A

Mini Project

on

SNAKE SPECIES IDENTIFICATION AND RECOGNITION

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING (AI&ML)

by

R.S.S. NAVANEETH (207R1A6645)

K. MAHESH (207R1A6630)

R. SAI TEJA (207R1A6643)

Under the Guidance of

Mr. M. Sreenu

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI&ML)

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

2020-2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI&ML)

CERTIFICATE

This is to certify that the project entitled **"SNAKE SPECIES IDENTIFICATION AND RECOGNITION"** being submitted by **R.S.S. NAVANEETH (207R1A6645), K. MAHESH (207R1A6630) & R. SAI TEJA (207R1A6643)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering (AI&ML) to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2022-23.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Mr. M Sreenu
Assistant Professor
INTERNAL GUIDE

Dr. S Rao Chintalapudi
HOD CSE(AI&ML)

EXTERNAL EXAMINER

Submitted for viva voice Examination held on _____

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **Mr. M Sreenu,** Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Dr. G. Vinoda Reddy, Dr. K. Mahesh, Md. Hafeena & D. Babu Rao** for their cordial support, valuable information, and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. S Rao Chintalapudi,** Head, Department of Computer Science and Engineering (AI&ML) for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

R.S.S. NAVANEETH (207R1A6645)

K. MAHESH (207R1A6630)

R. SAI TEJA (207R1A6643)

ABSTRACT

Snake Species Identification is a challenge as erroneous snake identification from the perceptible traits is a prime reason of death because of snake bites. The main objective of the proposed system is to be able to identify snake species from their visual traits in order to provide suitable treatment, thus preventing subsequent deaths.

The proposed system involves techniques based on Image Processing, Convolution Neural Networks and Deep Learning to achieve the mentioned purpose. CNN has been highly used in automatic image classification system. In most cases, extracting features and utilizing them for classification.

Deep learning successfully achieves recognition of objects in images as it is implemented using artificial neural networks. Image classification tasks have seen a rise with the introduction of deep learning techniques. So far, no automated method for classification has been suggested to categorize snakes. The system that would be developed will be useful to recognize snake species correctly and thus take necessary action.

LIST OF FIGURES

LIST OF TABLES

Table of Contents

8. REFERENCES

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

This project is titled “Snake Species Identification and Recognition.” The project scope involves developing an automated system that can accurately identify and recognize different species of snakes based on their physical characteristics such as color patterns, scales, head shape, and body size.

1.2 PROJECT PURPOSE

The project purpose of snake species identification and recognition is to develop an automated system that can accurately identify and recognize different species of snakes based on their physical characteristics. This system would serve as a valuable tool for snake researchers, conservationists, and wildlife managers to monitor different snake species in their natural habitats, which could help in understanding their ecology, behavior, and conservation status.

The Project aims to develop a tool that can aid in snake conservation and management efforts, as well as potentially save lives by improving the accuracy and efficiency of snakebite treatment.

1.3 PROJECT FEATURES

The snake species identification and recognition using deep learning project is a sophisticated project that leverages deep learning algorithms to accurately identify and classify different species of snakes based on their physical characteristics. The project features include the use of convolutional neural networks (CNNs), a type of deep learning algorithm that can learn and extract relevant features from snake images, improving the accuracy of classification.

Transfer learning is also employed to use pre-trained models to jumpstart training and improve the accuracy of classification with a limited amount of data.

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, "what must be done to solve the problem?" The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

The problem definition of snake species identification and recognition is to develop an automated system that can accurately identify and recognize different species of snakes based on their physical characteristics. This is a challenging task due to the high variability and complexity of snake morphology, as well as the fact that many snake species have similar-looking patterns and colorations.

2.2 EXISTING SYSTEM

The existing system presents the snake species identification by using natural language processing. It aims to help medical professionals in predicting the snake species for snake-bite treatments based on the patient's description of the snake. The decision in suitable anti-venom critically depends on the type of snake species. Wrong anti-venom may result in severe morbidity and mortality. This research investigates the human perception and the selection of words in describing a snake based on their visual view. The descriptions were presented in unstructured text, and the NLP processing involves pre-processing, feature extraction and classification. Four machine learning algorithms (naïve Bayes, k-Nearest Neighbor, Support Vector Machine, and Decision Trees J48) were used during training and classification.

2.2.1 DISADVANTAGES OF EXISTING SYSTEM

Following are the disadvantages of existing system:

- Limited data availability
- Identifying and classifying snake species based on text descriptions can be challenging due to the ambiguity and context-specific nature of natural language.
- Snake species identification may require support for multiple languages, which could add complexity to an NLP-based system.

2.3 PROPOSED SYSTEM

Snakes are highly diverse, and snake biologists still document and describe snake diversity, with a mean of 30 new species described annually since the year 2000. Taxonomically, snakes are categorized into 24 families, containing 528 genera and 3,709 species. Our database is composed of around 3050 RGB images of varying size, split into 28 species.

Following are the major components of our proposed architecture of the system:

• Preprocessing: In this step, Grab Cut algorithm is applied to the images and data augmentation is performed.

• Training: The system is trained to perform classification.

• Classification: Pre-trained model is combined with the classifier to predict class labels.

• Testing and verification: Once training is complete, testing is done on random snake images and they are verified.

• Accuracy check: After testing on the images, the accuracy of the model is calculated

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

Following are the advantages of proposed system:

- Unlike NLP, which relies on textual data, deep learning models can be trained on images from different geographic regions, allowing for identification and recognition of snake species from various parts of the world, regardless of language or dialect.
- Deep learning models can be deployed in real-time and used to automate the identification and recognition of snake species, reducing the need for manual identification, and increasing efficiency.
- Deep learning models can be deployed in real-time and used to automate the identification and recognition of snake species, reducing the need for manual identification, and increasing efficiency.

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and a business proposalis put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis:

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.4.1 ECONOMIC FEASIBILITY

The economic feasibility of a snake species identification and recognition project would depend on several factors, such as the target market, the cost of development, and the potential revenue streams. One potential market for the project could be herpetologists, snake enthusiasts, and wildlife management organizations. These organizations could benefit from a reliable and efficient system for snake species identification and recognition, as it could help them identify and track snake populations in the wild.

The cost of development for the project could include the cost of hardware such as cameras, and the cost of software development, which could involve the use of machine learning algorithms and image processing techniques. The cost could also include ongoing maintenance and updates to the system. A thorough analysis of these factors would need to be conducted to determine the viability of the project.

2.4.2 TECHNICAL FEASIBILITY

The technical feasibility of a snake species identification and recognition project would depend on the availability of suitable technology and expertise in machine learning and image processing. The project would require a significant investment in hardware, software, and data curation, as well as expertise in these areas to ensure accuracy and reliability of the system.

2.4.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The social feasibility of a snake species identification and recognition project would depend on several factors, including the social acceptance of the technology, public awareness of the benefits of snake species identification, and potential impacts on snake conservation efforts.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- **System** : Pentium IV or higher
- **Hard Disk** : 8 GB and above
- **Monitor** : 14’ Colour Monitor.
- **Mouse** : Optical Mouse.
- **Ram** : 8 GB and above

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

- **Operating system** : Windows 7 Ultimate.
- **Coding Language** : Python.
- **Front-End** : Python.
- **Designing** : Tkinter
- **Data Base** : MySQL.
- **Tools** : Python Launcher
- **Algorithms** : GrabCut

3. ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification, starting from input to final prediction.

