### **Step 1: Video Preprocessing**

* **Objective:** Prepare the surveillance video for change detection.
* **Actions:**
  + Convert the video frames to grayscale to simplify processing.
  + Apply Gaussian blur to reduce noise and minor fluctuations that could lead to false detections.

### **Step 2: Background Subtraction using Mixture of Gaussians (MOG)**

* **Objective:** Identify foreground objects that represent changes in the environment.
* **Actions:**
  + Implement the Mixture of Gaussians (MOG) model to separate foreground objects from the static background.
  + Fine-tune the MOG parameters (like the number of Gaussians, learning rate) to adapt to your specific environment.

### **Step 3: Temporal Differencing**

* **Objective:** Detect motion and changes by comparing consecutive frames.
* **Actions:**
  + Implement basic frame differencing between consecutive frames to capture moving objects.
  + Use a combination of two-frame and multi-frame differencing to detect both quick and slow changes.
* **Novelty:**
  + Introduce multi-frame differencing to identify gradual changes, like a stationary object slowly moving or environmental lighting changes.

### **Step 4: Morphological Operations**

* **Objective:** Clean up the detected foreground to remove noise and fill in gaps.
* **Actions:**
  + Apply morphological operations like erosion and dilation to remove small, irrelevant changes and to enhance the shape of detected objects.
  + Use contour detection to focus on continuous regions of change.

### **Step 5: Object Classification**

* **Objective:** Classify detected changes to understand their nature (e.g., person, vehicle, object).
* **Actions:**
  + Implement a simple object classification method (e.g., using a pre-trained model like MobileNet) to categorize the detected changes.
  + Focus on distinguishing between key categories relevant to your environment, such as vehicles, people, and objects.

### **Step 6: Contextual Filtering**

* **Objective:** Filter and prioritize detected changes based on their context within the scene.
* **Actions:**
  + Assign priority levels to different zones in your surveillance area (e.g., high-traffic areas, restricted zones).
  + Filter detected changes based on their location and type. For example, a person entering a restricted zone might be given higher priority than a person in a public area.
* **Novelty:**
  + Apply contextual filtering to provide more meaningful insights, focusing on the significance of changes within different areas of the scene.

### **Step 7: Contextual Annotations**

* **Objective:** Annotate detected changes with relevant metadata for deeper analysis.
* **Actions:**
  + Annotate each detected change with details like the type of change (person, vehicle), location within the scene, and time of occurrence.
  + Include additional contextual annotations, such as the duration of the change and proximity to critical areas (e.g., exits/entrances).
* **Novelty:**
  + Use this contextual annotation to analyze patterns, such as identifying high-traffic areas, repeated movements, or potential security risks.

### **Step 8: Post-Processing and Visualization**

* **Objective:** Provide a visual and analytical summary of the detected changes.
* **Actions:**
  + Generate visual outputs, like bounding boxes around detected changes and annotated frames.
  + Create summary reports that highlight the most significant changes, contextual information, and any detected patterns.

### **Step 9: System Testing and Calibration**

* **Objective:** Ensure the system is robust and accurate for your specific environment.
* **Actions:**
  + Test the system in different scenarios, such as varying lighting conditions, different times of day, and with various objects.
  + Calibrate the system parameters based on these tests to minimize false positives and negatives.