Automatic Case Report Generator by Image Detecting using AI

Aditya Waghmare[1], Shrutika Raghorte[1], Aayushi Ingole[1] , Divyani Jabalpure[1], Susmita Chide[1] ,Dinesh Jamthe[2]

[1,2]*Department of Computer Science and Engineering, Priydarshini Bhagwati Collage of Engineering Nagpur,India*

---***---

Abstract - The advancement of artificial intelligence (AI) and image detection technologies has opened new avenues for automating complex tasks that traditionally required significant human effort. This research presents the development of an Automatic Case Report Generator Using Image Detection, a comprehensive system designed to automate the generation of case reports through advanced image detection and AI-driven text generation. The system leverages React Native for the frontend, Expo for server-side operations, and MongoDB Atlas for cloud-based data storage. Central to this project are four RESTful APIs built using Node.js: the MongoDB Atlas connection, image detection via the Google Gemini API, location detection via the OpenCage API, and case report generation through ChatGPT.

1.INTRODUCTION

The automation of complex and time-consuming tasks has become increasingly important in various industries, driven by advancements in technology and the growing demand for efficiency. One such task is the generation of case reports, which are essential in fields such as law enforcement, healthcare, and insurance. Traditionally, creating these reports requires a significant amount of manual effort, including data collection, analysis, and documentation. The process can be prone to human error, time-consuming, and resource-intensive, making it a prime candidate for automation.

Automated report generation has been a focus of research for many years, particularly in domains such as journalism, finance, and healthcare. Early systems relied on rule-based methods to generate reports, which, while effective in specific contexts, were limited in scope and flexibility. The introduction of natural language processing (NLP) and AI-powered text generation models, such as GPT-2 and GPT-3, marked a significant leap forward in this field. These models, developed by OpenAI, are capable of generating coherent and contextually relevant text based on a given prompt, making them ideal for automating report writing tasks.

Several studies have demonstrated the effectiveness of AI in generating reports that are indistinguishable from those written by humans. For example, Lee et al. (2020) explored the use of AI-generated text in financial reporting, concluding that AI could significantly reduce the time required to produce accurate reports. Similarly, in the healthcare sector, Wang et al. (2021) showed that AI could assist in generating medical reports based on patient data, improving both the speed and accuracy of documentation.

2. Body of Paper

The Automatic Case Report Generator Using Image Detection is a complex system that integrates multiple technologies to automate the generation of case reports based on captured images, user inputs, and geolocation data. The architecture of the system is designed to be modular, scalable, and efficient, leveraging both frontend and backend technologies to provide a seamless user experience. This section outlines the system architecture, detailing the key components and their interactions.

The implementation of the Automatic Case Report Generator involves integrating various technologies and APIs to create a seamless system that captures images, processes them, and generates detailed case reports. The system is divided into multiple modules, each responsible for different aspects of the process.

1. Image Capture and Detection

A key feature of the system is its ability to capture images and extract useful information. This is achieved using the device’s camera through React Native’s Expo Camera API. Once the image is captured, it is sent to the Google Gemini API, which performs image recognition and detection tasks. The API identifies objects or relevant details in the image that are crucial for the case report.

2. Case Report Generation

The detected information from the image is used to automatically generate a case report. For this purpose, the system integrates ChatGPT API, which utilizes natural language processing to generate human-readable reports based on the extracted data. The AI model is provided with context from the image (via Gemini API) and generates structured text describing the case. This text is then formatted into a case report template and saved in the database.

The Automatic Case Report Generator Using Image Detection is a sophisticated system that integrates advanced technologies to automate the creation of detailed case reports based on image analysis, user inputs, and geolocation data. Designed for modularity, scalability, and efficiency, the system provides a seamless user experience by leveraging both frontend and backend technologies. This section outlines the architecture and key components of the system, describing how they interact to deliver comprehensive case reports.

