

# Crime Investigation using Camera-Mounted Rover and VR with 5G

## 5G Innovation Hackathon 2025

Vivek Mukundan

208124033, M. Tech. Communication Systems

Tanmay Joyashi

208124015, M. Tech. Communication Systems



**Mentored by**

Dr. E.S. Gopi

Professor, Department of ECE, NITT

- **Shortage of forensic experts:** In many regions there are too few trained forensic investigators to attend every crime scene quickly. This scarcity creates delays in initial evidence collection and slows case progress.
- **Risk of contamination:** Multiple people entering a crime scene increases the risk of disturbing or contaminating physical evidence. Even well-intentioned actions can alter trace evidence or the original layout.
- **Need for remote, immersive, real-time investigation tools:** By combining a camera-mounted rover with a VR interface, experts can virtually inspect scenes in first-person, capture detailed imagery, and provide guidance to on-site personnel without physically entering the scene.

# Problem Statement: Specific Objectives

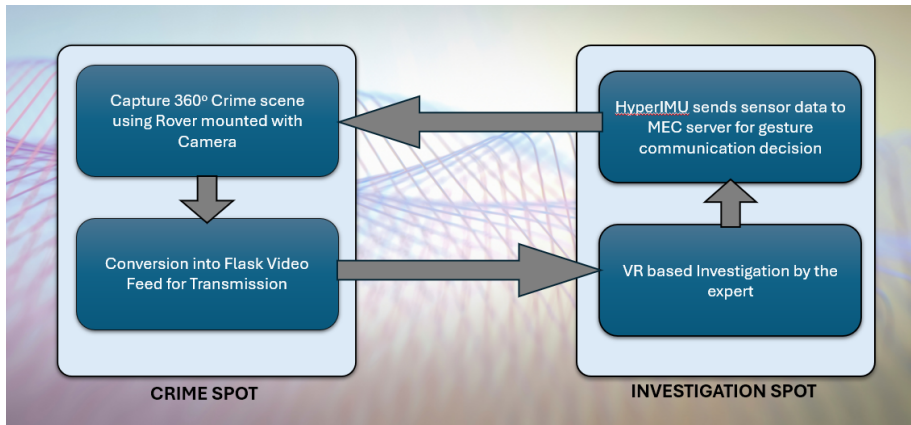
This project aims to build a preliminary prototype with the following objectives:

- ➊ To design and assemble a prototype system that demonstrates robust, bidirectional communication over a **private 5G network**.
- ➋ To validate the high-throughput data uplink by transmitting sensor data, simulating a live video feed.
- ➌ To validate the low-latency command downlink by transmitting control signals to actuate a simulated rover response.
- ➍ To establish this validated 5G communication link as the foundation for a future fully-featured remote investigation system using a VR headset and a camera-equipped rover.

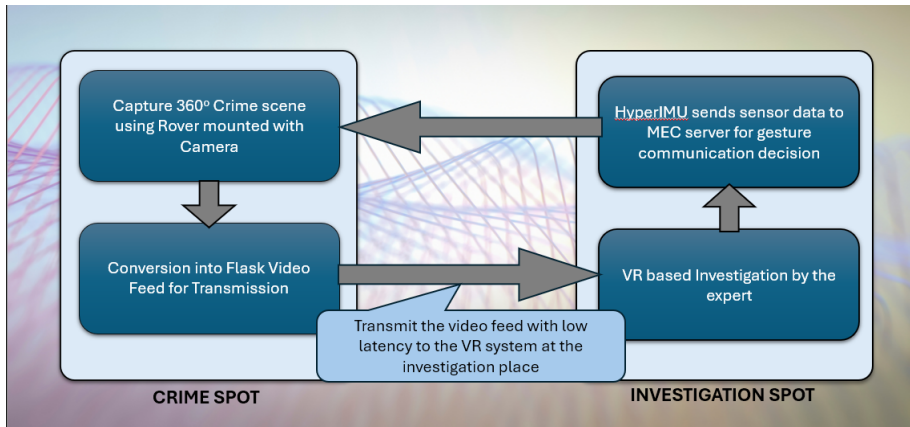
# Proposed Solution

- **Rover with Mounted Camera:** Controlled via VR headset for immersive, intuitive navigation.
- **Gesture-based Control (HyperIMU):** Smartphone sensors (gyro, accelerometer) map head movements to rover and camera actions.
- **Low-latency 5G MEC Communication:** Ensures smooth video streaming and instant response to commands.
- **Remote Expert Investigation:** Experts examine the crime scene virtually, reducing delays and contamination risk.

