**CCTV Surveillance Threat Detection (CSTD) Software/App**

****

**By Sandeep Singh Sajwan**

**(Final Project/FeynnLabs)**

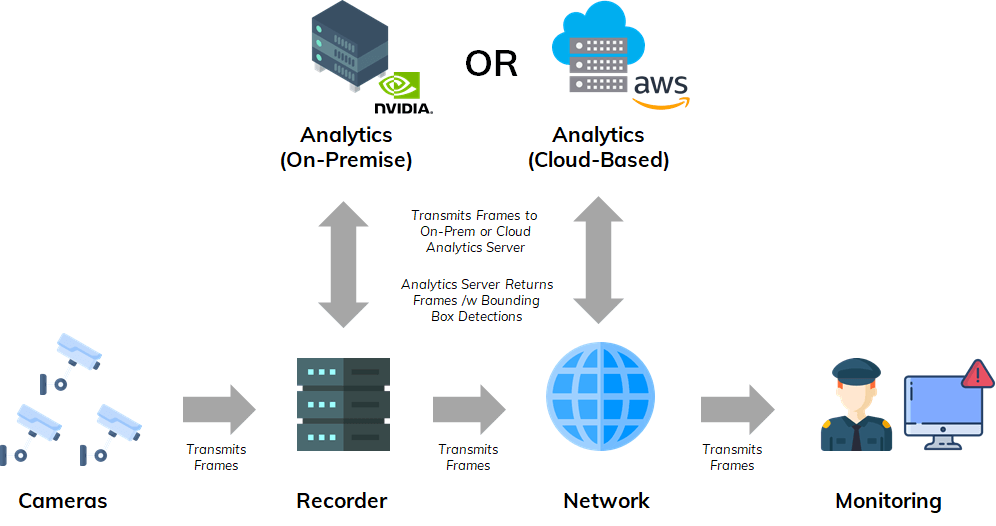
**Problem Statement:**

CCTV surveillance is needed for many reasons, including:

****

* **Crime prevention**: CCTV cameras can deter criminals from committing crimes like theft, vandalism, and break-ins.
* **Evidence collection**: CCTV footage can be used to identify criminals and as evidence in court.
* **Peace of mind**: CCTV cameras can provide peace of mind by continuously monitoring areas in real time.
* **Protecting assets**: CCTV cameras can help protect a business's assets, intellectual property, and sensitive information.
* **Protecting employees**: CCTV cameras can help protect employees from workplace violence, sexual harassment, and internal theft.
* **Monitoring activities**: CCTV cameras can monitor the activities of employees and visitors.
* **Dispute resolution**: CCTV footage can be used to settle disputes, such as employee feuds and altercations between staff and customers.
* **Insurance**: CCTV cameras can be used for insurance purposes.
* **Monitoring pets**: CCTV cameras can be used to monitor pets.
* **Accidents**: CCTV cameras can be used to monitor for accidents that may occur on a property.
* **Property value**: CCTV cameras can increase the value of a property.

**Prototype:**

****

Github Link: https://github.com/sandeep4seyeon/CCTV-Surveillance-Model

**How it works:**

### 1. ****Data Collection and Input****

* **Video Streams**: The CCTV cameras continuously capture footage in real-time.
* **Input Data**: The video frames from the cameras are fed into the system. They may include images, motion patterns, or events like a person entering a restricted area.

### 2. ****Preprocessing****

* **Frame Processing**: The video is broken down into individual frames.
* **Feature Extraction**: Key features such as motion, object recognition, facial detection, or behavior patterns are extracted from these frames.

### 3. ****Machine Learning Models****

* **Object Detection Models**: These models (e.g., YOLO, Faster R-CNN) identify objects (e.g., humans, vehicles, weapons) in the video feed.
* **Anomaly Detection Models**: Unsupervised ML models can detect unusual behavior by learning what "normal" activity looks like and flagging deviations.
* **Face Recognition Models**: Facial recognition models use deep learning to match faces against a database of known individuals (for authorized/unauthorized access).
* **Motion Detection**: Algorithms detect movement and classify it as normal (like walking) or suspicious (e.g., erratic movements, running).