The implementation of the Automatic Case Report Generator revolves around multiple modules, each responsible for different aspects of the process. The system efficiently captures and processes images, detects relevant details, and generates structured case reports with minimal user intervention.

One of the core features is the ability to capture images and extract meaningful information from them. Through React Native's Expo Camera API, users can easily capture images using the device's camera. These images are then processed using the Google Gemini API, a powerful image recognition tool. This API identifies key objects and relevant scene details, which are critical for compiling an accurate case report.

Once the image data has been analyzed by the Gemini API, it is used to automatically generate a detailed case report. The system utilizes the ChatGPT API, which employs natural language processing (NLP) to convert the extracted data into a well-structured, human-readable report. This generated report includes important information such as incident description, time, location, and objects identified in the image. The report is formatted into a predefined template and stored in the database for future reference.

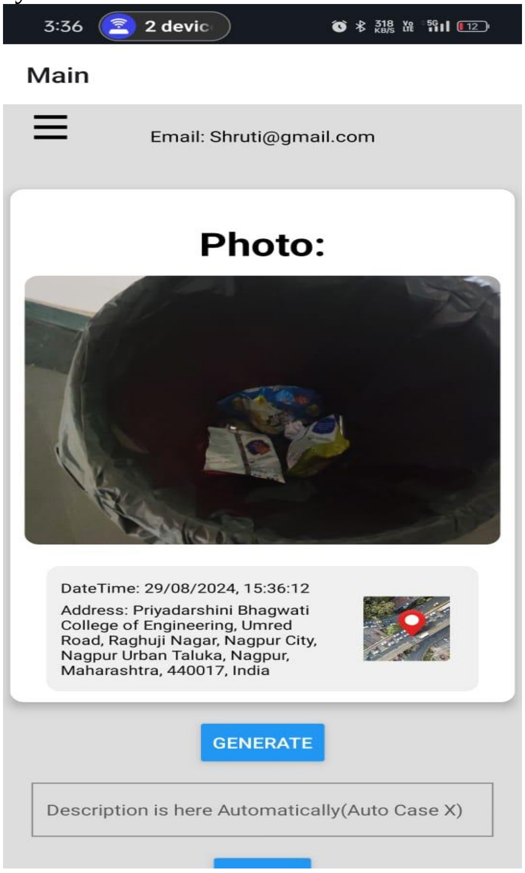
To further enhance the relevance of the case report, the system integrates geolocation services via the OpenCage API, capturing the user's precise location when the image is taken. This geolocation data adds valuable context, linking the detected objects in the image to a specific location and time, which is essential for creating a complete and accurate case report.

The generated reports, along with the images and corresponding data, are securely stored in a cloud-based database using MongoDB Atlas. The backend, developed with Node.js, ensures efficient data management and communication between the frontend, APIs, and database, making the process of report generation smooth and reliable. This architecture supports scalability, enabling the system to handle a large volume of users and data.

On the frontend, the user interface, built with React Native, offers an intuitive experience. Users can capture images, view generated reports, and access historical data through a clean, user-friendly interface. The navigation is simplified with a hamburger menu, which organizes different sections like new case creation, report history, and help resources.

Overall, the Automatic Case Report Generator Using Image Detection significantly reduces manual effort by automating the entire process from image capture to report generation. It ensures the accuracy and relevance of the case reports by using advanced APIs for image detection and natural language generation. With its modular architecture and cloud-based infrastructure, the system is scalable and adaptable to various use cases, making it an efficient and indispensable tool for industries requiring accurate case documentation.

