Occupant Monitoring System (OMS)

1. Objective of the POC

Objective of the POC

The objective of this POC is to design and validate a simplified **Occupant Monitoring System (OMS)** with integrated **Child Presence Detection (CPD)** that can classify seat occupancy as **ADULT, CHILD, OBJECT, or EMPTY** and make a corresponding airbag decision. The system also aims to detect when a child is left unattended in the vehicle and trigger an alert after a defined duration. This POC demonstrates:

- An end-to-end vision pipeline from image input to safety decision and alert
- Deterministic and explainable behavior suitable for automotive safety use-cases
- Feasibility of implementing OMS and CPD without AI or training data
- Real-time readiness for both frame-based (PC) and live camera-based (embedded) inputs
- Smooth migration from PC simulation to the OKT507 board with a real camera

This POC serves as a functional and technical foundation for a real in-vehicle OMS with CPD capability, aligned with embedded constraints and automotive safety principles.

---

2. Problem Statement

- Incorrect airbag deployment can cause serious injury to children or when the seat is empty or occupied by an object
- A child left alone inside a parked vehicle faces life-threatening risks
- Vehicles need a reliable way to identify seat occupancy as **EMPTY, OBJECT, CHILD, or ADULT**
- Airbag behavior must adapt safely based on occupant type
- Child presence must be monitored over time to trigger alerts when unattended
- The solution must be lightweight, deterministic, and suitable for embedded automotive hardware
- Heavy AI models are not always feasible on resource-constrained platforms

This POC addresses these problems by implementing a rule-based Occupant Monitoring System with integrated Child Presence Detection using classical computer vision, designed for migration to OKT507.

---

3. Input Structure

The system processes images from the following folder structure:

```
oms_input/
├── adult/
│   └── adult.jpg
```

```
│   └── 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

The OMS POC follows a classical vision pipeline:

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

For **Child Presence Detection (CPD)**, validation was performed using repeated CHILD frames (synthetic images):

Scenario	Observed Behavior	Result
Single CHILD frame	No CPD alert	Pass
CHILD repeated multiple times	CPD alert is triggered	Pass

---

9. Challenges Faced

1 False Occupant Classification

- Initial threshold values caused frequent misclassification
- CHILD and OBJECT were sometimes detected as ADULT
- Solved by iterative tuning using synthetic datasets and foreground ratios

2 Image Quality Variations

- Internet-downloaded images had different lighting and perspectives
- Caused unstable foreground extraction
- Reinforced the need for controlled camera placement and calibration

3 Background Dependency

- Background subtraction depends heavily on a clean “empty seat” reference
- Minor changes in lighting affected foreground detection
- Highlighted the importance of capturing a proper baseline frame

4 CPD Simulation Limitations

- Real child data could not be captured in office environment

- CPD had to be validated using synthetic and repeated frames
- Counter-based logic was introduced for PC testing

5 Embedded Constraints

- OMS logic had to remain lightweight for OKT507
- Avoided heavy AI models and training dependencies
- Required a deterministic and explainable approach

---

10. Current Status of POC

- OMS pipeline implemented using classical computer vision
- Occupant classification: EMPTY / OBJECT / CHILD / ADULT working
- Airbag enable/disable logic integrated
- CPD logic added using child-presence persistence
- Validated using synthetic image sets
- Ready for migration to OKT507 with camera input
- POC objectives achieved

---

11. Future Enhancements

Embedded(OKT507) 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.

---

12. Conclusion

This POC demonstrates a reliable **Occupant Monitoring System (OMS)** with integrated **Child Presence Detection (CPD)** using simple, explainable image processing. It can:

- Classify the seat as **EMPTY, OBJECT, CHILD, or ADULT**
- Control airbag activation safely

- Detect a child left alone and trigger a CPD alert
- Operate on synthetic inputs and is ready for real camera integration

The project proves that safety features can be built without AI, using deterministic logic suitable for embedded systems. It forms a strong foundation for migration to the OKT507 platform with real-time camera input and automotive-grade alerts.This ensures a smooth transition from PC-based validation to real-time embedded deployment.
