A Project Topic
On
“Snake recognition”

Submitted by:
Masud Ali (18CSE027)

Date: May, 2024

Department of Computer Science & Engineering

Bangabandhu Sheikh Mujibur Rahman Science and Technology University

Snake recognition

(This project is submitted in the partial fulfillment of the requirement for the project proposal of " Fourth Year 2nd Semester " in Computer Science & Engineering)

By

Masud Ali

ID: 18CSE027

Supervised by:

Dr. Saleh Ahmed

Associate Professor

Department of Computer Science & Engineering

Bangabandhu Sheikh Mujibur Rahman Science and Technology University

Certificate of Approval

It is hereby declared that the contents of this project proposal are original and any part it has not been submitted elsewhere for the award of any degree.

---------------------------------- ---------------------------------

Signature of the Candidate Signature of the Supervisor

Date: Date:

Acknowledgement

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them. I owe my deep gratitude to our project supervisor **Dr. Saleh Ahmed**, who took keen interest in our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system. I am thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staff of Computer Science and Engineering which helped me in successfully completing my project work.

Masud Ali

May, 2024

Abstract

This project introduces an integrated system for snake detection. Snakes are curved, limbless, warm-blooded reptiles of the phylum serpents. Any characteristics, including head form, body shape, physical appearance, texture of skin and eye structure, might be used to individually identify nonvenomous and venomous snakes, that are not usual among non-experts' peoples. A standard machine learning methodology has also been used to create an automated categorization of species of snake dependent upon the photograph, in which the characteristics must be manually adjusted. As a result, a Deep convolutional neural network has been proposed in this paper to classify snakes into two categories: venomous and non-venomous. A set of data of 1440 snake pictures but after the argumentation total images is 8594 used to implement seven Neural networks with our proposed model. The number of photographs even has been increased by utilizing various image enhancement techniques. Ultimately, the transfer learning methodology is utilized to boost the identification process accuracy even more. Snakebites, a global health issue, account for millions of envenomation's annually, posing a significant threat to human life, especially in regions with high snake incidence. This project introduces an integrated system for snake detection, coupled with an intelligent antivenom suggestion mechanism. The system leverages advanced technologies, including image recognition and machine learning, to accurately identify snake species, analyze symptoms, and recommend tailored antivenom treatments. By amalgamating real time monitoring with sophisticated algorithms, our project aims to revolutionize the management of snakes, enhancing early intervention, reducing mortality, and addressing the challenges posed by the diverse nature of snake venoms. This endeavor represents a crucial step towards improving healthcare outcomes in snakebite-affected regions worldwide

Table of Contents

Table of Figure:

Chapter 1

Introduction

Snakebites constitute a pervasive and life-threatening health issue, particularly in regions where venomous snake species are prevalent. The prompt and accurate management of snakebite incidents is paramount to mitigating the associated morbidity and mortality. Traditional methods of snake identification necessitate innovative approaches to enhance the efficiency and effectiveness of snake management.

The "Snake Detection" project addresses these challenges by leveraging advanced machine learning (ML) techniques. By integrating cutting-edge technology with a diverse dataset sourced from Roboflow's[9] "Snake Breed Identification - v1 2022-07-28 9:18pm," this project aims to develop a comprehensive system. This system not only detects the presence of snakes and predicts the severity of snake identification

Motivation:

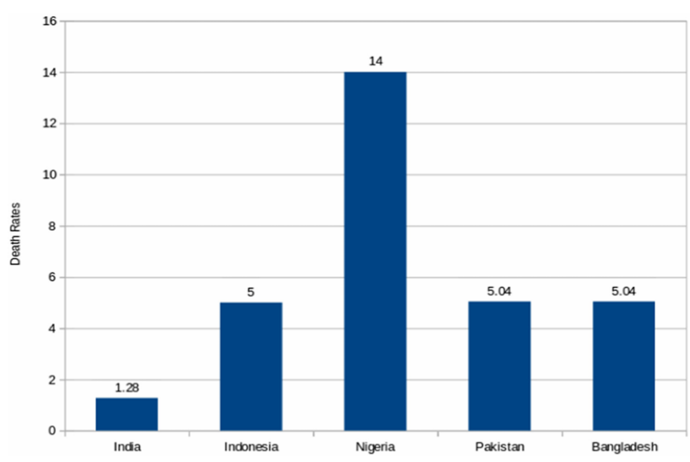

Figure 1 Top 5 countries with the highest rate of snake bite deaths per 100,000 people

A study reviewing 60 articles found that among 363 snakebite victims, 88% had venomous bites, and 12% had nonvenomous bites. Treatment was provided based on the presence of necrosis in

15.2% of cases. Interestingly, antibiotics were not used, suggesting that snakes might naturally carry antibiotics. Difficulty in identifying snakes by their visible characteristics contributes to mortality from snakebites.

Historically, snake venom, especially in Chinese culture, has been used in medicine. Drugs for conditions like high blood pressure and heart attacks have been developed using snake venom. Some FDA-approved medicines, such as Captopril, Integrilin, and Aggrastat, are derived from snake venom. Ongoing research is exploring more potential therapeutic uses of snake venom.

Snakebite envenoming is a significant health issue, with about 2 million people affected in Asia and 435,000 to 580,000 snakebites in Africa annually. Vulnerable populations, including women, children, and farmers in remote areas with limited healthcare access, face the highest risk.

Objectives

- Snake Detection:
  - Develop a machine learning model capable of accurately detecting snakes.
  - Utilize the comprehensive dataset containing images and bite-related features from Roboflow[9] for model training.
- Antivenom Recommendation:
  - Create a system that recommends suitable antivenoms based on the identified snake species and the severity of the bite.
  - Establish a knowledge base of antivenoms and their effectiveness against specific snake venoms.
- User Interface (UI):
  - Design an intuitive and user-friendly interface for healthcare professionals and individuals in snake-prone areas.
  - Provide clear information about the detected snake, potential danger, and recommended antivenom.
- Integration and Deployment:
  - Integrate the snake detection model with the antivenom recommendation system.
  - Deploy the system in relevant environments, such as healthcare facilities and emergency services.

Chapter 2

Related Work

Computer vision aims to provide computers with the human-like visual capability of understanding graphical information with no accompanying textual description [1]. Common tasks include object detection and object classification. Object classification involves categorizing an input image into several predefined classes [3]. This provides the foundation for multiple more advanced tasks such as semantic and instance segmentation [4]. Object detection aims to rapidly locate the region of an image where an object of interest may appear [5]. To resolve aforementioned problems, researchers have been using classical pattern matching in the 1970s to extract meaningful interpretation from graphical data [6]. This feature-based approach is generally combined with classical machine learning algorithms, including Support Vector Machines and K- Nearest Neighbors [7]. These methods perform well especially on simple classification with very promising efficiency. However, for complex classification in real life, it usually requires extremely complicated models to describe potential solutions, which limits both speed and accuracy of traditional machine learning methods. In recent decades, the most popular approach for visual recognition is based on Convolutional Neural Networks (CNN). Even though there are multiple alternatives to achieve image recognition, CNN has been demonstrated to have superior performance for image classification over them and even humans [8].