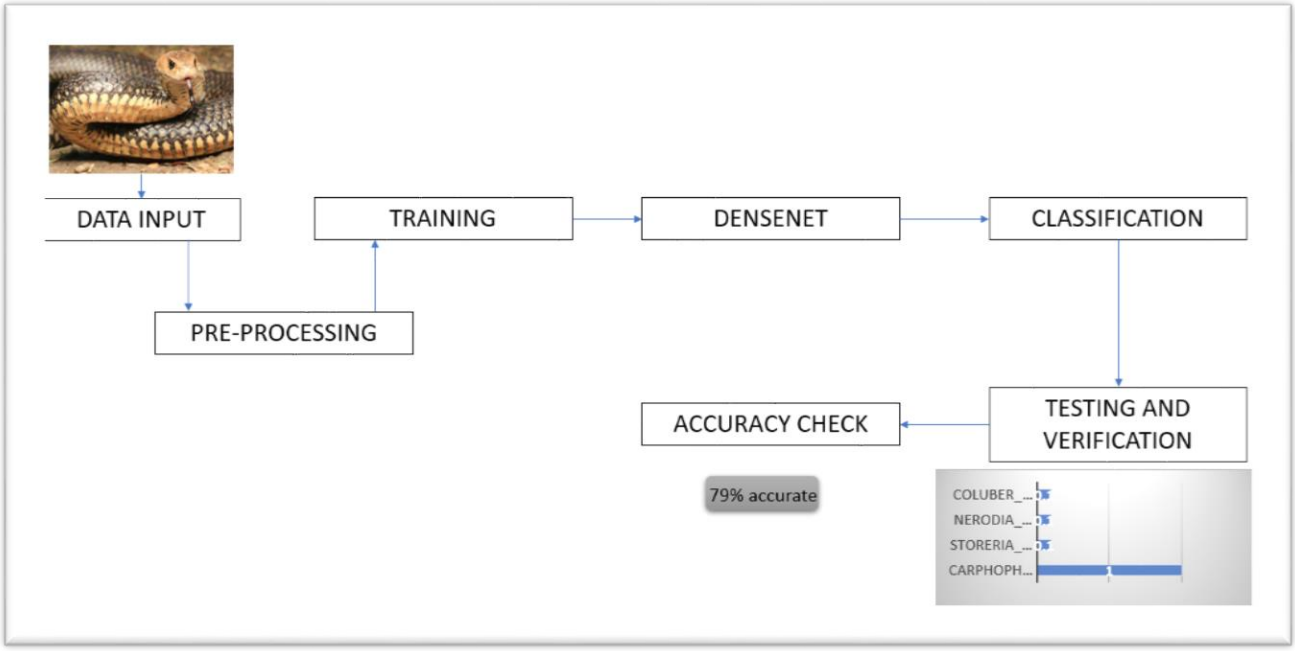

Figure 3.1: Project Architecture of Snake Species Identification and recognition

3.2 DESCRIPTION

Snake species identification and recognition using deep learning involves training a computer vision model to accurately identify and classify snake species based on their physical characteristics. This involves feeding a large dataset of snake images into a deep learning algorithm, which then learns to identify patterns and features that distinguish between different species of snakes. It represents a powerful tool for accurately and efficiently identifying and classifying snakes, which can be used to inform conservation efforts, public safety, and scientific research.

3.3 USE CASE DIAGRAM

In the use case diagram, we have basically one actor who is the user in the trained model.

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of usersthe system has. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.

