

**SPECIAL LAB INITIATIVES
REWARD POINTS**

FULL STACK & DEVOPS

**Name : Sabarish R
Roll : 7376222IT237**

**Name : NaveenKumar P
Roll : 7376222IT210**

**Name : FIDAL CASTRO SARAVANAN S
Roll : 7376222CB113**

Project Documentation for Chatbot with YOLOv5, BLIP, and Usage Analytics Dashboard

1. Introduction

This project involves the development of a chatbot that leverages advanced models such as YOLOv5 for object detection and BLIP for image captioning. The chatbot allows users to upload images, upon which it detects objects, identifies colors, generates captions, and provides responses. Additionally, a real-time usage analytics dashboard tracks interactions, object detections, and image uploads to provide key insights into the chatbot's performance.

2. System Architecture

The system consists of the following key components:

- ☐ **Frontend:** A chatbot UI where users can upload images or type queries. The interface includes features like voice input/output, quick replies, and buttons for enhanced user interaction.
- ☐ **Backend (Flask):** The server processes incoming requests, handles image uploads, performs object detection using YOLOv5, generates image captions with BLIP, and returns responses to the user. MongoDB is used to store interactions and analytics.
- ☐ **Database (MongoDB):** MongoDB stores both image and text data, including the captions, detected objects, and user queries.

3. Key Features

- 1. Image Object Detection:** Detects objects within uploaded images using YOLOv5.
- 2. Color Detection:** Identifies the dominant color of objects detected in images.
- 3. Image Captioning:** Uses BLIP to generate captions based on image content.
- 4. User Interaction Tracking:** Records user interactions and queries for later analysis.
- 5. Real-Time Analytics:** Provides visual insights into user activity, object detection statistics, and system performance.

4. Project Components

a. YOLOv5 Object Detection

YOLOv5 is used for detecting objects in uploaded images. The detected objects are returned to the user, and the color of each object is also identified. The detection results are saved in MongoDB for future queries.

b. BLIP Image Captioning

BLIP (Bootstrapping Language-Image Pretraining) generates a descriptive caption for uploaded images. This helps users understand the content of the image beyond object detection. Captions are also stored in the MongoDB database.

c. Flask Backend

The backend is built using Flask. It handles requests from the chatbot UI, processes images using YOLOv5 and BLIP, and returns the results to the user. Flask also manages data storage in MongoDB.

d. MongoDB Integration

MongoDB is used to store image data, detected objects, and user interactions. The system logs every image upload, caption generation, and query, enabling detailed usage analysis through the analytics dashboard.

5. Usage Analytics Dashboard

The Usage Analytics Dashboard tracks and displays key metrics:
Most Commonly Detected Objects: Provides insight into the most frequently detected objects in uploaded images.

User Interaction Logs: Tracks user queries, responses, and image uploads.
Image Upload History: Lists all uploaded images along with detected objects and captions. Data is displayed visually using charts and graphs to offer a comprehensive overview of system usage.

6. Real-Time Analytics

Real-time analytics enable monitoring of the chatbot's performance:
Response Time: Tracks how quickly the system responds to user queries.

Interaction Volume: Displays the number of user interactions over time.

Model Performance: Measures the accuracy of YOLOv5 and BLIP in handling image data and generating accurate captions. These analytics are updated in real-time to help monitor and optimize system performance.

7. Configuration

Pre-requisites:

- Python 3.7 or above
- MongoDB installed locally or on a cloud service
- Install necessary Python packages using the `requirements.txt` file
 - ❖ . Set up MongoDB locally or configure cloud-based MongoDB.
 - ❖ . Download YOLOv5 weights and place them in the `models` folder.

8. API Endpoints

a. `/chat` [POST]

- Handles user queries and image uploads.
- Accepts both text and image input.

b. `/analytics` [GET]

- Retrieves data for the Usage Analytics Dashboard.
- Provides insights into system performance and user interactions.

9. How to Use

1. User Interface: The chatbot can be interacted with via text or image uploads. Users can ask about the objects and colors detected in the uploaded images.

2. Admin Analytics: Admins can access the analytics dashboard to review system usage and performance metrics.

10. Conclusion

This chatbot integrates state-of-the-art object detection (**YOLOv5**) and image captioning (**BLIP**) models to provide an interactive experience. It offers not only real-time interaction capabilities but also a robust analytics dashboard for tracking and optimizing performance.