✍■Approach Documentation – FlytBase AI Engineer Assignment

■Problem Understanding

Thetask was to build a Drone Security Analyst Agent that can: - Monitor a video feed from a

fixed-location drone - Analyze visual content to identify people, vehicles, and potential threats - Detect and log security alerts such as loitering, assault, or crash - Allow summarization of the entire activity - Support natural language querying to search logs quickly

■System Architecture

```
Video
      → Frame Extraction → BLIP (Captioning) → Parse → Store in SQLite
                                                  ■
                                                Alert Detection (keywords)
Logs → BART (Summarization)
     → Gemini 1.5 (Natural language to SQL)
```

■Tool Selection & Justification

BLIP – Betterdescriptivepowerthan CLIPBART– Excellent for summarization tasks Gemini 1.5

Flash – Reliable for natural language to SQL conversion OpenCV – Lightweight and battle-tested for video processing SQLite – Local and fast database for event logging python-dotenv – For secure API key handling

■Development Approach

1. Video Processing (main.py):

- Used OpenCV to extract one frame every 2 seconds - Passed each frame through BLIP to get textual captions - Parsed captions for keywords and stored with timestamp - Logged captions and alerts in SQLite

2. Data Interaction (app.py):

- Loaded logs from database - Generated summary using BART - Took user queries and passed to Gemini - Executed SQL on DB and returned results

■Features Implemented

-Frame-by-frame loggingwithcaptions - Alert detection using keyword logic - Timestamped storage in SQLite - BART-powered summarization - Keyword and natural language search using Gemini

■Sample Use Cases

• Where was the car seen? → ■ Answer: parking • What frames had loitering? → ■ Answer: frame_0012.jpg, frame_0015.jpg • Summarize the video → A motorcycle drove through a tunnel and a man was seen at the gate.

■ Challenges Faced

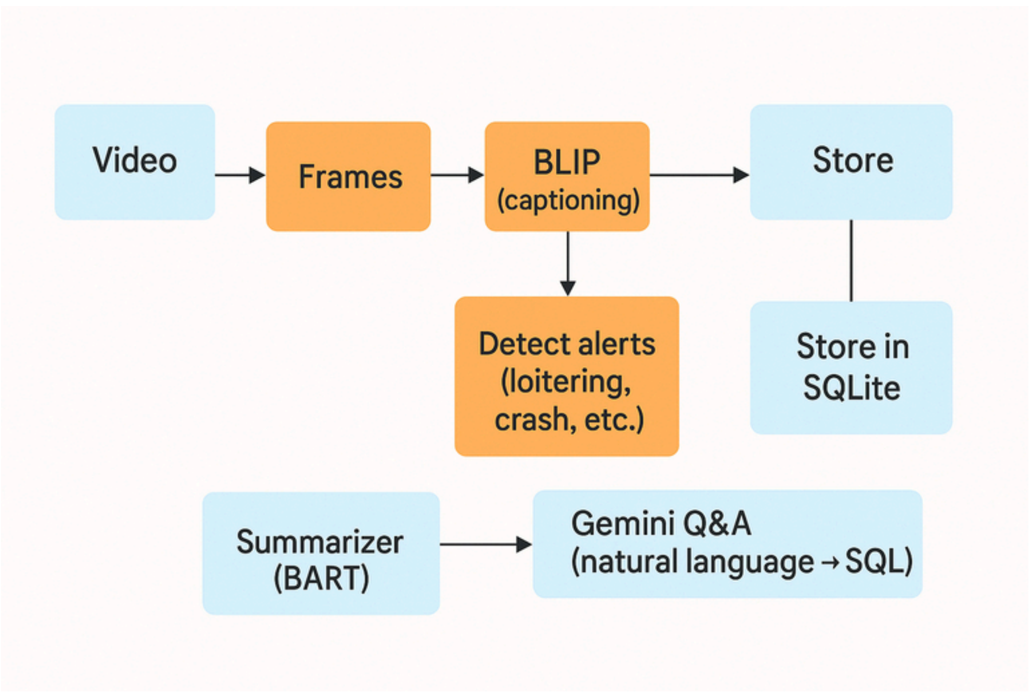
- Balancing frame rate with processing time - Prompt tuning for Gemini to generate valid SQL - Building reliable alert logic from textual captions

■ Improvements & Next Steps

- Add GUI for log and video interaction - Add GPS-tagged telemetry input - Use action detection models for smarter alerts - Integrate scene summarization & open-source LLM for Gemini replacement

■ AI Tools That Assisted

- Claude 3, Gemini – Prompt design and code scaffolding - Cursor AI, Replit – Debugging SQL queries - Hugging Face – Testing BLIP and BART before deployment


Video
Frames
BLIP (captioning)
Store
Detect alerts (loitering, crash, etc.)
Store in SQLite
Summarizer (BART)
Gemini Q&A (natural language → SQL)