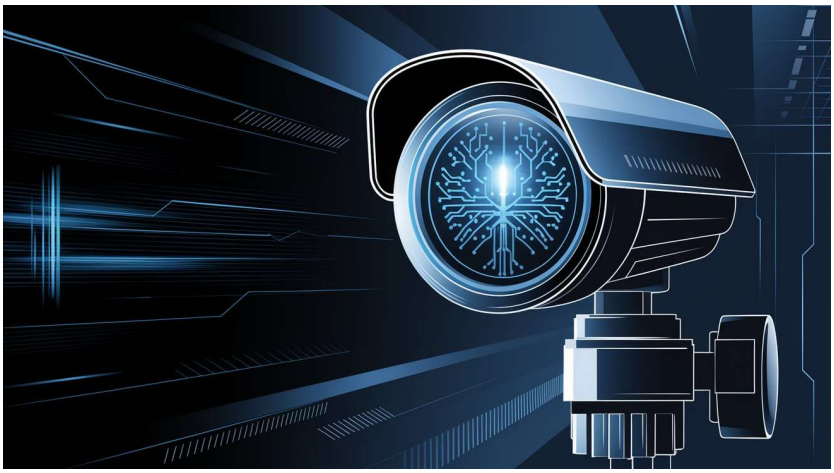
HONEYWELL HACKATHON

TOPIC: **AI-POWERED SURVEILLANCE**



SUBMITTED BY: RIYA

COLLEGE: UPES, Dehradun

COURSE: BTECH CSE AIML (4th Year)

SAP ID: 500101787

Introduction

Public safety in crowded places such as airports, railway stations, shopping malls, and public transport hubs is of utmost importance. One of the major security threats arises when suspicious or abandoned objects, such as bags, are left unattended for a certain duration. Detecting such objects can prevent incidents and allow authorities to take prompt action.

This project aims to build a computer vision-based system that detects abandoned bags in a video stream. The system uses background subtraction, contour detection, and motion analysis to identify static objects that remain in the scene beyond a defined time threshold.

Objectives

- To develop a surveillance-based system capable of detecting abandoned bags.
- To ensure detection by comparing a reference (first) frame with live video frames.
- To raise an alert in the terminal if an object remains stationary for more than 5 seconds.
- To export annotated video with bounding boxes highlighting the detected object.

Tools & Technologies

- **Programming Language:** Python 3.12
- **Libraries Used:**
    - OpenCV → Image processing and object detection
    - NumPy → Matrix and array operations
    - Time → For monitoring object duration
- **IDE / Environment:** VS Code / Command Prompt
- **Dataset:** Recorded surveillance video (video1.avi)

Methodology

Step 1: Input Video

- The system takes a surveillance video (video1.avi) as input.
- First frame is captured and saved as the reference frame.

Step 2: Preprocessing

- Convert frames to grayscale.
- Apply Gaussian blur to reduce noise.
- Perform Canny edge detection to highlight moving objects.

Step 3: Background Subtraction

- Compute absolute frame difference between the reference frame and the current frame.
- Apply thresholding and dilation to detect foreground objects.

Step 4: Contour Detection

- Identify contours of objects using cv2.findContours().
- Ignore small areas (< 500 pixels) to reduce noise.
- Draw bounding boxes around significant objects.

Step 5: Abandonment Detection

- Track each detected object's position.
- If an object remains at the same position for more than 5 seconds, trigger an alert in the terminal.

Step 6: Export Annotated Video

- Annotated video with bounding boxes is exported
- This video can be reviewed later for evidence or further analysis.

Results

- Successfully detected abandoned bag when left in the same position for >5 seconds.
- Alerts were displayed in the terminal.
- Output video was saved with bounding boxes around detected objects.
- The system works in real-time for small videos and can be extended to CCTV feeds.

Applications

- Airports and railway stations for unattended baggage detection.
- Shopping malls and public places for security surveillance.

Conclusion

This project demonstrates the use of computer vision and OpenCV for security surveillance. The system is capable of detecting abandoned bags and provides an alert mechanism to notify security threats.

