**INTERACTIVE PRODUCT ENQUIRY CHATBOT WITH VIDEO CALL INTEGRATION**

**Rekha M1, Madhan A2, Mahesh M3, Muzammil Sk4**

*1Assistant professor Dept of Computer Science and Engineering, Lingayas Institute of Management & Technology, Madalavarigudem, Vijayawada, Andhra Pradesh, India*

*2,3,4BTech students, Dept of Computer Science and Engineering, Lingayas Institute of Management & Technology, Madalavarigudem, Vijayawada, Andhra Pradesh, India*

---------------------------------------------------------------------\*\*\*---------------------------------------------------------------------

**ABSTRACT -** In today's digital era, customer engagement and accessibility are pivotal in ensuring a seamless shopping experience. This project presents the development of an Interactive Product Enquiry Chatbot with Video Call Integration, aimed at enhancing user interaction and accessibility in product inquiries. The chatbot leverages modern technologies such as image-based product identification, object detection, and text-to-speech conversion to give users a dynamic and engaging experience. The key features of the chatbot include real-time camera preview and picture capture functionalities, allowing users to capture images of products they wish to inquire about. These images are then processed using TensorFlow.js and the COCO-SSD model for object detection, enabling the chatbot to identify and display the names of objects within the images. Additionally, the detected object names are converted into speech using TextToSpeech functionality, enhancing accessibility for users with visual impairments. The project is implemented using Ionic React for the frontend interface, integrating various Ionic components for user interaction. Capacitor plugins are utilized to access device features such as the camera and text-to-speech functionality, while TensorFlow.js handles the object detection tasks. The backend manages communication between the front end and external services, facilitating seamless integration and operation of the chatbot. Challenges encountered during the project include the integration of TensorFlow.js and the COCO-SSD model into the front end, ensuring compatibility across different devices and platforms, and optimizing performance and responsiveness. Future enhancements may include additional features such as natural language processing for more advanced interactions and integration with external databases for product information. Overall, the Interactive Product Enquiry Chatbot with Video Call Integration offers a novel solution for enhancing customer engagement and accessibility in product inquiries, paving the way for more interactive and personalized shopping experiences.

***Key Words*:** TensorFlow, COCO-SSD, Capacitor.

**1. INTRODUCTION**

In the rapidly evolving landscape of e-commerce and digital communication, businesses strive to enhance customer engagement and accessibility to ensure a seamless and personalized shopping experience. The advent of chatbots has revolutionized customer interaction, offering an efficient and interactive platform for addressing inquiries and providing assistance. However, traditional text-based chatbots may sometimes lack the visual context necessary for certain types of inquiries, particularly in product-related queries. To address this challenge, we present the development of an Interactive Product Enquiry Chatbot with Video Call Integration. This project aims to augment traditional chatbot functionalities with advanced features such as real-time image-based product identification, object detection, and text-to-speech conversion. By integrating these technologies, the chatbot provides users with a dynamic and engaging platform for querying products and receiving personalized assistance. The primary objective of the project is to enhance user interaction and accessibility in product inquiries by leveraging modern technologies and innovative approaches. The chatbot allows users to capture images of products using their device camera and utilizes TensorFlow.js and the COCO-SSD model for object detection. This enables the chatbot to identify objects within the images and provide users with immediate feedback on the detected objects. Furthermore, the project incorporates text-to-speech functionality to convert the detected object names into speech, thereby enhancing accessibility for users with visual impairments. By offering both visual and auditory feedback, the chatbot caters to a diverse range of users and ensures a seamless and inclusive user experience. The implementation of the project utilizes Ionic React for the frontend interface, integrating various Ionic components for user interaction. Capacitor plugins enable access to device features such as the camera and text-to-speech functionality, while TensorFlow.js handles the object detection tasks. The backend facilitates communication between the front end and external services, ensuring seamless integration and operation of the chatbot. Through this project, we aim to demonstrate the potential of integrating advanced technologies to create interactive and personalized shopping experiences. By providing users with a platform that combines visual and auditory feedback, the Interactive Product Enquiry Chatbot with Video Call Integration redefines customer engagement in the digital age, paving the way for more immersive and efficient interactions in e-commerce environments.

## 2. LITERATURE SURVEY

Karanbir Singh Chahal, Kuntal Dey. Object detection is the identification of an object in the image along with its localisation and classification. It has widespread applications and is a critical component for vision-based software systems. This paper seeks to perform a rigorous survey of modern object detection algorithms that use deep learning. As part of the survey, the topics explored include various algorithms, quality metrics, speed/size trade-offs and training methodologies. This paper focuses on the two types of object detection algorithms- the SSD class of single-step detectors and the Faster R-CNN class of two-step detectors. Techniques to construct detectors that are portable and fast on low-powered devices are also addressed by exploring new lightweight convolutional base architectures. Ultimately, a rigorous review of the strengths and weaknesses of each detector leads us to the present state of the art. AppsMax Lynch, Capacitor is a cross-platform native runtime that makes it easy to build performant mobile applications that run natively on iOS, Android, and more using modern web tooling. Representing the next evolution of Hybrid apps, Capacitor creates Web Native apps, providing a modern native container approach for teams who want to build web-first without sacrificing full access to native SDKs when they need it.