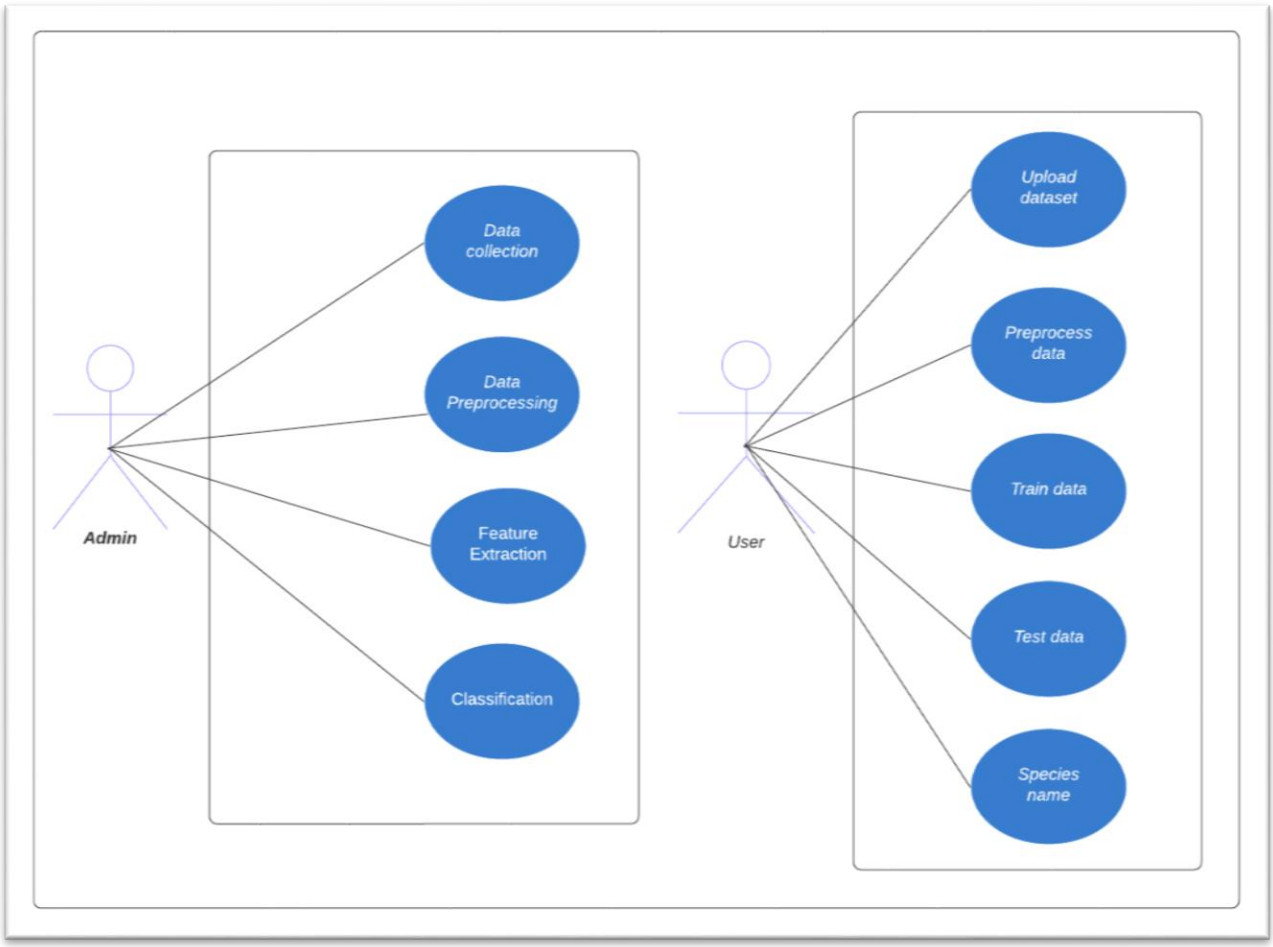

Figure 3.2: Use Case Diagram for Snake Species Identification and Recognition

3.4 CLASS DIAGRAM

Class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

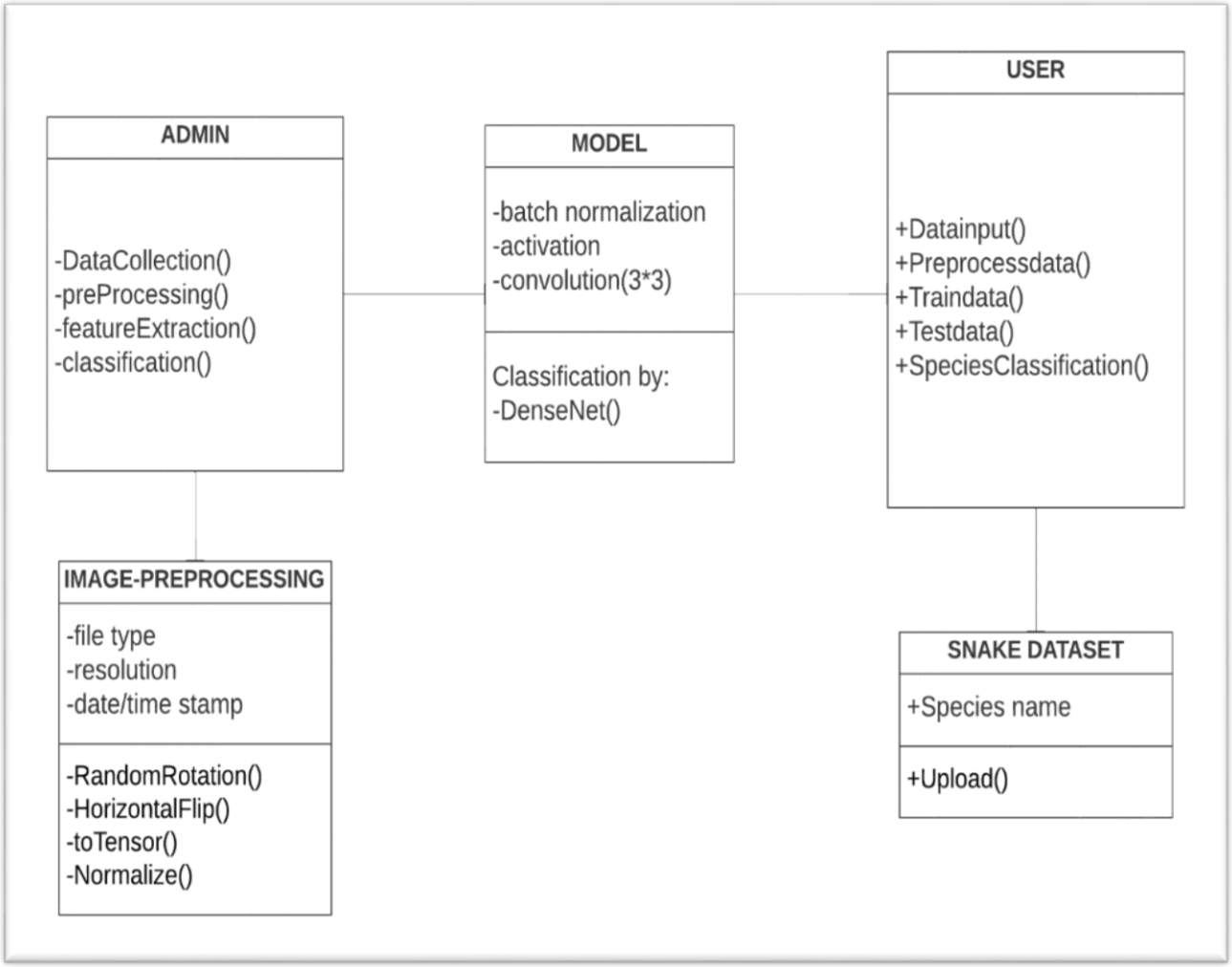

Figure 3.3: Class Diagram for Snake Species Identification and Recognition

3.5 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development.