Code Snippet

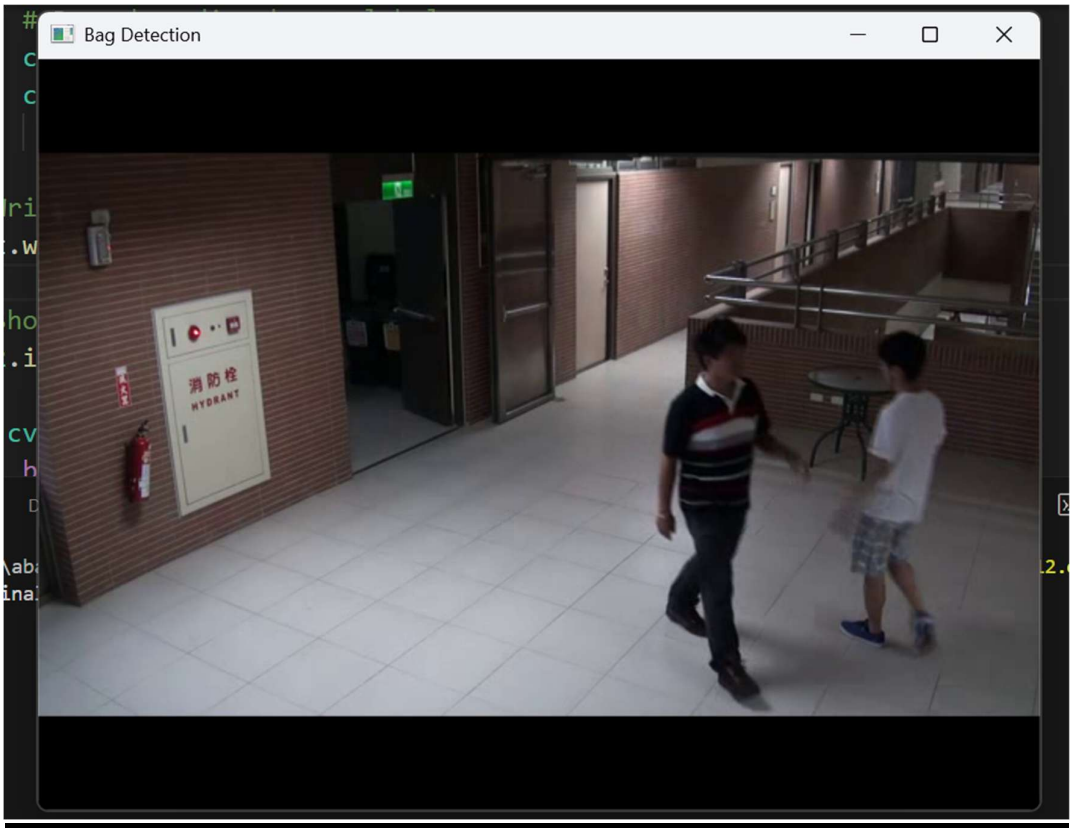
```python
import cv2
import time
import numpy as np
from ultralytics import YOLO

# Load YOLO model
model = YOLO("yolov8n.pt")

# Classes of interest
bag_classes = ["backpack", "handbag", "suitcase"]

# Open video
file_path = "video1.avi"
cap = cv2.VideoCapture(file_path)

# Video writer (original FPS)
fourcc = cv2.VideoWriter_fourcc(*'XVID')
fps = int(cap.get(cv2.CAP_PROP_FPS)) or 20
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
out = cv2.VideoWriter("output_original.avi", fourcc, fps, (width, height))  # same FPS
```