### 4. ****Threat Classification****

* The ML model analyzes the data for **patterns of threats** such as:
  + Unattended bags
  + Unusual crowd behavior
  + Intrusion into restricted areas
  + Violent behavior or object like weapons
* These can be classified as **low-risk** or **high-risk** threats.

### 5. ****Alert System****

* When a potential threat is detected, the system automatically triggers an **alert**. This could include sending notifications to security teams or sounding alarms.

### 6. ****Continuous Learning****

* The model learns continuously, improving threat detection accuracy over time. This happens by analyzing false positives/negatives and retraining on updated datasets.

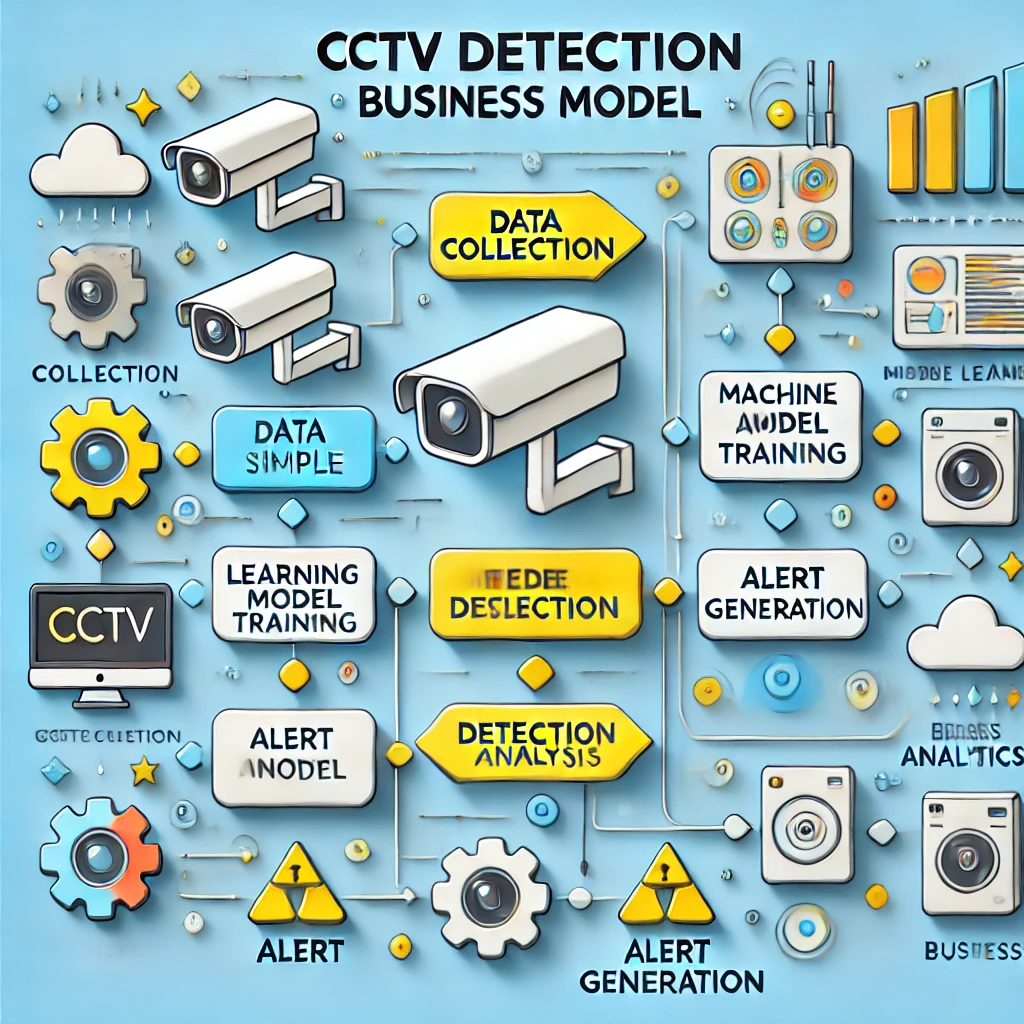
### 7. ****Decision Making****

* **Human Intervention**: If the system detects something uncertain, it might escalate the decision-making process to a human operator.
* **Automated Response**: For clear threats, the system can trigger immediate responses like lockdowns, alarms, or notifications.

### Example Models:

* **Convolutional Neural Networks (CNNs)**: Used for image recognition and object detection.
* **Support Vector Machines (SVMs)**: Sometimes used for classifying different behaviors (normal vs. suspicious).
* **Recurrent Neural Networks (RNNs)**: Can be applied for video sequence analysis to detect suspicious activity over time.

***Business Model (financial equations) :***

****

**1. Revenue Model (R)**

Revenue can be generated through different business models. Let’s define possible approaches:

* **Subscription-based Revenue (R\_sub):** Charge customers (businesses, homes, municipalities, etc.) a recurring monthly or annual subscription fee for using your AI surveillance system. The subscription could be tiered based on features or the number of cameras/locations monitored.

Rsub​=Subscription Fee×Number of Subscribers

* **One-time Sales (R\_sales):** You could sell the model as a one-time product or license the software to customers who want to run it on their own systems (e.g., large corporations or municipalities).

Rsales​=Sales Price per System×Number of Systems Sold

* **Custom Integrations (R\_custom}:** Offer custom integrations or add-on services for specific industries (e.g., retail, transport, warehouses) that need tailored surveillance solutions.

Rcustom​=Custom Integration Fee×Number of Custom Projects

* **Savings-based Model (R\_savings):** You may also offer a model where your revenue comes from a **shared savings agreement**. This is based on how much the client saves by using your system (e.g., reducing theft, damages, or insurance premiums). You take a percentage of these savings.

Rsavings​=Percentage of Client Savings×Total Savings

Thus, the **Total Revenue (R\_total)** becomes:

Rtotal​=Rsub​+Rsales​+Rcustom​+Rsavings​

**2. Cost Structure (C)**

Your costs will likely be in four primary categories:

* **Development Costs (C\_dev):** These include costs to build, train, and improve the AI model, the development of the platform (e.g., website, dashboards, APIs), and integrating with hardware.

Cdev​ = Initial Development Cost + Ongoing Model Improvements

* **Hardware and Deployment Costs (C\_hard):** If you provide hardware (e.g., cameras, servers, edge devices), you’ll need to factor in these costs. If you're a software-only business, clients bear this cost.

Chard​=Cost per Camera/Edge Device×Number of Cameras

* **Operational Costs (C\_op):** Recurring expenses for maintaining the system, such as cloud infrastructure, server hosting, API calls, model retraining, and customer support.

Cop​=Cloud/Server Cost+Customer Support+Maintenance

* **Sales, Marketing, and Customer Acquisition Costs (C\_mark}:** This covers marketing, outreach, sales team costs, and customer onboarding. You can estimate the cost to acquire each customer.

Cmark​=Cost per Customer Acquisition×Number of Customers

Thus, **Total Costs (C\_total)**:

Ctotal​=Cdev​+Chard​+Cop​+Cmark​

**3. Profit Equation (P)**

To determine the **monetary profit (P)** for your business, we subtract the total costs from the total revenue:

P=Rtotal​−Ctotal​

**4. Monetary schemes:**

* **Scaling Revenue:** Increasing the number of subscribers, systems sold, or custom projects leads to more revenue. You can also scale through **partnering** with hardware manufacturers or offering your system through **third-party distributors**.
* **Scaling Costs:** As your subscriber base grows, operational and customer support costs may increase. However, economies of scale will reduce the per-user cost, especially in terms of cloud infrastructure, marketing, and support.