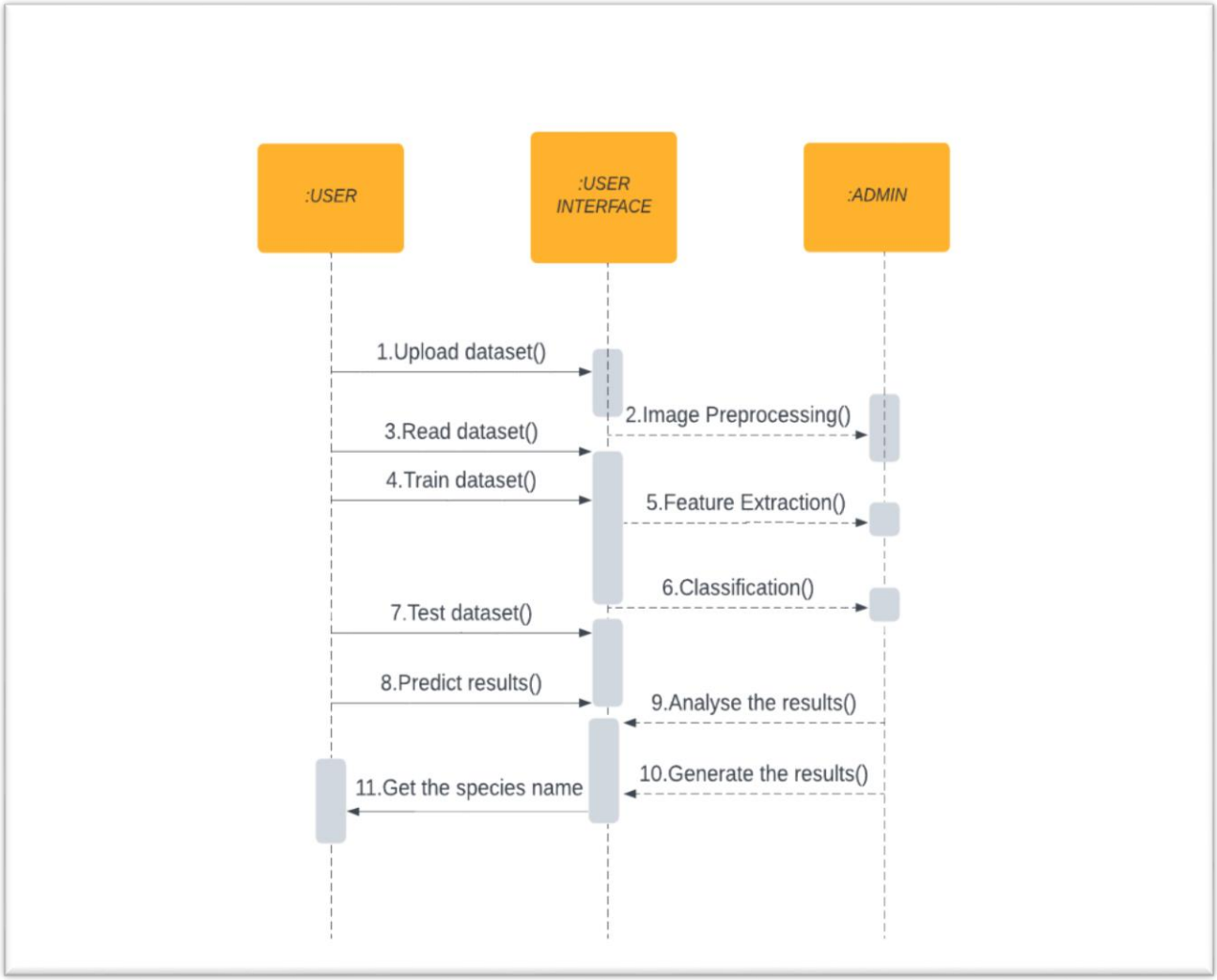

Figure 3.4: Sequence Diagram for Snake Species Identification and Recognition

3.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration, and concurrency. They can also include elements showing the flow of data between activities through one or more data stores.

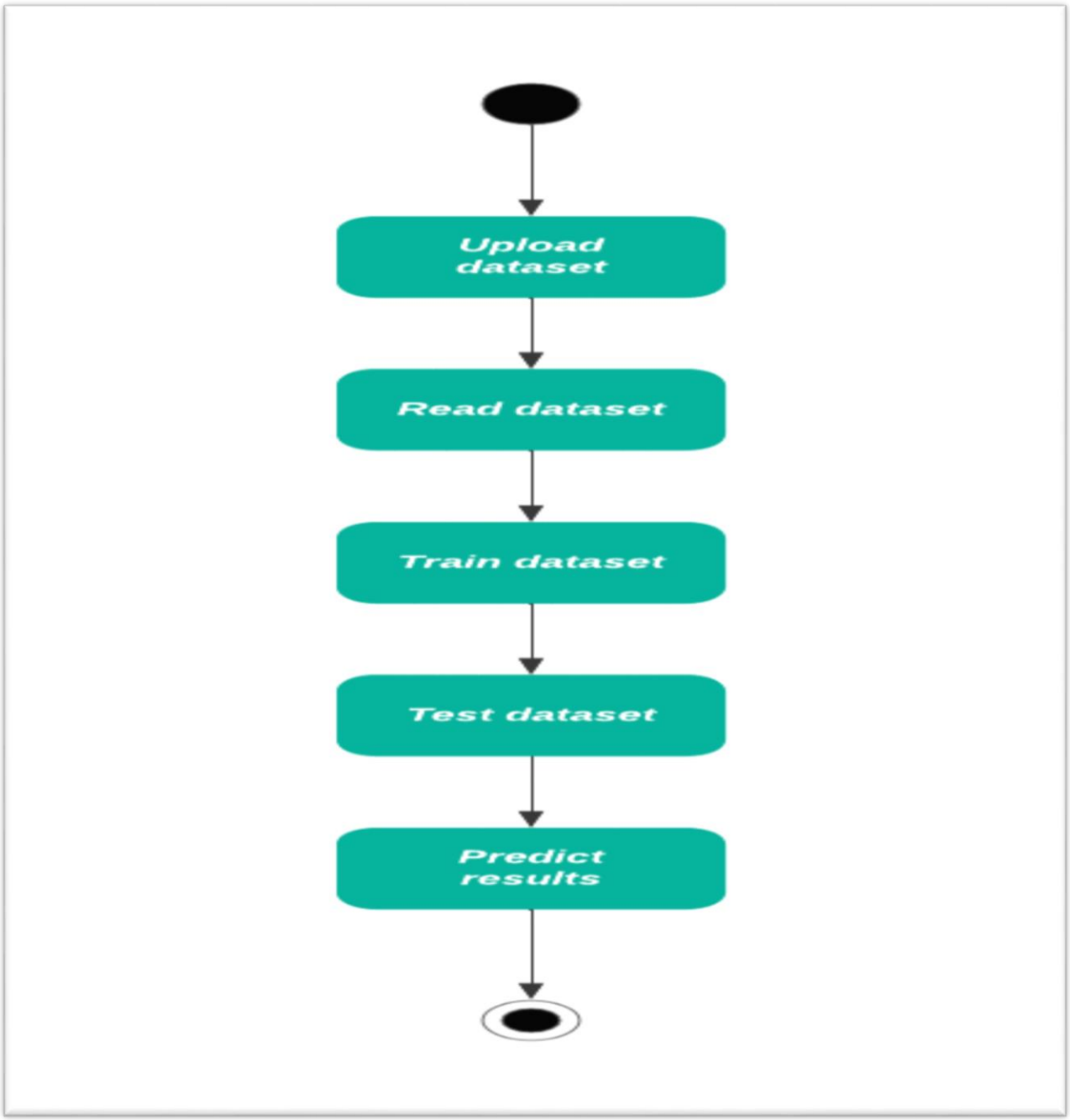

Figure 3.5: Activity Diagram for Snake Species Identification and Recognition

4. IMPLEMENTATION

4.1 SAMPLE CODE

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
from tkinter.filedialog import askopenfilename
import cv2
import random
import numpy as np
from keras.utils.np_utils import to_categorical
from keras.layers import  MaxPooling2D
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Convolution2D
from keras.models import Sequential
from sklearn.model_selection import train_test_split
from keras.applications import DenseNet169
from sklearn.metrics import accuracy_score
from keras.callbacks import ModelCheckpoint
import pickle
import os
from keras.models import load_model
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
import seaborn as sns
```

```
main = tkinter.Tk()
main.title("Snake Species Identification & Recognition")
main.geometry("1300x1200")

global filename
global classifier
global labels, X, Y, X_train, y_train, X_test, y_test, classifier

def readLabels(filename):
    global labels
    labels = []
    for root, dirs, directory in os.walk(filename):
        for j in range(len(directory)):
            name = os.path.basename(root)
            if name not in labels:
                labels.append(name)



def uploadDataset():
    global filename
    global labels
    labels = []
    filename = filedialog.askdirectory(initialdir=".")
    pathlabel.config(text=filename)
    text.delete('1.0', END)
    text.insert(END,filename+" loaded\n\n");
    readLabels(filename)
    text.insert(END,"Snake Species found in dataset are\n\n")
    for i in range(len(labels)):
        text.insert(END,labels[i]+"\n")

def processDataset():
    text.delete('1.0', END)
    global filename, X, Y, X_train, y_train, X_test, y_test
```

