Image Classification

Using Deep Learning & Streamlit for Real-Time Image Classification

Developer : Aditya Wagh

Introduction

- Image Classification Using CNN: The project utilizes a Convolutional Neural Network (CNN) to classify images from the CIFAR-10 dataset into 10 categories such as airplanes, cars, and animals.

- Web-Based Application: Built with Streamlit, the application allows users to upload images and receive real-time predictions, making it easy to interact with the model.

- User-Friendly Interface: The purpose is to provide an accessible and visually appealing platform for users to explore image classification without needing deep technical knowledge.

Objective

- Easy Image Classification: Provide a simple platform where users can upload images and get instant predictions using a Convolutional Neural Network (CNN).

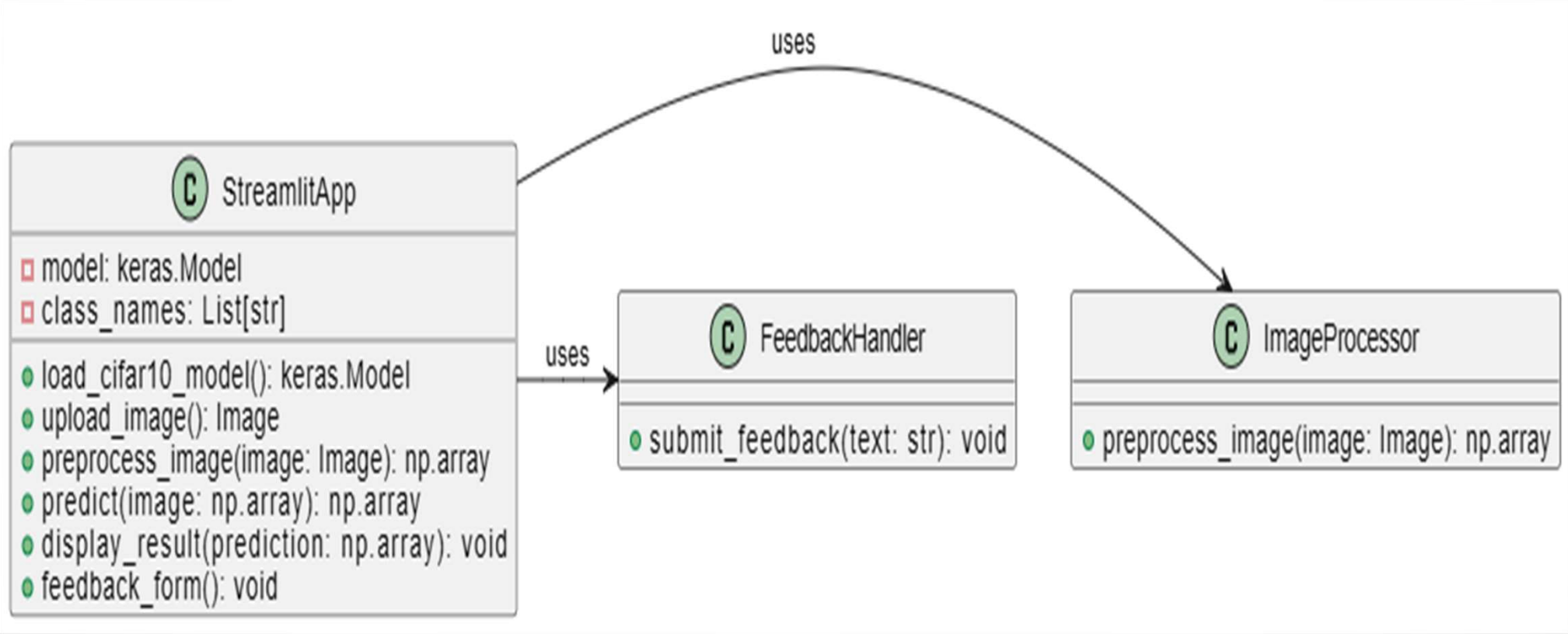
- Accessible to Everyone: Make advanced image classification available to non-technical users through a user-friendly interface built with Streamlit.

- Real-Time Interaction: Allow users to experience real-time image classification with minimal effort, making it both interactive and engaging.