Chapter 3

Goals of the project

The primary goal of the "Snake Detection" project is to develop an integrated system that addresses the complexities associated with snakebite envenomation. The project aims to achieve the following objectives:

Accurate Snake Detection:

Develop a robust system using advanced image recognition and machine learning techniques to accurately identify snake species in real-time. This will enhance early intervention and reduce the time between the snakebite incident and medical response.

Symptom Analysis Algorithm:

Implement an intelligent algorithm to analyze symptoms exhibited by snakebite victims. By considering a range of physiological responses, the algorithm aims to provide a comprehensive understanding of the envenomation, facilitating more precise treatment recommendations.

- **CNNs:** deep neural networks primarily used in image recognition.
    - ★ Accuracy score: 78.72%
    - ★ Epochs: 50
- **Xception:** is a convolutional neural network that is 71 layers deep.
    - ★ Accuracy score: 89.30%
    - ★ Epochs: 15
- **VGG-16:** is a convolutional neural network that is 16 layers deep.
    - ★ Accuracy score: 76.30%
    - ★ Epochs: 50
- **ResNet-50:** is a 50-layer convolutional neural network
    - ★ Accuracy score: 64.80%
    - ★ Epochs: 50

Real-time Monitoring in High-Risk Areas:

Integrate the detection system with surveillance cameras for continuous real-time monitoring in areas prone to snakes. This proactive approach enhances the overall responsiveness to snake incidents, especially in remote or underserved regions.

Reducing Mortality and Improving Treatment Efficacy:

The ultimate goal is to significantly reduce mortality rates associated with snakebites by ensuring prompt and accurate treatment. Tailored antivenom recommendations based on the specific snake species and symptoms aim to enhance treatment efficacy and minimize adverse effects.

Enhancing Accessibility to Treatment:

Increase accessibility to timely treatment in regions with limited access to medical facilities. By leveraging technology, the project aims to bridge the gap between snake incidents and medical intervention, particularly in areas where healthcare resources are scarce.

Contribution to Public Health:

Contribute valuable data and insights to the broader field of public health by establishing an integrated system that not only addresses the immediate challenges of snakebite envenomation but also serves as a model for proactive healthcare solutions in resource-constrained environments.

User-Friendly Interface:

Develop an intuitive and user-friendly interface for healthcare professionals to interact with the system, ensuring easy adoption and integration into existing healthcare workflows.

By achieving these goals, the project seeks to make a significant impact on the management of snakes, improving patient outcomes, and advancing the field of snakebite treatment and prevention.

Chapter 4

Data Pre-processing

First, we will discuss the dataset. To initiate, the data set and model creation plan will be discussed in depth to aid in the planning of the proposed model. Following that, we'll go through the proposed model architecture simulation method in detail, as well as the training methodology for determining the best parameter modifications. Ultimately, we'll use modeling approaches to demonstrate critical flaws in visual indicators in addition to creating a reported snake more identifiable.

About Dataset

The set of data comes from Roboflow[9] and contains about 8598 snake images. Per-photo was assigned to a category and was divided into groups by the respective class labels such as non-venomous and venomous. Since reformatting is among the most important phases of data preprocessing, all images are reformatted to 224×224 pixels. Figure 2 and Fig. 3 depicts several photographs from the benchmark dataset. Figures 3 and 4 demonstrate that there are numerous differences between venomous and non-venomous snakes in terms of physical appearances such as head structure, eye shape, skin color, and so on. The mentioned features will aid our proposed model in learning the distinctions between poisonous and non-poisonous snakes. The set of data has been split into train, validation, and test segments in an appropriate proportion

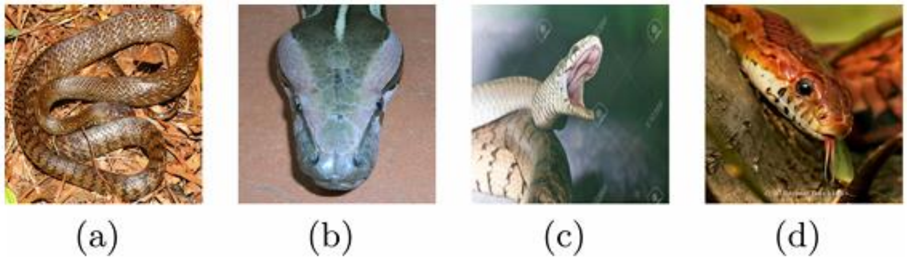

Figure 2; non-venomous snake image

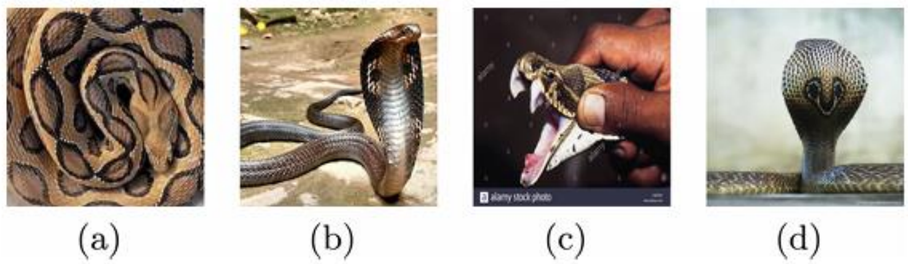

Figure 3: Venomous snake image

Chapter 5

Methodology

Convolutional neural networks are inspired by neurological mechanisms. A convolutional neural system is made up of many layers, including convolution layers, pooling layers, and fully connected layers, and it uses a back-propagation algorithm to obtain features to train the model properly

Model Construction

In this study, the framework was implemented using a Convolutional Neural Network (CNN),Xception,VGG-16,ResNet-50 and data augmentation. First, the architecture uses the dataset to take the pictures. Then preprocessing begins. Then some augmentation parameters are used to enlarge the dataset. Ultimately, the enlarged set of data is