```
    if os.path.exists("model/X.txt.npy"):
        X = np.load('model/X.txt.npy')
        Y = np.load('model/Y.txt.npy')
    else:
        for root, dirs, directory in os.walk(filename):
            for j in range(len(directory)):
                name = os.path.basename(root)
                if 'Thumbs.db' not in directory[j]:
                    img = cv2.imread(root+"/"+directory[j])
                    img = cv2.resize(img, (80,80))
                    im2arr = np.array(img)
                    im2arr = im2arr.reshape(80,80,3)
                    X.append(im2arr)
                    label = getID(name)
                    Y.append(label)
                    print(name+" "+str(label))
        X = np.asarray(X)
        Y = np.asarray(Y)
        np.save('model/X.txt',X)
        np.save('model/Y.txt',Y)
    X = X.astype('float32')
    X = X/255
    text.insert(END,"Dataset Preprocessing Completed\n")
    text.insert(END,"Total images found in dataset : "+str(X.shape[0])+"\n\n")
    indices = np.arange(X.shape[0])
    np.random.shuffle(indices)
    X = X[indices]
    Y = Y[indices]
    Y = to_categorical(Y)
    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2) #split dataset
into train and test
    text.insert(END,"80% images are used to train DenseNet169 :
"+str(X_train.shape[0])+"\n")
```

```
    text.insert(END,"20% images are used to train DenseNet169 :
"+str(X_test.shape[0])+"\n")


def trainDensenet():
    text.delete('1.0', END)
    global filename, X, Y, X_train, y_train, X_test, y_test, classifier, labels
    densenet = DenseNet169(include_top=False, weights='imagenet',
input_shape=(X_train.shape[1], X_train.shape[2], X_train.shape[3]))
    for layer in densenet.layers:
        layer.trainable = False
    classifier = Sequential()
    classifier.add(densenet)
    classifier.add(Convolution2D(32, 1, 1, input_shape = (X_train.shape[1],
X_train.shape[2], X_train.shape[3]), activation = 'relu'))
    classifier.add(MaxPooling2D(pool_size = (1, 1)))
    classifier.add(Convolution2D(32, 1, 1, activation = 'relu'))
    classifier.add(MaxPooling2D(pool_size = (1, 1)))
    classifier.add(Flatten())
    classifier.add(Dense(output_dim = 256, activation = 'relu'))
    classifier.add(Dense(output_dim = y_train.shape[1], activation = 'softmax'))
    classifier.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics =
['accuracy'])
    if os.path.exists("model/model_weights.hdf5") == False:
        model_check_point = ModelCheckpoint(filepath='model/model_weights.hdf5',
verbose = 1, save_best_only = True)
        hist = classifier.fit(X, Y, batch_size = 32, epochs = 20, validation_data=(X_test,
y_test), callbacks=[model_check_point], verbose=1)
        f = open('model/history.pckl', 'wb')
        pickle.dump(hist.history, f)
        f.close()
    else:
        classifier = load_model("model/model_weights.hdf5")
    predict = classifier.predict(X_test)
```

```
    predict = np.argmax(predict, axis=1)
    testY = np.argmax(y_test, axis=1)
    p = precision_score(testY, predict,average='macro') * 100
    r = recall_score(testY, predict,average='macro') * 100
    f = f1_score(testY, predict,average='macro') * 100
    a = accuracy_score(testY,predict)*100
    text.insert(END,"DenseNet169 Accuracy  : "+str(a)+"\n")
    text.insert(END,"DenseNet169 Precision : "+str(p)+"\n")
    text.insert(END,"DenseNet169 Recall    : "+str(r)+"\n")
    text.insert(END,"DenseNet169 FSCORE    : "+str(f)+"\n\n")
    conf_matrix = confusion_matrix(testY, predict)
    plt.figure(figsize =(6, 6))
    ax = sns.heatmap(conf_matrix, xticklabels = labels, yticklabels = labels, annot = True,
cmap="viridis" ,fmt ="g");
    ax.set_ylim([0,len(labels)])
    plt.title("DenseNet169 Confusion matrix")
    plt.ylabel('True class')
    plt.xlabel('Predicted class')
    plt.show()

def graph():
    f = open('model/history.pckl', 'rb')
    graph = pickle.load(f)
    f.close()
    accuracy = graph['val_accuracy']
    error = graph['val_loss']

    plt.figure(figsize=(10,6))
    plt.grid(True)
    plt.xlabel('EPOCH')
    plt.ylabel('Accuracy/Loss')
    plt.plot(accuracy, 'ro-', color = 'green')
    plt.plot(error, 'ro-', color = 'red')
    plt.legend(['DenseNet169 Accuracy', 'DenseNet169 Loss'], loc='upper left')
```

```
    plt.title('DenseNet169 Training Accuracy & Loss Graph')
    plt.show()


def classifySpecies():
    global classifier, labels
    filename = filedialog.askopenfilename(initialdir="testImages")
    image = cv2.imread(filename)
    img = cv2.resize(image, (80, 80))
    im2arr = np.array(img)
    im2arr = im2arr.reshape(1,80,80,3)
    img = np.asarray(im2arr)
    img = img.astype('float32')
    img = img/255
    preds = classifier.predict(img)
    predict = np.argmax(preds)

    img = cv2.imread(filename)
    img = cv2.resize(img, (700,400))
    cv2.putText(img, 'Snake Species Recognized as : '+labels[predict], (10, 25),
cv2.FONT_HERSHEY_SIMPLEX,0.7, (0, 0, 255), 2)
    cv2.imshow('Snake Species Recognized as : '+labels[predict], img)
    cv2.waitKey(0)

def close():
    main.destroy()

font = ('arial', 16 , 'bold')
title = Label(main, text='SNAKE SPECIES IDENTIFICATION &
RECOGNITION',anchor=CENTER, justify=CENTER)
title.config(bg='green', fg='white')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)
```

```
font1 = ('times', 13, 'bold')
upload = Button(main, text="Upload Snake Species Dataset", command=uploadDataset)
upload.place(x=50,y=100)
upload.config(font=font1)

pathlabel = Label(main)
pathlabel.config(bg='white', fg='black')
pathlabel.config(font=font1)
pathlabel.place(x=50,y=150)

processButton = Button(main, text="Preprocess Dataset", command=processDataset)
processButton.place(x=50,y=200)
processButton.config(font=font1)

trainButton = Button(main, text="Train DenseNet169 Algortihm",
command=trainDensenet)
trainButton.place(x=50,y=250)
trainButton.config(font=font1)

graphButton = Button(main, text="DenseNet169 Training Graph", command=graph)
graphButton.place(x=50,y=300)
graphButton.config(font=font1)

classifyButton = Button(main, text="Snake Species Classification",
command=classifySpecies)
classifyButton.place(x=50,y=350)
classifyButton.config(font=font1)

exitButton = Button(main, text="Exit", command=close)
exitButton.place(x=50,y=400)
exitButton.config(font=font1)

font1 = ('times', 12, 'bold')
```

```
text=Text(main,height=25,width=78)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=465,y=100)
text.config(font=font1)

