International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 12 | December - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

Real Time Object Detection and Security Surveillance

[1] Aditya Pasalkar, [2] Shashank Khankari, [3] Shubham Raut [4] Ganesh Koli
Guide of: Prof S.W Puranik
Undergrad. Student, Dept. of Information Technologies SKN Sinhgad Institute of Technology & Science,Lonavala, Maharashtra

Aditya Pasalkar
adityapasalkar.sknsits.it@gmail.com

Ganesh Koli
ganeshkoli.sknsits.it@gmail.com

Shubham Raut
shubhamraut.sknsits.it@gmail.com

Shashank Khankari
shashankkhankari.sknsits.it@gmail.com

***Abstract*—Real-time movement detection and video recording systems are becoming increasingly popular in a variety of applications, including security, surveillance, and traffic monitoring. These systems can provide valuable information about the activities taking place in each area and can be used to identify and track people and objects of interest.**

This paper presents a review of the state-of-the-art in real-time movement detection and video recording systems. It discusses the different types of movement detection algorithms that are available, as well as the different hardware and software components that are needed to implement a complete system. The paper also discusses some of the challenges that are associated with developing and deploying real-time movement detection and video recording systems.

I. Introduction

Movement detection is the process of identifying and tracking moving objects in a video sequence. It is a key component of many video surveillance and security systems. Real-time movement detection systems can detect moving objects in a video stream as it is being captured, without any delay. This allows for immediate action to be taken in response to detected movement, such as triggering an alarm or recording video footage.

There are a variety of different movement detection algorithms that can be used in real-time movement detection systems. Some of the most common algorithms include:

Background subtraction: This algorithm compares the current video frame to a background model, which is typically created by averaging several consecutive video frames. Any pixels that differ significantly from the background model are flagged as moving objects.

Optical flow: This algorithm tracks the movement of pixels in a video sequence. Moving objects are identified by tracking pixels that move significantly between consecutive frames.

Temporal differencing: This algorithm compares consecutive video frames to identify pixels that have changed. Moving objects are identified by tracking pixels that change significantly between consecutive frames.

The choice of movement detection algorithm depends on several factors, including the type of environment that is being monitored, the type of objects that are being tracked, and the desired accuracy and performance of the system.

Hardware and Software Components

A real-time movement detection and video recording system typically consists of the following hardware and software components:

Camera: The camera is responsible for capturing the video footage that is analysed by the movement detection algorithm.

Processing unit: The processing unit is responsible for running the movement detection algorithm and recording the video footage.

Storage device: The storage device is used to store the recorded video footage.

(Near Field Communication) tag. A key exchange technique has been proposed to verify the details of the medicine from its NFC tag [8]. None of the above methods use the automatic verification of product authenticity, and manufacturer legitimacy. When it comes to preventing counterfeit drugs in the drug supply chain, blockchain technology stands out to ensure an immutable chain of transaction ledger, tracking each step of the supply chain at the individual drug level [5].

Motivation:- Real-time movement detection and video recording systems are becoming increasingly important in a variety of applications, including:

Security: These systems can be used to monitor homes, businesses, and other public spaces for unauthorized activity.

Surveillance: These systems can be used to monitor traffic flow, identify, and track suspects, and collect evidence of crimes.

Traffic monitoring: These systems can be used to monitor traffic conditions, identify congestion, and detect accidents.

Despite the significant advances that have been made in recent years, there are still several challenges that need to be addressed to develop and deploy more effective real-time movement detection and video recording systems. These challenges include:

Improving the accuracy of movement detection algorithms in challenging conditions, such as low light or poor weather conditions.Developing more efficient movement detection algorithms that can run on lower-cost hardware.

Developing new video compression and storage techniques to reduce the cost of storing large amounts of video footage.The research paper

on real-time movement detection and video recording systems will provide a valuable contribution to the field by addressing these challenges and identifying opportunities for future research.

II. Proposed Framework

Video capture:

The first step is to capture the video footage that will be analysed by the movement detection algorithm. This can be done using a variety of different cameras, including IP cameras, webcams, and CCTV cameras.

Preprocessing:

The next step is to preprocess the video footage. This may involve tasks such as resizing the video, converting it to grayscale, and denoising it.

Real-Time Processing:

Implement real-time processing algorithms to track and record movement.

Consider using computer vision techniques or motion capture systems.

Video Recording:

Integrate video recording functionality to capture the visual representation of the movement.

Ensure synchronization with the captured movement data.The proposed structure for storage of transaction data.

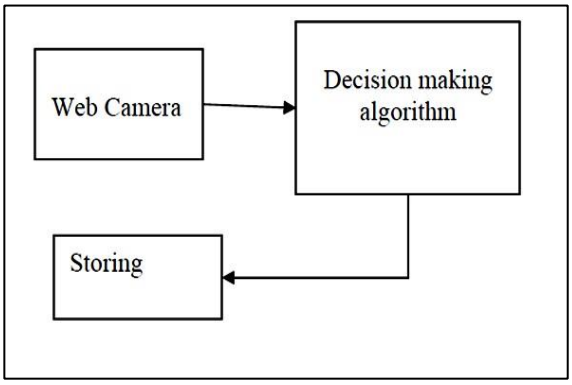

Fig 1 Basic architecture of the system

To enable real-time object detection, we will integrate an appropriate object detection model into the application. Models like YOLO, SSD, or Faster R-CNN, known for their real-time capabilities, can be chosen based on the specific project requirements. These models can identify objects and their positions in each frame.
The model will be integrated into the application using a deep learning framework like TensorFlow or PyTorch. This integration will facilitate the continuous processing of each frame from the video feed, enabling the identification of objects and their positions. A predefined confidence threshold will be set to filter out false positives, ensuring that only genuine object detections trigger further actions.

The model will be integrated into the application using a deep learning framework like TensorFlow or PyTorch. This integration will facilitate the continuous processing of each frame from the video feed, enabling the identification of objects and their positions. A predefined confidence threshold will be set to filter out false positives, ensuring that only genuine object detections trigger further actions.

The user interface is an essential component of the application, providing users with the ability to configure settings, view the live feed, and access recorded clips in a user-friendly manner. This can be implemented using graphical user interface (GUI) libraries or web interfaces, depending on the intended platform.

Additionally, the system can include alerts and notifications. Users can receive real-time alerts or notifications (e.g., via email or SMS) when an object is detected, even if recording is not activated. This feature is valuable for applications requiring immediate response to detected objects.

To optimize the system's performance, hardware acceleration, such as GPUs, can be employed for efficient object detection, ensuring that the application operates smoothly in real-time.

Thorough testing and validation are essential to ensure the accuracy and reliability of the object detection and recording components. The system should be tested under various conditions and with different objects to identify and resolve potential issues.

Comprehensive documentation and user guides will be prepared to help users understand how to use and configure the application effectively.

The deployment of the application will depend on the chosen platform, whether it's a desktop computer, an embedded system, or a cloud server.

Finally, legal and ethical considerations should not be overlooked, particularly when the application is used for surveillance or security purposes. Compliance with privacy laws and obtaining the necessary permissions for recording video are crucial.

Continuous maintenance and updates will be carried out to monitor the application's performance and address any issues that may arise, ensuring the system remains effective and reliable over time. This proposed framework provides a structured plan to achieve the project's goal of real-time object detection and recording in a continuous camera live stream.

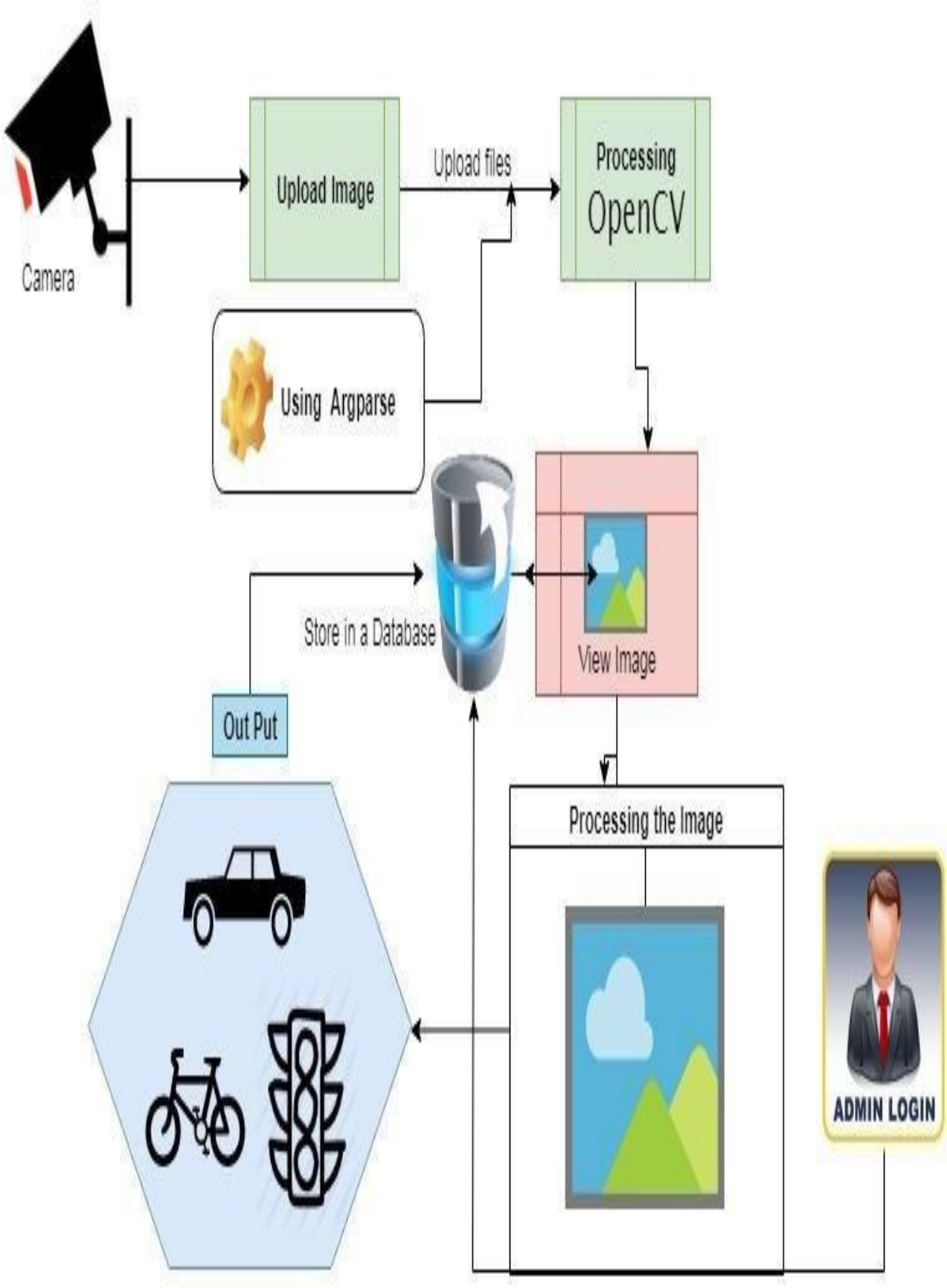

Fig. 2. Real-Time Object Detection and Smart Video Recording System

III. Conclusion

A background subtraction technique for detecting object has been proposed here

We have got accuracy for the proposed method 94.3204 % which is better than the Kim & Hwang method and Dewan & Chae method

Moving object detection is always a challenging task and there is a room to improve this method

References

[1] OpenCV Documentation: https://docs.opencv.org/
[2] YOLO (You Only Look Once) Paper: https://arxiv.org/abs/1506.02640
[3] SSD (Single Shot MultiBox Detector) Paper: https://arxiv.org/abs/1512.02325
[4] Python Video Streaming with Flask: https://blog.miguelgrinberg.com/post/video-streaming-with-flask
[5] SQLite Documentation: https://www.sqlite.org/docs.html
[6] Jenkins Documentation: https://www.jenkins.io/doc/
[7] Travis CI Documentation: https://docs.travis-ci.com/
[8] GDPR (General Data Protection Regulation) Overview: https://gdpr.eu/
[9] OWASP (Open Web Application Security Project) Website: https://owasp.org/
[10] Sphinx Documentation: https://www.sphinx-doc.org/en/master/