Capacitor provides a consistent, web-focused set of APIs that enable an app to stay as close to web standards as possible while accessing rich native device features on platforms that support them. If it works in the browser, it probably works in a mobile app when using Capacitor. Adding native functionality is straightforward with a Plugin API for Swift on iOS, Java on Android, and JavaScript for the web.

**3. PROBLEM STATEMENT**

The existing system for product inquiry and assistance lacks the capability to provide users with a rich and interactive experience, leading to limitations in user engagement, accessibility, and effectiveness. Users face challenges in accurately conveying their inquiries, especially when visual context is required, and the system's dependency on manual text-based input hinders usability and efficiency. Furthermore, scalability issues and dependencies on external services pose challenges in accommodating growing user demands and maintaining reliability. In light of these limitations, there is a need for a more advanced and user-centric solution that enhances user interaction, provides visual context, improves accessibility, and addresses scalability challenges. The proposed system aims to overcome these shortcomings by leveraging advanced technologies and interaction modalities to create an interactive product inquiry chatbot with video call integration, offering users a seamless and engaging experience while effectively addressing their queries and needs.

**4. OBJECTIVE & SCOPE OF THE PROJECT**

**OBJECTIVE:** The “Interactive Product Enquiry Chatbot With Video Call Integration” to develop an interactive chatbot system equipped with camera functionality to assist users in obtaining detailed product information. By combining text-based conversation with live video communication, the aim is to enhance user experience and satisfaction when querying product details.

**SCOPE:** The "Interactive Product Enquiry Chatbot With Video Call Integration" The completed project can be deployed on e-commerce platforms to provide users with an interactive and personalized shopping experience. Users can utilize the chatbot for product inquiries, assistance, and recommendations, thereby enhancing their overall shopping experience. also be integrated into the customer service portals of various businesses and organizations. Customer service representatives can use the chatbot's video call integration to engage with users in real-time, assist, and address queries more effectively.

**5. MATERIALS & METHODS**

|  |  |
| --- | --- |
| Operating System | Windows 7 or High |
| Main Technology | Machine Learning (TensorFlow) |
| Programming Language | HTML, CSS, JavaScript |
| Tools | Vscode |

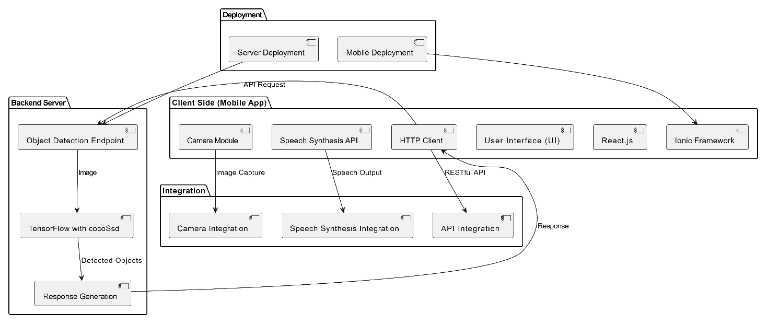
**Table -1:** Software requirements

|  |  |
| --- | --- |
| Processor | Intel Core Or Ryzen |
| Speed | 2.42 GHz |
| HDD | 120 GB |
| RAM | 4 GB (Minimum) |

**Table -2:** Hardware requirements

**5. SYSTEM DESIGN**

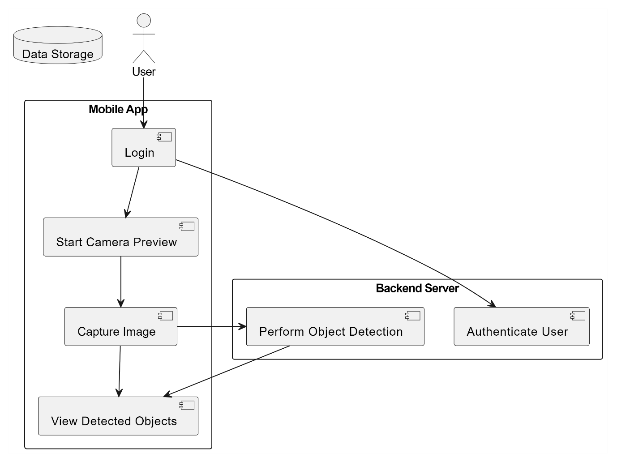
**5.1 SYSTEM ARCHITECTURE**



**Fig -1**: System architecture

**5.2 DATA FLOW DIAGRAM**

A data flow diagram (DFD) illustrates the flow of data within a system, showing how information moves between processes, data stores, and external entities. the DFD for the object detection app illustrates how data flows from the user through the mobile app to the backend server for object detection. it may include processes like user authentication, capturing images, performing object detection, and displaying results, along with data stores for storing images or detection results.



**Fig -2**: Level of flow diagram

**5. IMPLEMENTATION AND RESULTS**

Identify the specific requirements and objectives of the project, including the target platform (e.g., Android, iOS), desired object detection performance metrics, and integration constraints.

Research and Selection: Research different capacitor types, camera systems, and integration techniques to determine the most suitable components and approaches for the project.

Select appropriate hardware components, such as capacitors, camera modules, and development boards, based on performance, compatibility, and availability.

Hardware Setup: Set up the hardware environment for capacitor-camera integration, including connecting the capacitor to the power supply circuitry of the camera system.

Ensure proper wiring, voltage levels, and electrical connections to prevent damage to the components and ensure safe operation.

Software Development: Develop software components for controlling the camera module, capturing images, and performing object detection algorithms.

Integrate the capacitor control logic into the software to manage power supply and voltage stabilization during camera operation. Implement object detection algorithms using machine learning frameworks such as TensorFlow Lite, YOLO (You Only Look Once), or OpenCV (Open Source Computer Vision Library).

Integration and Testing: Integrate the hardware and software components into a cohesive system, ensuring compatibility and functionality. Conduct thorough testing to validate the performance, accuracy, and reliability of the object detection system with capacitor-camera integration.

Test the system under various conditions, including different lighting conditions, object sizes, and camera angles, to assess its robustness and effectiveness.

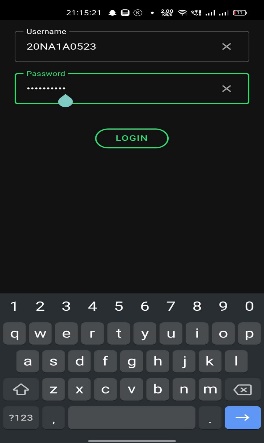


Fig -3 Login

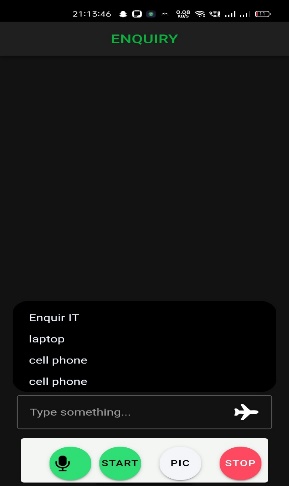


Fig -4 Object Detection

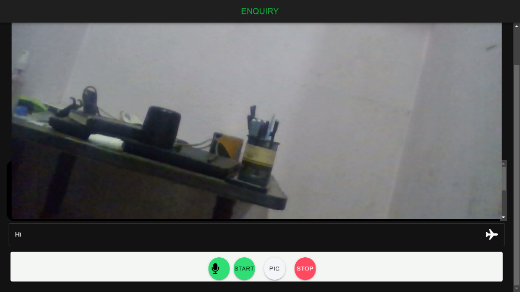


Fig – Page overview

**6. CONCLUSION**

Android Ionic app for object detection using TensorFlow's cocoSsd model represents a significant achievement in leveraging cutting-edge technology to address real-world challenges. Throughout the development process, careful consideration was given to the design, functionality, and usability of the application to ensure its effectiveness and user satisfaction.By implementing features such as user authentication, camera preview, image capture for object detection, and display of detected objects in an HTML div, the app provides users with a seamless and intuitive experience. The integration of speech capabilities further enhances accessibility and user engagement by providing spoken feedback at each step of the process.

**6. CODING**

The total coding is in the below hyperlink, kindly check it.

[Coding](Coding.pdf)

**7. REFERENCES**

[1] TensorFlow. (n.d.). TensorFlow Lite (v2.16.1)2024. Retrieved from <https://www.tensorflow.org/lite>.

[2] Ionic Framework. (n.d.) (v7.8.6)2024. Ionic Documentation. Retrieved from <https://ionicframework.com/docs>.

[3] React. (n.d.) (v8.0.1)2024. React Documentation. Retrieved from <https://reactjs.org/docs/getting-started.html>.

[4] OpenAI. (n.d.). GPT-3.5(v3.5) 2022. Retrieved from <https://openai.com/>.

[5] PlantUML. (n.d.) (v2.0). PlantUML Documentation. Retrieved from <https://plantuml.com/>.

[6] Capacitor. (n.d.). Capacitor Documentation. Retrieved from <https://capacitorjs.com/docs>.

[7] IEEE. (n.d.). IEEE Xplore Digital Library. Retrieved from <https://ieeexplore.ieee.org/Xplore/home.jsp>.

[8] Google Chrome. (n.d.). Google Chrome Help Center. Retrieved from <https://support.google.com/chrome>