'''canv = Canvas(root, width=80, height=80, bg='white')
canv.grid(row=2, column=3)
img = PhotoImage(file="ui1.jpeg")
canv.create_image(20,20, anchor=NW, image=img)
img= PhotoImage(file="ui")
label=Label(ws,Image=img)
label.place(x=100,y=100)'''
main.config(bg='sky blue')
main.mainloop()
```

5. SCREENSHOTS

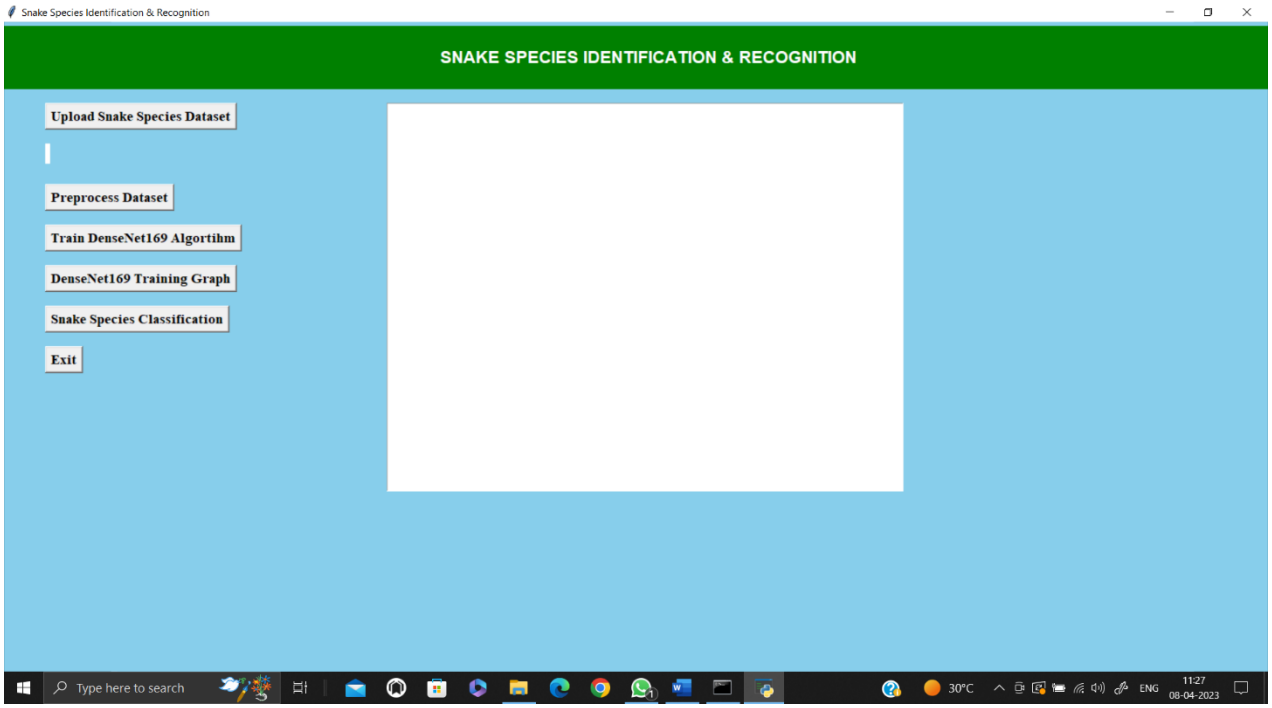

Figure 5.1: User Interface

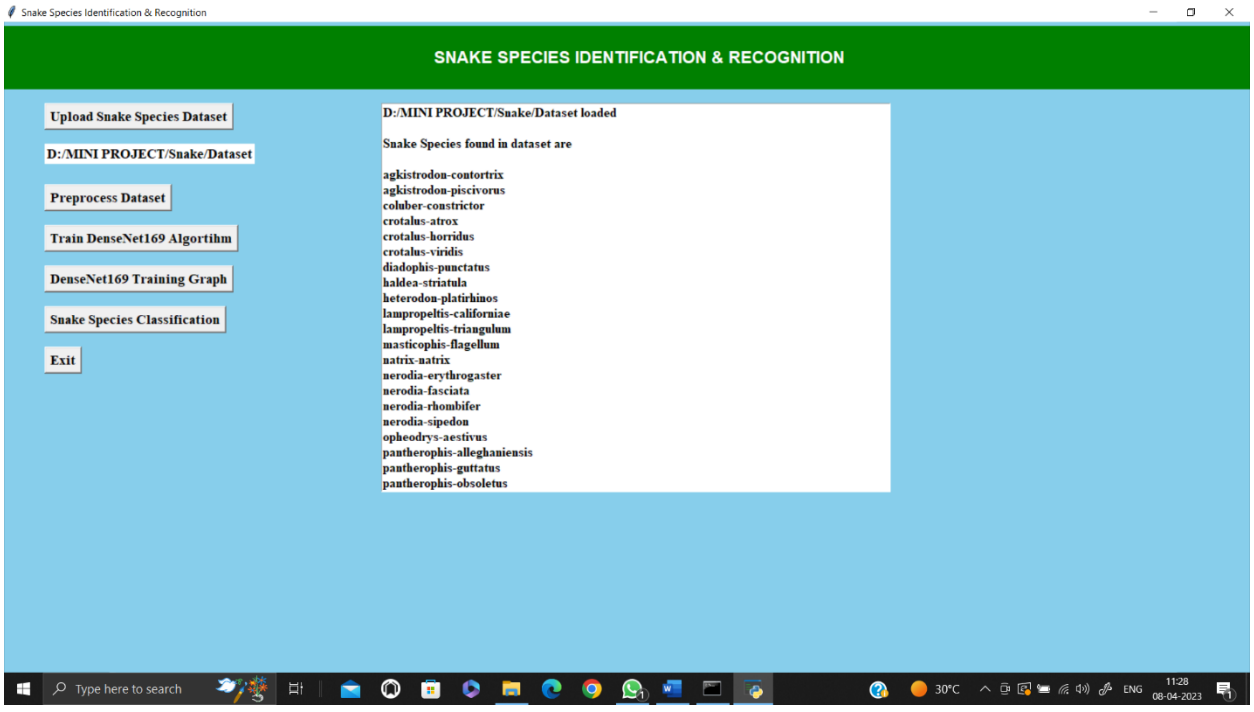

Figure 5.2: Dataset is uploaded and the species are listed

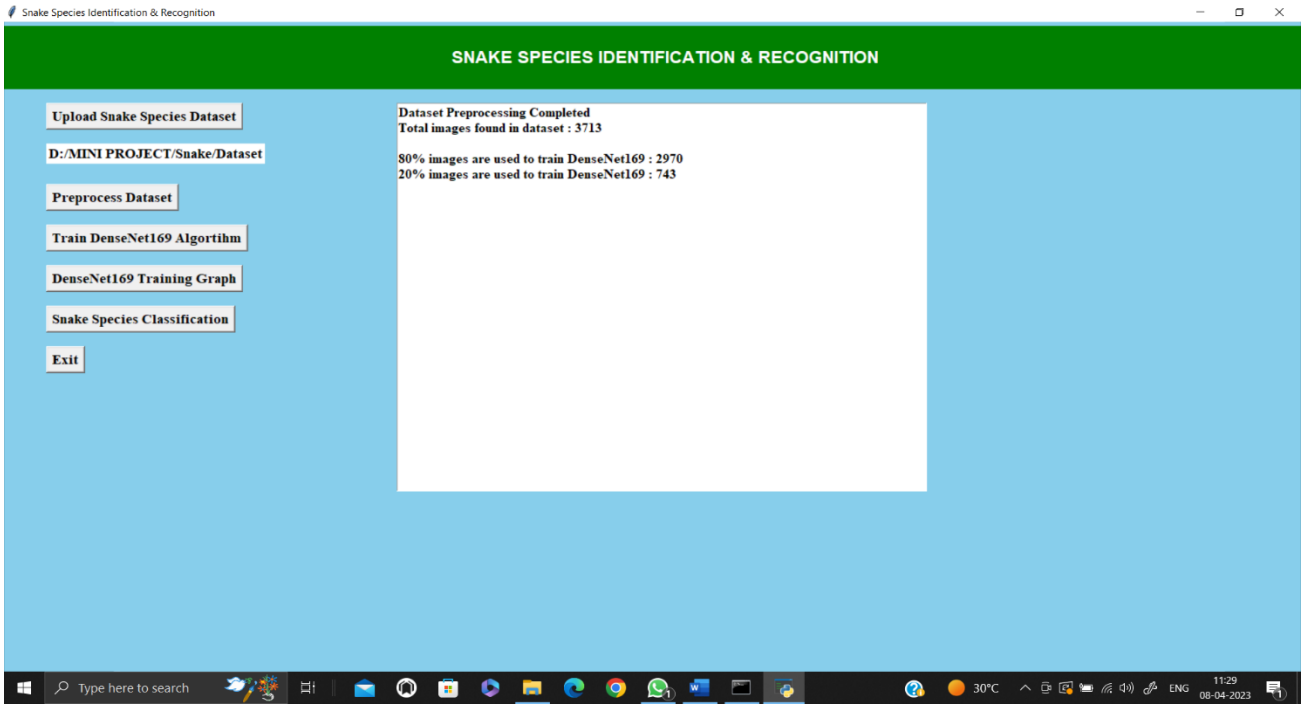

Figure 5.3: Dataset is preprocessed

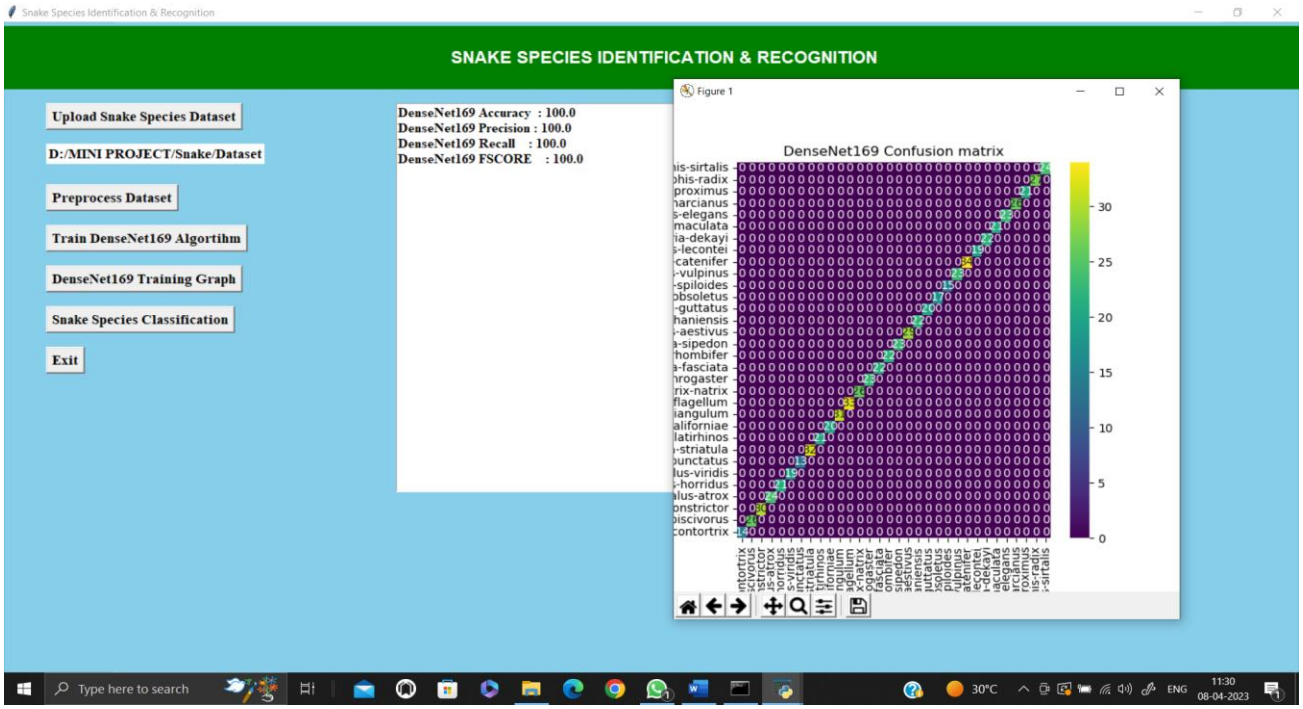

Figure 5.4: Images are trained

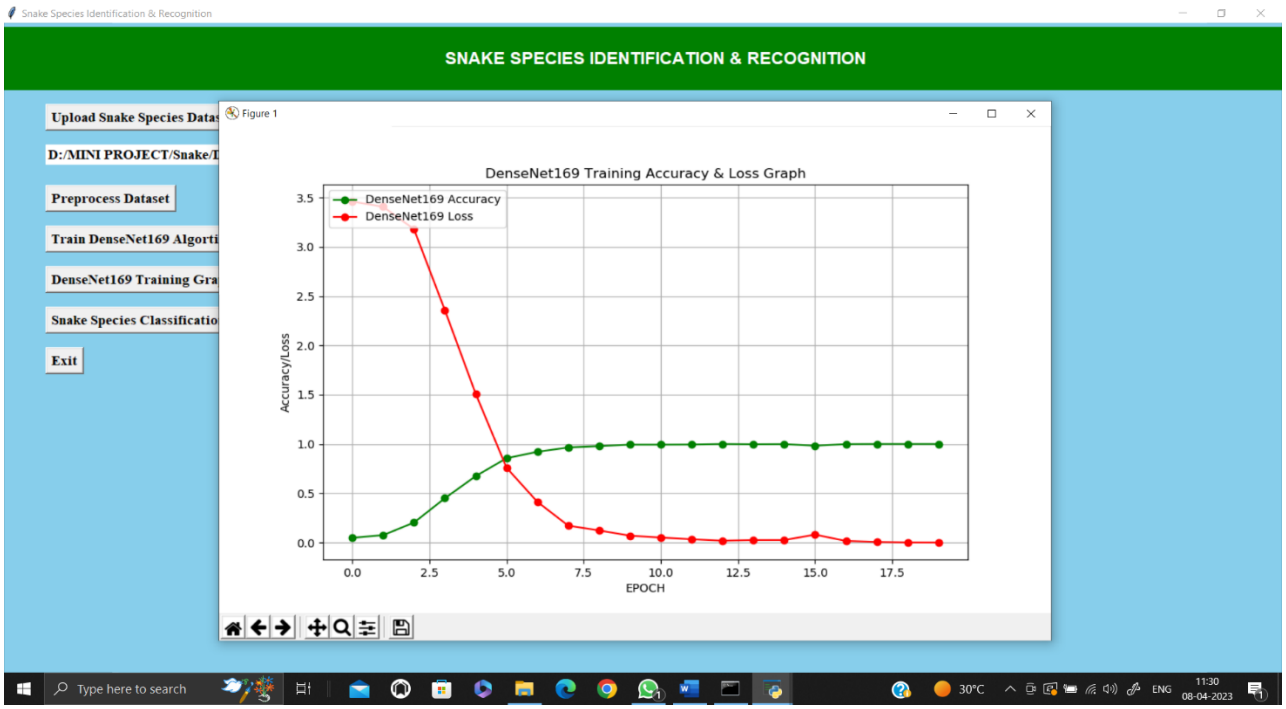

Figure 5.5: Images are tested

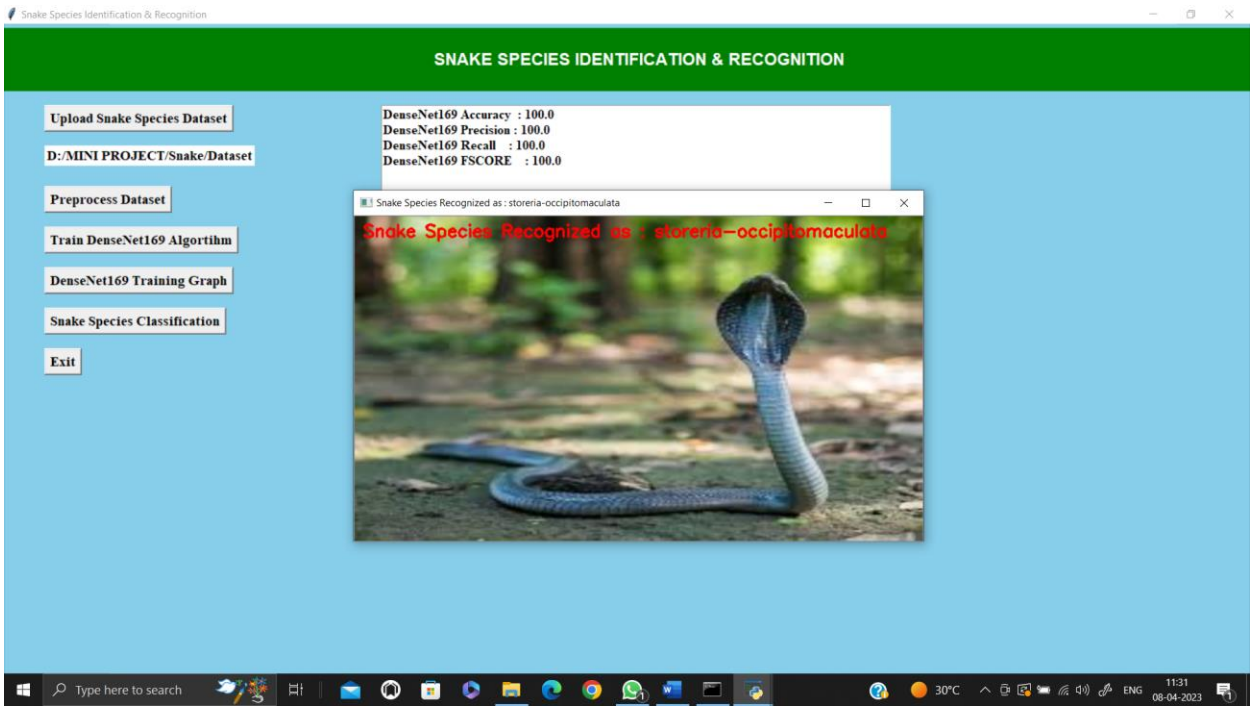

Figure 5.6: Snake species are identified

6. TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

Testing is an important aspect of the snake species identification and recognition project, as it ensures that the deep learning model is accurate and reliable in identifying different species of snakes. The main objective of testing in this project is to evaluate the performance of the model and to identify any errors or inaccuracies in the identification process in order to improve the performance and reliability. Testing involves a range of activities, including data collection, model training, and evaluation of model performance.

6.2 TYPES OF TESTING

6.3 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. Unit testing is a type of testing that focuses on verifying the correctness and functionality of individual units or components of the software application. Here, unit testing can be applied to the deep learning model to ensure that it functions correctly and accurately in identifying different species of snakes. Unit testing involves separation of the model components, design of test cases, implementation of test cases, and evaluation of the results.