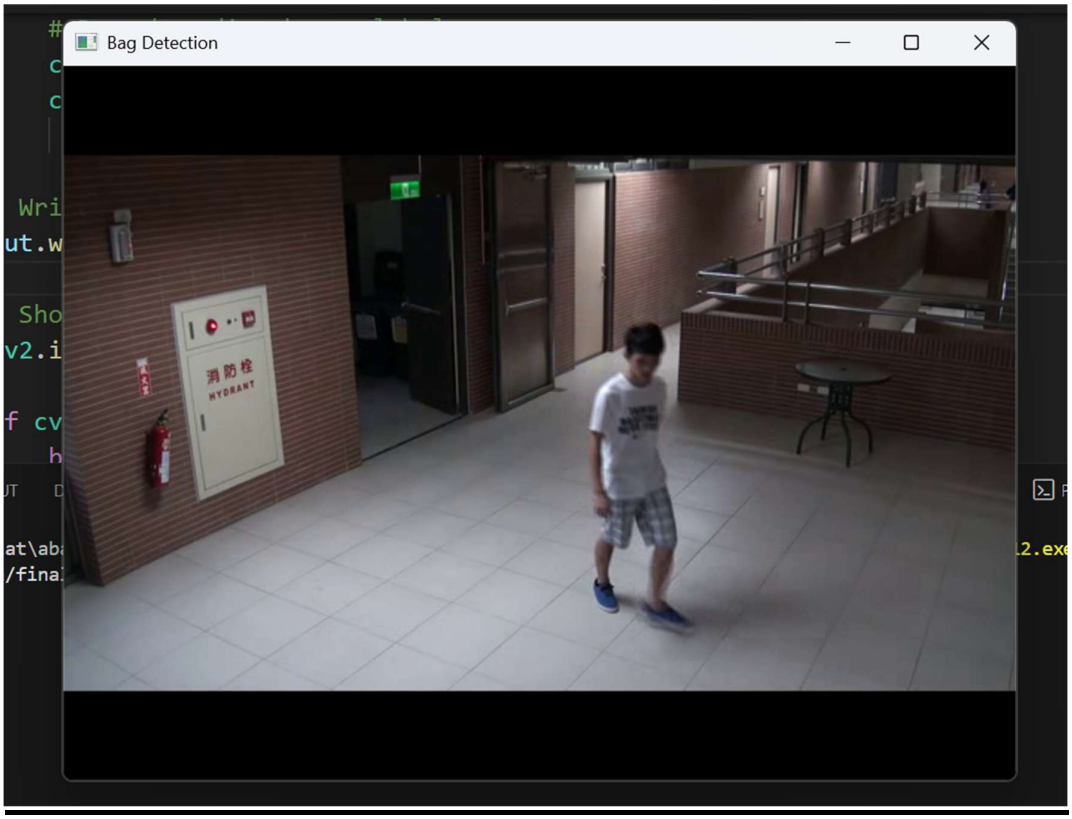
```python

# Track static bags {id: (center, start_time, alerted)}
object_tracker = {}
object_id = 0
alert_time = 5  # seconds

while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break

    results = model(frame, verbose=False)
    detections = results[0].boxes.data.cpu().numpy()

    for *xyxy, conf, cls in detections:
        class_name = model.names[int(cls)]
        if class_name not in bag_classes:
            continue

        x1, y1, x2, y2 = map(int, xyxy)
        center = ((x1 + x2) // 2, (y1 + y2) // 2)
```

