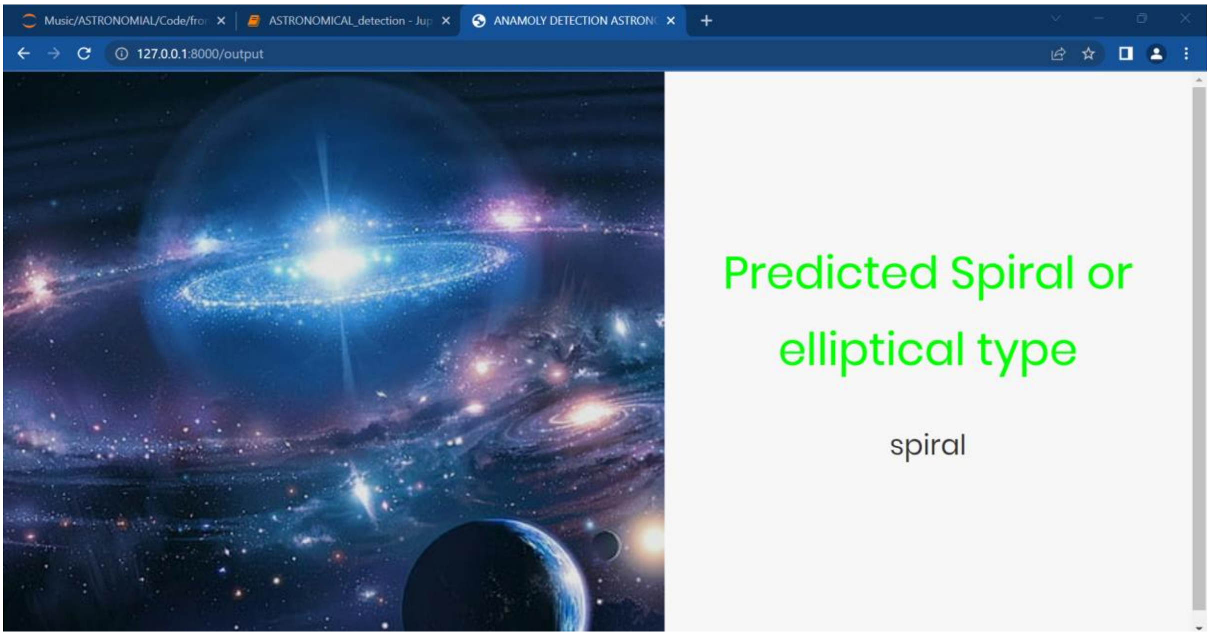
The model predicted that the image is spiral (Based on VGG19)



Now the spiral image is choosed and INCEPTIONV3 Algorithm is selected.



The model predicted that the image is spiral (Based on INCEPTIONV3)



Accuracy of Both Algorithms

VGG19 :

```
Test accuracy: 0.9655555486679077
```

INCEPTIONV3 :

```
Test accuracy: 0.9333333373069763
```

These shows that VGG19 Algorithm predict More Accurately then INCEPTIONV3 Algorithm.

6.4 TESTING

TEST CASE :

Test Case Id	Test Scenario Name	Test Scenario Description	Expected Result	Actual Result	Comment
T01	Test case for Latitude and longitude	The user need to enter both Latitude and Longitude , if not entered We cannot predict.	Latitude and Longitude need to be entered.	Working As expected	passed
T02	Test case for Predict button	If the User entered the Latitude and Longitude the user click the predict button.to predict according to the Latitude and Longitude.	The system need to predict.	Working As expected	passed
T03	Test case for Selecting image	User need to select the image in the available dataset.	Image need to be accepted.	Working As expected	passed
T04	Test case for Selecting algorithm	User need to select the either of the algorithm.if the user not selected the algorithm.it cannot be predict algorithm.	User need to select the algorithm	Working As expected	passed
T05	Test case for Inception v3	User select the inception v3 algorithm.it must predict the image based on the Algorithm	it must predict based on inception v3 algorithm	Working As expected	passed
T06	Test case for VGG19	User select the VGG19 algorithm.it must predict the image based on the Algorithm	it must predict based on VGG19 algorithm	Working As expected	Passed
T07	Test case for Predict Button	After selecting the image and Algorithm the user needs to enter the predict button and the system finds the image is in elliptical or spiral image.	Image need to be predicted if the button pressed.	Working As expected	Passed

WHITE BOX TESTING:

White Box Testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing.

White box testing techniques analyze the internal structures the used data structures, internal design, code structure and the working of the software rather than just the functionality as in black box testing. It is also called glass box testing or clear box testing or structural testing.

Working process of white box testing:

- Input: Requirements, Functional specifications, design documents, source code.
- Processing: Performing risk analysis for guiding through the entire process.
- Proper test planning: Designing test cases so as to cover entire code. Execute rinse-repeat until error-free software is reached. Also, the results are communicated.
- Output: Preparing final report of the entire testing process

UNIT TESTING :

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

FUNCTIONAL TESTING :

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/ Procedures: interfacing systems or procedures must be invoked.