6.3.1 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.3.2 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.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.4 TEST CASES

Test case ID	Test case name	Purpose	Input	Output
1	Common Species Identification	To check if the model can correctly identify a common garter snake.	An image of a common garter snake	Common garter snake
2	Rare and uncommon species	To check if the model can correctly identify a rare snake species	An image of a rare and uncommon snake species	Rare and uncommon snake species
3	Images with low resolution or poor lighting	To check if the model can handle images with low resolution or poor lighting	An image of a snake species with low resolution or poor lighting	Correct identification of the snake species
4	Images with occlusions or partial views	To check if the model can handle images with occlusions or partial views	An image of a snake species with occlusions or partial views	Correct identification of the snake species

Table 6.1 Testcases for classification of snake species

7. CONCLUSION

7. CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

The proposed system helps in snake species identification. The combination of a pretrained DenseNet model with the classifier shows good accuracy in classification of snake species. It can also be observed that with the help of the transfer learning technique, we can save time and computation resources. As there are millions of snake species around the world, this system could easily be adapted by training a greater number of snake species images to recognize them. Thus, a part of the future work would be to construct a larger database with a variety of images and apply distinctive techniques for recognizing the species. The crucial part in building such a system is the training dataset that needs to be developed either by manually taking pictures of the snakes from around the world or by using public datasets.