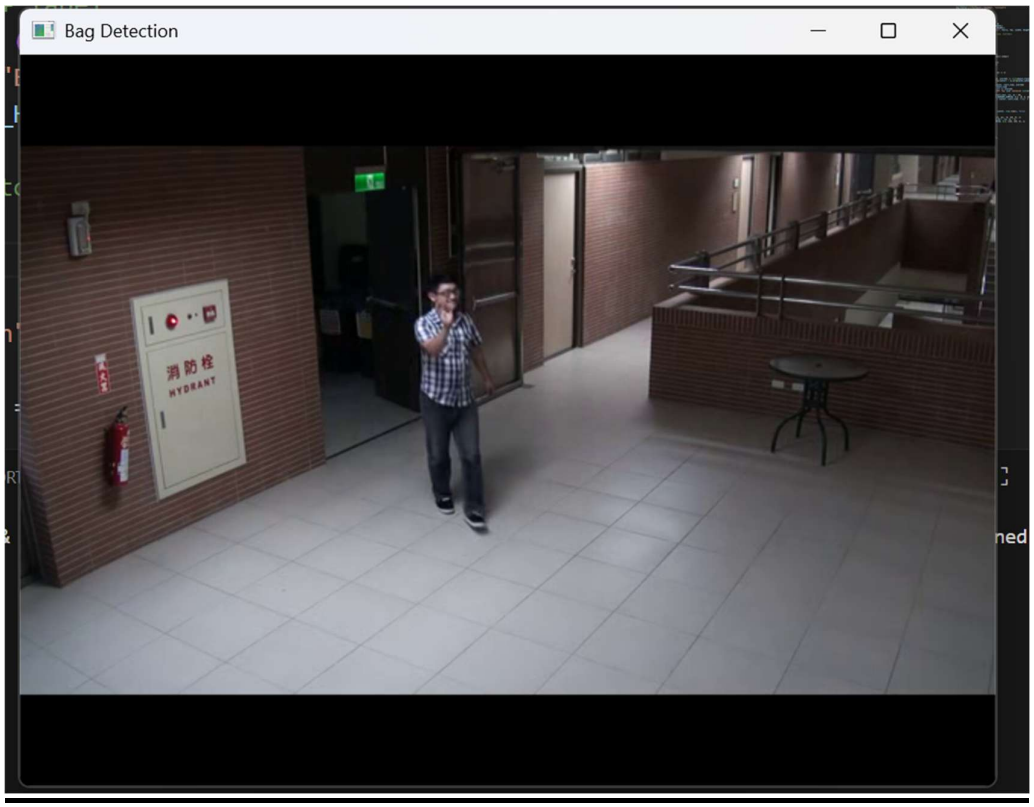
```
        found = False
        for oid, (prev_center, start_time, alerted) in list(object_tracker.items()):
            dist = np.linalg.norm(np.array(center) - np.array(prev_center))
            if dist < 40:  # same object
                object_tracker[oid] = (center, start_time, alerted)
                elapsed = time.time() - start_time
                if elapsed > alert_time and not alerted:
                    alert_text = f"⚠️ ALERT: Bag {oid} abandoned {int(elapsed)}s"
                    print(alert_text)
                    cv2.putText(frame, alert_text, (x1, y1 - 10),
                                cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255), 2)
                    object_tracker[oid] = (center, start_time, True)  # mark alerted
                found = True
                break

        if not found:
            object_id += 1
            object_tracker[object_id] = (center, time.time(), False)
```

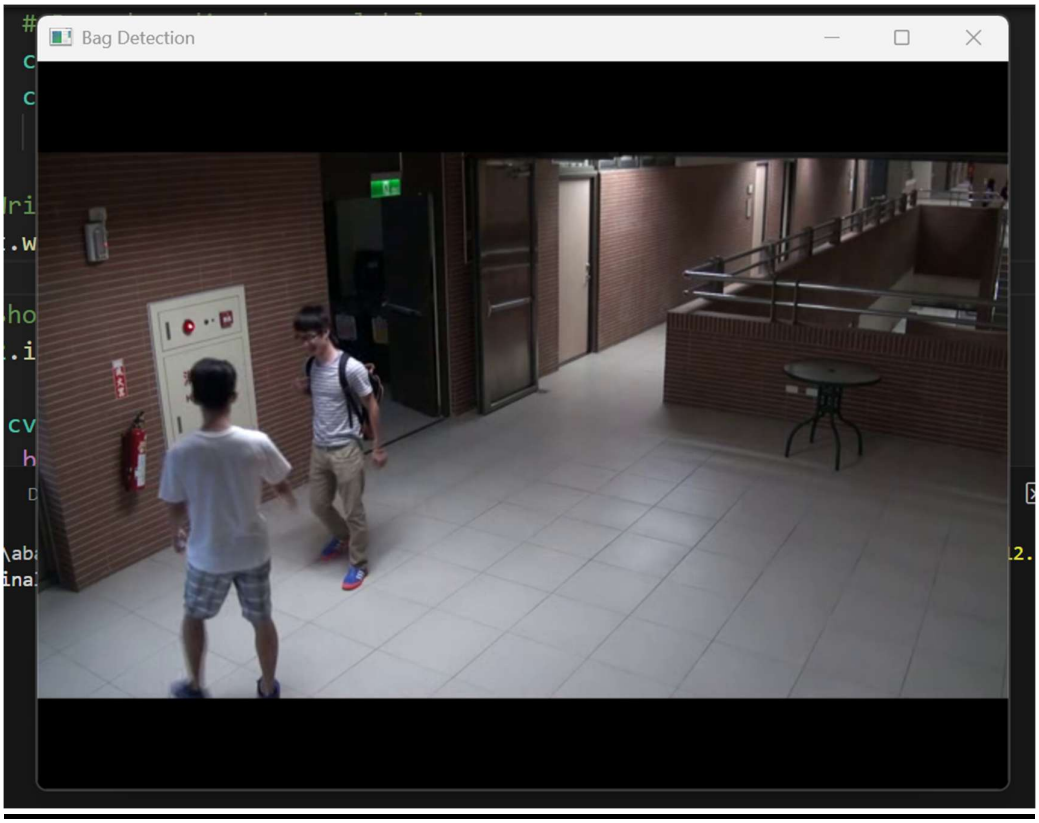
```
            # Draw bounding box + label
            cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
            cv2.putText(frame, f"Bag {object_id}", (x1, y1 - 5),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 0), 2)

    # Write annotated frame to video
    out.write(frame)

    # Show live
    cv2.imshow("Bag Detection", frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
out.release()
cv2.destroyAllWindows()
```

