|  |  |
| --- | --- |
| **Requirement Area** | **Linked Test Case** |
| **1. Image Capture and Quality** | **Test Case 1** |
| **2. Object and Person Recognition** | **Test Case 3** |
| **3. People Counting** | **Test Case 5** |
| **4. Alerts and Notifications** | **Test Case 6** |
| **5. Data Management** | **Test Case 7** |
| **6. User Interface and Accessibility** | **Test Case 10** |
| **7. Remote Access** | **Test Case 11** |
| **8. Labeling and Training Data** | **Test Case 9** |

**User Requirements (Functional)**

**1. Image Capture and Quality**

1.1 The system should capture clear and detailed images.  
1.2 The system shall operate effectively in a variety of lighting conditions (bright, dim, etc.) and capture clear and detailed images.

**2. Object and Person Recognition**

2.1 The system shall accurately identify specific objects and persons.  
2.2 The system shall support incremental learning to recognize new objects over time.  
2.3 The system shall recognize and identify objects in real time.  
2.4 The system shall validate object recognition performance using academic principles of machine learning and computer vision.  
2.5 The system shall use YOLO as the preferred model for object detection.

**3. People counting**

3.1 The system shall count the number of people entering a defined area.  
3.2 The system shall allow evaluation using test videos of known crowd sizes (validated through manual inspection).

**4. Alerts and notifications**

4.1 The system shall allow users to configure alerts for specific events or detections.  
4.2 The system shall send alerts via multiple channels (email, SMS, in-app notification).

**5. Data Management**

5.1 The system shall integrate with existing data systems (where applicable).  
5.2 The system shall provide easy access to historical and real-time data.  
5.3 The system shall transmit data to a central server using the MQTT protocol.

**6. User Interface and Accessibility**

6.1 The system shall have a simple user interface for setup and monitoring.  
6.2 The interface shall clearly display detected objects and alert history.  
6.3 A web-based dashboard shall present relevant insights and metrics.

**7. Remote access**

7.1 The system shall allow remote access to the live video feed and system settings.

**8. Labeling and Training Data**

8.1 The system shall use labelling tools (e.g., Label Img) to annotate images and train recognition models.

Spectra VisionAi Test Cases

Image Quality and Environmental Adaptation

Test Case 1: Lighting Condition Robustness

* Objective: Validate image clarity in varying lighting.
* Steps:
  + Capture images in bright sunlight, dim indoor lighting, and mixed conditions (e.g., backlit scenes).
  + Use resolution charts to measure sharpness, noise levels, and dynamic range.
  + Compare results against predefined thresholds for acceptable clarity.
* Success Criteria: Maintains >90% clarity in all conditions

Test Case 2: Low-Light Performance

* Objective: Ensure usable images in near darkness.
* Steps:
  + Activate low-light mode and capture images with minimal ambient light.
  + Measure object recognition accuracy using YOLO.
* Success Criteria: Recognizes simple objects (e.g., people, products) with >80% accuracy.

Object Recognition and Learning

Test Case 3: Baseline Object Detection

* Objective: Validate accuracy for predefined objects (products, people).
* Steps:
  + Feed images with labeled objects into the YOLO model[8](https://encord.com/blog/yolo-object-detection-guide/).
  + Compare detected objects against ground-truth labels.
* Success Criteria: Achieves mAP (mean Average Precision) ≥0.85 on COCO dataset benchmarks.

Test Case 4: Incremental Learning

* Objective: Test system’s ability to recognize new objects over time.
* Steps:
  + Introduce a new object class (e.g., "sewer valve") using LabelImg9.
  + Retrain the model and validate detection accuracy.
* Success Criteria: New class detection accuracy ≥75% after retraining.

Test Case 5: Real-Time Processing

* Objective: Ensure object labelling occurs in real-time.
* Steps:
  + Stream live video to the system.
  + Measure latency between frame capture and object labelling.
* Success Criteria: Latency <200ms per frame.

Alert System and Integration

Test Case 6: Custom Alert Triggers

* Objective: Validate alert customization and delivery.
* Steps:
  + Set alerts for specific objects (e.g., "untreated sewage").
  + Trigger events and verify alerts via email, SMS, and app.
* Success Criteria: Alerts delivered within 5 seconds of detection.

Test Case 7: MQTT Data Transmission

* Objective: Ensure seamless data integration with servers.
* Steps:
  + Transmit detection logs via MQTT.
  + Verify data integrity and latency at the central server.
* Success Criteria: Zero packet loss; latency <1 second.

Specific Task Validation

|  |  |  |
| --- | --- | --- |
| Task | Test Scenario | Success Metric |
| Smart Alarm | Detect unauthorized entry in a restricted zone | 95% detection rate; <2% false positives |
| People Counting | Count individuals entering/exiting a space with varying crowd densities. | Accuracy ≥90% vs. manual counts |
| Watering Monitoring | Detect Illegal sewage discharge in river footage | Identify discharge events within 10 seconds |
| Sewer Valve monitoring | Track valve position changes in real-time | Position accuracy +5° |

Model and Data Validation

Test Case 8: YOLO Performance Benchmarking

* Objective: Compare YOLO against alternatives (e.g., Faster R-CNN).
* Steps:
  + Train both models on custom-labelled data9.
  + Evaluate speed (FPS) and accuracy (mAP).
* Success Criteria: YOLO achieves ≥2× faster inference with comparable accuracy.

Test Case 9: Data Annotation Workflow

* Objective: Ensure LabelImg compatibility and export accuracy.
* Steps:
  + Annotate 100 images with bounding boxes.
  + Verify YOLO-compatible TXT files are generated correctly.
* Success Criteria: 100% format compliance; annotations align with objects.

Usability and Remote Access

Test Case 10: Dashboard Clarity

* Objective: Validate intuitive display of alerts and object labels.
* Steps:
  + Simulate detection events.
  + Survey users on interface readability.
* Success Criteria: ≥90% user satisfaction on clarity.

Test Case 11: Remote Configuration

* Objective: Test camera settings adjustment via web/app.
* Steps:
  + Change resolution, alert rules, and model parameters remotely.
  + Verify settings apply without system reboot.
* Success Criteria: Changes within 10 seconds.

Academic and Technical Considerations

* Edge Case Handling: Test occlusion robustness using synthetic data.
* Computational Efficiency: Monitor GPU/CPU usage during peak loads.
* Ethical Compliance: Anonymize faces in public spaces to adhere to privacy laws.

By systematically addressing these test cases, the AI camera system can meet both functional requirements and real-world operational demands.