Motion-Activated Camera System

Applications of ICT (CS-117)

Department of Aerospace Engineering, SMME, NUST

Group Members:
Danyal Hassan
Eisa Hassan
Abdul Moeed

The Challenge & The Solution

The Problem

Continuous video recording generates massive amounts of redundant data. This leads to inefficient storage usage, increased power consumption, and significant difficulty in reviewing footage to find critical events.

Our Solution

An intelligent **Motion-Activated Recording System** using Python and OpenCV. It continuously monitors a live feed but only records when meaningful activity is detected, preserving context through smart buffering.

System Architecture

1

Initialization

Camera setup & Reference frame capture

2

Idle & Buffer

Continuous monitoring with rolling deque buffer

3

Detection

MOG2 Algorithm identifies motion

4

Recording

Pre-event + Active Event + Post-event saved

Core Logic: Background Subtraction

We implemented the **MOG2** algorithm for robust detection.

- ⚡ **Why MOG2?** It is significantly faster and more memory-efficient than KNN (K-Nearest Neighbors).
- ☁️ **Shadow Detection:** Unlike simple subtraction, MOG2 can detect shadows, preventing false positives from lighting changes.
- 🚫 **Alternatives:** GMG (too heavy), CNT (less robust).

Optimized Buffering with Deque

The Library

We utilized Python's collections library to implement a **deque** (Double-Ended Queue) for our frame buffer.

Efficiency

Deque provides time complexity for appends and pops from both ends, making it far superior to standard lists ( ).

Context Preservation

This allows us to maintain a "rolling window" of the last few seconds of video, ensuring we capture the moment *before* motion starts.

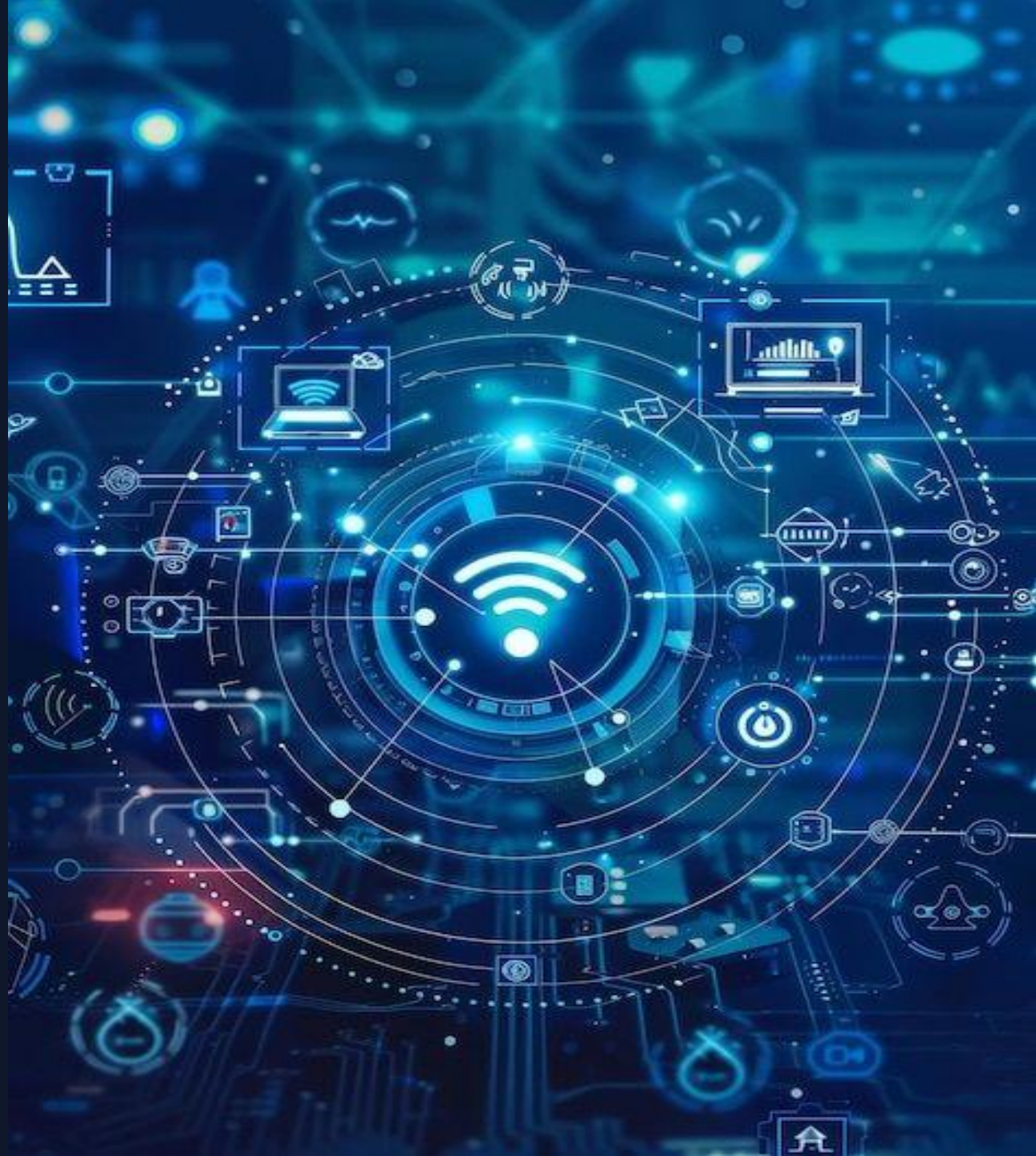
Implementation

The system is built on **Python** using the **OpenCV** library for image processing.

Key Components:

- Frame subtraction loop
- Contour detection (filtering small movements)
- Uses Deque data structure for buffer

The code handles lighting anomalies via threshold tuning and manages storage automatically.



Sustainability & Impact

Reduced Storage

Saves only meaningful events, drastically reducing hard drive usage.

Energy Efficient

Less data writing means lower power consumption for storage servers.

Cost Effective

Built on open-source software, making surveillance accessible.

Conclusion

This project successfully demonstrates how ICT tools can solve real-world engineering problems. By integrating **MOG2 algorithms** with **deque-based buffering**, we achieved a surveillance system that is efficient, reliable, and sustainable.

Questions?

Thank you for your attention.