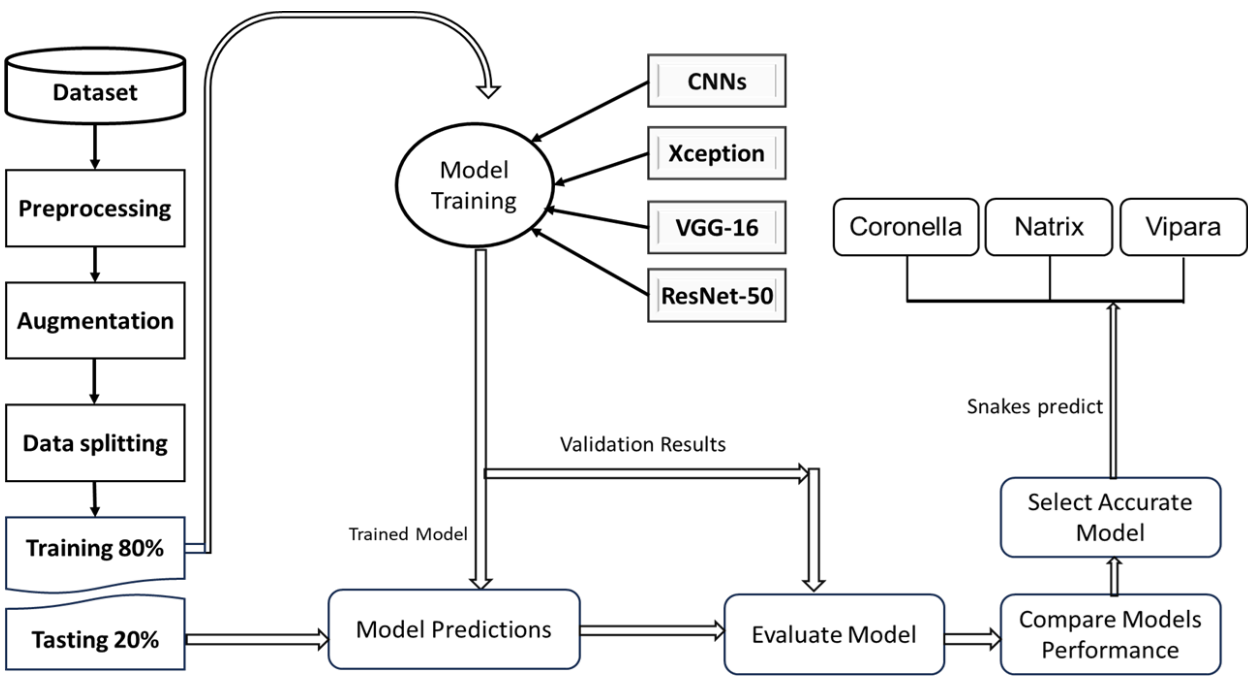

Figure 4 System flowchart

used to forecast the class by the CNN architecture. The pictures were standardized to some extent to be properly categorized. CNN itself performed the characteristic retrieval of the pictures.

The Implementation Procedure

The coding for the framework was produced and implemented in Google Colab [6] using the python programming language. Keras, Tensorflow, NumPy, and Matplotlib,streamlit,android studio were the libraries utilized in the whole study. The back end of the framework was chosen as Tensorflow, and keras has been utilized to offer additional built-in functionality such as activation functions, optimizers, layers, and so on. Keras API was used to enhance the dataset. NumPy is a Python library for mathematical evaluation. Confusion matrix, split train and test files, modelcheckpoint, callback mechanism, as well as other schematic representations like confusion matrix, loss against epochs, graphs, accuracy against epochs curves, and many more, are all generated using Sklearn. The matplotlib library is also needed to create visual representations of the previously mentioned diagrams, such as the confusion matrix.

Chapter 6

Success Criteria

Innovative Model Development:

Indicator: Development of a novel model integrating snake recognition and bite severity prediction.

Benefit: Advancement of snakebite management by addressing the dual challenge of snake identification severity.

Seamless Integration of Components:

Indicator: Cohesive integration of snake recognition and easily indentify snake and take a proper antivenom.

Benefit: Provision of a unified and comprehensive snake management system for efficient decision-making.

Improved Public Health Outcomes:

Indicator: Contribution to enhanced public health outcomes by reducing morbidity and mortality associated with snakebites.

Benefit: Empowerment of healthcare professionals and individuals with a quick, accurate, and user-friendly tool.

User Accessibility and Education:

Indicator: Design of an intuitive user interface ensuring accessibility for diverse users.

Benefit: Promotion of user education and awareness, empowering communities with actionable information.

These success criteria encapsulate the project's innovative goals, emphasizing the development of a holistic and impactful solution for snakebite management.

CNN:

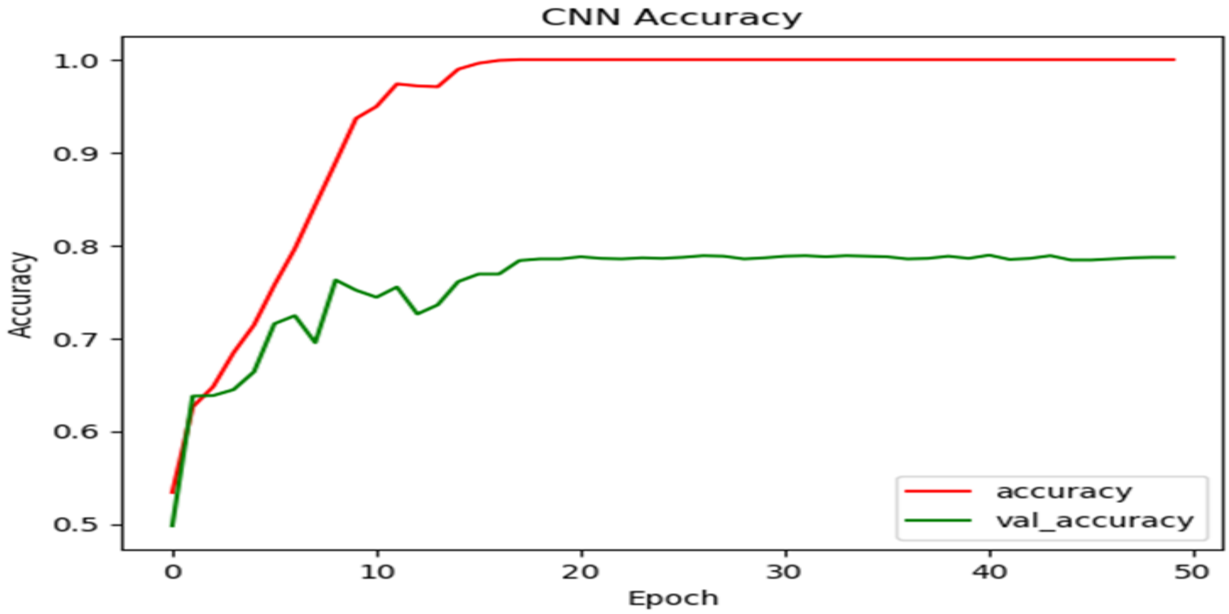

Figure 5 CNN accuracy

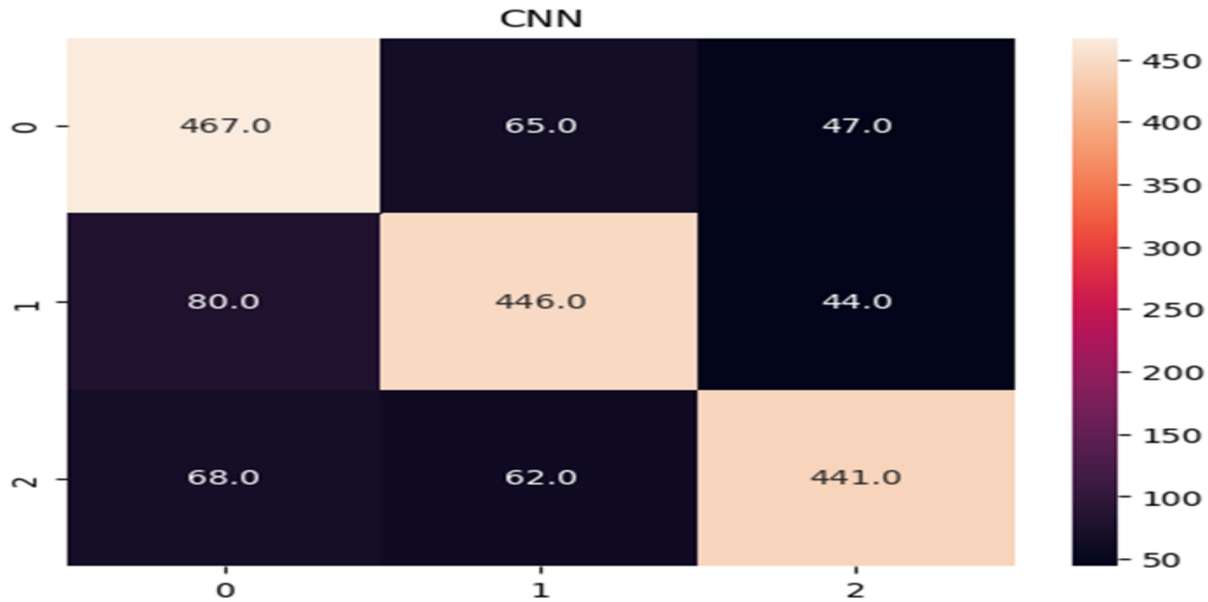

Figure 6 CNN confusion matrix

The CNN model for multi-class image classification using Keras. The model consists of four convolutional blocks with max pooling, two dense layers, and an output layer with softmax activation. The input images are rescaled by 1/255 for normalization. The model is compiled with the Adam optimizer and categorical cross-entropy loss, and accuracy is used as a performance metric. The model summary provides details on the layers, output shapes, and trainable parameters.

Xception:

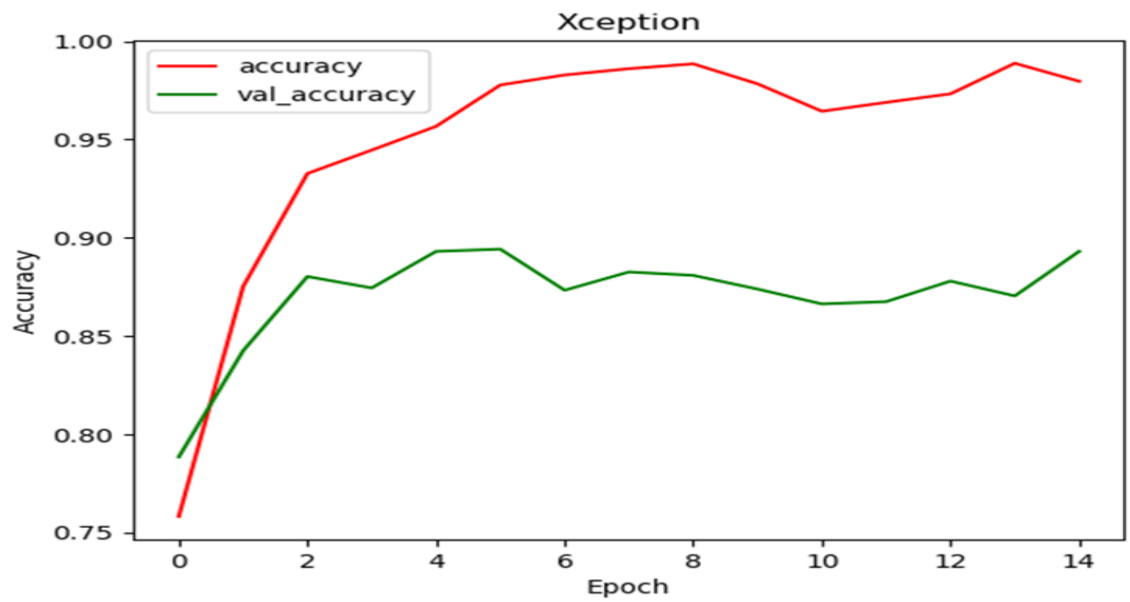

Figure 7 Xception accuracy

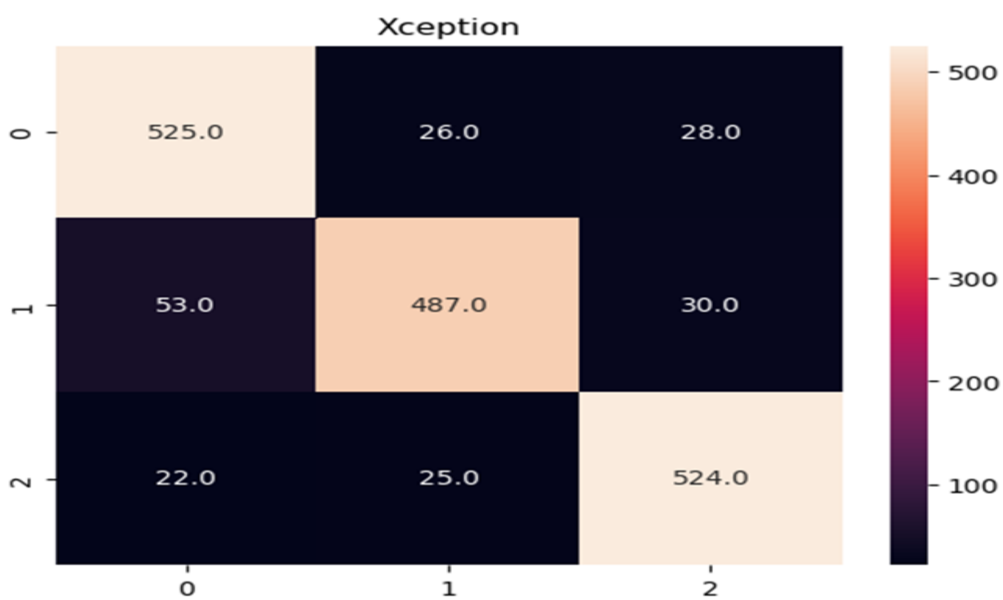

Figure 8 Xception confusion matrix

The Xception architecture pre-trained on ImageNet. The base model is frozen to keep its pre-trained weights fixed during training. An input layer is created with image normalization using a rescaling layer to normalize pixel values between 0 and 1. The input layer is connected to the base model, and a new output layer is added using a flatten layer and a dense layer with softmax activation. The new model is created by connecting the input and output layers, and it is compiled using categorical cross-entropy loss, the Adam optimizer, and accuracy as a performance metric. This approach leverages the pre-trained weights of the Xception architecture to improve performance on a smaller dataset.

VGG-16:

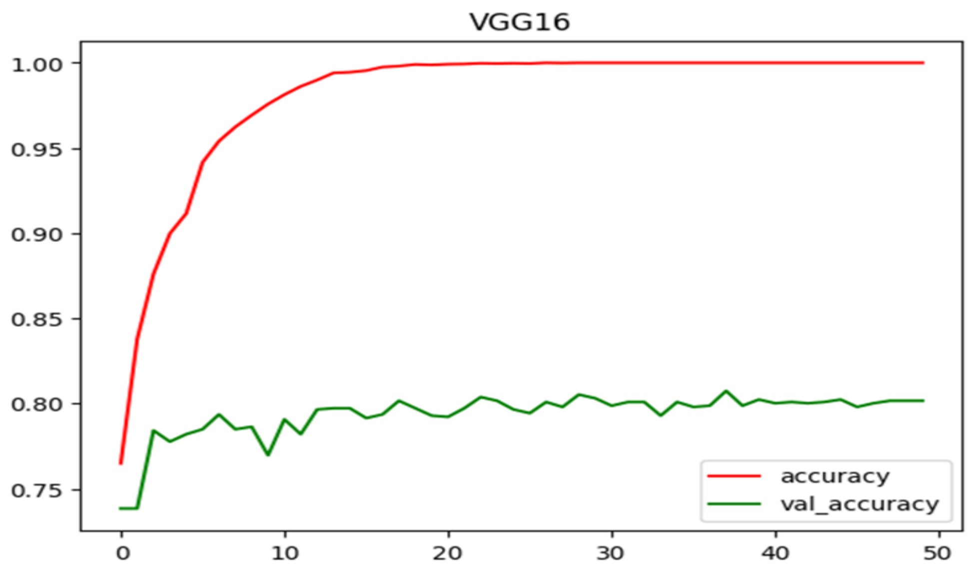

Figure 9 VGG-16 accuracy

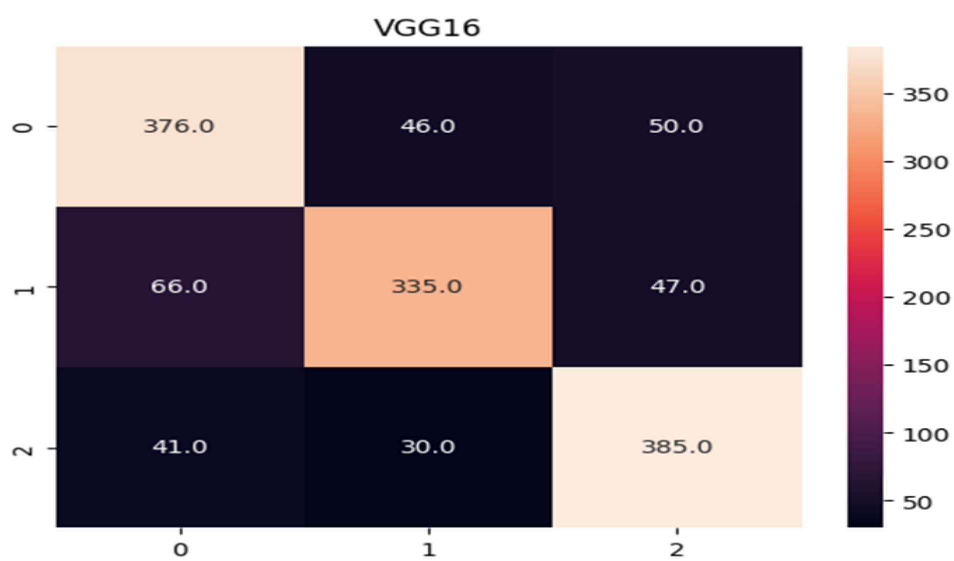

Figure 10 VGG-16 confusion matrix

The VGG16 architecture pre-trained on ImageNet. The base model is frozen to keep its pre-trained weights fixed during training. An input layer is created with image normalization using a rescaling layer to normalize pixel values between 0 and 1. The input layer is connected to the base model, and a new output layer is added using a flatten layer and a dense layer with softmax activation. The new model is created by connecting the input and output layers, and it is compiled using categorical cross-entropy loss, the Adam optimizer, and accuracy as a performance metric. This approach leverages the pre-trained weights of the VGG16 architecture to improve performance on a smaller dataset.

ResNet-50:

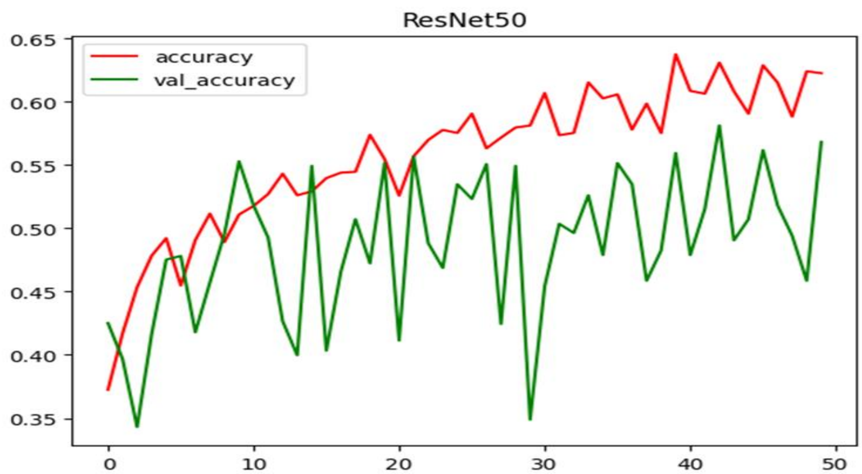

Figure 11 ResNet-50 accuracy

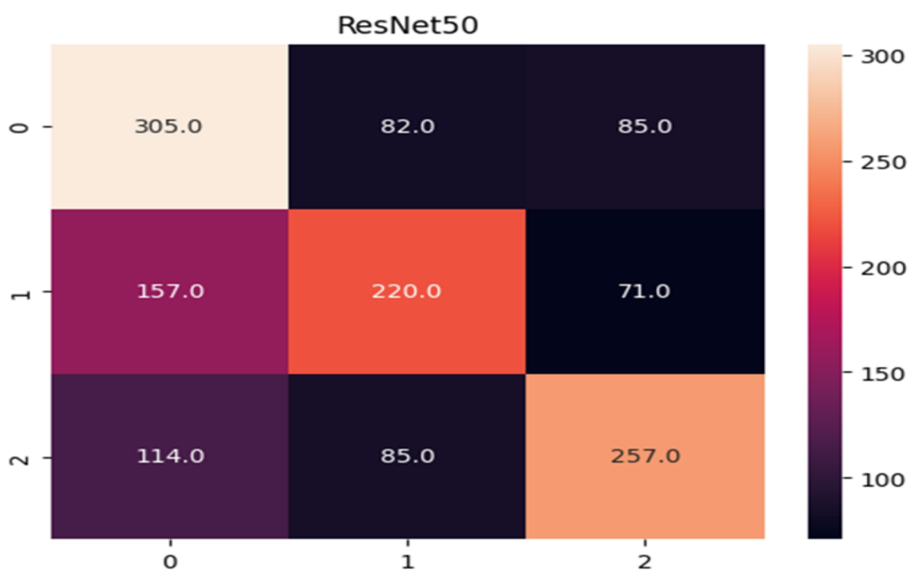

Figure 12 ResNet-50 confusion matrix

The ResNet50 architecture pre-trained on ImageNet. The base model is frozen to keep its pre-trained weights fixed during training. An input layer is created with image normalization using a rescaling layer to normalize pixel values between 0 and 1. The input layer is connected to the base model, and a new output layer is added using a flatten layer and a dense layer with softmax activation. The new model is created by connecting the input and output layers, and it is compiled using categorical cross-entropy loss, the Adam optimizer, and accuracy as a performance metric. This approach leverages the pre-trained weights of the ResNet50 architecture to improve performance on a smaller dataset.

Chapter 7

Web application

This Streamlit application provides a user-friendly interface for image classification using a pre-trained CNN model. The application is designed to classify images of snakes into three possible classes: "natrix", "coronella", and "vipera". The application allows users to upload an image of a snake, which is then preprocessed and fed into the model for prediction. The model returns a probability distribution over the three classes, and the application displays the uploaded image and a bar chart of the predicted probabilities for each class. The result is also displayed as a title above the bar chart.

The application uses OpenCV to preprocess the uploaded image. The image is first decoded from bytes to a NumPy array and then converted from BGR to RGB color space. The image is then resized to 224x224 pixels and normalized to have pixel values between 0 and 1. The normalized image is expanded to have a batch dimension and fed into the model for prediction. The model returns a probability distribution over the three classes, which is used to create a bar chart using Matplotlib.

The application uses the Streamlit library to create a user interface. The file uploader widget is used to allow users to upload an image. The uploaded image is displayed using the st.image() function. The bar chart is saved as an image and displayed using the st.image() function as well. The Lottie library is used to add an animation to the application to make it more engaging and visually appealing.

The pre-trained model used in this application is a CNN model saved in the "CNN.h5" file. The model was trained on a dataset of snake images and achieved high accuracy on a validation set. The model architecture consists of several convolutional layers with ReLU activation, max pooling layers, and a fully connected layer with softmax activation. The model was compiled with categorical cross-entropy loss and the Adam optimizer.

In summary, this Streamlit application provides an easy-to-use interface for snake image classification using a pre-trained CNN model. The application uses OpenCV for image preprocessing, Matplotlib for visualization, and Streamlit for the user interface. The pre-trained model used in this application is a CNN model that achieved high accuracy on a validation set. The application can be used for educational or research purposes to identify snakes in images.

Web View:

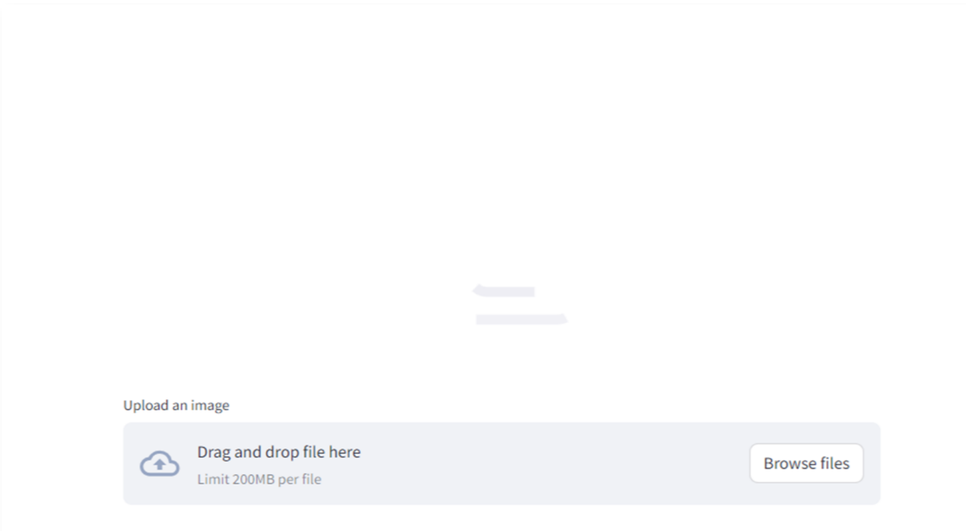

Figure 13 Web application view

Result:

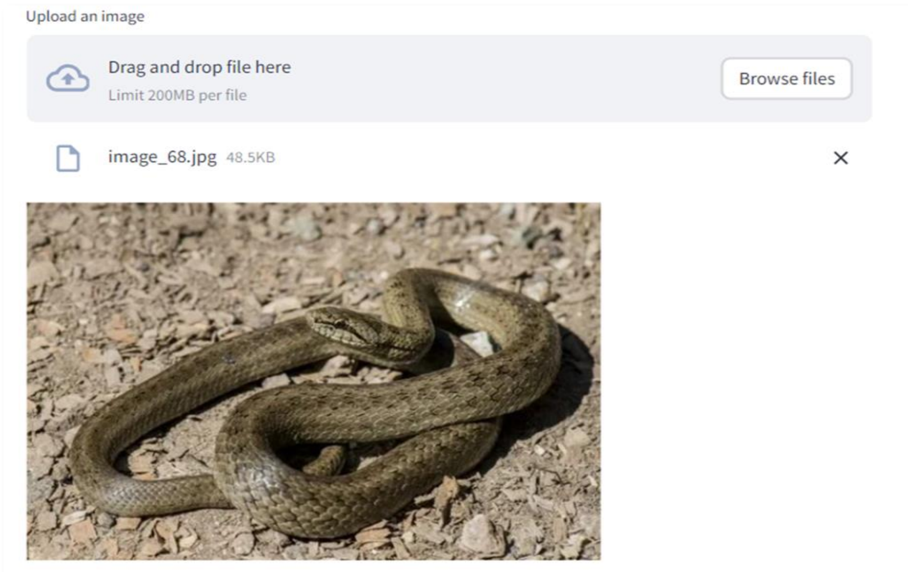

Figure 14 Application and result

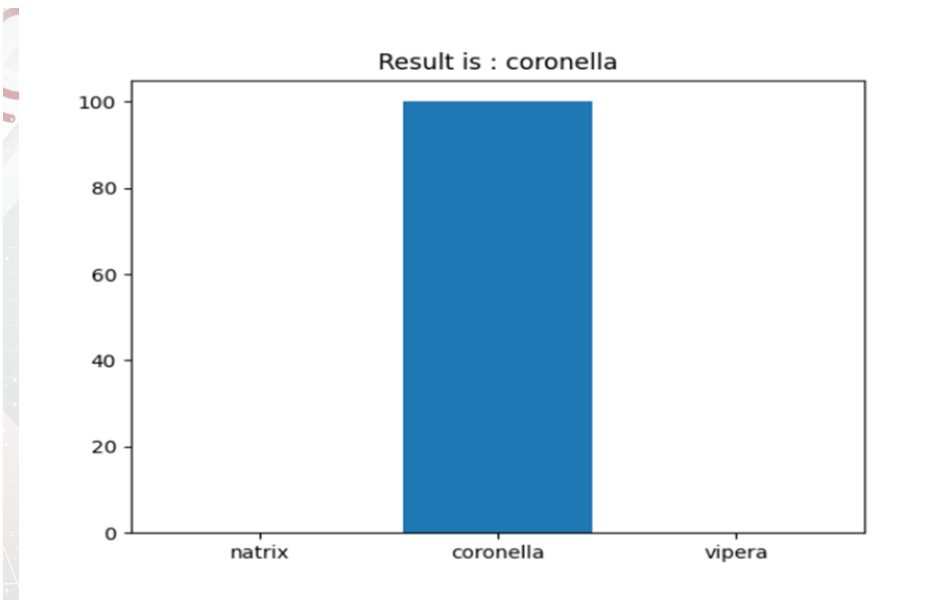

Figure 15 Application and result

Chapter 8

Android application

This is an Android application written in Kotlin for snake recognition using TensorFlow Lite. The application allows the user to select an image from their device and displays the predicted snake species and the corresponding probability. The main activity of the application sets up the user interface and defines the functionality for selecting an image and making a prediction.

The user interface consists of two buttons, one for selecting an image and another for making a prediction, an image view for displaying the selected image, and two text views for displaying the predicted snake species and the corresponding probability. The select button is set up to open the device's image gallery and allow the user to select an image. The predict button is set up to make a prediction using the selected image.

The application uses an ImageProcessor to preprocess the selected image before making a prediction. The ImageProcessor normalizes the image by subtracting the mean and dividing by the standard deviation of the ImageNet dataset, and then resizes it to 224x224 pixels using bilinear interpolation. The preprocessed image is then converted to a TensorImage, which is used as input to the TensorFlow Lite model.

The TensorFlow Lite model is loaded using the Model class, which is generated from a .tflite file. The model takes a single input tensor of shape [1, 224, 224, 3] and returns a single output tensor of shape [1, 3], where each element corresponds to the probability of the corresponding snake species. The output tensor is processed to find the index of the element with the highest probability, which is used to look up the predicted snake species in an array of snake labels.

The predicted snake species and the corresponding probability are displayed in the user interface. The application also displays the probability for each snake species in the array of snake labels.

The application overrides the onActivityResult method to handle the result of selecting an image from the device's image gallery. The selected image is loaded using the MediaStore.Images.Media.getBitmap method and then scaled down to 224x224 pixels using the Bitmap.createScaledBitmap method. The scaled image is then displayed in the image view.

In summary, this Android application uses TensorFlow Lite to recognize snake species from images selected by the user. The application preprocesses the selected image using an ImageProcessor and then passes it to the TensorFlow Lite model for prediction. The predicted snake species and the corresponding probability are displayed in the user interface, along with the probability for each snake species in the array of snake labels. The application handles the selection of an image using the onActivityResult method and displays the selected image in the image view.

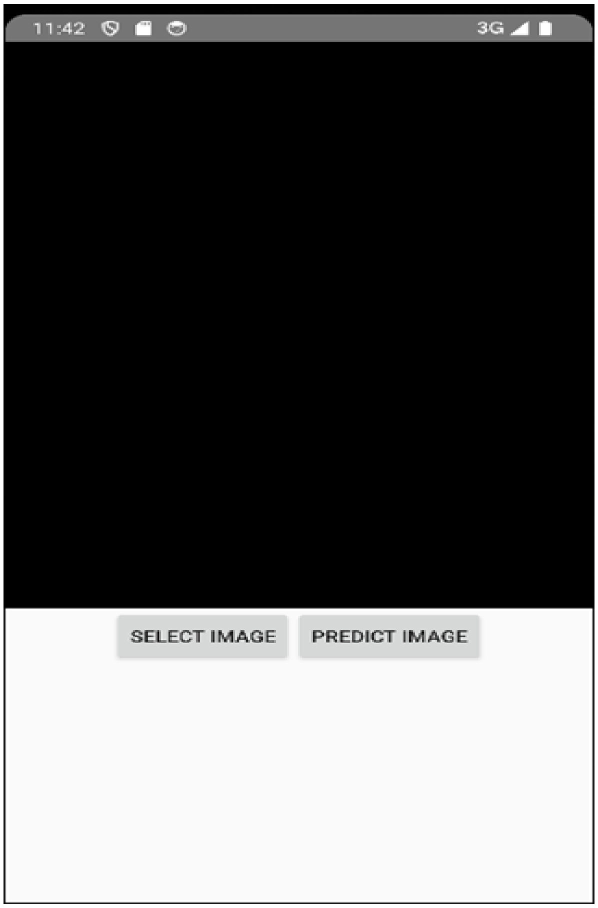

Figure 17 Android user interface and application

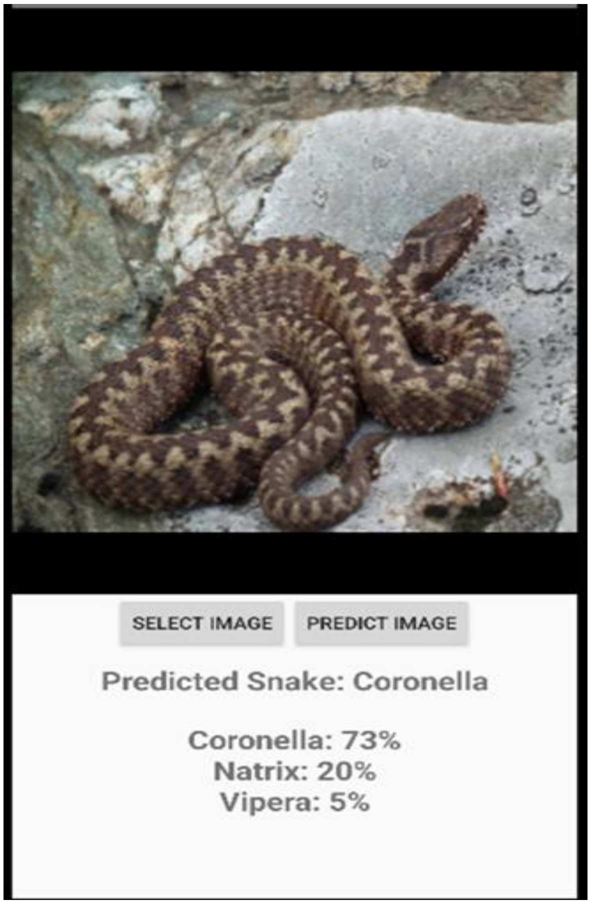

Figure 16 Android user interface and application

Chapter 9

Technology Stack/ Resources

Technology Stack:

In the realm of machine learning, the project relies on established frameworks such as TensorFlow or PyTorch, with a specific focus on Convolutional Neural Networks (CNNs) to facilitate image classification tasks. For image preprocessing, the project incorporates OpenCV, a library known for its versatility in tasks like resizing, normalization, and noise reduction. The backend development is orchestrated using either Streamlit, providing the necessary infrastructure for handling model inference. Here using the android studio for mobile application.

The deployment phase leverages cloud services such as AWS, Azure, or Google Cloud for scalability and reliability. Containerization is facilitated through Docker, ensuring seamless deployment across different environments. Git serves as the version control system, enabling collaborative development, and documentation is managed using tools like Swagger or ReDoc.

Resources:

Data forms the backbone of the project, and the dataset utilized is sourced from Roboflow, specifically the "Snake Breed Identification - v1 2022-07-28 9:18pm" dataset. The development environment is established using popular integrated development environments (IDEs) such as Vs Code or PyCharm. Communication and collaboration among the project team are facilitated through tools like Slack or Microsoft Teams.

Continuous integration and continuous deployment (CI/CD) pipelines are implemented using Jenkins, GitLab CI, or GitHub Actions, ensuring automated testing and deployment processes. Monitoring and logging are addressed with tools like Prometheus for monitoring and the ELK Stack (Elasticsearch, Logstash, and Kibana) for comprehensive logging capabilities. Security measures include the implementation of SSL certificates and adherence to OWASP best practices to safeguard the system from potential vulnerabilities.

This comprehensive technology stack and set of resources provide a solid foundation for the development and deployment of the "Snake and Bite Detection with Antivenom Suggestion" project. The choices made in each component reflect a balance between efficiency, scalability, and the specific requirements of the project.

Chapter 10

Risks and Dependencies

Risks:

The development of the "Snake Recognition" project is not without its challenges. One significant risk involves the quality of the training data for the machine learning model. Inadequate or biased data could compromise the model's accuracy. To address this, the project will implement thorough data preprocessing, ensuring the dataset is comprehensive and representative.

Another risk pertains to the model's ability to generalize across diverse snake species. Overcoming this challenge is crucial for the project's success. Leveraging a diverse dataset and incorporating transfer learning techniques during model development are key strategies to mitigate this risk.

Ensuring the reliability of medical decisions made by the system is paramount. Incorrect antivenom recommendations could have severe consequences. Continuous updates to the antivenom knowledge base and collaboration with medical experts form a core strategy to enhance the accuracy of recommendations and mitigate this risk.

Deployment introduces its own set of challenges, where issues during this phase could impact the overall availability of the system. Thorough testing procedures and a gradual rollout strategy will be implemented to minimize potential disruptions during deployment.

User adoption, particularly among healthcare professionals, poses a risk to the project. Overcoming resistance to adopting a new system is a common challenge. To address this, the project will develop comprehensive user training programs and establish mechanisms for continuous user feedback, fostering a user-friendly environment.

Ethical considerations are also a significant risk. There is a risk of the system being misused for unintended purposes. To mitigate this, the project will implement strict access controls and adhere to ethical guidelines.

Dependencies:

The success of the project is dependent on several crucial factors. The availability of a diverse dataset for model training is a fundamental dependency. Collaboration with data providers and exploration of additional datasets are essential to ensure the robustness of the model.

Continuous updates to the antivenom knowledge base represent another critical dependency. The project relies on up-to-date information, necessitating ongoing partnerships with medical institutions and engagement in continuous research efforts.

Collaboration with healthcare professionals is a crucial dependency for input and validation. Establishing partnerships with medical practitioners and conducting regular consultations will ensure that the system aligns with real-world medical practices.

Adherence to healthcare regulations and ethical guidelines is a foundational dependency, requiring engagement with legal and compliance experts. Staying informed about regulatory changes is essential to ensure compliance throughout the project.

Lastly, continuous user feedback for system improvement is a vital dependency. Implementing mechanisms and conducting user surveys will foster a collaborative environment, ensuring that the system remains responsive to user needs and concerns.

Chapter 11

Conclusions

In summary, the "Snake Recognition" project integrates advanced technologies, including machine learning and geolocation services, to create a sophisticated system. The innovative antivenom recommendation system, supported by a dynamic knowledge base, distinguishes this project in addressing the complexities of snakebites.

The chosen technology stack reflects a balance between efficiency and user-friendliness. As the project advances, collaboration with healthcare professionals, ethical considerations, and user feedback will be crucial. The system's potential impact on public health outcomes, through rapid snake detection and informed interventions, is significant.

The project stands at the forefront of innovation, offering a holistic solution to reshape snake management. By providing timely and effective interventions, it holds the promise of reducing morbidity and mortality associated with snakebites, benefiting both healthcare professionals and individuals in snake-prone areas.

References

1. Abadi, M., et al.: Tensorflow: a system for large-scale machine learning. In: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 2016), pp. 265–283 (2016)
2. Abdurrazaq, I.S., Suyanto, S., Utama, D.Q.: Image-based classification of snake species using convolutional neural network. In: 2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), pp. 97–102. IEEE (2019)
3. Abeysinghe, C., Welivita, A., Perera, I.: Snake image classification using siamese networks. In: Proceedings of the 2019 3rd International Conference on Graphics and Signal Processing, pp. 8–12 (2019)
4. Albawi, S., Mohammed, T.A., Al-Zawi, S.: Understanding of a convolutional neu ral network. In: 2017 International Conference on Engineering and Technology (ICET), pp. 1–6. IEEE (2017)
5. Annesa, O.D., Kartiko, C., Prasetiadi, A., et al.: Identification of reptile species using convolutional neural networks (CNN). Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi) 4(5), 899–906 (2020)
6. Bisong, E.: Google colaboratory. In: Building Machine Learning and Deep Learning Models on Google Cloud Platform, pp. 59–64. Springer (2019)
7. Bloch, L., et al.: Combination of image and location information for snake species identification using object detection and efficientnets. CLEF working notes (2020)
8. Chauhan, R., Ghanshala, K.K., Joshi, R.: Convolutional neural network (CNN) for image detection and recognition. In: 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), pp. 278–282. IEEE (2018)
9. snake breed identification - v1 2022-07-28 9:18pm (roboflow.com)