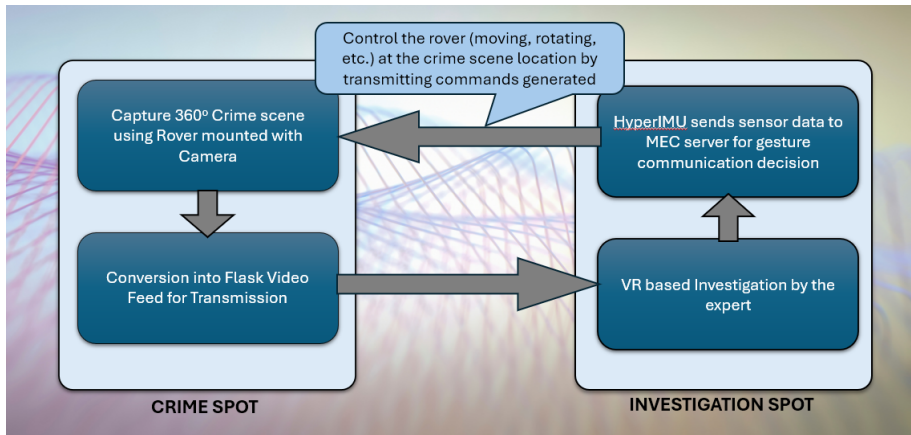
# Proposed Solution



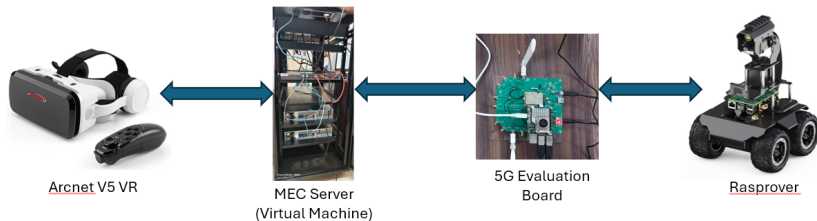
# Proposed Solution



# Proposed Solution

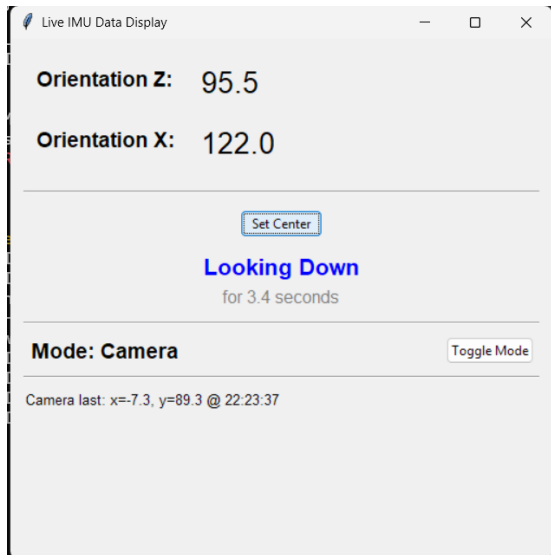
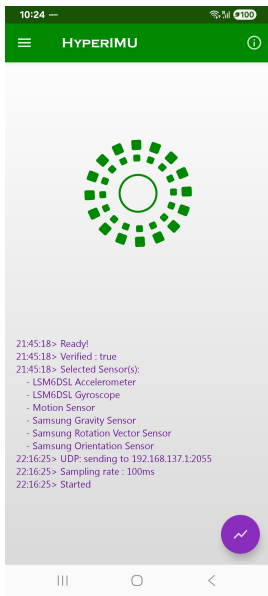