SYSTEM TESTING :

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

PERFORMANCE TESTING :

The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

INTEGRATION TESTING :

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

ACCEPTANCE TESTING :

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. Acceptance testing for Data Synchronization:

- The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
- The Route add operation is done only when there is a Route request in need
- The Status of Nodes information is done automatically in the Cache Updation process

BUILD THE TEST PLAN :

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

CHAPTER 7

RESULTS

7.1 RESEARCH FINDINGS :

In the Recent research Detection was done using the DG and GMM models together with a histogram based method and the aritificial injections are used to evaluate the performance of algorithms. The histogram based methods (H) is repeating the process of the injection , In some Research two techniques are used for astronomical problems : a star-galaxy separator for the UKIRT Infrared Deep Sky Survey (UKIDSS) and a novel anomaly detection method for cross-matched astronomical datasets. Previous System was made using the machine learning algorithms (SVM and KNN). As there is always new data being generated every day in large amounts. In the proposed system we implement the deep learning models such as Inceptionv3 and VGG19 model to predict the shape of an astronomical object shape which will be more accurate then the existing system which was made using the machine learning algorithms.

7.2 EVALUATION METRICS :

PERFORMANCE METRICS

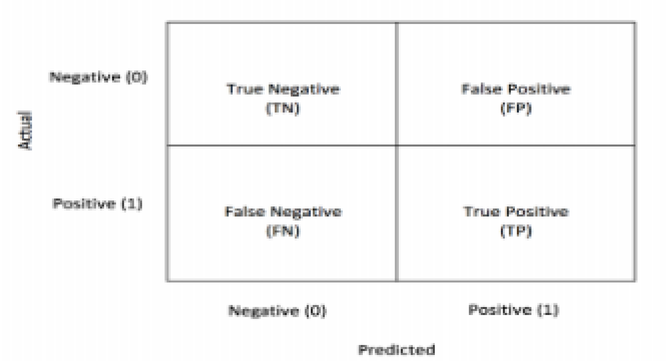
Data was divided into two portions, training data and testing data, both these portions consisting 70% and 30% data respectively. All these six algorithms were applied on same dataset using Enthought Canaopy and results were obtained.

$$\textbf{Accuracy} = (TP+TN) / (P + N)$$

Predicting accuracy is the main evaluation parameter that we used in this work. Accuracy can be defied using equation. Accuracy is the overall success rate of the algorithm.

CONFUSION MATRIX

It is the most commonly used evaluation metrics in predictive analysis mainly because it is very easy to understand and it can be used to compute other essential metrics such as accuracy, recall, precision, etc. It is an NxN matrix that describes the overall performance of a model when used on some dataset, where N is the number of class labels in the classification problem.



CONCLUSION

After acquiring adequate information regarding our subject matter, supervised learning can be implemented to enhance our expertise. Our aim is to explore the vastness of the universe by comprehending the large amount of astronomical data employing various machine learning methods. The approach of deep learning has been accomplished to obtain the objective of galaxy classification and detection. The galaxy can be classified as an entirely round-shaped, cigar-like appearance, or a mixture of both these types, on edge, has signs of spiral, spiral barred, spiral, elliptical, irregular.

FUTURE WORK

In these project we have done a front end in which the users can upload a image and check whether the shape is normal or a anomaly one with the help of the algorithm we have done at the backend (INCEPTIONV3 and VGG19). Till now the model can only predict the elliptical and spiral images . In Future we are going to make a model which will predict all the other rare shapes like irregular and peculiar and also we are going to make a application which will be available for all the end users .

And also in future we are going to automate the model which will choose a most accurate algorithm by own once the user uploaded the image to give the best accurate prediction.

REFERENCES

[1] Roberto E. Gonz aleza, Roberto P. Mu noza, Cristian A. Hern andez, “Galaxy detection and identification using deep learning and data augmentation.”, 2018

[2] Wahyono, Rahman, M. A., and Azhari, .SN. (2018). Classification of Galaxy Morphological Image Based on Convolutional Neural Network. (Vol. 5, Issue 6). International Journal of Advanced Research in Science, Engineering and Technology.

[3] Mittal A., Soorya A., Nagrath P., Hemant D.J. (2019). Data augmentation based morphological classification of galaxies using deep convolutional neural network. (part of Springer Nature 2019). Springer-Verlag GmbH Germany.

[4] Khalifa N. E. M., Taha M. H. N., Hassanien A. E., Selim I. M. Deep Galaxy: Classification of Galaxies based on Deep Convolutional Neural Networks. Faculty of Computers and Information, Cairo University. National Research Institute of Astronomy and Geophysics, Cairo, Egypt. Computer science Department Integrated Thebes Institutes Cairo, Egypt. Scientific Research Group in Egypt, (SRGE).

[5] Lukic V., Bruggen M., (2017) Galaxy Classifications with Deep Learning Hamburger Sternwarte, International Astronomical Union, University of Hamburg, Gojenbergsweg 112, 21029, Hamburg, Germany.

[6] Gauthier A., Jain A., Noordeh E. (2016). Galaxy Morphology Classification, Stanford University.

[7] Dai J. M., Tong J. (2018). Galaxy Morphology Classification with Deep Convolutional Neural Networks National Space Science Center, Chinese Academy of Sciences, China. University of Chinese Academy of Sciences, Beijing, China.