Smart CCTV model for Change and Object Detection using Open CV

Shreya Todmal, MTech GI
Indian Institute of Technology, Kanpur

Introduction

The goal of this project is to develop a Smart CCTV system that can efficiently detect significant changes in the environment and identify objects within those changes using OpenCV. The system processes video frames in real-time, identifies and discards irrelevant background movements, and isolates frames with meaningful changes for further analysis. This approach helps in reducing the amount of data stored while ensuring that critical incidents are captured and analyzed.

Problem Statement

Traditional CCTV systems continuously record video footage, leading to vast amounts of data, much of which is redundant. This project addresses the inefficiency of traditional surveillance systems by developing a method to detect and store only significant changes and objects in the footage, thereby optimizing storage and improving the relevance of recorded data.

Objective

- Efficient Frame Processing: To process each frame of the video in real-time and identify significant changes.
- Background Subtraction: To differentiate between static background and dynamic foreground elements.
- Noise Reduction: To remove irrelevant background movements, such as swaying trees, to avoid false detections.
- Change Detection: To identify and save frames where meaningful changes occur.
- Object Detection: To detect and label objects within the frames where changes are detected.
- Optimized Storage: To save only the frames with significant changes, reducing storage requirements.

Methodology

Frame Selection

Each frame of the video is read and resized to a uniform dimension (600x400 pixels) for consistent processing. This ensures that the system works efficiently regardless of the original video resolution.

Background Subtraction

The system uses a background subtractor model (cv2.createBackgroundSubtractorMOG2) to distinguish between background and foreground elements in each frame. This model adapts over time to handle dynamic backgrounds.

Figure 1 Background subtraction from video

Noise Reduction

To remove noise from the foreground mask, morphological operations are applied using a rectangular structuring element. This step helps in eliminating small, irrelevant changes (e.g., leaves moving).

Figure 2 Subtracting background noise

Connected Component Analysis

The system identifies and retains only large connected components in the foreground, discarding smaller, irrelevant movements. This is done using the f_keepLargeComponents function, which keeps components above a certain size threshold.

Figure 3 Keeping connected components in background

Change Detection and Frame Saving

Frames where significant changes occur are shortlisted and saved for further analysis. If consecutive frames have changes, they are grouped together and saved as a sequence using the f_saveSeq function.

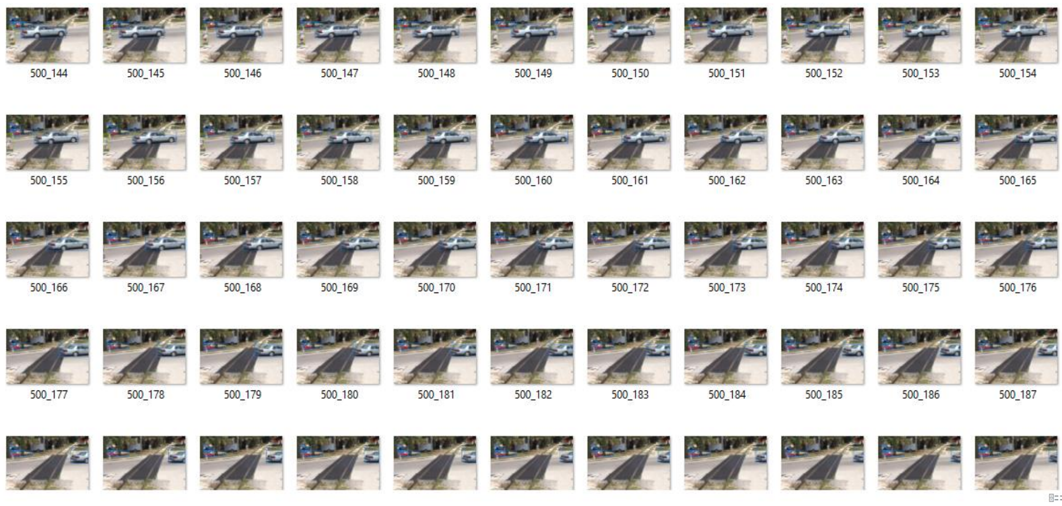

Figure 4 Saved change detected frames

Object detection

In the frames with detected changes, objects are identified using cvlib. Bounding boxes are drawn around detected objects, and the frames are saved with these annotations.

Figure 5 Changed detected frames with object detection

Visualization

The system displays the original frame alongside the foreground mask, allowing for a visual comparison between the processed output and the input video.

Result

- The system successfully identifies and isolates significant changes in the video, ignoring irrelevant background movements.

- Objects within the frames where changes occur are detected and labeled, providing actionable insights for surveillance.
- The storage requirements are reduced by saving only the relevant frames, optimizing data management.

Testing on different data set:

Figure 6 Background subtraction and processed image

Figure 7 Object detection

Insights and Future Scope

Future enhancements could include integrating machine learning models for more advanced object detection and real-time alerts. Additionally, the system could be extended to handle different types of environments and conditions, improving its robustness and applicability in various scenarios.