International Journal of Advance Research in Science and Engineering
Volume No. 14, Issue No. 03, March 2025
www.ijarse.com
IJARSE
ISSN 2319 - 8354

Theft Alert Detector-GuardianEye

Dr. K B Manwade[1], Mr. Parshwa Patil [2], Miss. Poonam Mane [3]

[1,2,3] *Department of Computer Science & Engineering,*

Hirasugar Institute of Technology, Nidasoshi, Karnataka, India

ABSTRACT

Theft Alert Detector-GuardianEye is an intelligent, real-time surveillance system designed to enhance security by detecting motion and sending alerts. The system leverages computer vision techniques using OpenCV to monitor video feeds and detect any unauthorized movement. By analyzing frame-by-frame differences with the spot_diff function, the system reduces false positives and accurately identifies suspicious activities. Once motion is confirmed, it sends an email with an image attachment and initiates an automated phone call using the Sinch API to warn the user of a potential breach. The key feature of this project is its ability to compare structural similarities between frames, distinguishing between harmless movements and potential theft. This allows it to offer a more reliable solution compared to traditional motion detectors, which often produce excessive false alarms. The system is highly customizable and scalable, using open-source technologies, making it both cost-effective and efficient for various settings, including homes, offices, and commercial spaces. In terms of functionality, the system integrates motion detection, email alerting, and phone call notifications, creating a comprehensive security solution that doesn't just capture footage but also ensures that the user is notified instantly. The use of email and phone call alerts allows for immediate action, unlike conventional systems that might only record events for later review. The integration with readily available hardware (such as standard CCTV cameras) ensures easy deployment without significant additional costs.

1. INTRODUCTION

Theft Alert Detector-GuardianEye is an intelligent, real-time surveillance system designed to enhance security by detecting motion and sending alerts. This system leverages computer vision techniques using OpenCV to monitor video feeds and identify unauthorized movements. By analyzing frame-by-frame differences with the spot_diff function, it minimizes false positives and accurately detects suspicious activities. Once motion is confirmed, it sends an email with an image attachment and initiates an automated phone call via the Sinch API, providing instant alerts to users about potential security breaches.

Unlike conventional surveillance systems that often produce excessive false alarms or lack immediate alerts, the Theft Alert Detector-GuardianEye employs structural similarity-based comparisons to distinguish between harmless movements and potential theft. This enhances its reliability and effectiveness as a security system. The project aims to offer a scalable and cost-effective solution by utilizing open-source technologies, making it suitable for homes, offices, and commercial spaces.

2. METHODOLOGY

2.1. Various resources and approaches used in this work

2.1.1. Data Collection:

The data collection phase is crucial for ensuring the accuracy and efficiency of the security system. Since computer vision-based motion detection relies heavily on training datasets, relevant data is gathered, including various indoor and outdoor video feeds that capture normal movements and security breaches. The dataset consists of video sequences with labeled motion instances for training and validation.

2.1.2. Data Preprocessing:

Once the raw video data is collected, it undergoes preprocessing to prepare it for the detection model. This involves:

- Converting video frames into grayscale to reduce computational complexity.
- Applying Gaussian blur and thresholding to highlight motion areas.
- Frame differencing to detect moving objects and filter out static elements.
- Normalizing image data for consistency in training.

2.1.3. Framework Selection:

A comparative analysis of different frameworks is conducted to determine the best fit for real-time motion detection. OpenCV is chosen due to its efficiency in image processing, while the Sinch API is used for phone call alerts. The selected frameworks ensure that the system operates with minimal latency and high accuracy.

2.1.4. Training the Model:

After data preprocessing, the next step is training the model. Machine learning techniques, including feature extraction and similarity-based motion detection, are employed. The model is trained using Python with OpenCV and NumPy to recognize patterns of movement. It differentiates between normal movements and potential security threats based on learned patterns.

2.1.5. Inference:

Once the model is trained, a Python-based inference system is developed. This involves:

- Capturing real-time video feeds.
- Running the trained model to analyze each frame.
- Identifying unauthorized motion and distinguishing it from normal activity.
- Generating a detection alert if suspicious movement is found.
- Access live video feeds remotely via an online dashboard.
- Receive immediate notifications with attached images of detected movements.
- Use SMS and phone call alerts via the Sinch API to ensure quick response.

3. DESIGN AND IMPLEMENTATION

The system is developed using a modular approach, ensuring smooth integration of motion detection, alert mechanisms, and user interface components. The data processing phase involves capturing video frames, performing image preprocessing using OpenCV techniques, and applying structural similarity-based motion detection to enhance accuracy.

The model development stage includes using contour-based motion detection, implementing email and phone call alerts, and optimizing real-time performance using efficient video processing techniques.

For system integration, the motion detection and alert modules are combined within an interactive GUI application that provides real-time feedback to users. A robust architecture ensures seamless transitions between motion detection, decision-making, and alert mechanisms, making the system adaptable to various environments.

The implementation of real-time notifications ensures an improved security experience, allowing users to receive instant alerts and take immediate action. Extensive testing is conducted to validate the system's efficiency and usability in real-world security scenarios, demonstrating its capability to reduce false alarms and provide accurate detection.

