Using an existing CCTV network for ,crowd management, crime prevention, and work monitoring

Rajdeep Singh Mulae, Chirag, Rahul Kumar, Khushi
CHANDIGARH ENGINEERING COLLEGE , MOHALI
Bachelor of Technology in Robotics and Artificial Intelligence

ABSTRACT
Anomaly detection in video surveillance has become a sophisticated field of study, attracting considerable attention from researchers. There is a growing demand for smart systems that can automatically spot unusual events in real-time video streams. As a result, various techniques have been introduced to create effective models aimed at improving public safety. Many reviews have examined different facets of anomaly detection, such as network anomalies, financial fraud, and the analysis of human behavior. The use of deep learning has shown remarkable effectiveness across numerous areas of computer vision. Importantly, the swift progress of generative models has placed them at the forefront of modern methodologies. This paper aims to provide a comprehensive review of deep learning-based approaches for detecting anomalies in video. These methods are categorized according to their goals and learning metrics.

Closed-Circuit Television (CCTV) systems play a vital role in contemporary security frameworks, providing uninterrupted surveillance that serves both as a deterrent and an essential mechanism for monitoring and gathering evidence. In contrast to human security personnel, who may experience fatigue and have limitations in their field of vision, CCTV cameras deliver reliable, round-the-clock observation of critical locations. They address deficiencies in existing security measures by facilitating real-time monitoring and recording of incidents for subsequent analysis, thereby ensuring that potential security threats are identified and managed more efficiently. This not only enhances the overall effectiveness of security measures but also diminishes the dependence on human oversight. The incorporation of Artificial Intelligence and Machine Learning (AIML) technologies into current CCTV systems offers a promising strategy to tackle significant challenges in urban settings. This initiative explores the utilization of AIML for crowd management, crime deterrence, and workplace oversight through CCTV infrastructure. In terms of crowd management, AIML allows for automated counting and density assessment of crowds, which aids in the effective distribution of resources during events and emergencies. For crime prevention, AIML algorithms process video feeds in real-time to identify suspicious behaviors and detect anomalies, thereby assisting law enforcement in taking proactive measures. Furthermore, AIML improves workplace monitoring by evaluating productivity indicators, ensuring adherence to safety regulations, and streamlining operational processes. The fusion of AIML with existing CCTV systems signifies a groundbreaking evolution in urban surveillance and management strategies, providing scalable solutions to a variety of urban challenges.

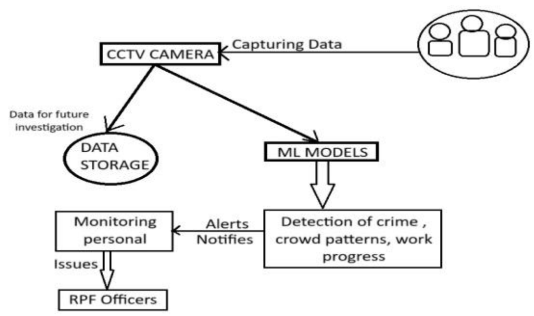
The use of current CCTV systems for efficient crowd management, crime prevention, and employee monitoring by applying artificial intelligence and machine learning (AI/ML) technologies. This strategy utilizes real-time video analytics to boost situational awareness, allowing for proactive responses to potential criminal activities while also enhancing workforce management. By combining AI/ML algorithms with existing surveillance systems, the proposed framework seeks to enhance safety and operational efficiency across different settings.

I. INTRODUCTION

In today's world, where cities are growing and populations are booming, we face some pretty tough challenges when it comes to public safety and keeping things running smoothly. That's why it's become so important to make the most of what we already have. Take Closed-Circuit Television (CCTV) networks, for example. They started out as simple surveillance tools, but now, when we combine them with Artificial

Intelligence (AI) and Machine Learning (ML), they turn into something much more powerful.

With these advancements, traditional CCTV systems can help us manage crowds in real-time, prevent crimes before they happen, and monitor workplaces more effectively—all with improved accuracy and efficiency. Thanks to AI and ML algorithms, these networks can spot unusual activities, recognize faces, analyze how crowded a space is, track movements, and even check if safety rules are being followed. This not only boosts security but also helps everyone make better decisions, whether in public or private sectors. Embracing AI-powered CCTV systems is a smart move toward creating smarter cities, safer neighborhoods, and more efficient workplaces, all while reducing the need for costly new infrastructure.



II. A. METHODOLOGY USED IN CROWD MANAGEMENT

The application under discussion is a comprehensive solution for automated crowd counting and management. It is specifically designed to detect and count individuals in images, video files, and live camera streams, offering a robust tool for real-time crowd analysis. The system supports various use cases such as event oversight, public safety monitoring, and analytics-driven decision-making. Its main goal is to provide users with accurate crowd statistics, real-time visualizations, and timely alerts when predefined thresholds are exceeded.

The application is built using a combination of advanced technologies. At its core is YOLOv3 (You Only Look Once), a fast and accurate deep learning-based object detection algorithm. For processing images and videos, OpenCV is used, enabling frame manipulation and data preprocessing. The user interface is developed with Streamlit, a Python-based web framework that allows the creation of interactive applications with ease. For performance enhancement, CUDA is optionally employed, allowing for GPU acceleration on compatible NVIDIA hardware. Additionally, the Telegram API is integrated to send real-time alerts when crowd limits are breached.

The system architecture includes several important components. During model integration, the YOLOv3 configuration and weights are loaded using OpenCV's DNN module. Detection is based on the COCO dataset, which includes the class label for "person." Input data, whether images or video frames, undergo preprocessing that involves resizing and normalization before being passed through the detection model. After inference, Non-Maximum Suppression (NMS) is applied to eliminate redundant or overlapping detections and retain the most confident ones. The final output includes bounding boxes drawn around detected individuals along with a live count.

A significant feature of the system is its real-time processing capability. It captures video streams from a webcam, processes them through the detection model, and displays output along with statistical data such as current person count, average count, and frame rate (FPS). In addition to real-time input, the application supports file uploads, allowing users to process static images and pre-recorded videos. Processed outputs can be reviewed on-screen and downloaded for further use.

The alert mechanism is an essential part of the system, especially for safety-critical environments. Users can define a crowd size threshold, and if the detected count surpasses this value, the system sends out a notification via Telegram. The alert process is implemented in a thread-safe manner using Python's threading and queue modules to ensure stable performance.

The interface is designed to be user-friendly, offering customization options for detection settings such as confidence thresholds, minimum object size, and NMS filtering parameters. Users can switch between optimized presets depending on scene density and choose whether or not to enable alerts.

Performance optimization is handled via GPU support when available, resolution tuning to balance speed and accuracy, and batch frame processing in video mode. The system also includes robust error handling to catch issues in model loading, inference, or API usage, and validates user input to avoid misconfigurations.

Finally, the application emphasizes privacy and ethical considerations by ensuring that all processing occurs locally without uploading data to external servers. It is designed for responsible use, supporting scenarios like crowd control, safety audits, and data-driven event planning.

B.METHODOLOGY FOR CRIME PREVENTION

This crime prevention application uses real-time video analysis to detect weapons, fights, and suspicious objects in surveillance footage or webcam feeds. It is designed to improve public safety by identifying potential threats through computer vision. The system analyzes each frame of a video stream to detect aggressive behavior or dangerous items, helping with proactive crime detection and prevention.

At its core is the YOLOv4-Tiny model, a lightweight and efficient object detection architecture optimized for speed and real-time performance. Supporting technologies include OpenCV for frame processing, Streamlit for the user interface, and NumPy for computations. The system primarily runs on CPU but supports GPU acceleration for improved performance.

The application supports two input modes: a live webcam feed or an uploaded video. Users can select the input source through a sidebar in the Streamlit interface. Each video frame is preprocessed by resizing, normalizing, and converting it into a blob format using cv2.dnn.blobFromImage, preparing it for YOLO processing.

Once preprocessed, frames are passed through the YOLOv4-Tiny network, which detects objects and returns bounding boxes with confidence scores. Non-Maximum Suppression (NMS) is used to remove overlapping detections and retain the most reliable results. The system monitors specifically for weapons like guns, knives, and rifles, as well as suspicious items like backpacks and bags.

A key feature is the fight detection mechanism, which analyzes motion patterns between video frames. It calculates frame differences, applies thresholding to highlight movement, and uses contour detection to identify significant activity. By tracking motion over time, the system can detect behavior that suggests a physical fight.

When a threat is detected—such as a weapon or a fight—the system generates real-time alerts. These alerts are shown on the video frames with bounding boxes and labels like “Weapon Detected” or “FIGHT DETECTED!”, using different colors for different types of alerts. The processed video is displayed live in the app.

The user interface is interactive and allows customization. Users can adjust detection thresholds, choose video input, and view live feedback through the annotated video. The system includes error handling, alerting users about missing model files, inaccessible webcams, or invalid video uploads.

To boost performance, the app uses YOLOv4-Tiny for fast inference, background subtraction for lightweight motion detection, and Streamlit’s caching to avoid reloading the model unnecessarily. These optimizations ensure smooth real-time performance without sacrificing accuracy.

In terms of output, the system delivers an annotated video stream that highlights weapons, suspicious items, and aggressive behavior. By combining object detection with motion analysis, this tool helps improve situational awareness in places like schools, public venues, and security checkpoints.

C. METHODOLOGY OF WORK MMONITORING

The Workforce Monitoring Pro system is designed to detect and track the presence of workers in real-time using advanced computer vision techniques. By utilizing YOLOv3 object detection and integrating real-time alerting mechanisms, it ensures efficient workforce supervision across various environments such as construction sites, warehouses, and office spaces.

The system begins by addressing a key issue—monitoring worker presence accurately and providing timely alerts when certain conditions are met. These include worker absence for a predefined period and a decrease in worker count below a set threshold. To address these needs, the system integrates a range of technologies including Streamlit for the user interface, OpenCV for image and video processing, YOLOv3 for object detection, Telegram API for instant notifications, NumPy for calculations, and Pillow for image handling. It also leverages Python’s threading and queue modules for smooth, non-blocking alert delivery.

Upon initialization, session variables such as alert settings, detection logs, and Telegram credentials are stored using Streamlit’s session state, allowing persistent configurations throughout the session. The YOLOv3 model is loaded using configuration and weights files, with optional GPU acceleration via CUDA for improved performance. Class names are retrieved from the COCO dataset to identify human subjects.

The system supports two input sources: live webcam feeds for continuous monitoring, and uploaded files (images or videos) for offline analysis. When processing frames, the system filters YOLOv3 detections to focus on people, applying non-maximum suppression to refine results. Each detection is displayed with bounding boxes and confidence labels, alongside real-time timestamps and worker count overlays.

For real-time video feeds, the camera_capture function drives continuous monitoring, applying the detection pipeline to each frame. It tracks attendance by logging the presence or absence of workers, the time of absence, and calculates metrics like average worker count and frames per second. If no workers are detected for a configured duration or if the count falls below a threshold, alerts are triggered and sent through Telegram asynchronously to avoid blocking the main application flow.

In addition to real-time monitoring, the system can process uploaded images and videos. It detects and marks workers in images, offers side-by-side visual comparisons, and lets users download the annotated results. Video processing applies the detection logic frame-by-frame, shows progress updates, and outputs a downloadable processed video.

Environment-specific presets further optimize detection. These include configurations for construction sites, factory floors, offices, and warehouses. Each preset adjusts detection thresholds, bounding box parameters, and minimum object sizes to suit different surveillance conditions. Users can also configure work shift hours, including overnight shifts, ensuring alerts are only triggered during working hours.

Comprehensive logs of worker presence, with timestamps and status updates, are maintained. The Streamlit interface provides an intuitive layout, with sidebar controls for configuration and expandable sections for advanced settings. Error handling is built-in across all modules—from camera access and model loading to file processing and Telegram integration.

In conclusion, the system blends real-time object detection, intelligent alerting, customizable settings, and robust error management to create a dependable workforce monitoring tool. Its adaptability to different work environments and ease of use make it highly practical for modern workforce supervision and safety enforcement.

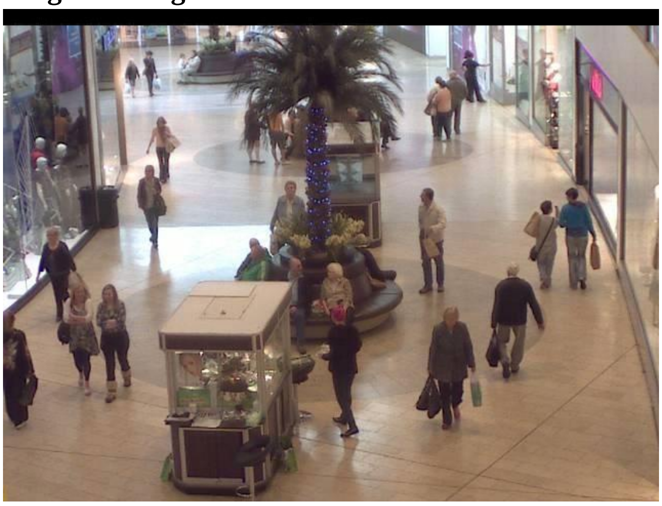
III. RESULTS AND DISCUSSION

A.RESULT OF CROWD MANAGEMENTS :-

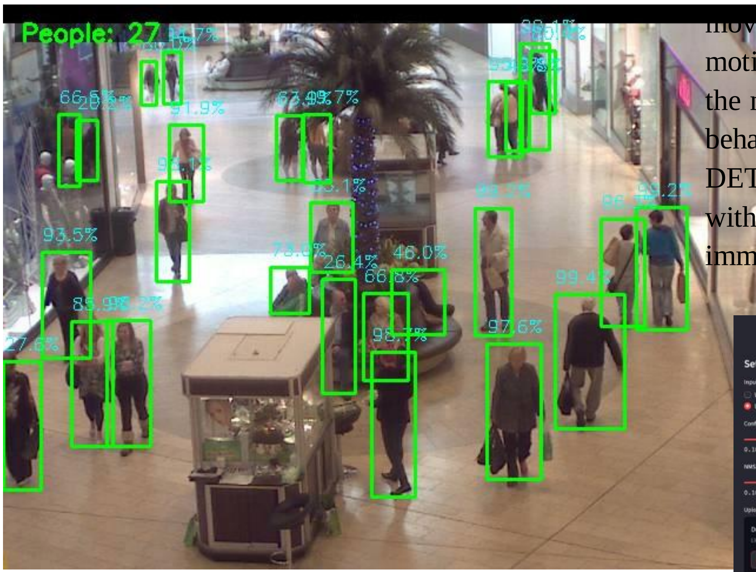
This streamlight-based benefits from the Counter app to identify the Yolov3 object detection model accurately and count individuals in photos, videos and live camera feed, which benefits fromshowcasing real-time results with a delimitation box and confidence percentage. It provides customization settings (including Vishwas threshold and size parameters), many input methods and GPU acceleration for optimal performance. When activated, the Telegram notification system sends when the overload number exceeds the user-defined threshold, while maintaining a configured cold period between information..

With a strong error handling for model loading, credentials and hardware compatibility, this application acts versatile solution for audience monitoring in different scenarios.All paragraphs must be indented. All paragraphs must be justified, i.e. both left-justified and right-justified.

Original image:-



output

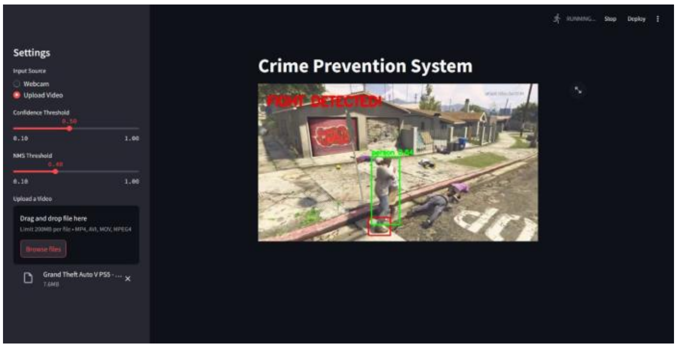


B.RESULT OF CRIME PREVENTION

The crime prevention system demonstrates effective real-time video surveillance capabilities by detecting weapons, suspicious items, and aggressive behavior such as fights. Utilizing the YOLOv4-Tiny model for object detection, the application provides responsive and interactive outputs, making it suitable for proactive monitoring in public and high-security environments.

One of the core functionalities is object detection, where the system identifies specific threats such as knives, guns, pistols, rifles, and suspicious items like backpacks and bags. Leveraging the COCO dataset classes, it can also recognize a wide range of other non-threatening objects. For every detected item, the system draws a bounding box around it and displays the object's name along with a confidence score, helping users quickly identify and assess potential threats within the frame.

In addition to object detection, the system incorporates a fight detection module that analyzes motion patterns between video frames. Using frame differencing, thresholding, and contour analysis, the software detects areas with significant and unusual movement. By maintaining and analyzing a history of motion across frames, the application can determine if the movement suggests aggressive behavior. When such behavior is detected, the system generates a "FIGHT DETECTED!" alert and highlights the area of interest with a red bounding box and bold text, signaling immediate attention.



Real-time processing is a key feature. The system supports two input modes: live webcam feeds and uploaded video files. For webcam inputs, it continuously captures and analyzes frames, providing real-time feedback. In the case of uploaded files, it processes the video frame by frame and displays the annotated output with bounding boxes and alert messages within the Streamlit interface. This allows seamless monitoring, review, and documentation of events.

The application also emphasizes user interaction and configurability. Through a sidebar interface, users can select their input source (webcam or uploaded video) and adjust settings such as the confidence and Non-Maximum Suppression (NMS) thresholds. These customizable parameters help adapt the detection behavior based on different environmental conditions and use-case scenarios.

The system provides clear visual alerts whenever weapons or suspicious behavior are detected. Weapons are marked with distinct bounding boxes, and labeled appropriately. Aggressive motion triggers prominent fight alerts, both aiding in threat recognition and guiding necessary actions for security personnel.

Performance-wise, the system ensures responsive feedback by processing frames as efficiently as the hardware allows. It calculates frame rates and motion intensity metrics, further aiding fight detection accuracy. Additionally, the system incorporates error handling routines. It alerts users if essential model files are missing, if the webcam cannot be accessed, or if no video file is uploaded, ensuring a smoother user experience.

The application's effectiveness is illustrated through various test scenarios: detecting a knife in a video results in a labeled red bounding box; identifying a fight triggers a red alert box with a "FIGHT DETECTED!" label; spotting a backpack results in an orange bounding box with the item label. These examples confirm the system's capability to detect threats accurately and deliver results in real time.

In summary, the application successfully performs real-time object and behavior detection, offers user-friendly interaction, and generates timely alerts, making it a valuable tool for enhancing surveillance and crime prevention.

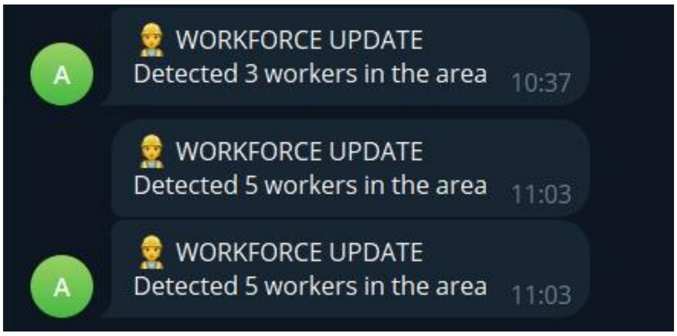
C.RESULT OF WORK MONITORING

The Workforce Monitoring system is a comprehensive solution for tracking worker presence using real-time video analysis and object detection. It uses YOLOv3 and is integrated into a Streamlit web interface. The system provides useful results to help supervisors manage workers across different environments.

The core feature is real-time Worker Detection. It analyzes live webcam feeds or uploaded images and videos to detect people using YOLOv3. Detected individuals are marked with bounding boxes and confidence scores. The system shows the number of workers per frame, tracks their presence over time, and calculates statistics like frames per second (FPS) and the average number of workers over recent frames

Original image:-

Output:-



The system includes Alert Mechanisms to inform supervisors of critical issues. Alerts are triggered if no workers are detected for a set time (e.g., 10 minutes) or if the number of workers drops below a user-defined threshold. Alerts appear in the Streamlit interface and can also be sent to a Telegram account for real-time updates.

For Real-Time Monitoring, the system supports live webcam feeds. Users can configure camera settings such as resolution and rotation. Detection data is displayed directly on the live feed, showing current worker activity, shift status, and any periods of absence. If the camera fails, clear error messages are shown.

The File Processing module lets users upload images or videos for offline analysis. Images are processed and shown with detection annotations. Videos are analyzed frame by frame, generating a new annotated video that can be downloaded. Live statistics and previews are available during video processing.

For Worker Presence Logging, the system keeps a time-stamped history of the last 50 detection events. Each log includes the number of detected workers and a status (present or absent), with color-coded highlights (e.g., green for presence, red for absence) for easier viewing.

Environment Presets make the system adaptable to various workplaces. Presets like Construction Site, Factory Floor, Office Space, and Warehouse adjust detection parameters such as minimum height, confidence level, and overlap suppression to fit different scenarios.

With Shift Scheduling, users can define work hours, including overnight shifts. This ensures alerts are only triggered during active shifts, avoiding false alarms outside of work hours.

Telegram Integration allows the system to send real-time alerts about worker absence or low presence. Messages include details like how long workers have been absent and how many were detected, helping managers respond quickly.

Performance is improved with GPU Acceleration when available, and frame skipping helps maintain a smooth 30 FPS experience. Errors such as missing model files, camera issues, or Telegram configuration problems are clearly shown to the user.

In summary, the Workforce Monitoring Pro system offers real-time detection, alerts, logging, and customization through an easy-to-use interface. It is ideal for managing worker presence in industries like construction, manufacturing, warehousing

IV. CONCLUSION

The integration of Artificial Intelligence and Machine Learning with traditional CCTV systems marks a significant advancement in video surveillance, enabling real-time anomaly detection and enhancing public safety. By automating tasks such as crowd analysis, crime detection, and workplace monitoring, AI/ML technologies overcome the limitations of human oversight and conventional surveillance methods. The use of deep learning techniques, particularly generative models, has proven highly effective in identifying unusual patterns and behaviors in video feeds. This approach not only improves situational awareness and response times but also supports efficient resource allocation in complex urban environments. Overall, the proposed AI/ML-driven surveillance framework offers a scalable and intelligent solution for addressing modern security and operational challenges.

REFERENCE

1). Banerjee, D. C., Krishna, K. V. G., Murthy, G. V. G. K., Srivastava, S. K., & Sinha, R. P. (1994). Occurrence of Spodumene in the Rare Metal-Bearing Pegmatites of Mariagalla-Allapatna Area, Mandya Dist., Karnataka. Journal Geological Society of India, 44(2), 127-139.

2). Mahalakshmi, A., Goud, N. S., & Murthy, G. V. (2018). A survey on phishing and it’s detection techniques based on support vector method (Svm) and software defined networking (sdn). International Journal of Engineering and Advanced Technology, 8(2), 498-503.

3). Murthy, G., & Shankar, R. (2002). Semiconductors II-Surfaces, interfaces, microstructures, and related topicsHamiltonian theory of the fractional quantum Hall effect: Effect of Landau level mixing. Physical Review-Section BCondensed Matter, 65(24), 245309-24530

4). Murthy, G. V. K., Sivanagaraju, S., Satyanarayana, S., & Rao, B. H. (2014). Optimal placement of DG in distribution system to mitigate power quality disturbances. International Journal of Electrical and Computer Engineering, 7(2), 266-271.
5). Muraleedharan, K., Raghavan, R., Murthy, G. V. K., Murthy, V. S. S., Swamy, K. G., & Prasanna, T. (1989). An investigation on the outbreaks of pox in buffaloes in Karnatak