JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Species Identification Using Image Recognition

Mr. Pratharv Surve, Mr. Vinayak Pandey, Mr. Utkarsh Pandey

Assistant Professor, Undergraduate Student, Undergraduate Student
Department of Information Technology
University of Mumbai, Mumbai, India

Abstract: This research paper presents the development of a species identification system using image recognition. The project leverages modern front-end technologies, including React Native, JavaScript, React Native Expo, Gemini API, and Snack Expo to enable accurate species detection from images. Unlike traditional systems that rely on complex backend infrastructure and databases, this approach ensures a lightweight, efficient, and user-friendly mobile application. React Native Expo simplifies development and deployment, making the app cross-platform and easily accessible. The Gemini API processes images and provides species identification results, eliminating the need for manual classification or local computation. JavaScript ensures smooth application functionality, while Snack Expo facilitates quick testing and debugging in a browser. This system is designed for nature enthusiasts, students, and researchers who require quick and accurate species identification on the go. By leveraging cloud-based AI models, the project eliminates the need for heavy computation on mobile devices, maintaining high accuracy and efficiency.

Index Terms – Species identification, Image recognition, React Native, React Native Expo, JavaScript, Gemini API, Snack Expo, Mobile application, AI-powered recognition.

I. Background and Motivation

Species identification plays a crucial role in biodiversity research, wildlife conservation, and environmental education. Traditionally, identifying species required expert knowledge, field guides, and laboratory analysis. With advancements in technology, automated species recognition has become a powerful tool, allowing researchers, students, and nature enthusiasts to identify species efficiently using digital platforms.

However, many existing species identification systems rely on complex machine learning models, backend databases, and high computational resources, making them difficult to implement on mobile devices. These systems often require continuous internet connectivity and large storage capacities, limiting accessibility and ease of use.

This project aims to bridge this gap by developing a lightweight, real-time species identification mobile application. By leveraging React Native, JavaScript, React Native Expo, Gemini API, and Snack Expo, the system eliminates the need for backend infrastructure while maintaining high accuracy. The motivation behind this research is to create an accessible, fast, and user-friendly tool that allows individuals to identify species effortlessly, promoting wildlife awareness and conservation efforts.

II. INTRODUCTION

The advancement of mobile technology has revolutionized species identification, making it more accessible to researchers and enthusiasts. Traditional methods often rely on large datasets and complex machine learning models, which require high computational power and backend infrastructure. This project introduces a lightweight, API-driven approach, leveraging React Native for mobile development, JavaScript for application logic, and the Gemini API for AI-powered image recognition.

Users can capture or upload images, which are processed by the Gemini API to identify species accurately. Unlike traditional methods, this eliminates the need for machine learning models, databases, or heavy backend infrastructure, ensuring a fast, efficient, and user-friendly experience. The application, built with React Native Expo, provides a seamless cross-platform experience, running smoothly on both Android and iOS. Snack Expo is used for testing and debugging, ensuring a streamlined development process.

This research focuses on enhancing accessibility, usability, and scalability in species identification, making it a practical tool for education, wildlife research, and conservation efforts. Future enhancements may include offline functionality, multilingual support, and improved user experience to expand accessibility.

III. Literature Review

III.1 React Native & React Native Expo

React Native is a widely used framework for building cross-platform mobile applications using JavaScript. It allows developers to maintain a single codebase while ensuring native-like performance on both Android and iOS. React Native Expo simplifies development by providing built-in tools for testing, debugging, and deployment, eliminating the need for complex native configurations.

Key libraries used in this project include:

- expo-image-picker – Allows users to select images from their gallery or capture new ones using the camera.
- expo-camera – Provides real-time access to the device's camera for capturing images.
- React Navigation – Manages app navigation efficiently, ensuring smooth transitions between screens.

III.2 Gemini API

The Gemini API provides AI-powered image recognition, enabling real-time species identification. Instead of relying on complex machine learning models, backend databases, or extensive datasets, this API processes images remotely and returns accurate results. This approach reduces computational overhead on mobile devices while ensuring high precision in species identification.

III.3 JavaScript for Application Logic

JavaScript serves as the core programming language for this project, handling user interactions, managing API calls, and ensuring smooth application functionality. Its dynamic nature allows for efficient data handling and seamless UI updates, improving the overall user experience.

III.4 Snack Expo for Testing

Snack Expo is an online platform used for testing React Native applications directly in a browser. It allows developers to experiment with UI changes, debug components, and test API integrations without requiring a physical device or emulator setup. This tool enhances development efficiency by providing real-time feedback on code modifications.

IV. Methodology

IV.1 Modular Architecture

The application follows a modular architecture to enhance scalability, maintainability, and efficiency. It consists of four key components:

- Image Upload Module – Enables users to select images from the gallery or capture new ones using expo-camera or expo-image-picker.
- API Communication Module – Manages communication with the Gemini API, sending images and retrieving species identification results.
- Results Display Module – Presents species identification results in a structured and user-friendly format, including common and scientific names.

- Testing & Debugging Module – Utilizes Snack Expo for real-time testing and debugging without requiring a physical device or emulator.

IV.2 UI/UX Considerations

The user interface is designed for simplicity and ease of use, ensuring an intuitive experience with:

- A single-click upload button for image selection.
- A visually clean layout to display species identification results clearly.
- A refresh button to allow users to initiate new searches instantly.

IV.3 Security Measures

To ensure data privacy and security, the application implements the following measures:

- Local Pre-Processing – Images undergo minimal pre-processing before being sent to the Gemini API.
- Secure API Requests – All communication with the API is encrypted to prevent unauthorized access.
- No Authentication Required – The app functions without storing personal data, ensuring user anonymity.

V. Results and Discussion

The application was successfully developed using React Native, React Native Expo, JavaScript, Gemini API, and Snack Expo for real-time species identification. Extensive testing confirmed the following:

- High Accuracy – The Gemini API provides precise species identification when given clear and well-lit images.
- Seamless Performance – React Native and React Native Expo ensure smooth functionality and cross-platform compatibility on Android and iOS.
- User-Friendly Interface – The intuitive UI allows users to quickly upload images and receive identification results with minimal delays.

V.1 Security & Challenges

While the application performs effectively, certain security measures and challenges were observed:

- Secure API Communication – The app ensures data integrity and user privacy through encrypted API requests.
- Challenges Encountered:
  - Blurry or low-quality images resulted in less accurate species identification.
  - Network dependency affected API response times, leading to occasional delays.

V.2 Future Enhancements

To further improve usability and performance, the following enhancements are proposed:

- Offline Identification – Implementing local AI models for species identification without internet dependency.
- UI/UX Enhancements – Refining the user experience to improve engagement and accessibility.
- Expanded Species Database – Incorporating a larger dataset to improve identification accuracy and species coverage.

VI. Ethical Considerations and Limitations

VI.1 Ethical Considerations

The implementation of an AI-driven species identification system introduces several ethical concerns that must be addressed to ensure responsible use. The following key ethical considerations are taken into account:

- Data Privacy – Since the application processes user-uploaded images, maintaining user anonymity is a priority. No personal data is stored, shared, or misused, ensuring compliance with privacy standards.
- Bias in AI Models – AI-based recognition systems may have limitations in identifying lesser-known or rare species, potentially leading to inaccurate classifications. Continuous updates and improvements to the Gemini API are necessary to mitigate biases.

- Environmental Impact – Cloud-based AI models require server operations and energy consumption. While the application minimizes computational demands on mobile devices, sustainable AI practices should be considered to reduce environmental impact.

VI.2 Limitations

Despite its advantages, the application has certain limitations that need to be acknowledged:

- Dependence on Internet Connectivity – The Gemini API requires an active internet connection for real-time species identification. Users in remote or low-connectivity areas may face challenges in obtaining results.
- Image Quality Sensitivity – Identification accuracy is highly dependent on image quality. Blurry, poorly lit, or incomplete images may lead to misclassification or failed recognition.
- Limited Offline Functionality – Since the system does not utilize local AI models, it cannot perform species identification offline. Future improvements may involve integrating lightweight AI models for offline functionality.

VII. Conclusion

This project introduces a lightweight and efficient approach to species identification by leveraging React Native, JavaScript, React Native Expo, Gemini API, and Snack Expo. By eliminating complex machine learning models and backend databases, the application provides a real-time, user-friendly solution for species identification through images.

The Gemini API delivers highly accurate results when provided with clear and well-lit images. React Native Expo simplifies app development and deployment, while Snack Expo facilitates real-time testing and debugging, ensuring an efficient development workflow. However, challenges such as network dependency may affect response times and accessibility in low-connectivity areas.

Future Enhancements:

- Offline species identification using lightweight AI models to reduce dependency on internet connectivity.
- Enhanced UI/UX features to improve user experience and engagement.
- Expanded species information, providing additional details beyond basic identification, such as habitat, behavior, and conservation status.

VIII. Acknowledgment

We would like to express our sincere gratitude to our project guide, Mr. Prathav Surve, for his invaluable guidance, encouragement, and continuous support throughout this project. His expertise and insights have been instrumental in shaping the development of this species identification application.

We also extend our appreciation to our college faculty and mentors for their constructive feedback, which has helped refine this project. Special thanks to the developers and contributors behind React Native, React Native Expo, JavaScript, Gemini API, and Snack Expo, whose technologies made this implementation possible.

We are grateful to our friends and peers for their assistance in testing and providing valuable feedback to enhance the application's usability. Lastly, we would like to thank our families for their unwavering support and motivation throughout this journey.

This project is the result of collaborative efforts, and we sincerely appreciate everyone who contributed directly or indirectly to its success.

References

[1] React Native Documentation, "Building Mobile Applications with React Native," Available: https://reactnative.dev/docs/getting-started, Accessed Feb. 2025.

[2] Snack Expo Documentation, "Testing React Native Applications," Available: https://snack.expo.dev/, Accessed Feb. 2025.

[3] JavaScript Documentation, "JavaScript Guide for Developers," Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript, Accessed Feb. 2025.

[4] Gemini API Guide, "AI-Powered Image Recognition API," Available: https://ai.google.dev/, Accessed Feb. 2025.

[5] J. Resig and B. Bibeault, *Secrets of the JavaScript Ninja*, 2nd ed., Manning Publications, 2016.

[6] D. Crockford, *JavaScript: The Good Parts*, O'Reilly Media, 2008.

[7] N. Zakas, *Understanding ECMAScript 6: The Definitive Guide for JavaScript Developers*, No Starch Press, 2016.

[8] W. T. Freeman and E. H. Adelson, "The Design and Use of Steerable Filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891-906, 1991. Available: https://ieeexplore.ieee.org/document/873290, Accessed Feb. 2025.

[9] M. Fowler and K. Beck, *Refactoring: Improving the Design of Existing Code*, 2nd ed., Addison-Wesley, 2018.