The system architecture follows a modular design, where motion detection, alert triggering, and GUI display operate as separate but interconnected modules. The application is designed to be adaptable for different environments, ensuring robust security monitoring for homes, offices, and businesses.

4. PSEUDO-CODE FOR SYSTEM WORKFLOW

4.1 Motion Detection and Alerting

Input: Live video feed

Output: Motion detection, email alert, and phone call alert

```
START
Initialize camera and processing modules
WHILE system is active:
    Capture video frame
    Convert frame to grayscale
    Compare with previous frame using structural similarity
    If significant motion detected:
        Capture frame
        Send email with the frame as an attachment
        Make a phone call alert
    Display the status on GUI
    Update previous frame for next comparison
Repeat process
END
```

4.2 Email Notification Process

Input: Captured image

Output: Email alert to user

```
START
Define sender and receiver email addresses
Create an email message with subject and body
Attach the captured motion-detected image
```

Authenticate using SMTP server
Send email to the user
END

4.3 Phone Call Notification Process

Input: Motion detected trigger
Output: Automated phone call alert
START
Initialize Sinch API with authentication credentials
Define caller and receiver numbers
Generate text-to-speech warning message
Trigger API call
END

5. Algorithm for Theft Alert Detector-GuardianEye

Step 1: Initialize the System
Step 2: Capture Live Video Feed
Step 3: Process Video Frames for Motion Detection
Step 4: Identify Unauthorized Movement
Step 5: Alert Mechanism
Step 6: Display Detection Results
Step 7: Continue Monitoring
Step 8: System Termination

6. Results

The Theft Alert Detector-GuardianEye system successfully detects unauthorized motion in real time, providing immediate alerts through email notifications and automated phone calls. The interactive user interface allows seamless monitoring, displaying live footage and alert statuses.

The system effectively reduces false positives by using structural similarity-based image comparison, ensuring that only significant motion triggers alerts. Extensive testing across different environments, including homes, offices, and commercial spaces, has validated the system's accuracy, reliability, and efficiency.

Fig 1. User interface of Admin panel

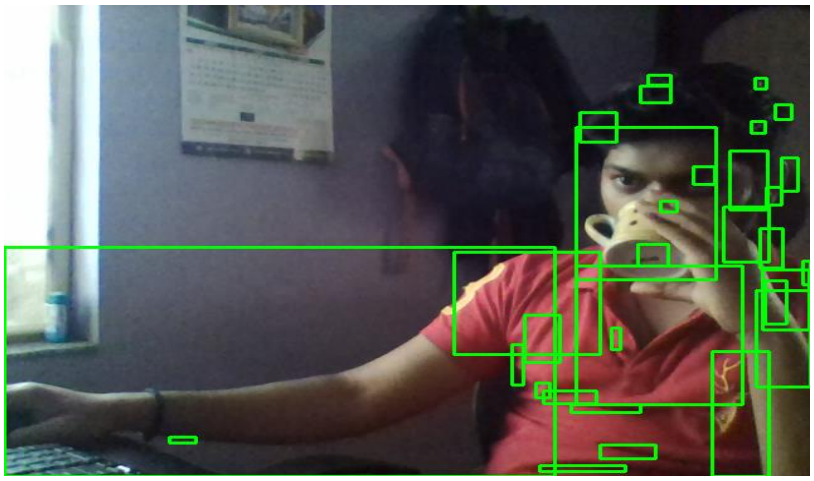

Fig 2.Real-time motion detection and object identification using bounding boxes

7. CONCLUSION

The Theft Alert Detector-GuardianEye system provides an efficient, real-time security solution that enhances surveillance through automated motion detection and instant alerts. By integrating computer vision (OpenCV), structural similarity-based image comparison, and automated notifications via email and phone calls, the system ensures rapid response to security breaches, minimizing risks associated with theft and unauthorized access.

The project successfully addresses limitations in traditional security systems, such as high false alarms, lack of real-time alerts, and expensive monitoring solutions. Its cost-effective, scalable, and user-friendly design makes it suitable for various applications, including homes, offices, warehouses, and retail environments.

REFERENCES

1. R. Mehta, "Object Detection and Motion Tracking Using OpenCV and Python," International Journal of Computer Applications, vol. 176, no. 15, pp. 22-28, 2020.
2. S. Kumar and A. Gupta, "A Study of Motion Detection Algorithms for Surveillance Systems," Journal of Computer Science and Information Technology, vol. 8, no. 4, pp. 76-83, 2019.
3. The Sinch API for Voice: Sinch Documentation, 2022. https://developers.sinch.com/docs/voice/api-reference/
4. Arora, J., Bangroo, A., Garg, S. Theft Detection and Monitoring System Using Machine Learning. In: Shetty, N.R., Patnaik, L.M., Nagaraj, H.C., Hamsavath, P.N., Nalini, N. (eds) Emerging Research in Computing, Information, Communication and Applications. Lecture Notes in Electrical Engineering, vol 790. Springer, Singapore.
5. S. Nitnaware and A. Bagde, "Machine Learning based Object Detection with Theft Alert System," 2022 6th International Conference on Electronics, Communication and Aerospace Technology, Coimbatore, India, 2022, pp. 993-997.