# System Flow Diagram





# Gesture Communication and GUI



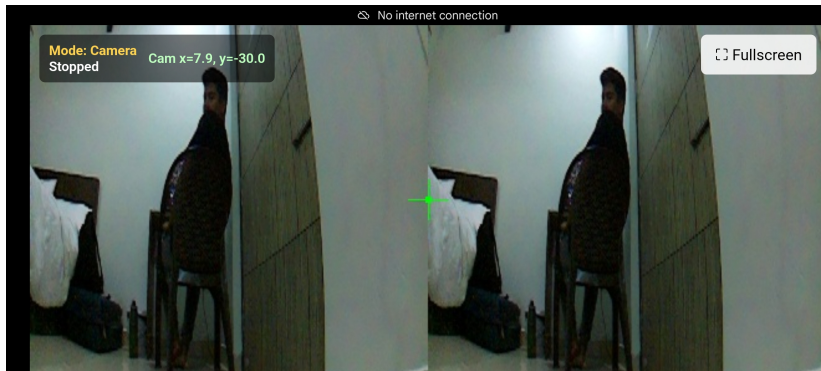
# Rover Homepage and Server Script

The screenshot displays a JupyterLab environment. The top browser window shows the URL `192.168.50.5:8888/lab/tree/tutorial_en/rover_server1.py`. The left sidebar shows a file explorer with a list of files and folders, including `001 Manual_Control.ipynb`, `01 Introduction to Robotics Basics.ipynb`, `02 Python Chassis Motion Control.ipynb`, `03 Pan-tilt Control And LED Control.ipynb`, `04 OLED Screen Control.ipynb`, `05 Building UI Interfaces in JupyterLab.ipynb`, `06 Retrieving Chassis Feedback Information.ipynb`, `07 Controlling Subordinate Devices with JSON Commands.ipynb`, `08 Microcontroller JSON Command Set.ipynb`, `09 Automatic Command Execution on Boot.ipynb`, `10 Play Audio Files.ipynb`, `11 Text-to-Speech (TTS).ipynb`, `12 Image Transmission Based on Flask.ipynb`, `13 Displaying Real-Time Video Stream in Jupyter Lab.ipynb`, `14 Time-lapse Photography.ipynb`, `15 OpenCV Motion Detection.ipynb`, `16 Controlling Photo Capture with Buttons.ipynb`, `17 Face Recognition Based on OpenCV.ipynb`, `18 Object Recognition Based on DNN (Deep Neural Network).ipynb`, and `19 OpenCV Color Recognition.ipynb`. The main area shows a code editor with the following Python code:

```
1 import socket, threading, json
2 from flask import Flask, Response
3 from picamera2 import Picamera2
4 import cv2
5 from base_ctl import BaseController
6
7 PORT_CHD = 8889
8
9 def is_raspberry_pi5():
10     try:
11         with open('/proc/cpuinfo', 'r') as file:
12             for line in file:
13                 if 'Model' in line:
14                     return 'Raspberry Pi 5' in line
15     except:
16         return False
17     return False
18
19 base = BaseController('/dev/ttyAM0', 115200) if is_raspberry_pi5() else
20 BaseController('/dev/serial0', 115200)
21
22 def translate_to_base_command(msg):
23     if msg.get("type") == "camera_reset":
24         try:
25             base.gimbal_ctl(0,0,0,0)
26             print("Camera reset to (0,0,0)")
27         except Exception as e:
28             print("Camera reset failed:", e)
29         return None
30
31 status, mode = msg.get("status"), msg.get("mode", "Rover")
32 if mode == "Rover":
33     if status == "looking lin": return ("T=1 P=1 A=0 S=0")
```

The bottom status bar shows the file is `rover_server1.py`, line 18, column 1, with 4 spaces.

# Video Feeds



[https://drive.google.com/file/d/  
1cfebv1ZIK08DT3IVsfPlri3VJTodSJRY/view?usp=sharing](https://drive.google.com/file/d/1cfebv1ZIK08DT3IVsfPlri3VJTodSJRY/view?usp=sharing)

- Forensic crime scene investigation.
- Search and rescue operations in hazardous areas.
- Defense and border security surveillance.
- Remote inspection of inaccessible environments.

# Future Work and Conclusion

- Enhance autonomy with AI-based evidence detection.
- Improve video quality with higher-bandwidth 5G slices.
- Expand multi-user VR collaboration.
- Conclusion: A scalable, secure, and immersive tool for forensic experts.

# Thank You!