7.2 FUTURE SCOPE

The future scope of snake species identification and recognition project is wide, with the potential for further developments and collaborations that can benefit both humans and snakes. In terms of the expansion of dataset, the accuracy of the deep learning model can be improved by training it on a larger and more diverse dataset of snake images. This can help the model to learn more subtle variations and features that distinguish between different species of snakes. Collaboration with experts in the field of snake identification and conservation can help to refine the project and provide valuable insights into the behavior and habitats of different snake species. This can inform conservation efforts and help to protect snake populations around the world.

8. BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 REFERENCES

[1] Mrugendra Vasmatkar, Ishwari Zare, Prachi Kumbla, Shantanu Pimpalkar, Adithya Sharma, “Snake Species Identification and Recognition”, 2020 IEEE Bombay Section Signature Conference (IBSSC).

[2] Karen Simonyan, Andrew Zisserman VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION ICLR 2015.

[3] Nur Liyana Izzati Rusli, Amiza Amir, Nik Adilah Hanin Zahri, R. Badlishah Ahmad, “Snake species identification by using natural language processing” Indonesian Journal of Electrical Engineering and Computer Science, December 2018.

[4] Francis Jesmar, P. Montalbo and Alexander A. Hernandez, "Classification of fish species with augmented data using deep convolutional neural network", *IEEE 9th International Conference on System Engineering and Technology*, 2019.

[5] Rohan Sapkal, Pranalini Joshi, Dhanesh Sarpale and Apeksha Rajput. “Snake species recognition using tensor flow machine learning algorithm and effective convey system” 2019.

8.2 GITHUB LINK

1. **Project Code:**
   https://github.com/rentala-navaneeth/SNAKE-SPECIES-IDENTIFICATION-AND-RECOGNITION