The integration of AI-driven technologies into web and mobile applications via RESTful APIs is a growing trend, enabling developers to incorporate sophisticated features without the need for deep expertise in AI. This approach has been widely adopted in various fields, including e-commerce, healthcare, and social media, where AI-driven features such as recommendation systems

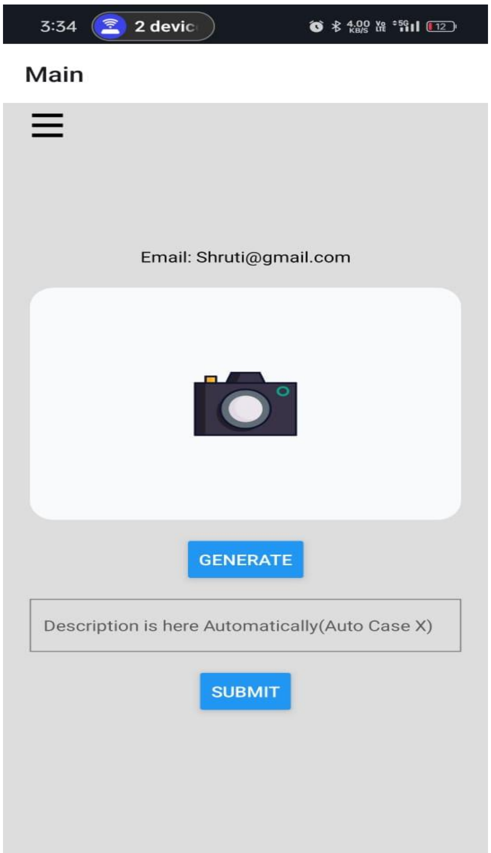


Email: Shruti@gmail.com

Photo:

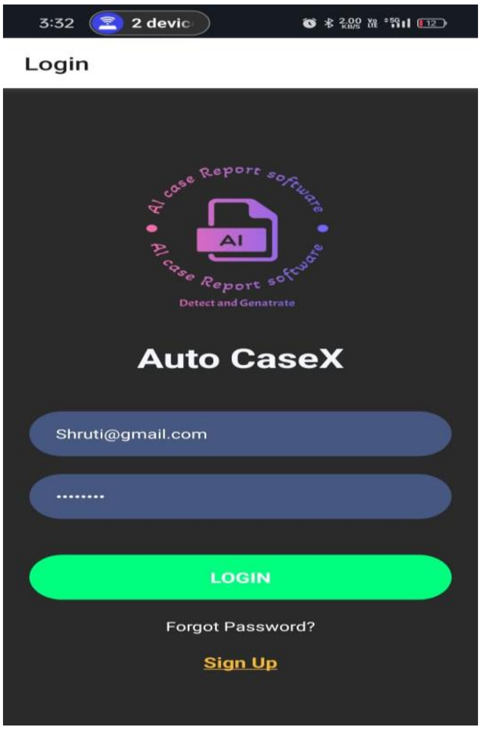
DateTime: 29/08/2024, 15:36:12
Address: Priyadarshini Bhagwati College of Engineering, Umred Road, Raghuji Nagar, Nagpur City, Nagpur Urban Taluka, Nagpur, Maharashtra, 440017, India

Description is here Automatically(Auto Case X)



Email: Shruti@gmail.com

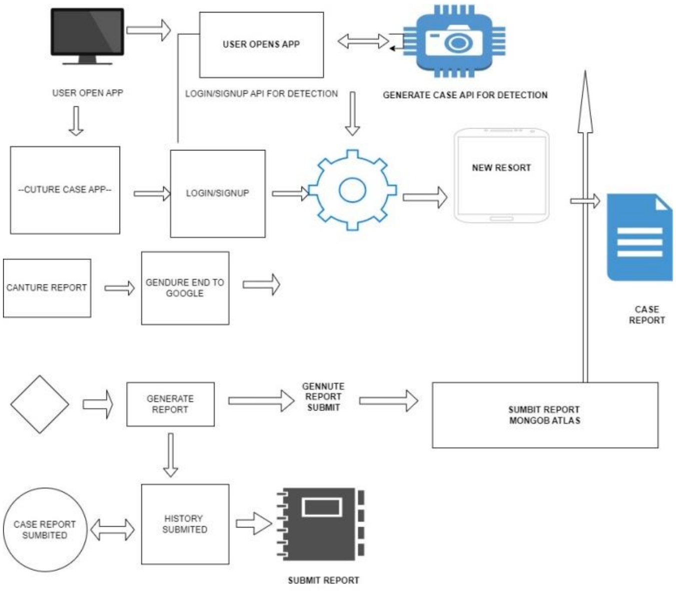
Description is here Automatically(Auto Case X)