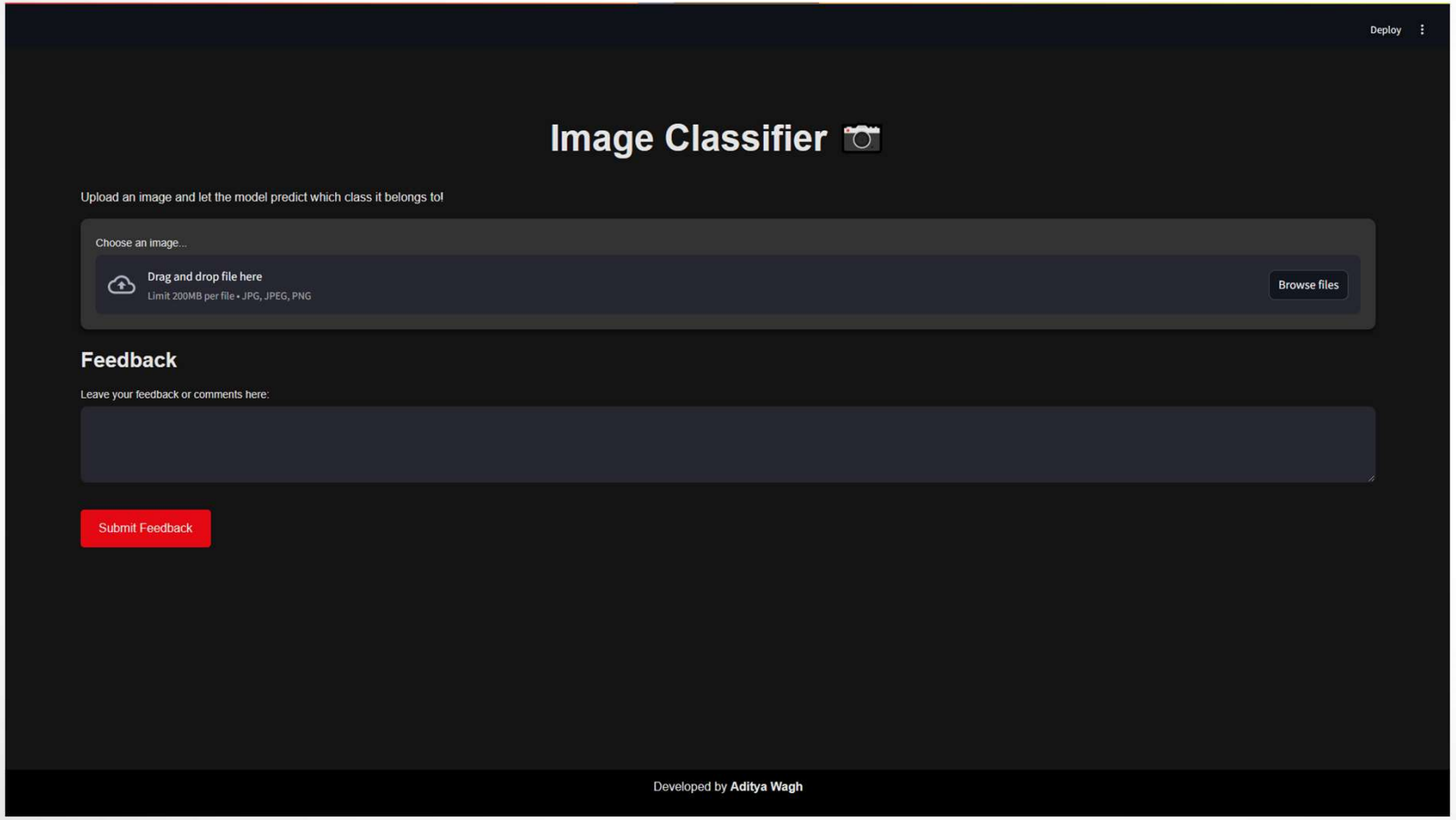
Technology Stack

- Python
- TensorFlow
- Streamlit
- CNN (Convolutional Neural Network)
- CIFAR-10 dataset

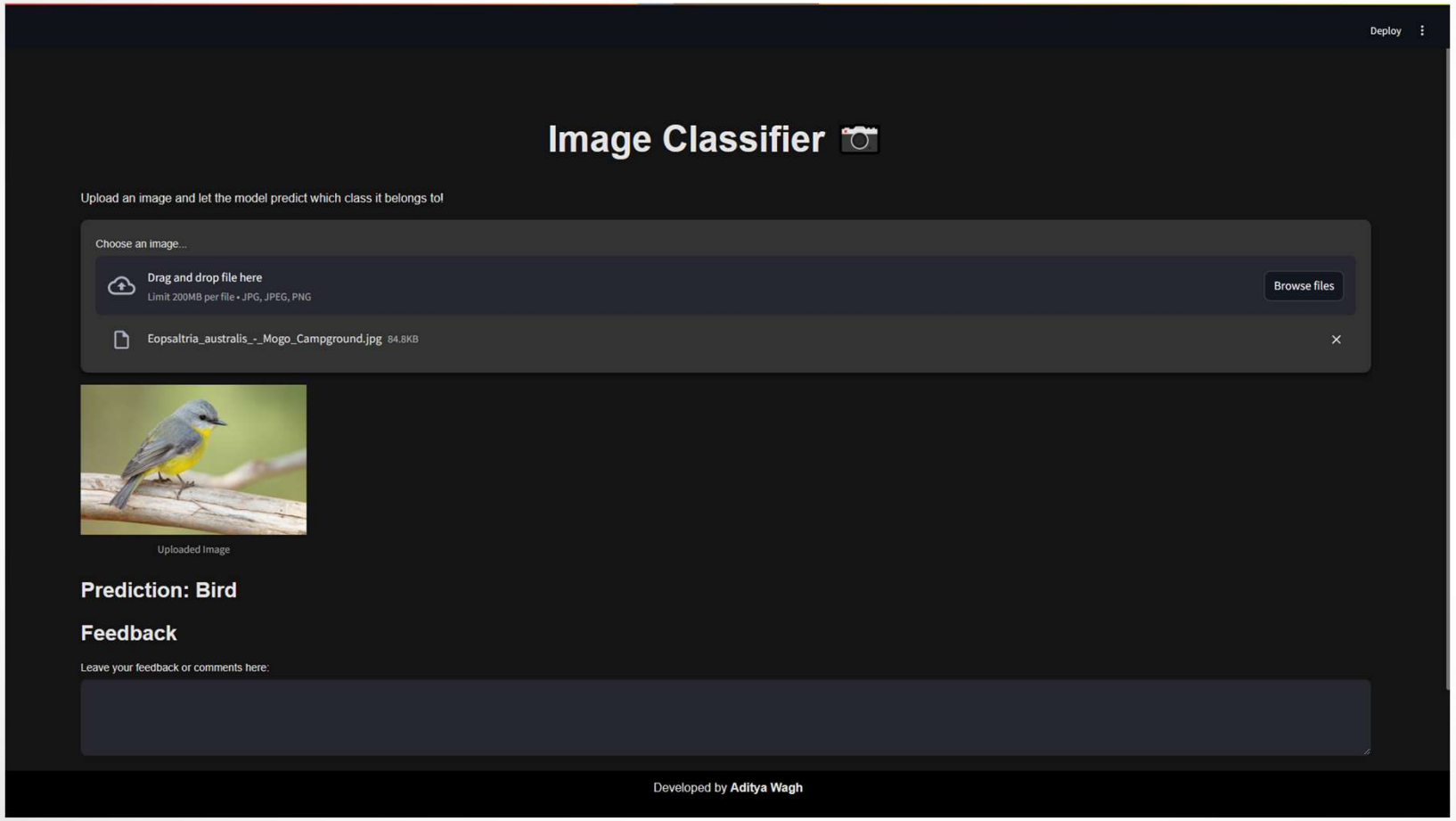
UML Diagram


uses
StreamlitApp
model: keras.Model
class_names: List[str]
load_cifar10_model(): keras.Model
upload_image(): Image
preprocess_image(image: Image): np.array
predict(image: np.array): np.array
display_result(prediction: np.array): void
feedback_form(): void
uses
FeedbackHandler
submit_feedback(text: str): void
ImageProcessor
preprocess_image(image: Image): np.array

Webpage


Deploy
Image Classifier
Upload an image and let the model predict which class it belongs to!
Choose an image...
Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG
Browse files
Feedback
Leave your feedback or comments here:
Submit Feedback
Developed by Aditya Wagh

Webpage (After classification)



Feedback Option

Key Features

- Real-Time Predictions: Instantly classify uploaded images using a pre-trained CNN model.
- User-Friendly Interface: Simple and intuitive web design, making it accessible for users without technical expertise.
- Support for Multiple Image Types: Handles common formats like JPG, JPEG, and PNG for easy uploads.

Future Enhancements

- Improved Model Accuracy: Integrate advanced models like ResNet or EfficientNet for better image classification accuracy.
- Mobile and Cloud Integration: Expand the app to support mobile devices and cloud-based processing for scalability and enhanced accessibility.
- Enhanced User Interface: Add interactive features like drag-and-drop uploads and real-time visual feedback to improve the overall user experience.

Conclusion

- Effective Image Classification: The app successfully classifies images in real-time using a CNN model, making advanced technology easy to use for anyone.

- Smooth User Experience: With a simple and interactive design, users can easily upload images and get instant predictions.

- Room for Improvement: There are many possibilities to improve the app, like using better models, making it work on mobile devices, and enhancing the user interface.

THANKYOU !