Occupant Monitoring System (OMS)

1. Overview

This POC demonstrates a simplified **Occupant Monitoring System (OMS)** for an automotive environment. The goal is to classify the seat state as:

- **ADULT** → Airbag Activated
- **CHILD / OBJECT** → Airbag Deactivated
- **EMPTY** → Airbag Deactivated

The POC is implemented on **PC (Linux)** using **C++ and OpenCV**, with offline image inputs. It simulates how a real in-vehicle camera-based OMS would behave and is designed to be portable later to the **OKT507 embedded platform**.

---

2. Objectives

- Build an end-to-end OMS pipeline using classical computer vision.
- Classify occupant type using rule-based logic.
- Validate behavior with multiple test images.
- Provide deterministic, explainable results (no black-box ML).
- Prepare architecture for later migration to embedded hardware.

---

3. Input Structure

The system processes images from the following folder structure:

```
oms_input/
├── adult/
│   └── adult.jpg
├── child/
│   └── child.jpg
├── object/
│   └── object.jpg
└── empty/
    └── empty.jpg
```

Each image represents a seat state:

- Adult sitting
- Child sitting
- Object on seat
- Empty seat

---

4. Technical Approach

1. **Predefined Seat Region of Interest (ROI)**
   A fixed rectangle representing the seat area is defined in the image.
2. **Background Modeling**
   An empty-seat image is used as the reference background.
3. **Foreground Extraction**
   The current frame is compared against the background using:
   - Grayscale conversion
   - Absolute difference
   - Thresholding
   - Morphological cleanup
4. **Blob Analysis within ROI**
   Only pixels inside the seat ROI are considered.
5. **Geometric Feature Extraction**
   - Foreground area ratio
   - Bounding box height ratio
   - Aspect ratio of detected blob
6. **Rule-Based Classification**

Example decision logic:

- If foreground ratio ≈ 0 → EMPTY
- Else if
  - Large area
  - Tall bounding box
  - Human-like aspect ratio → ADULT
- Else → CHILD / OBJECT

This makes the system:

- Deterministic
- Explainable
- Suitable for safety-oriented domains

---

5. Output

For each image, the application prints:

```
Processing: oms_input/adult/adult.jpg
Foreground ratio: 0.80
Occupant type: ADULT
Airbag Activated
------------------------
```

The image is displayed with:

- Seat ROI overlay
- Foreground mask visualization

User interaction:

- Press any key to move to the next image

---

6. Advantages of the Selected Approach

- Works without AI or training data
- Scales well on embedded platforms
- Robust to stationary occupants
- Easy to tune and maintain
- Suitable for early-stage OMS POCs

Aligns with real automotive OMS development practices

7. Tools & Environment

- Language: C++
- Library: OpenCV (PC build)
- OS: Linux (Ubuntu)
- Build: g++ / CMake
- Auxiliary: Python (used only to generate synthetic test images)

Python is **not required** in the embedded target. It was used only to:

- Auto-generate consistent test images
- Validate algorithm behavior

---

8. Validation Strategy

Testing was done using:

- Real downloaded images
- Synthetic images generated via Python

Each class was validated:

Input Type	Expected Output	Result
Adult	ADULT	Pass
Child	CHILD	Pass
Object	OBJECT	Pass
Empty	EMPTY	Pass

Edge cases were refined by:

Improving background subtraction

- Normalizing ROI area
- Fixing thresholding errors

- Correcting feature normalization

---

9. Limitations of POC

- Works with fixed camera angle
- ROI is manually defined
- Lighting changes not modeled
- Single-frame decision (no temporal filtering)
- Not trained on real in-car datasets

10. Path to Embedded (OKT507) Migration

For migration:

- Replace file input with camera frames
- Remove Python dependency
- Use same C++ OpenCV pipeline
- Replace imshow() with framebuffer / HDMI rendering
- Optimize:
    - Image resolution
    - Memory usage
    - Processing time

The algorithm itself remains unchanged.

---

11. Conclusion

This POC demonstrates:

- Understanding of automotive safety requirements
- End-to-end computer vision pipeline
- Deterministic classification suitable for safety
- Embedded-ready design
- Clear separation between algorithm and platform

This OMS POC successfully demonstrates a complete image-to-decision pipeline using a lightweight, rule-based vision approach. The design is platform-independent and modular, making it ready for migration to the OKT507 embedded target. Only the image input source (camera instead of files) and the display/output mechanism need to be adapted; the core OMS algorithm remains unchanged. This ensures a smooth transition from PC-based validation to real-time embedded deployment.