Shruti@gmail.com

••••••••

Charts:



To use the Automatic Case Report Generator Using Image Detection, begin by downloading the application, whether it's for mobile or web, from the appropriate platform such as the Play Store or a specific web URL. Once installed, either register as a new user or log in with your existing credentials.

Upon accessing the application, you will be directed to the main interface. Here, you can start by using the built-in camera functionality, powered by React Native's Expo Camera API. Simply point the camera at the object or scene relevant to the case you are documenting, and capture the image. After the image is captured, it will automatically be processed using the Google Gemini API, which identifies key objects and details within the image.

Following image detection, the system will analyze the data and begin generating a detailed case report. This is done using the ChatGPT API**,** which transforms the information from the image into a well-structured, human-readable report. The generated report will include important details such as the description of the detected objects, relevant contextual data, and the overall summary of the case.

In addition, the system will automatically capture your geolocation using the OpenCage API, adding location-based context to the report. This helps provide a full picture of where and when the image was taken, ensuring that the case report is accurate and comprehensive.

Once the case report is generated, it will be formatted according to the predefined template and stored in the MongoDB Atlas database. The case can then be accessed in the future, reviewed, or shared as needed. The application also allows you to view past case reports through the user-friendly interface, making it easy to manage and track all generated reports.

3. CONCLUSIONS

The Automatic Case Report Generator Using Image Detection successfully automates the traditionally time-consuming task of generating case reports by leveraging advanced technologies such as image detection, AI-driven text generation, and cloud storage. By integrating the Google Gemini API for precise image recognition and the ChatGPT API for detailed, human-readable report creation, the system significantly reduces human effort while improving accuracy and efficiency.

This project demonstrates the potential for AI and automation to streamline workflows in sectors like law enforcement, healthcare, and administrative reporting, where case documentation is critical. The system proved to be accurate, user-friendly, and scalable, with positive feedback from users who appreciated the time-saving benefits and ease of use. Additionally, the admin panel offers robust case management capabilities, allowing for effective tracking and reviewing of reports.

In the future, this project could be expanded to support more complex image types, improve detection accuracy through advanced machine learning models, and extend the system's application across different industries. Overall, the Automatic Case Report Generator represents a valuable contribution to the automation of documentation processes, reducing manual errors and enhancing productivity.

4.ACKNOWLEDGEMENT

I would like to express my deepest gratitude to all those who supported and guided me throughout the development of the Automatic Case Report Generator by Using Image Detection. This project has been a valuable learning experience, and it wouldn't have been possible without the contributions of several individuals and institutions.

First and foremost, I would like to thank Priyadarshini Bhagwati College of Engineering, Nagpur, for providing me with the resources and platform to undertake this project as part of my final year.

I extend my sincere thanks to my project guide, Prof. Dinesh Jamthe , for their consistent guidance, insightful suggestions, and encouragement, which were invaluable in completing this project.

I am also grateful to my project team members for their collaborative efforts, dedication, and constant motivation that helped shape the project. Special thanks to Team Members for their unique contributions in each phase of the project.

I would also like to acknowledge the support from the MongoDB Atlas**,** Google Gemini API**,** OpenCage API**,** and Vercel platforms, whose tools and technologies played a crucial role in making the technical implementation successful.

7. REFERENCES

[1] Deep Learning for Image Recognition*Authors:* Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton *Published In:* Communications of the ACM

[2] Automated Report Generation Using Natural Language Processing**,** Xingxing Zhang, Mirella Lapata *Published In:* Association for Computational Linguistics (ACL)

[3] Real-Time Object Detection with Deep Learning*Authors:* Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi