**Anomaly Detection in Video Surveillance**

## **Project Overview:**

The project aims to detect anomalies in CCTV footage using machine learning techniques, specifically focusing on environmental changes like fog, haze, and other atmospheric conditions in both indoor and outdoor settings. The goal is to build a model that can differentiate between normal and anomalous events while adapting to environmental factors that might obscure or alter the footage.

## **Tasks Completed So Far:**

### **1. Initial Research and Understanding of Problem Scope:**

* Researched relevant machine learning models suitable for anomaly detection in video data.
* Identified potential challenges with environmental factors (e.g., fog, haze) in the videos, as they can affect the clarity and recognition of objects and events.
* Determined that the task requires both spatial (frame-level) and temporal (video sequence) analysis for accurate detection of anomalies.

### **2. Data Collection:**

* Collected a dataset consisting of 15 videos, each ranging from 15 seconds to 1 minute in duration.
* These videos contain a mixture of both normal and anomalous events (e.g., unusual activity, movement, or environmental conditions).
* Identified the need for preprocessing the dataset to add environmental changes such as fog and haze for additional complexity and realism.

### **3. Data Preprocessing:**

* Developed Python scripts to preprocess the videos, including:
  + Frame Extraction: Extracted frames from videos for easy manipulation.
  + Resizing and Normalization: Resized video frames to a standard size (e.g., 224x224 pixels) to ensure uniformity for model training.
* Resolved issues with kernel crashes during resizing by offloading the task to an external Python script.

### **4. Augmenting Videos with Fog and Haze:**

* Implemented augmentation techniques to add synthetic fog and haze effects to the videos to simulate real-world environmental conditions.
* Applied fog/haze to selected videos in the dataset to create a balanced variety between clear and foggy/hazy scenarios.
* Successfully handled video processing using OpenCV to add fog and haze effects, creating variations in the dataset for training.

### **5. Labelling the Data:**

* Created a CSV file to label each video as either Normal or Anomaly based on the content.
* Developed a Python function to load video data and associate it with the corresponding labels from the CSV file.
* Ensured labels are correctly converted to binary format for supervised learning (0 for Normal, 1 for Anomaly).