CCTV Video

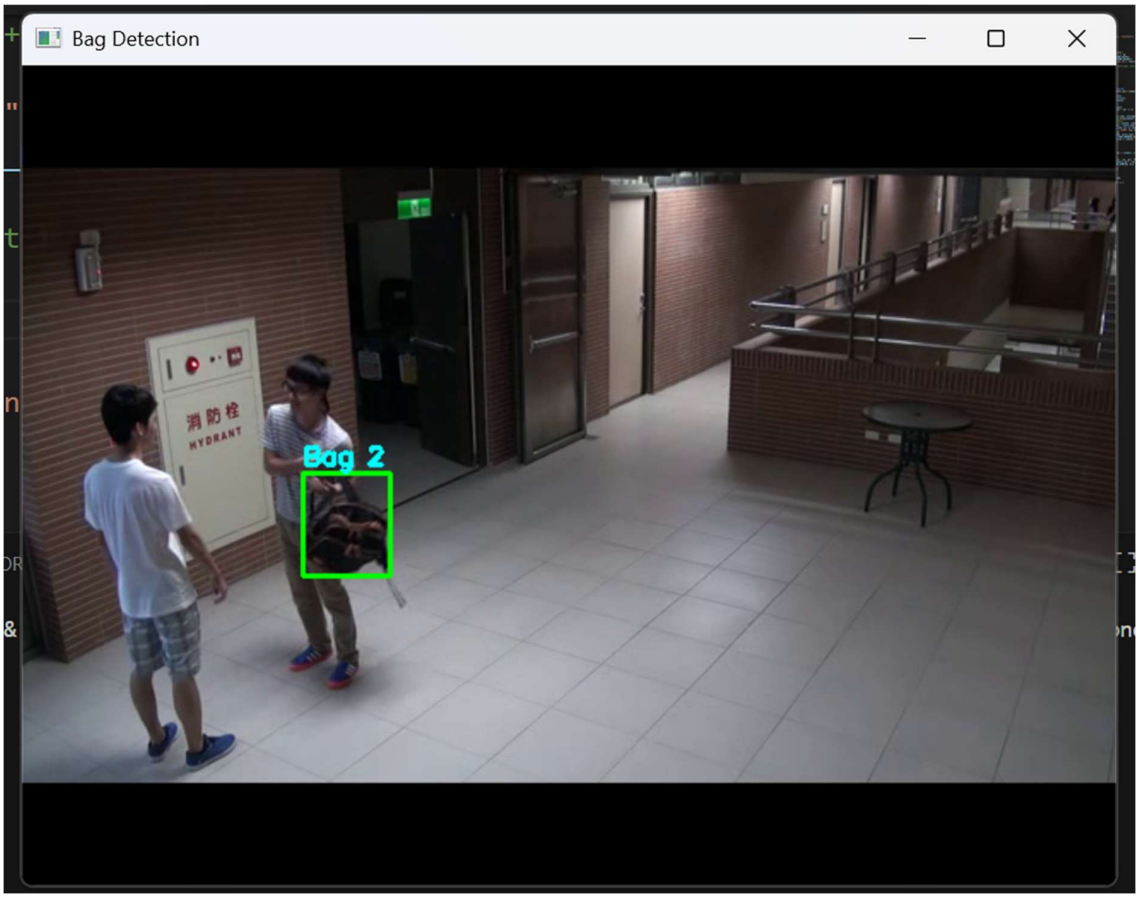





Bag Detection
消防栓
HYDRANT


Bag Detection

OUTPUT:



ALERT SENT: